Clustering

Nishant Mehta

Lecture 16

Clustering documents by main topic

Tesla free Supercharging program extended to all stations across Poland and Slovakia

3 hours ago

Young investors are impacting Wall Street and they're betting big on T

7 hours ago

Tesla defends Autopilot, FSD Beta in response to senators' "significant concerns"

8 hours ago

Mask mandate Ontario: Requirement to end in most places on March 21 | CTV News

CTV News Toronto · 7 hours ago

'We had no idea': Unvaccinated B.C. doctor stuns patients over COVID-19 beliefs

Global News · 15 hours ago

Moore to update Ontario's 'plan to live with and manage COVID-19,' Ford to speak

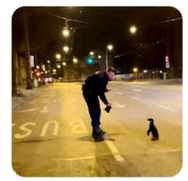
CityNews Toronto · 19 hours ago

Lion, Monkeys and Red Wolves Escape Kharkiv Zoo After Heavy Shelling

Newsweek · 7 days ago

Penguin escapes zoo, wanders Budapest streets

UPI News · 1 hour ago

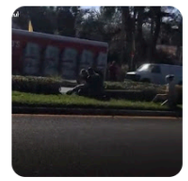


Ukrainian Zoo animals safe in Poland after escape from Russia invasion

Business Insider · 6 days ago

Florida Zoo Crocodile Breaks Out of Van and Races Down Road in Escape Attempt Caught on Video

PEOPLE · Feb 24



Clustering documents by main topic

Tesla free Supercharging program extended to all stations across Poland and Slovakia

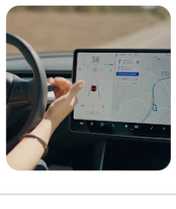
3 hours ago

Young investors are impacting Wall Street and they're betting big on T

7 hours ago

Tesla defends Autopilot, FSD Beta in response to senators' "significant concerns"

8 hours ago



Mask mandate Ontario: Requirement to end in most places on March 21 | CTV News

CTV News Toronto · 7 hours ago

'We had no idea': Unvaccinated B.C. doctor stuns patients over COVID-19 beliefs

Global News · 15 hours ago

Moore to update Ontario's 'plan to live with and manage COVID-19,' Ford to speak

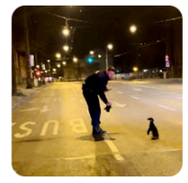
CityNews Toronto · 19 hours ago

Lion, Monkeys and Red Wolves Escape Kharkiv Zoo After Heavy Shelling

Newsweek · 7 days ago

Penguin escapes zoo, wanders Budapest streets

UPI News · 1 hour ago

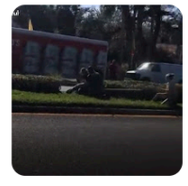


Ukrainian Zoo animals safe in Poland after escape from Russia invasion

Business Insider · 6 days ago

Florida Zoo Crocodile Breaks Out of Van and Races Down Road in Escape Attempt Caught on Video

PEOPLE · Feb 24



Clustering documents by main topic

Tesla free Supercharging program extended to all stations across Poland and Slovakia

3 hours ago

Young investors are impacting Wall Street and they're betting big on T

7 hours ago

Tesla defends Autopilot, FSD Beta in response to senators' "significant concerns"

8 hours ago

Mask mandate Ontario: Requirement to end in most places on March 21 | CTV News

CTV News Toronto · 7 hours ago

'We had no idea': Unvaccinated B.C. doctor stuns patients over COVID-19 beliefs

Global News · 15 hours ago

Moore to update Ontario's 'plan to live with and manage COVID-19,' Ford to speak

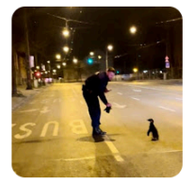
CityNews Toronto · 19 hours ago

Lion, Monkeys and Red Wolves Escape Kharkiv Zoo After Heavy Shelling

Newsweek · 7 days ago

Penguin escapes zoo, wanders Budapest streets

UPI News · 1 hour ago

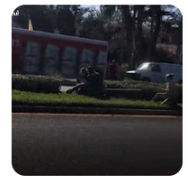


Ukrainian Zoo animals safe in Poland after escape from Russia invasion

Business Insider · 6 days ago

Florida Zoo Crocodile Breaks Out of Van and Races Down Road in Escape Attempt Caught on Video

PEOPLE · Feb 24



Clustering documents by main topic

"Elon Musk"

Tesla free Supercharging program extended to all stations across Poland and Slovakia

3 hours ago

Young investors are impacting Wall Street and they're betting big on T

7 hours ago

Tesla defends Autopilot, FSD Beta in response to senators' "significant concerns"

8 hours ago

"COVID-19"

Mask mandate Ontario: Requirement to end in most places on March 21 | CTV News

CTV News Toronto · 7 hours ago

'We had no idea': Unvaccinated B.C. doctor stuns patients over COVID-19 beliefs

Global News · 15 hours ago

Moore to update Ontario's 'plan to live with and manage COVID-19,' Ford to speak

CityNews Toronto · 19 hours ago

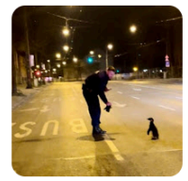
"Zoo escape"

Lion, Monkeys and Red Wolves Escape Kharkiv Zoo After Heavy Shelling

Newsweek · 7 days ago

Penguin escapes zoo, wanders Budapest streets

UPI News · 1 hour ago

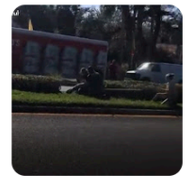


Ukrainian Zoo animals safe in Poland after escape from Russia invasion

Business Insider · 6 days ago

Florida Zoo Crocodile Breaks Out of Van and Races Down Road in Escape Attempt Caught on Video

PEOPLE · Feb 24

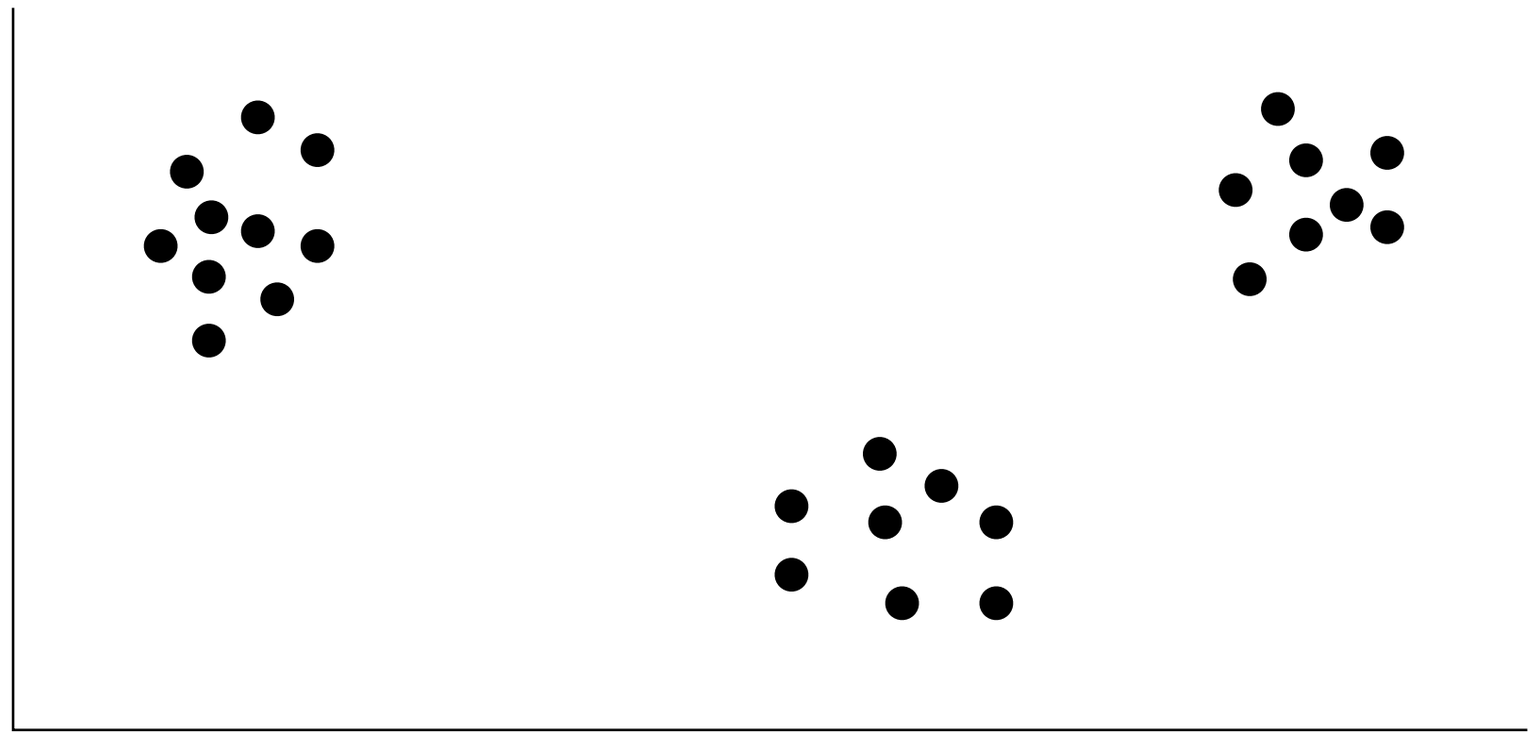


Clustering

Given: A set of unlabeled examples

Goal: Group the examples into clusters such that

- Examples in the same cluster are all similar
- Examples in different clusters are dissimilar

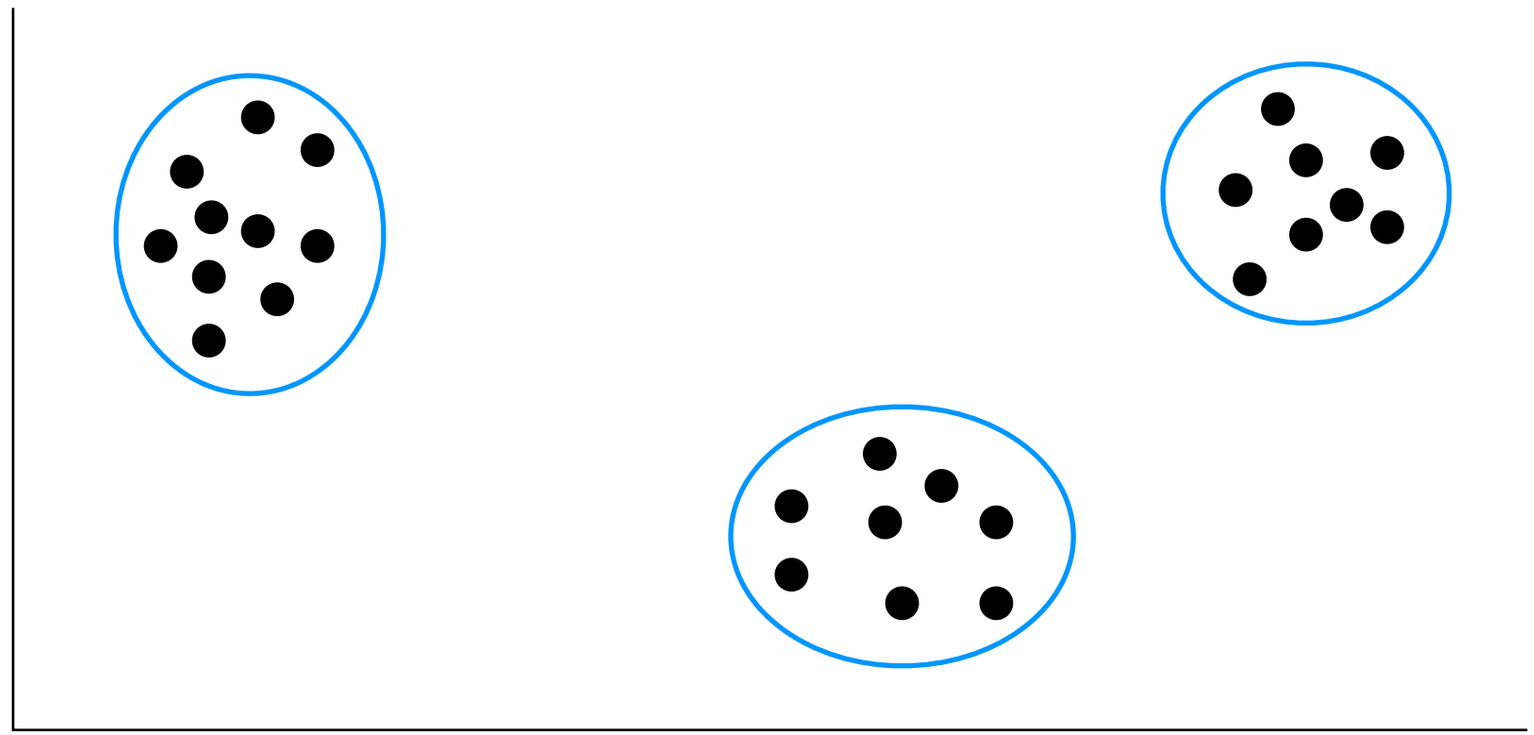


Clustering

Given: A set of unlabeled examples

Goal: Group the examples into clusters such that

- Examples in the same cluster are all similar
- Examples in different clusters are dissimilar



Applications

Note: Clustering often is not the final objective but rather is done to help tackle some (potentially supervised) learning task

Clustering users into types

Typical underlying goal: Predict how users rate movies (by predicting how each type of user rates movies)

Clustering in social networks

Typical underlying goal: Identifying communities

Clustering news articles to identify trends

Clustering images via image segmentation

Typical underlying goal: Object recognition

The *k*-means problem

Classic clustering problem, based on Euclidean distance as dissimilarity measure

A clustering is identified by the mean (center) of each cluster C_j :

$$\mu_j = \frac{1}{|C_j|} \sum_{x \in C_j} x$$

The ***cost*** of a clustering is the sum of the squared (Euclidean) distances of the points from their respective cluster centers:

$$W(C) = \sum_{j=1}^{k} \sum_{x \in C_j} \|x - \mu_j\|^2$$

The *k*-means problem:

Find a minimum-cost clustering of the examples into *k* clusters

The *k*-means problem

The *k*-means problem:

Find a minimum-cost clustering of the examples into *k* clusters

$$\underset{C_1,\ldots,C_k}{\operatorname{argmin}} \sum_{j=1}^{k} \sum_{x \in C_j} \|x - \mu_j\|^2$$

Some bad news

k-means problem is NP-hard, even when:

- $k = 2$ with general dimension
- dimension two (points in the plane) with general *k*

Lloyd's Algorithm

Choose (somehow…) k initial cluster centers $\mu_1, \ldots, \mu_k$

repeat

1. Assign each point to the cluster with the closest center

$$\left(\text{Assign each } x_i \text{ to cluster } C_{j^*} \text{ for } j^* = \operatorname*{argmin}_{j \in [k]} \|x_i - \mu_j\| \right)$$

2. Update the center of each cluster

$$\left(\text{For each } j \in [k] \,,\ \mu_j = \frac{1}{|C_j|} \sum_{x \in C_j} x \right)$$

until cluster memberships do not change

Lloyd's Algorithm

Choose (somehow…) ***k*** initial cluster centers $\mu_1, \ldots, \mu_k$

repeat

1. Assign each point to the cluster with the closest center
2. Update the center of each cluster

until cluster memberships do not change

How to initialize cluster centers? A common choice is to select them uniformly at random from the set of examples.

This turns out to be horrible in practice! More on this later…

*k-means is the **problem**, while Lloyd's algorithm is the above **algorithm**.*
This algorithm is so popular for k-means that it's often called the "k-means algorithm".

Example of Lloyd's algorithm

Lloyd’s Algorithm - Convergence

Lloyd’s Algorithm - Convergence

Does Lloyd’s algorithm converge?

Lloyd’s Algorithm - Convergence

Does Lloyd’s algorithm converge?

Yes!

Lloyd's Algorithm - Convergence

Does Lloyd's algorithm converge?

Yes!

Intuition: There are only finitely many ways to assign *n* examples to *k* clusters, and it makes progress in each iteration.

Lloyd's Algorithm - Convergence

Does Lloyd's algorithm converge?

Yes!

Intuition: There are only finitely many ways to assign *n* examples to *k* clusters, and it makes progress in each iteration.

E-step (updating assignments)

Can only decrease objective. Why? Each point is assigned to a center that is no farther than its current center.

Lloyd's Algorithm - Convergence

Does Lloyd's algorithm converge?

Yes!

Intuition: There are only finitely many ways to assign *n* examples to *k* clusters, and it makes progress in each iteration.

E-step (updating assignments)

Can only decrease objective. Why? Each point is assigned to a center that is no farther than its current center.

M-step (updating centers)

For a given cluster with fixed assignments, the sum of the squared errors for the points assigned to that cluster is minimized by taking the center to be the mean of the points assigned to that cluster

Lloyd's Algorithm - Convergence

Does Lloyd's algorithm converge?

Yes!

Intuition: There are only finitely many ways to assign *n* examples to *k* clusters, and it makes progress in each iteration.

E-step (updating assignments)

Can only decrease objective. Why? Each point is assigned to a center that is no farther than its current center.

M-step (updating centers)

For a given cluster with fixed assignments, the sum of the squared errors for the points assigned to that cluster is minimized by taking the center to be the mean of the points

You can also use this to remember how the algorithm is defined!
When each decision is viewed as a local choice (in isolation of other steps of the algorithm):
1) the assignments chosen by Lloyd's algorithm in the E-step are the best possible
2) the centers chosen by Lloyd's algorithm in the M-step are the best possible

Practical issues

How many iterations are required?

In practice, most of the progress is made in the first few iterations

Usually finishes in at most 10 iterations

Runtime (for naive implementation)?

Each iteration costs $O(n\ k\ d)$, for d features and n examples

How to initialize?

Random initialization

Choose centers uniformly at random from the examples

Performance: can be horrible in practice

How to initialize?

González algorithm

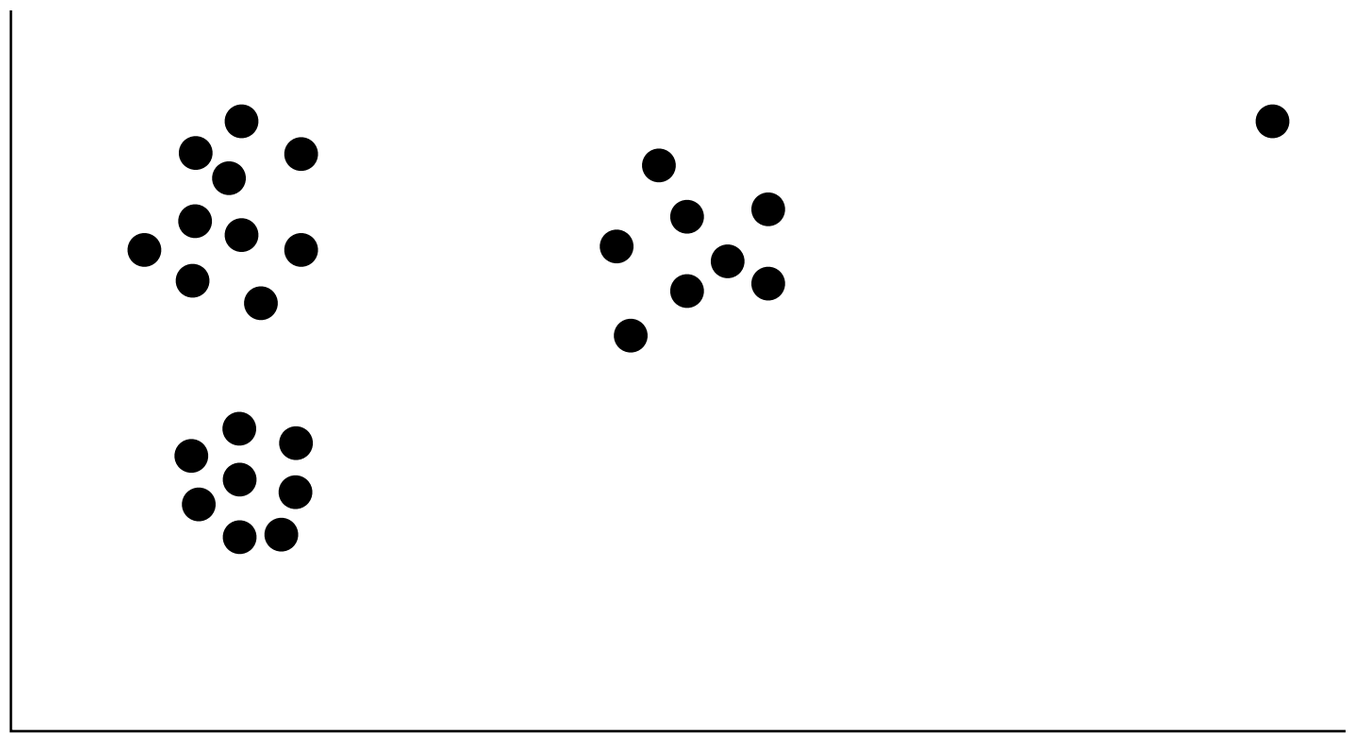
Choose first center to be arbitrary example (can be uniform at random)

For each successive center, choose it to be the farthest example from the current set of centers

Performance: Tends to be sensitive to outliers

But, for the related k-centers problem, it immediately gives a solution with at most twice the optimal cost

k-centers problem:
Find *k* centers such that the maximum distance of any point to its closest center is minimized



How to initialize?

González algorithm

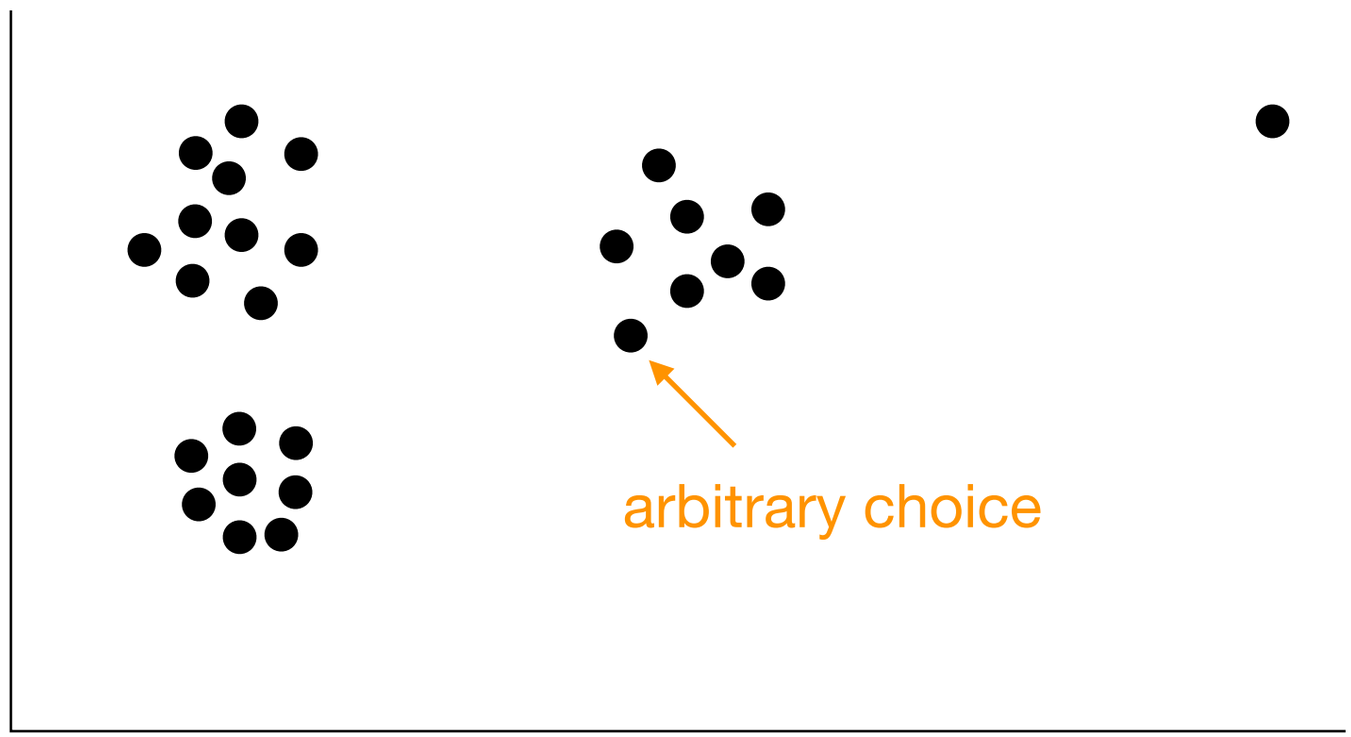
Choose first center to be arbitrary example (can be uniform at random)

For each successive center, choose it to be the farthest example from the current set of centers

Performance: Tends to be sensitive to outliers

But, for the related k-centers problem, it immediately gives a solution with at most twice the optimal cost

k-centers problem:
Find *k* centers such that the maximum distance of any point to its closest center is minimized



How to initialize?

González algorithm

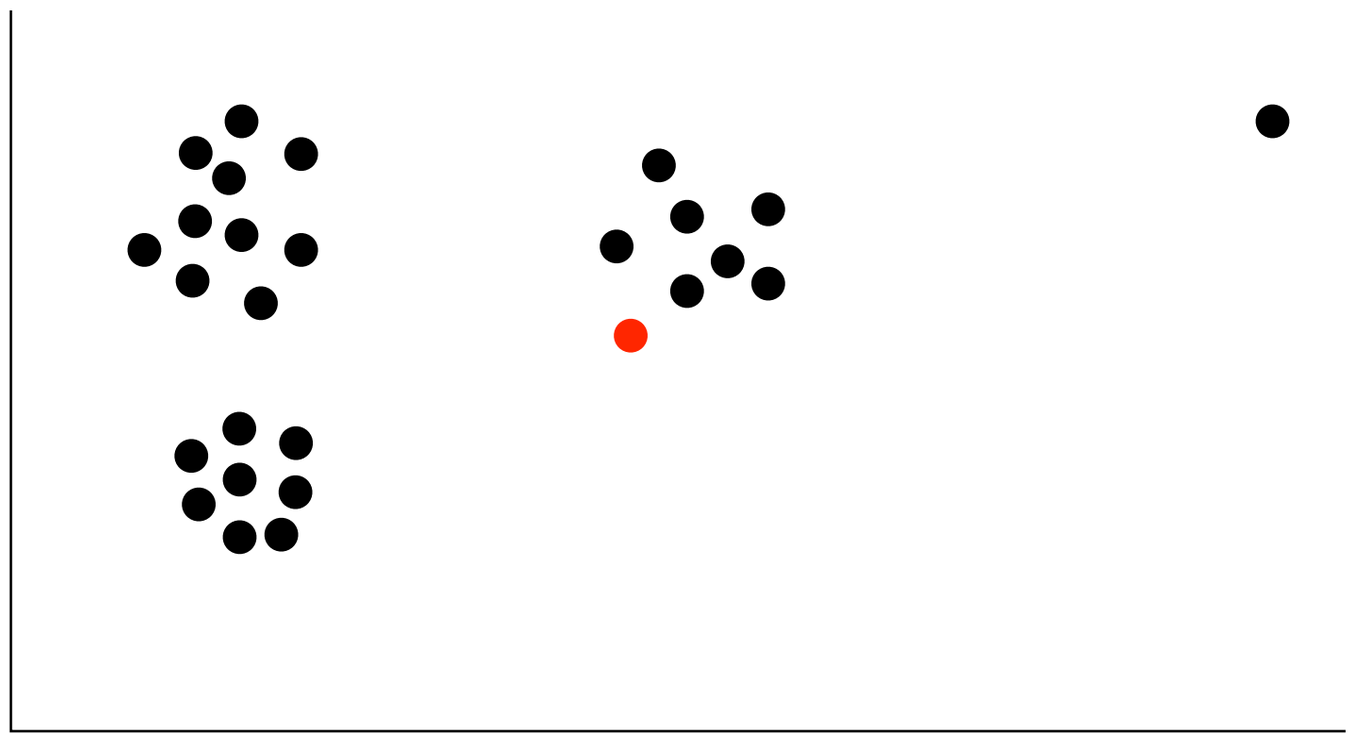
Choose first center to be arbitrary example (can be uniform at random)

For each successive center, choose it to be the farthest example from the current set of centers

Performance: Tends to be sensitive to outliers

But, for the related k-centers problem, it immediately gives a solution with at most twice the optimal cost

k-centers problem:
Find *k* centers such that the maximum distance of any point to its closest center is minimized



How to initialize?

González algorithm

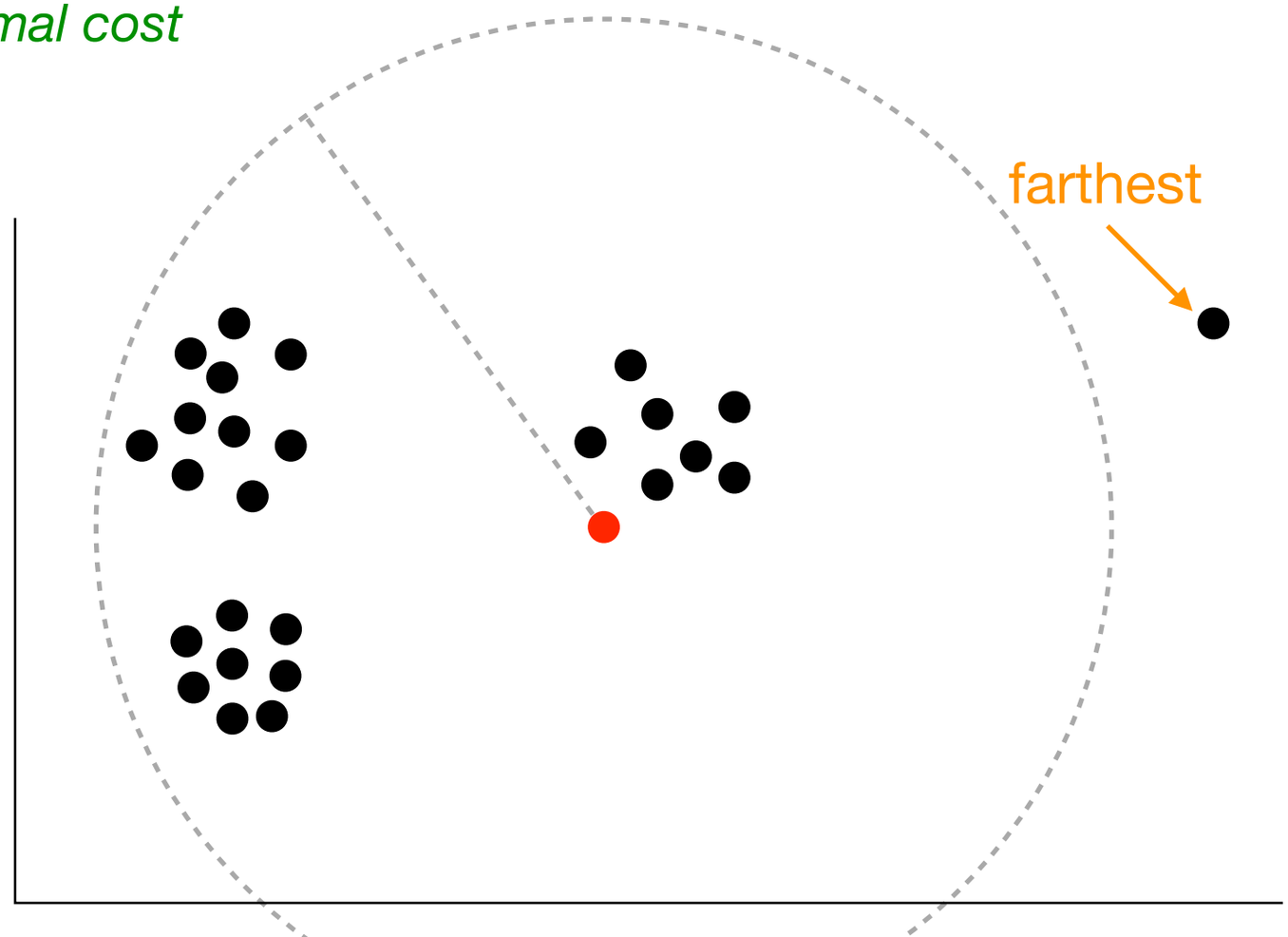
Choose first center to be arbitrary example (can be uniform at random)

For each successive center, choose it to be the farthest example from the current set of centers

Performance: Tends to be sensitive to outliers

But, for the related k-centers problem, it immediately gives a solution with at most twice the optimal cost

k-centers problem:
Find *k* centers such that the maximum distance of any point to its closest center is minimized



How to initialize?

González algorithm

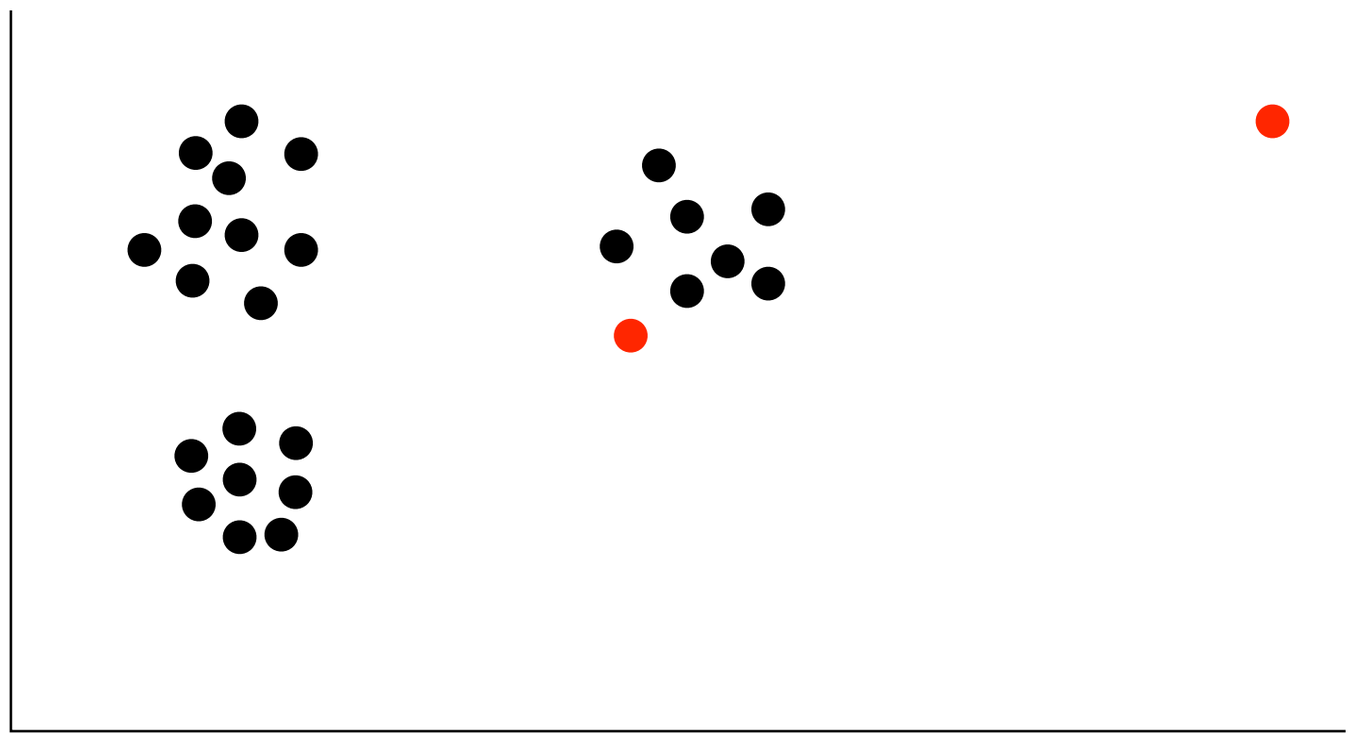
Choose first center to be arbitrary example (can be uniform at random)

For each successive center, choose it to be the farthest example from the current set of centers

Performance: Tends to be sensitive to outliers

But, for the related k-centers problem, it immediately gives a solution with at most twice the optimal cost

k-centers problem:
Find *k* centers such that the maximum distance of any point to its closest center is minimized



How to initialize?

González algorithm

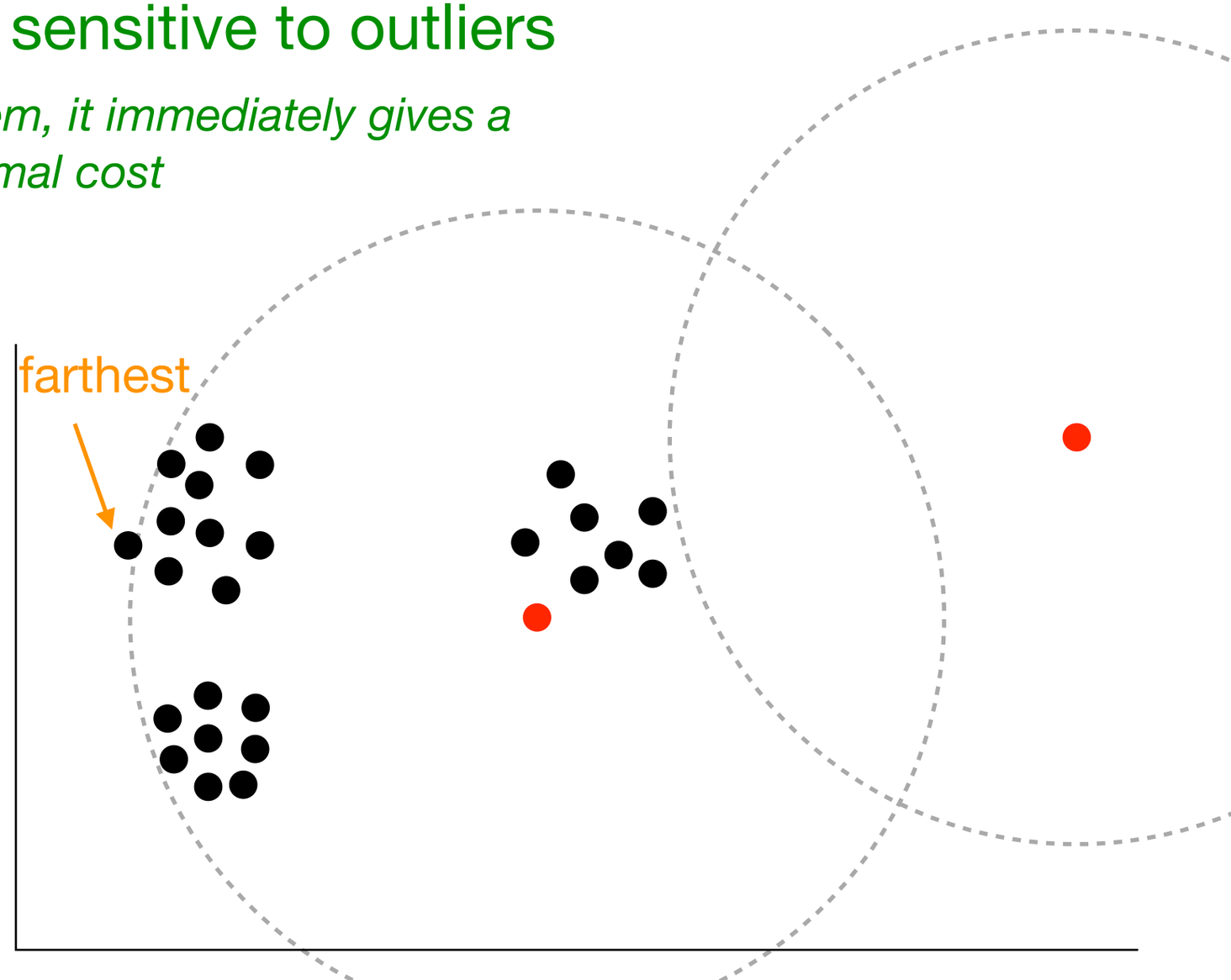
Choose first center to be arbitrary example (can be uniform at random)

For each successive center, choose it to be the farthest example from the current set of centers

Performance: Tends to be sensitive to outliers

But, for the related k-centers problem, it immediately gives a solution with at most twice the optimal cost

k-centers problem:
Find *k* centers such that the maximum distance of any point to its closest center is minimized



How to initialize?

González algorithm

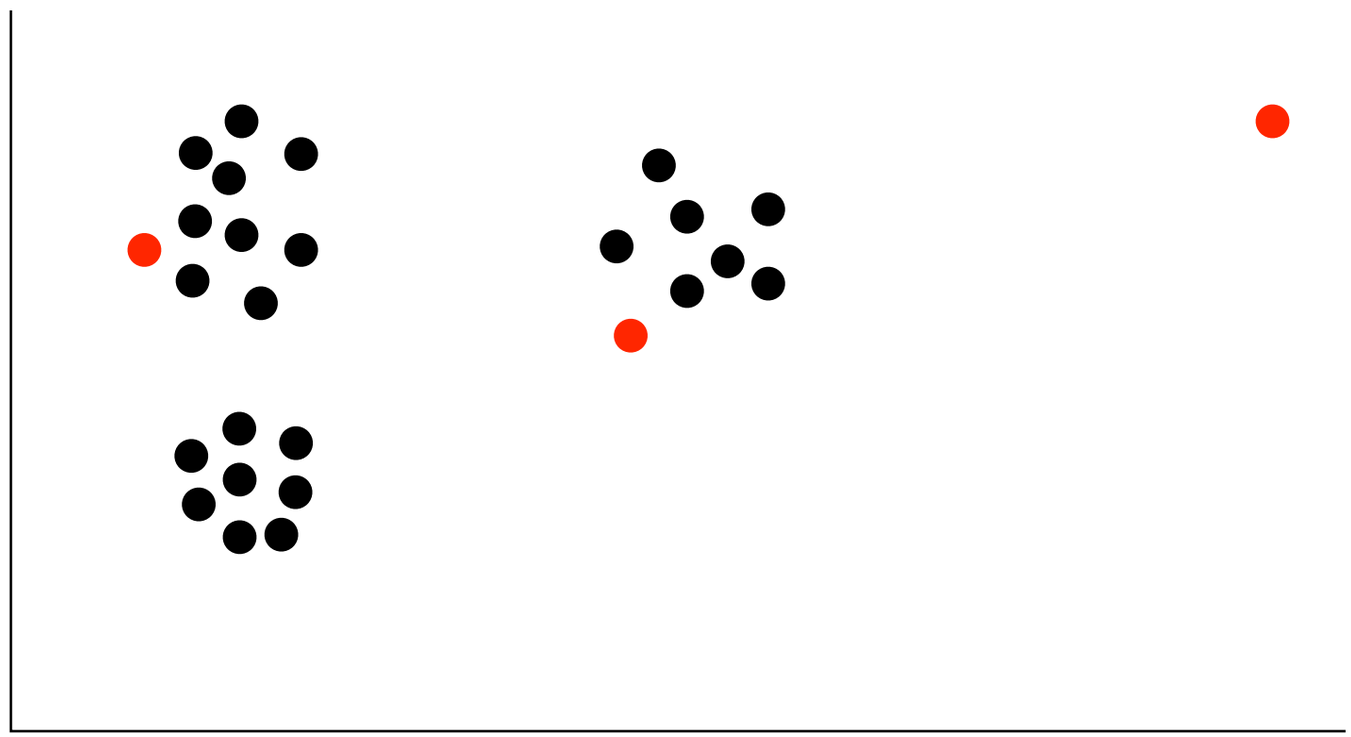
Choose first center to be arbitrary example (can be uniform at random)

For each successive center, choose it to be the farthest example from the current set of centers

Performance: Tends to be sensitive to outliers

But, for the related k-centers problem, it immediately gives a solution with at most twice the optimal cost

k-centers problem:
Find *k* centers such that the maximum distance of any point to its closest center is minimized



How to initialize?

González algorithm

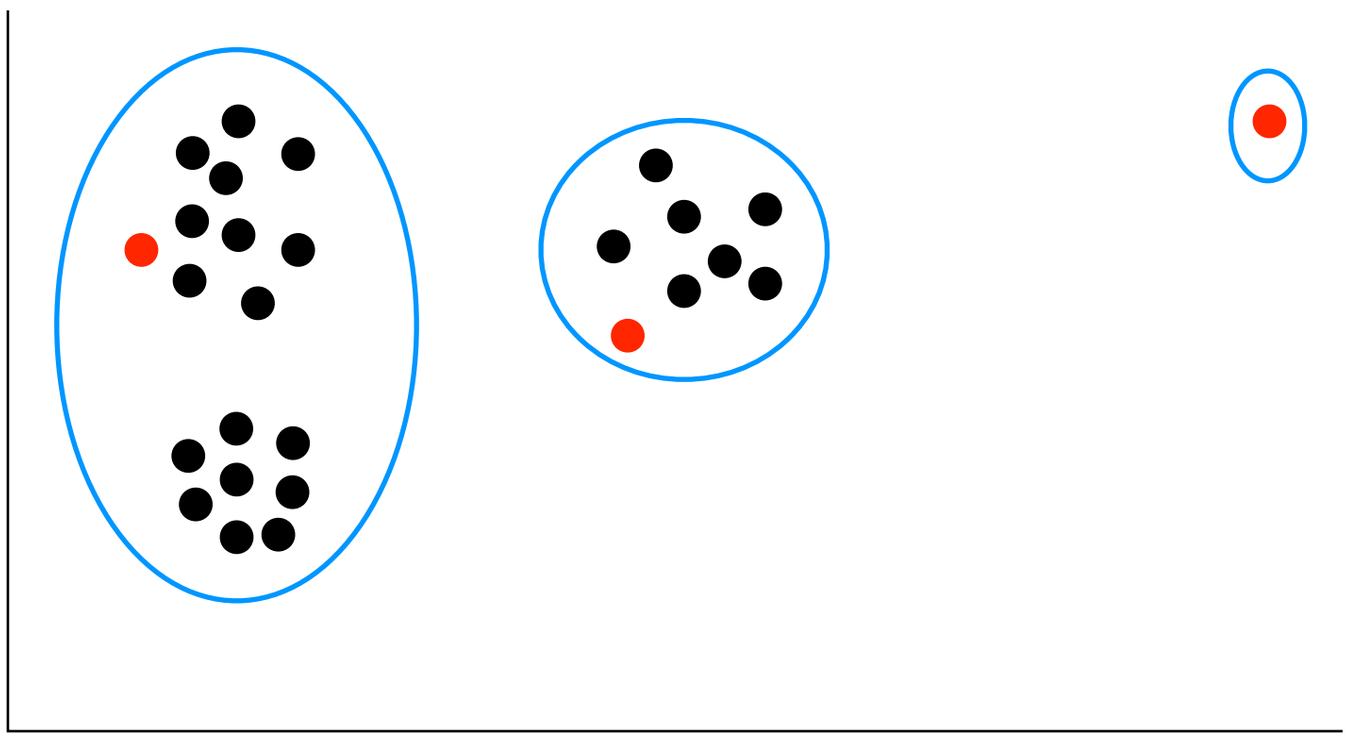
Choose first center to be arbitrary example (can be uniform at random)

For each successive center, choose it to be the farthest example from the current set of centers

Performance: Tends to be sensitive to outliers

But, for the related k-centers problem, it immediately gives a solution with at most twice the optimal cost

k-centers problem:
Find *k* centers such that the maximum distance of any point to its closest center is minimized



How to initialize?

_k_-means++

Idea: Like the González algorithm, but less sensitive to outliers

Choose first center to be arbitrary example (can be uniform at random)

Choose each successive center randomly from the remaining examples as follows:

Choose remaining example ***x*** with probability proportional to the squared distance of that example to its closest existing center

Performance: *Expected value* of the cost of the solution (from initialization alone) is guaranteed to be at most **O(log *k*)** times the optimal cost. Running Lloyd's algorithm thereafter can only improve the solution further!

Fighting poor initialization

Initialization is random, so with some probability any single initialization could be one that gives a bad clustering

Standard trick:

Run Lloyd's algorithm multiple times (e.g. for five repetitions, run ***k***-means++ and then Lloyd's algorithm)

Pick the clustering that achieves the lowest cost

How to choose *k*?

How low can the objective be?

How to choose *k*?

How low can the objective be?

Assuming all the examples are distinct, when will the objective reach this minimum value?

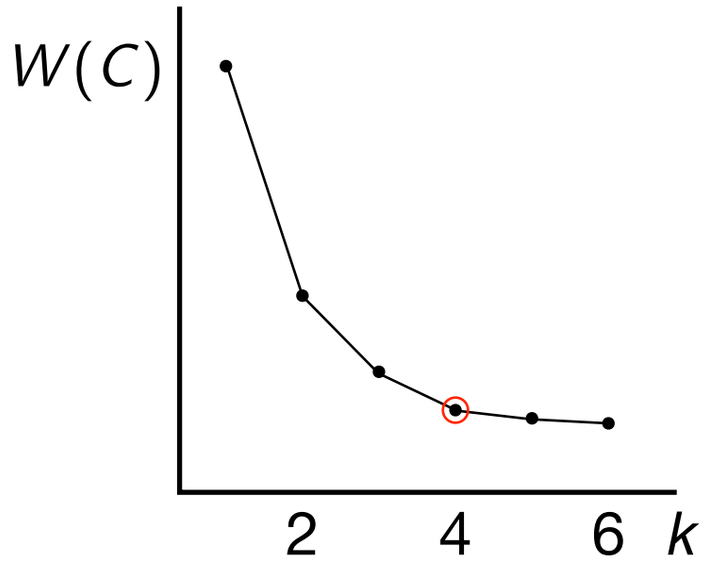
How to choose *k*?

How low can the objective be?

Assuming all the examples are distinct, when will the objective reach this minimum value?

Idea for selecting *k*:

Plot objective value (sum of squared distances) as *k* increases.
Objective typically decreases rapidly at first.
Select value *k* corresponding to a kink in the curve
(point at which objective stops decreasing rapidly)



But how to know where the kink is?
Not always easy to see.
Further reading for an automated method

J. R. Statist. Soc. B (2001)
63, *Part* 2, *pp.* 411–423

Estimating the number of clusters in a data set via the gap statistic

Robert Tibshirani, Guenther Walther and Trevor Hastie

Stanford University, USA

[Received February 2000. Final revision November 2000]

Summary. We propose a method (the 'gap statistic') for estimating the number of clusters (groups) in a set of data. The technique uses the output of any clustering algorithm (e.g. *K*-means or hierarchical), comparing the change in within-cluster dispersion with that expected under an appropriate reference null distribution. Some theory is developed for the proposal and a simulation study shows that the gap statistic usually outperforms other methods that have been proposed in the literature.

Keywords: Clustering; Groups; Hierarchy; *K*-means; Uniform distribution

Issues with the *k*-means problem

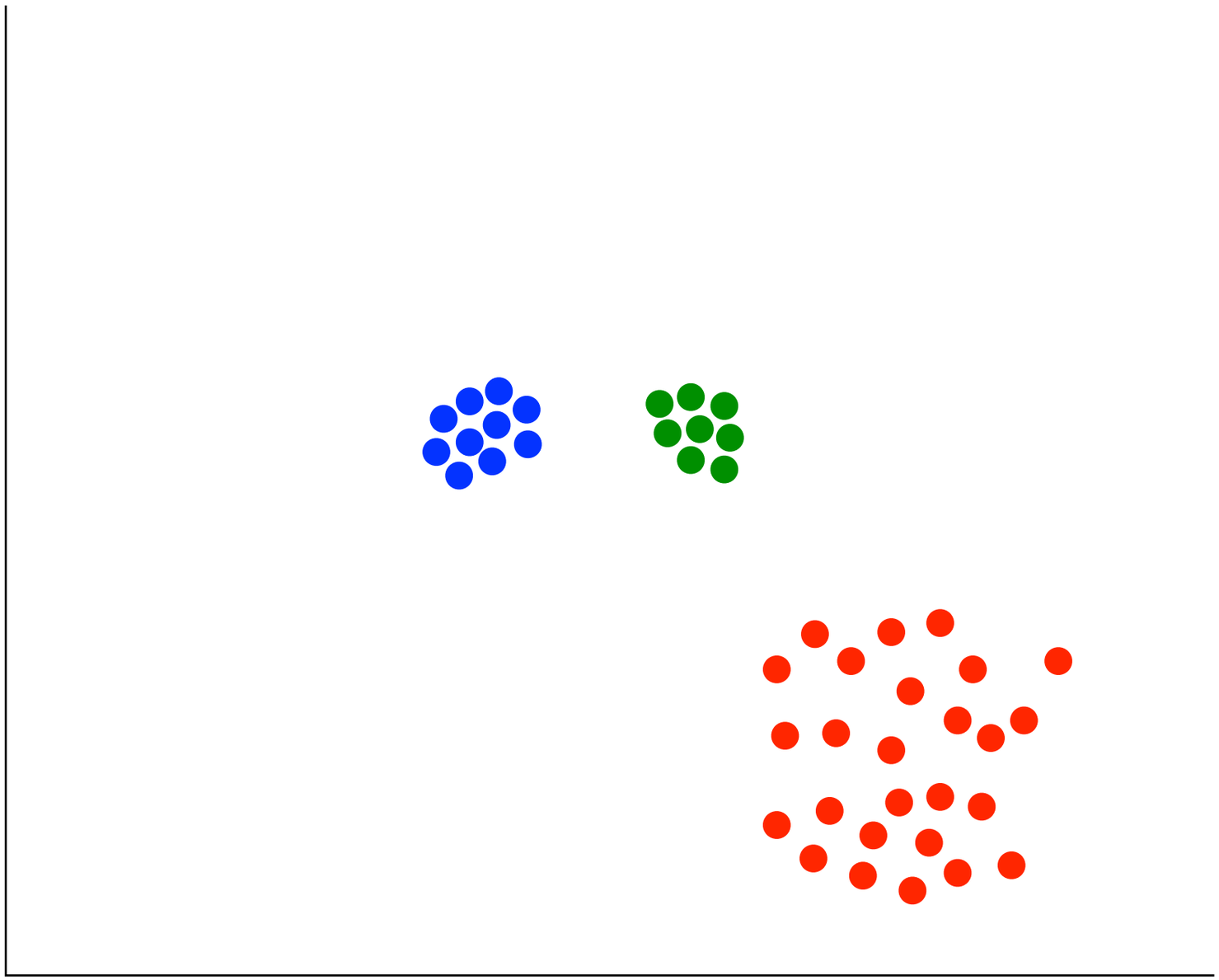
k-means isn't a good idea when the natural clusters:

- Have drastically different sizes
- Have different densities
- Are non-globular (aren't well-described by centers)

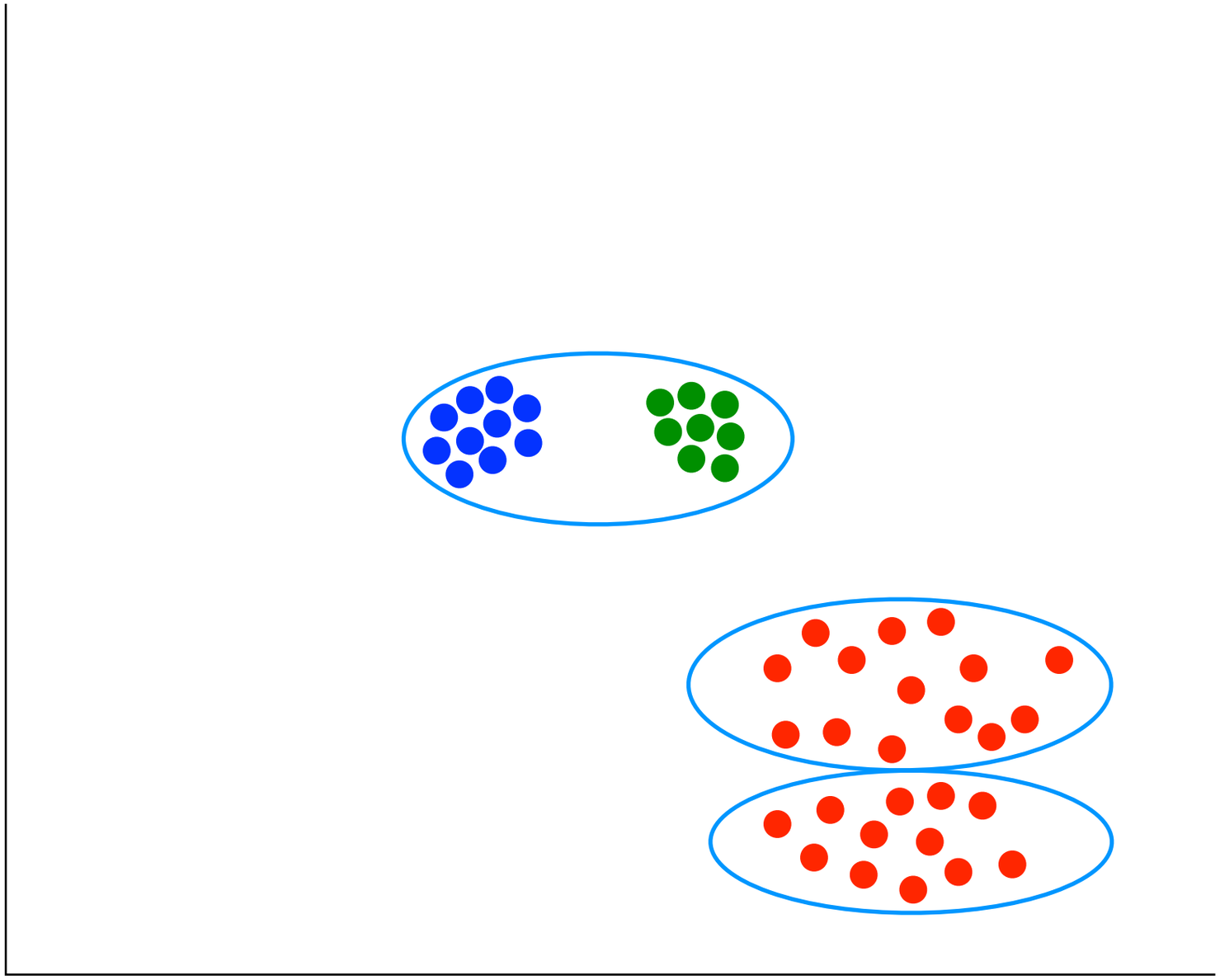
  If a cluster has graph-theoretic structure
  (points in a cluster have nearest neighbors in the cluster),
  a good method is *spectral clustering*

(advanced topic; not covered in this course)

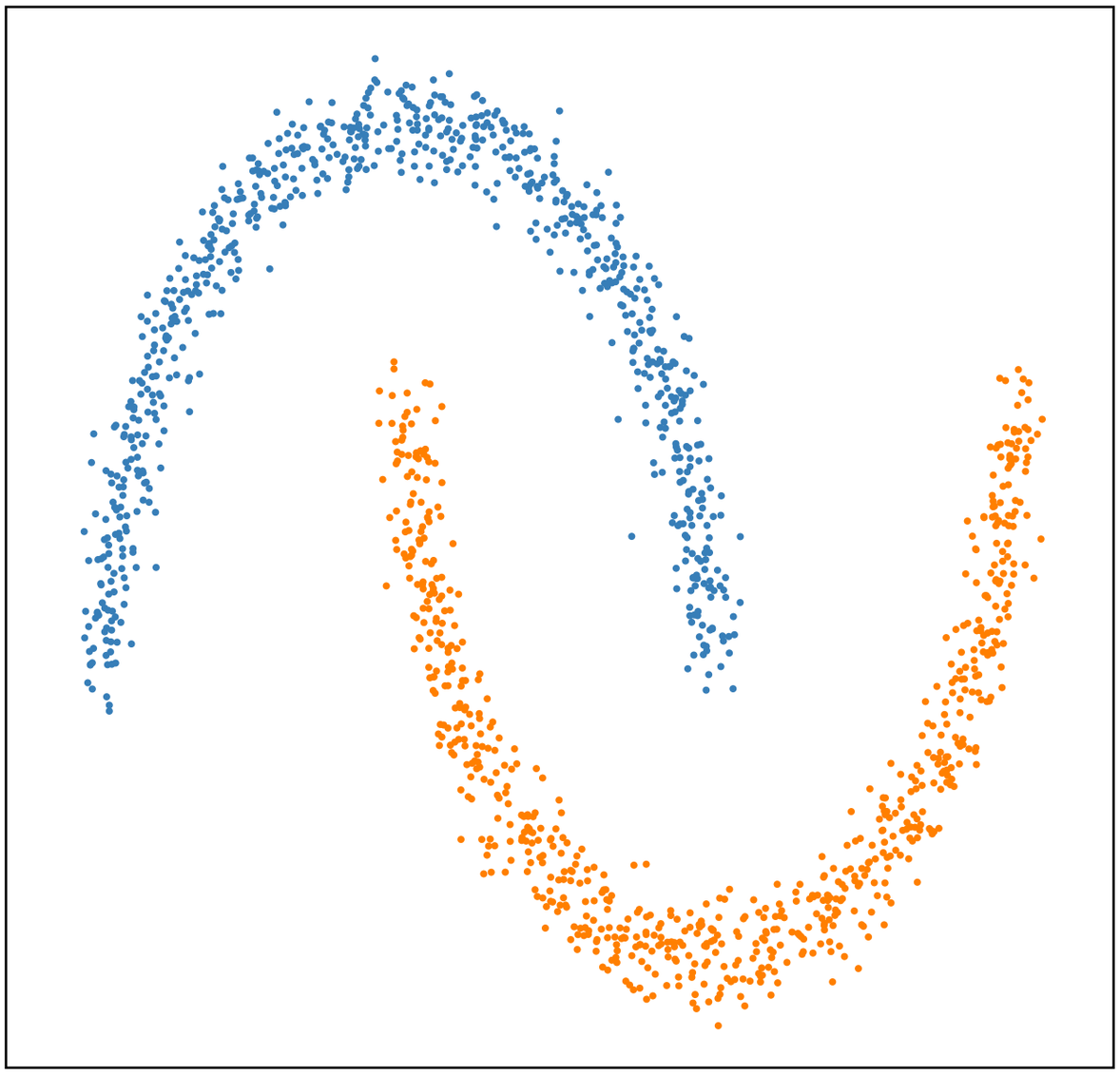
k-means fail: different sizes or densities



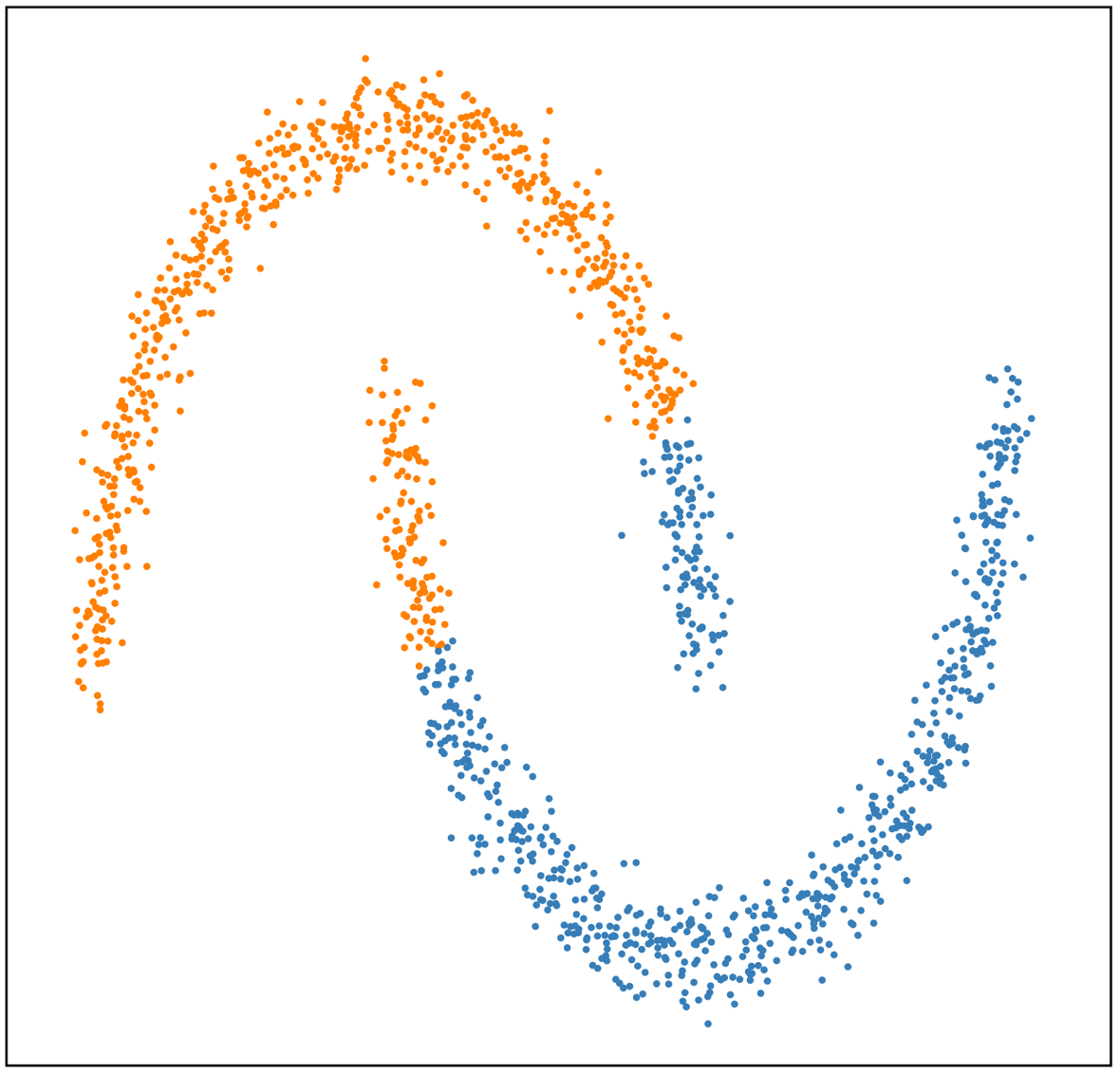
k-means fail: different sizes or densities



k-means fail: non-globular

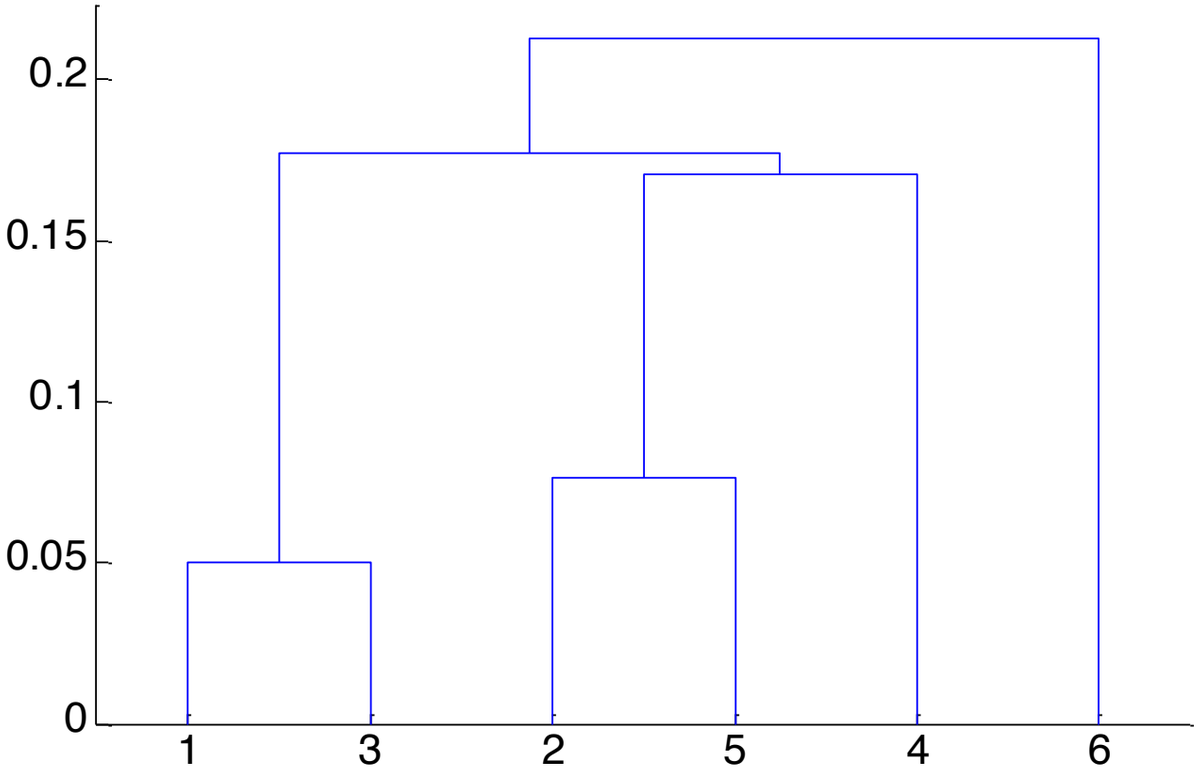


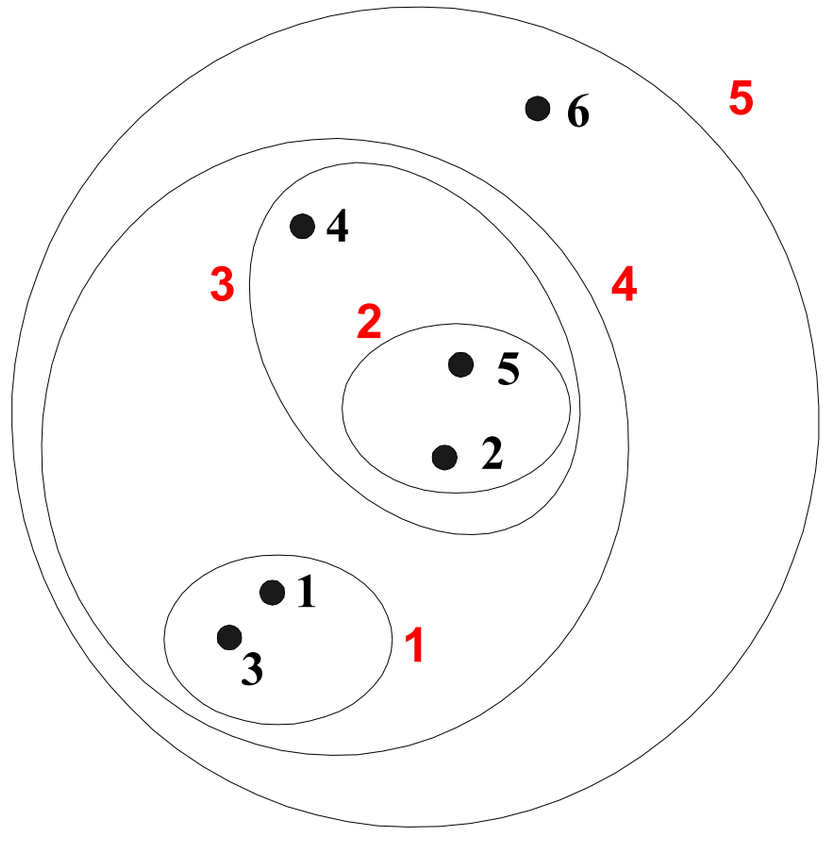
Ideal



k-means

Hierarchical clustering





Slide credits: Alona Fyshe

Hierarchical clustering

Two approaches:

Top-down (“Divisive clustering”)

All points initially in the same cluster. In each iteration, split a cluster (using some rule).

Bottom-up (“Hierarchical agglomerative clustering”)

Initially, make each point its own cluster. In each iteration, merge two clusters (using some rule).

Hierarchical Agglomerative Clustering (HAC)

Initialize each cluster to be a singleton: $C_i = \{i\}$ for $i = 1, 2, \ldots, n$

$A = \{1, 2, \ldots, n\}$

while $|A| > 1$

Select two most similar clusters: $(j, k) = \underset{j,k \in A}{\text{argmin}}\, d(C_j, C_k)$

Merge cluster k into cluster j : $A = A \setminus \{k\}$

$C_j = C_j \cup C_k$

Hierarchical Agglomerative Clustering (HAC)

How to measure dissimilarity between clusters?

For now, assume that for two examples, we measure dissimilarity as Euclidean distance:

$$d(x, x') = \|x - x'\| = \sqrt{\sum_{j=1}^{d} (x_j - x'_j)^2}$$

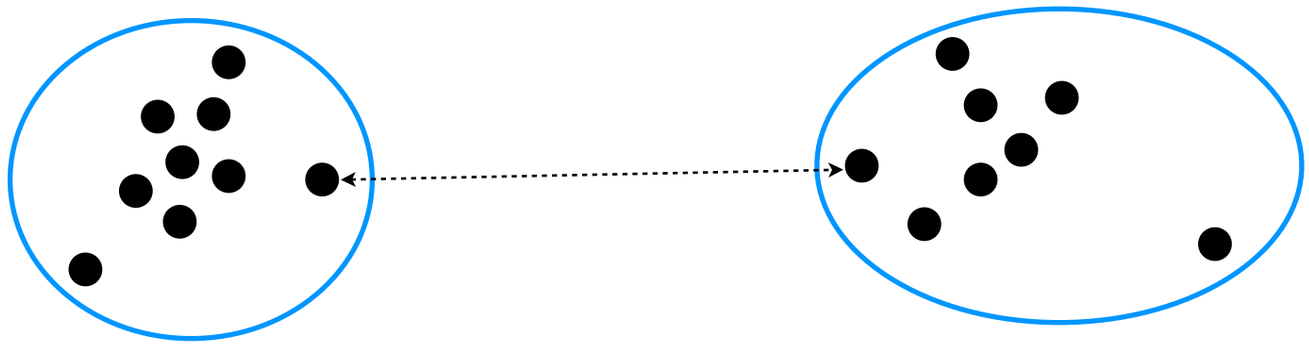
(but HAC easily handles other dissimilarity measures)

3 typical ways:

Single linkage Complete linkage Average linkage

Single linkage

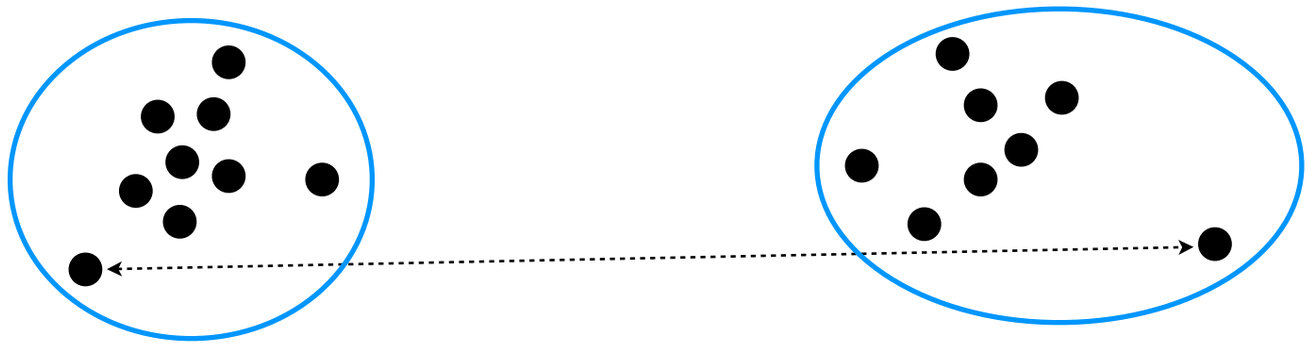
$$d_{\mathsf{SL}}(C, C') = \min_{x \in C, x' \in C'} d(x, x')$$



Measure distance between two nearest points

Complete linkage

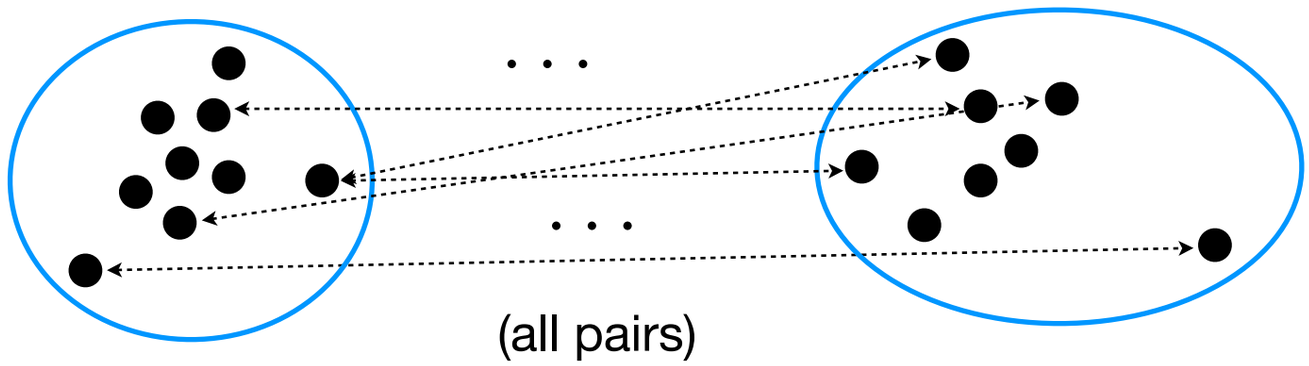
$$d_{\mathrm{CL}}(C, C') = \max_{x \in C, x' \in C'} d(x, x')$$



Measure distance between two farthest points

Average linkage

$$d_{\mathsf{AL}}(C, C') = \frac{1}{|C||C'|} \sum_{x \in C, x' \in C'} d(x, x')$$



Measure average distance between all pairs of points

Hierarchical Agglomerative Clustering (HAC)

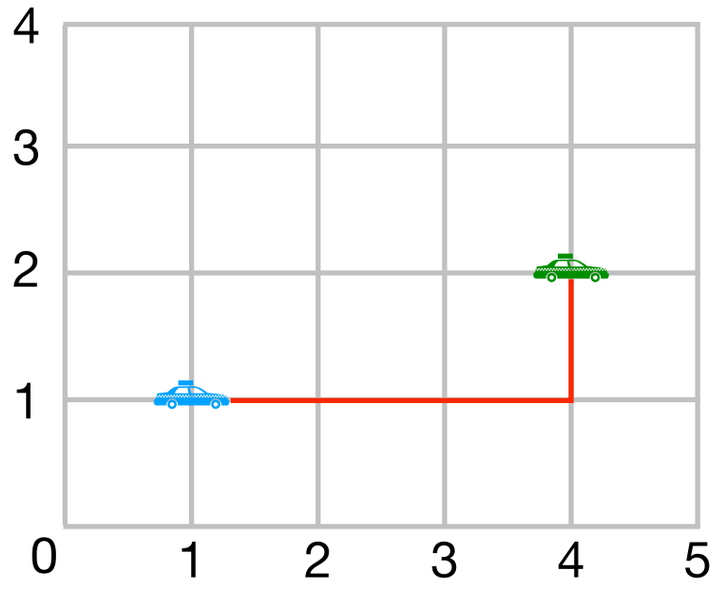
Lloyd's algorithm is fundamentally based on squared Euclidean distance

Why? Think of the form of update in the M-step

HAC can naturally handle other types of dissimilarity between individual points

Example: Manhattan distance: $d(x, x') = \sum_{j=1}^{d} |x_j - x'_j|$

"taxicab geometry"

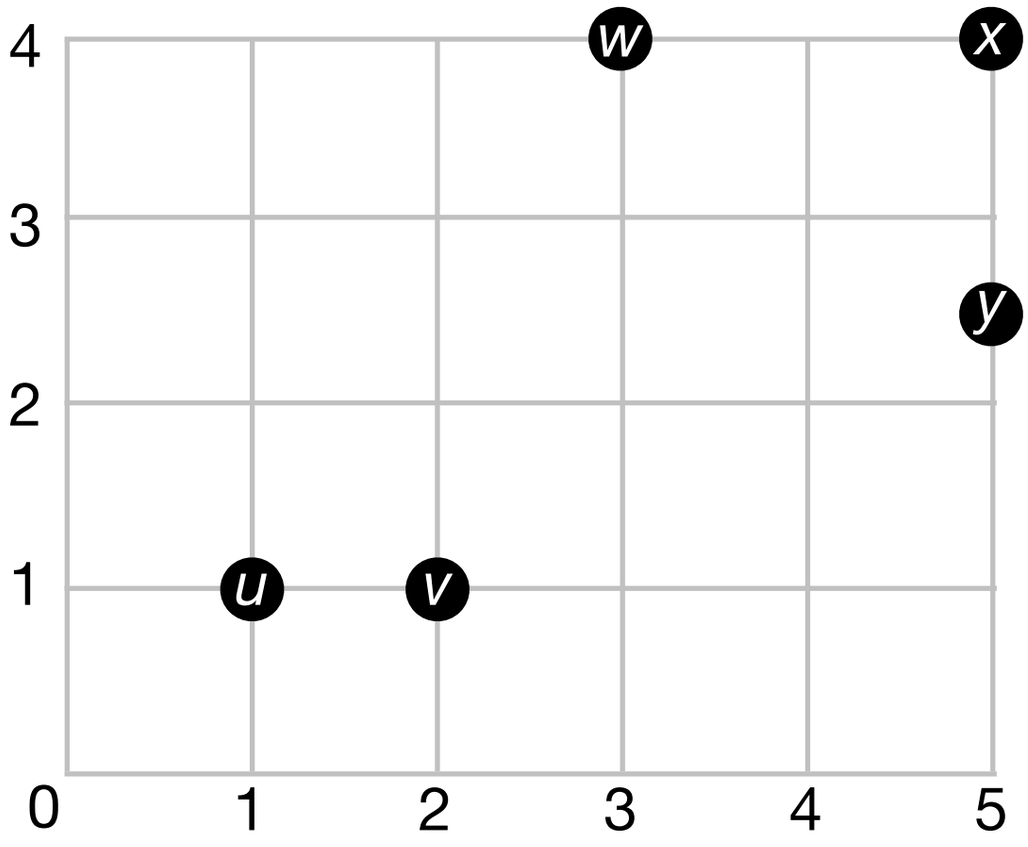


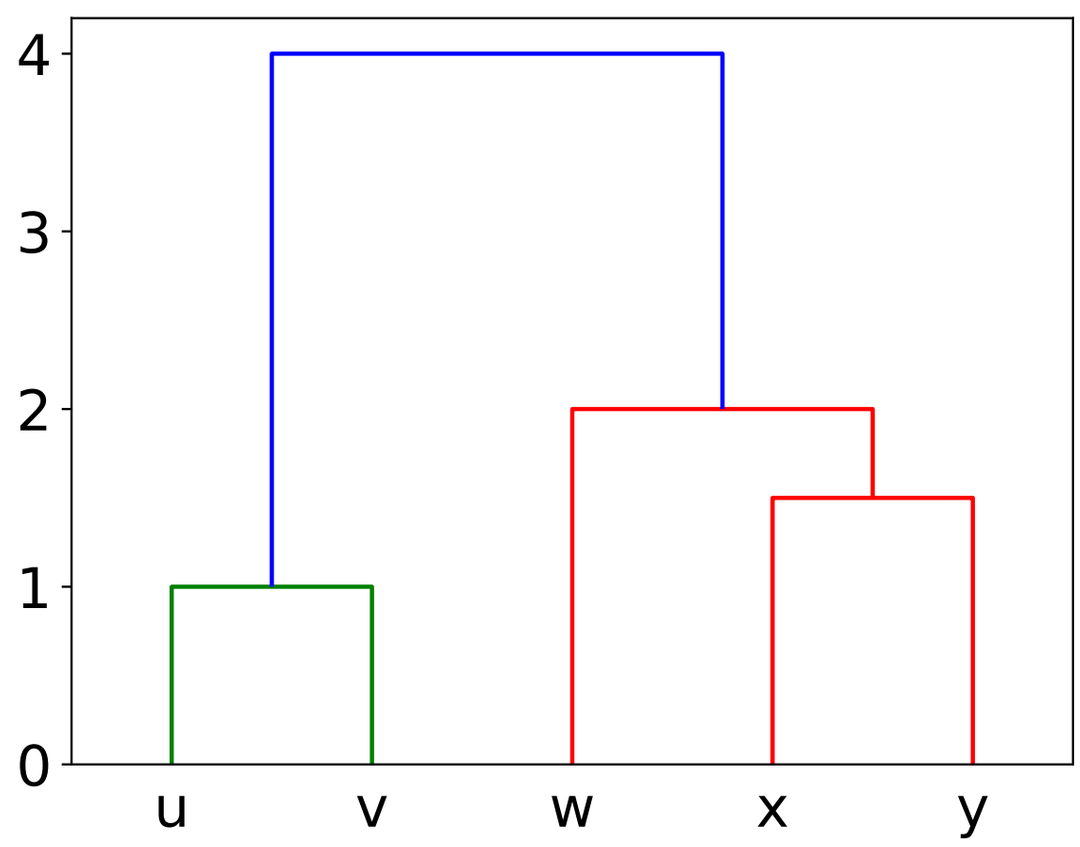
$d(\text{🚕}, \text{🚕}) = 3 + 1 = 4$

Dendrograms

Dendrogram: interpretable, binary tree depicting hierarchical structure

Height of a node indicates the dissimilarity between its two child nodes





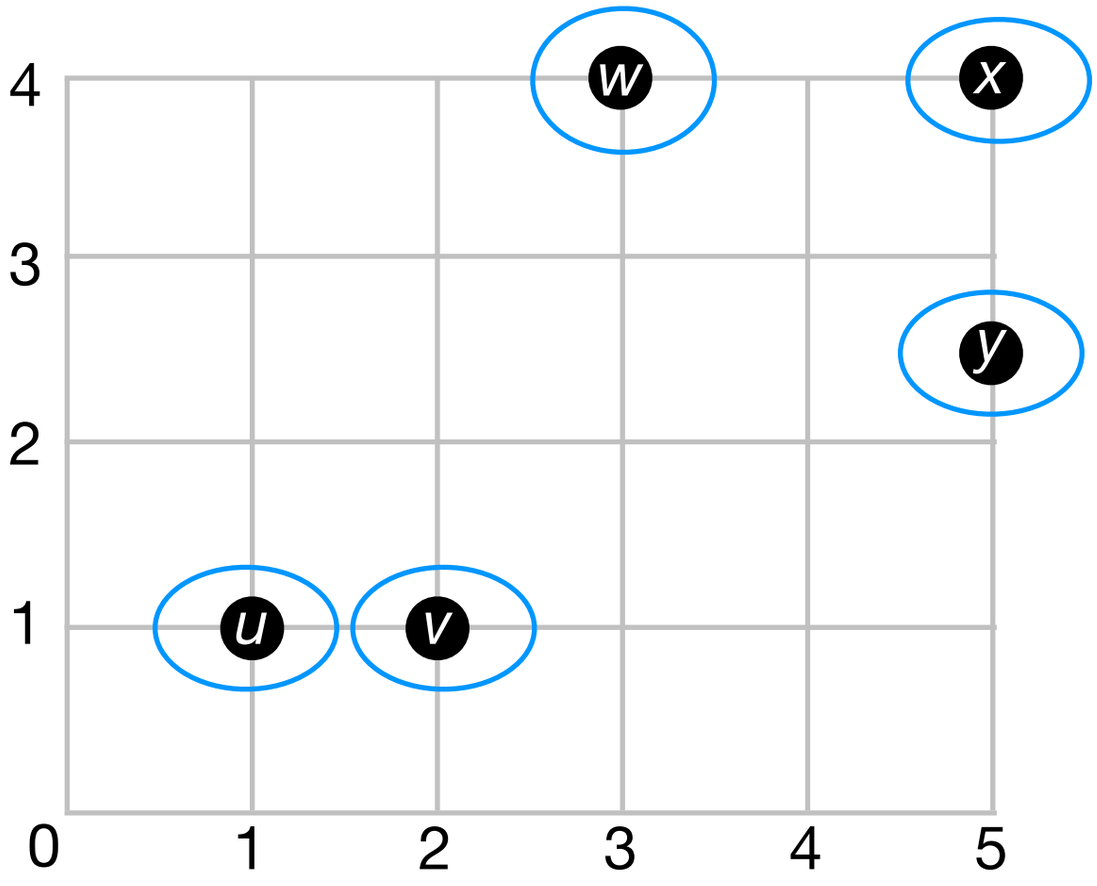
(based on single linkage HAC with Manhattan distance)

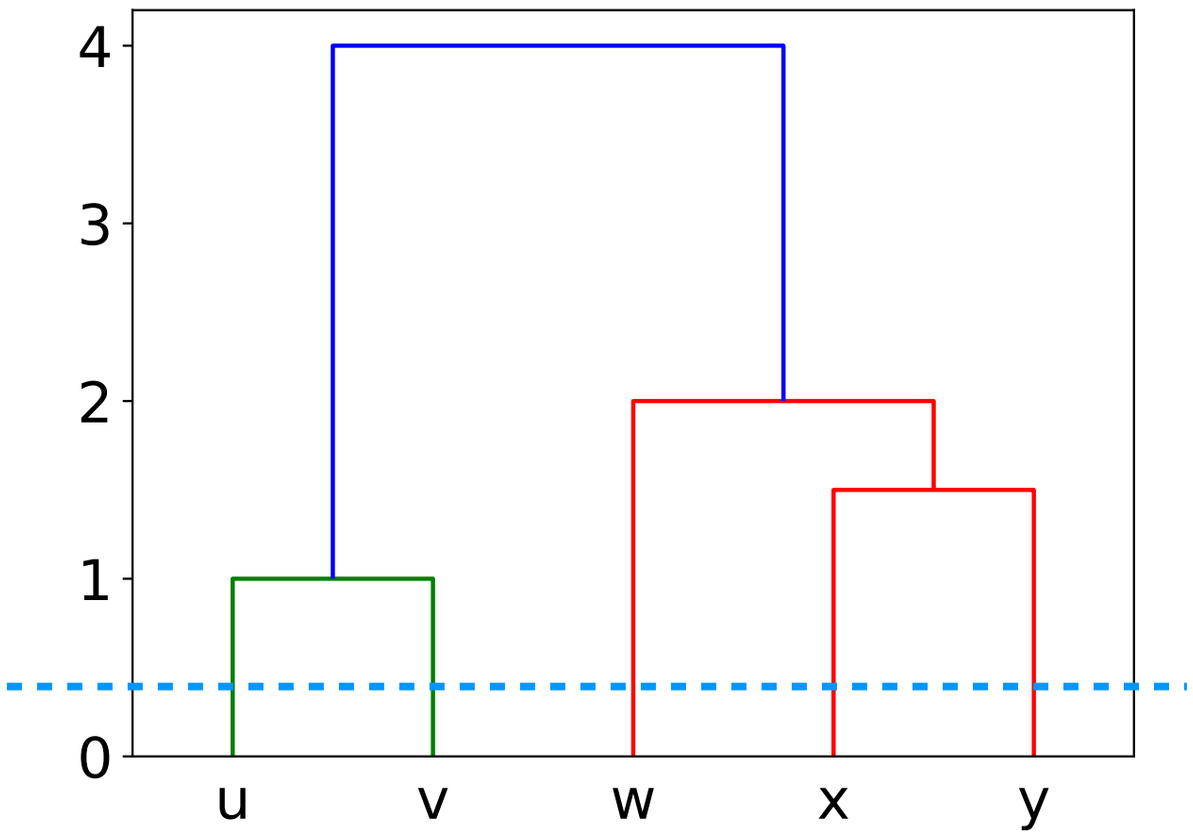
Dendrograms

Dendrogram: interpretable, binary tree depicting hierarchical structure

Height of a node indicates the dissimilarity between its two child nodes

Any horizontal cut in the dendrogram corresponds to a clustering





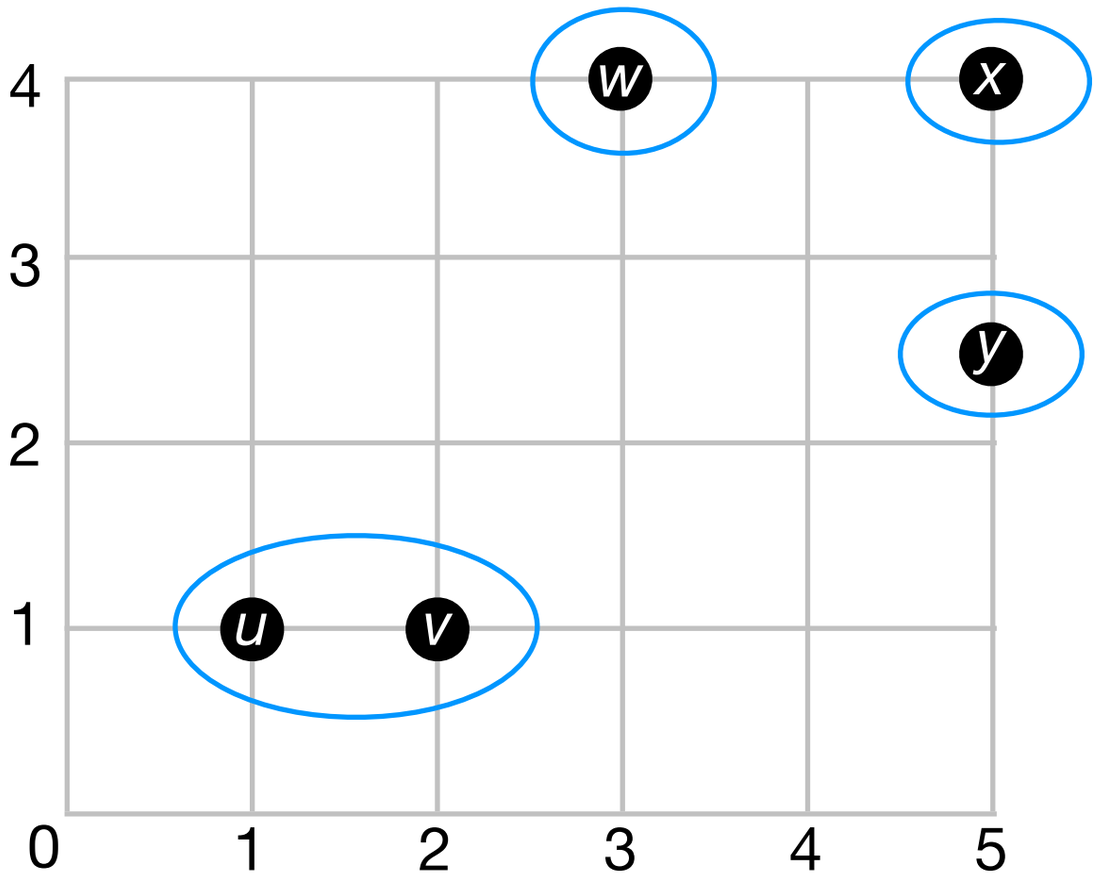
(based on single linkage HAC with Manhattan distance)

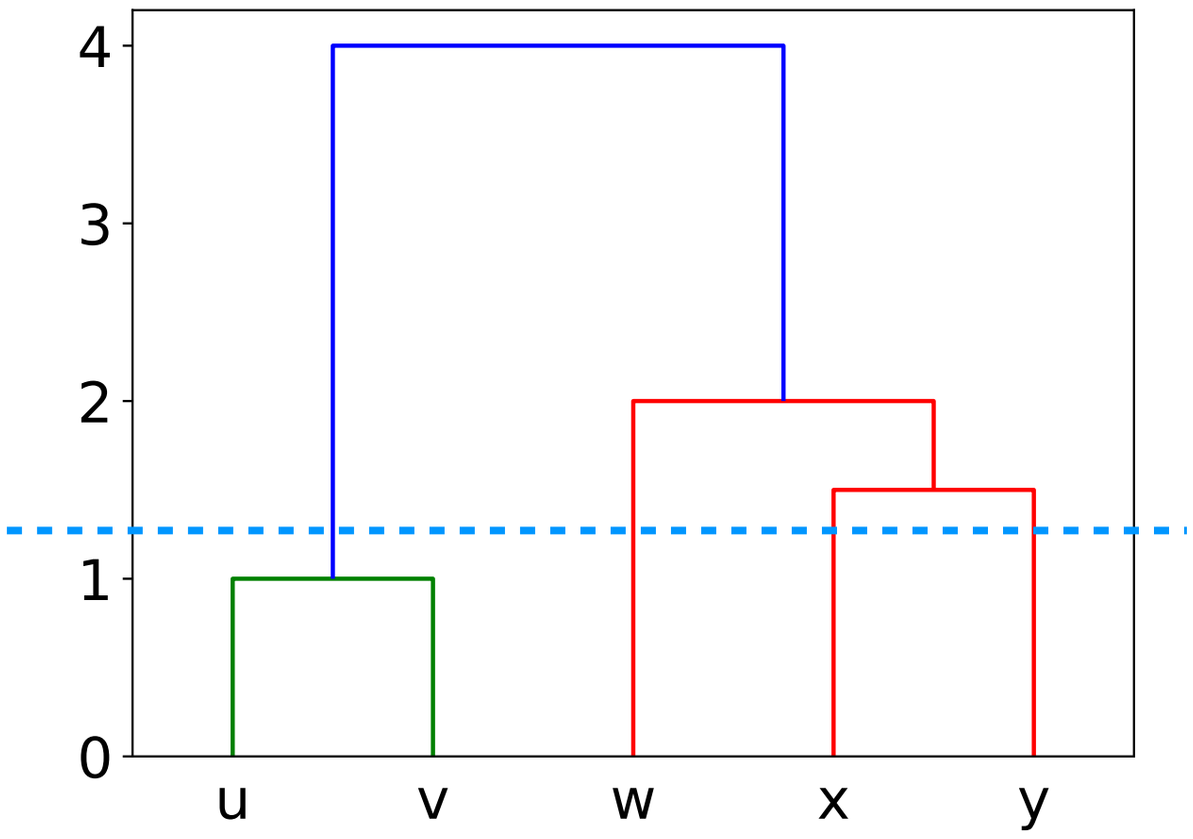
Dendrograms

Dendrogram: interpretable, binary tree depicting hierarchical structure

Height of a node indicates the dissimilarity between its two child nodes

Any horizontal cut in the dendrogram corresponds to a clustering





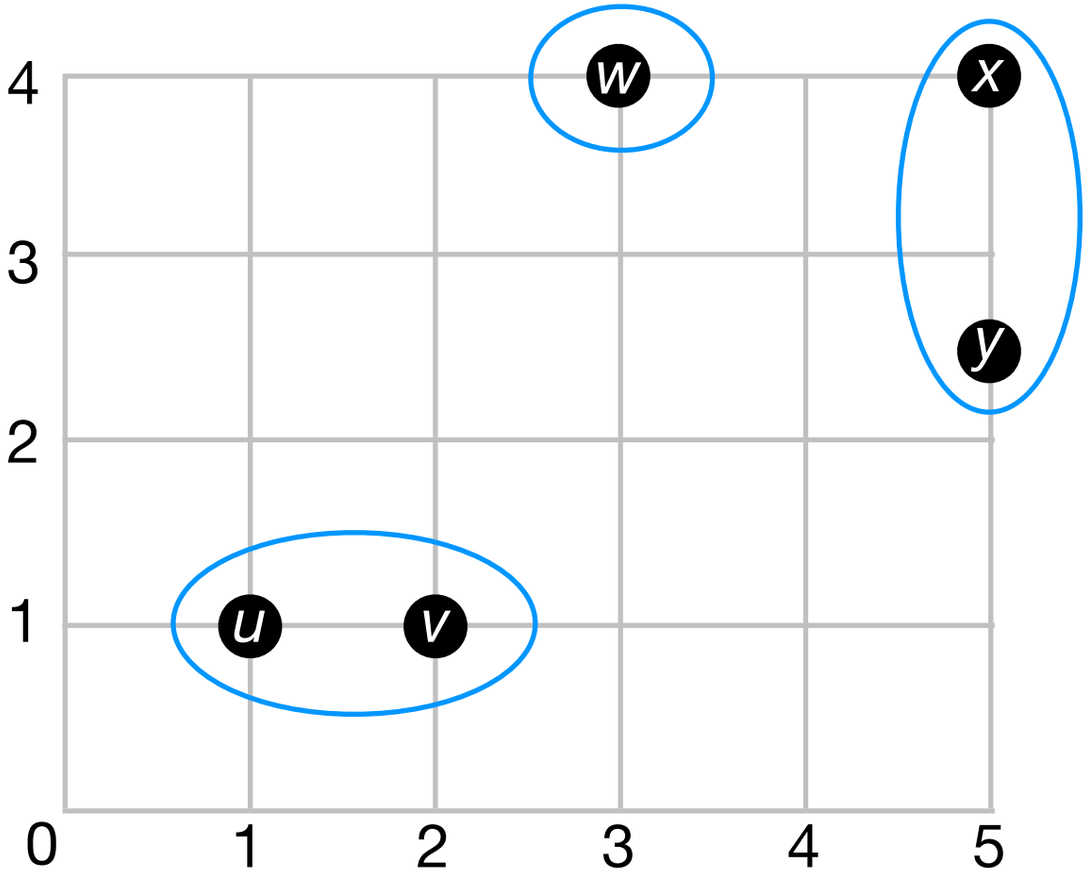
(based on single linkage HAC with Manhattan distance)

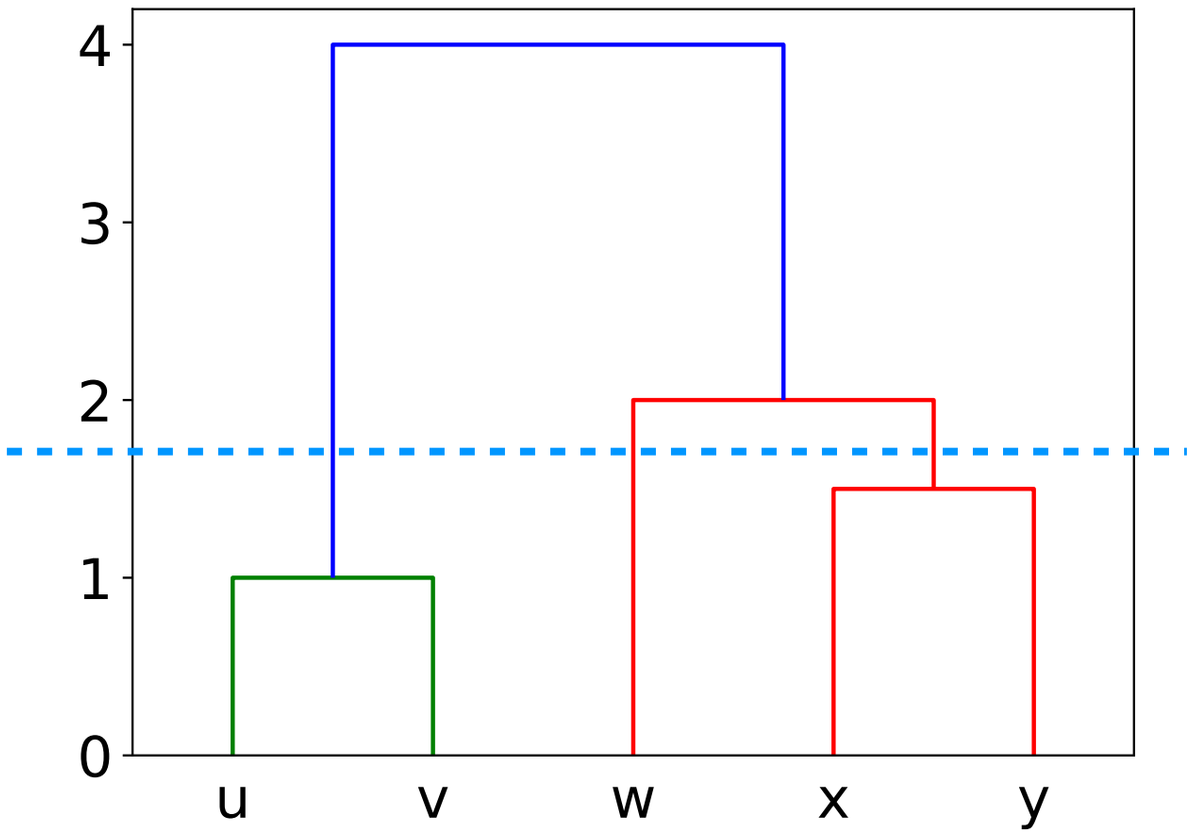
Dendrograms

Dendrogram: interpretable, binary tree depicting hierarchical structure

Height of a node indicates the dissimilarity between its two child nodes

Any horizontal cut in the dendrogram corresponds to a clustering





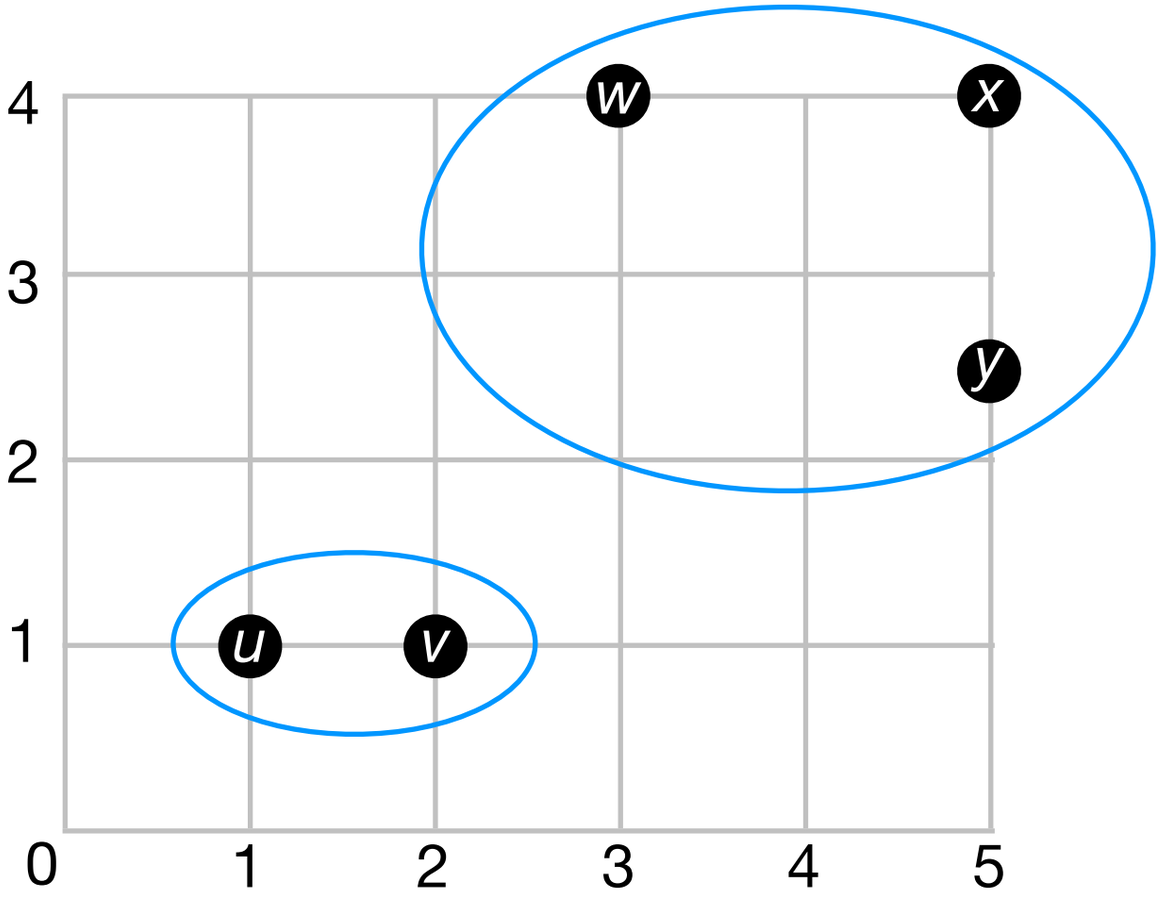
(based on single linkage HAC with Manhattan distance)

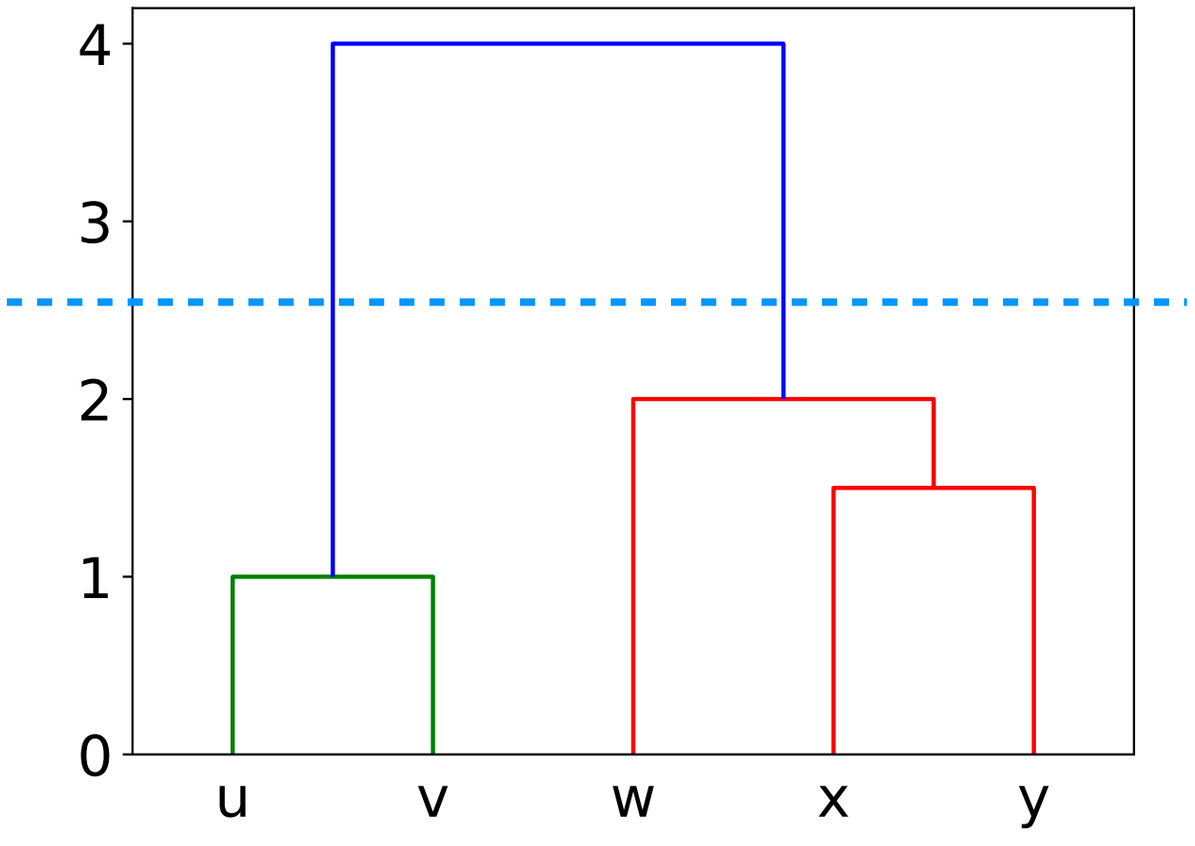
Dendrograms

Dendrogram: interpretable, binary tree depicting hierarchical structure

Height of a node indicates the dissimilarity between its two child nodes

Any horizontal cut in the dendrogram corresponds to a clustering





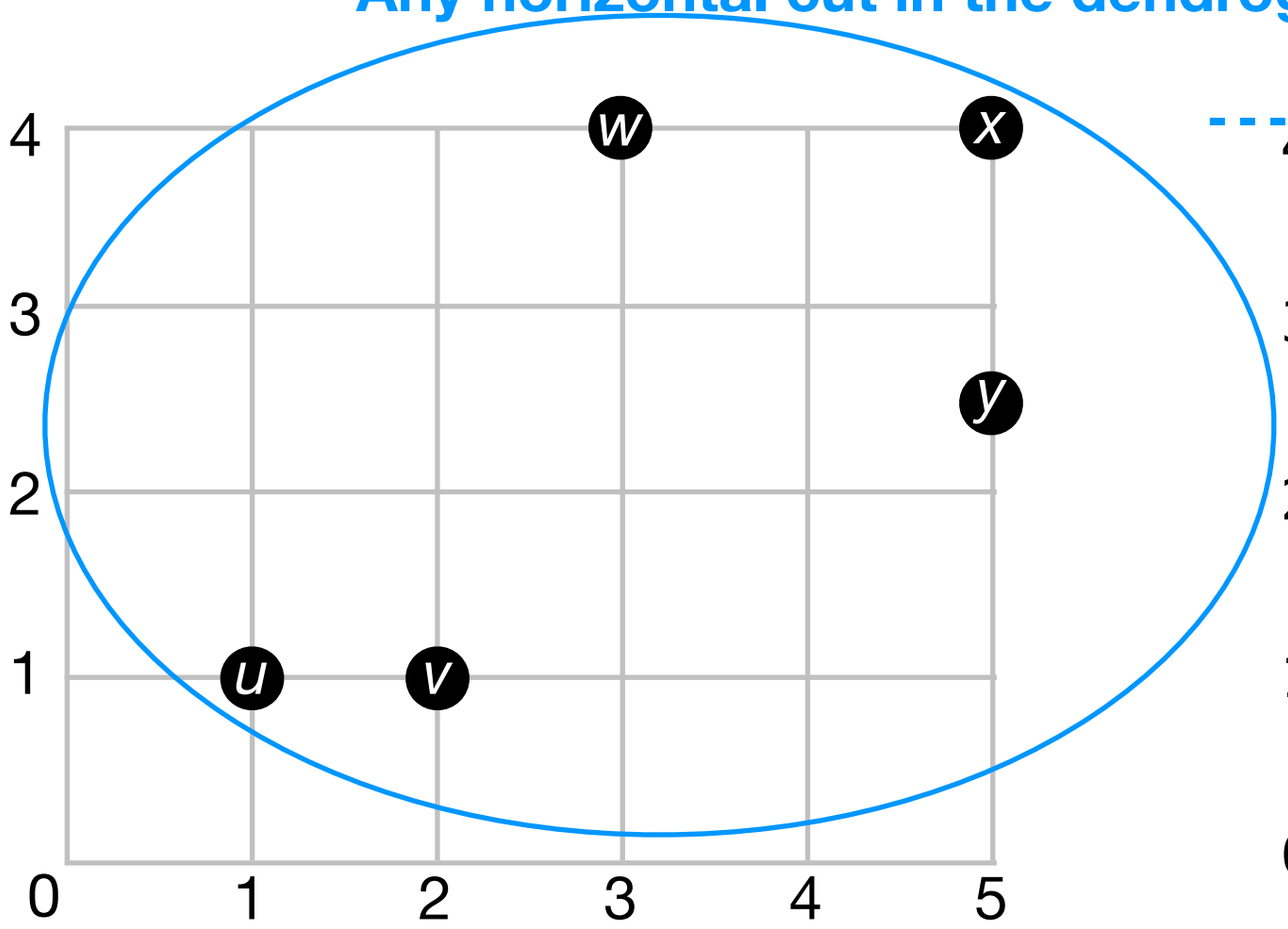
(based on single linkage HAC with Manhattan distance)

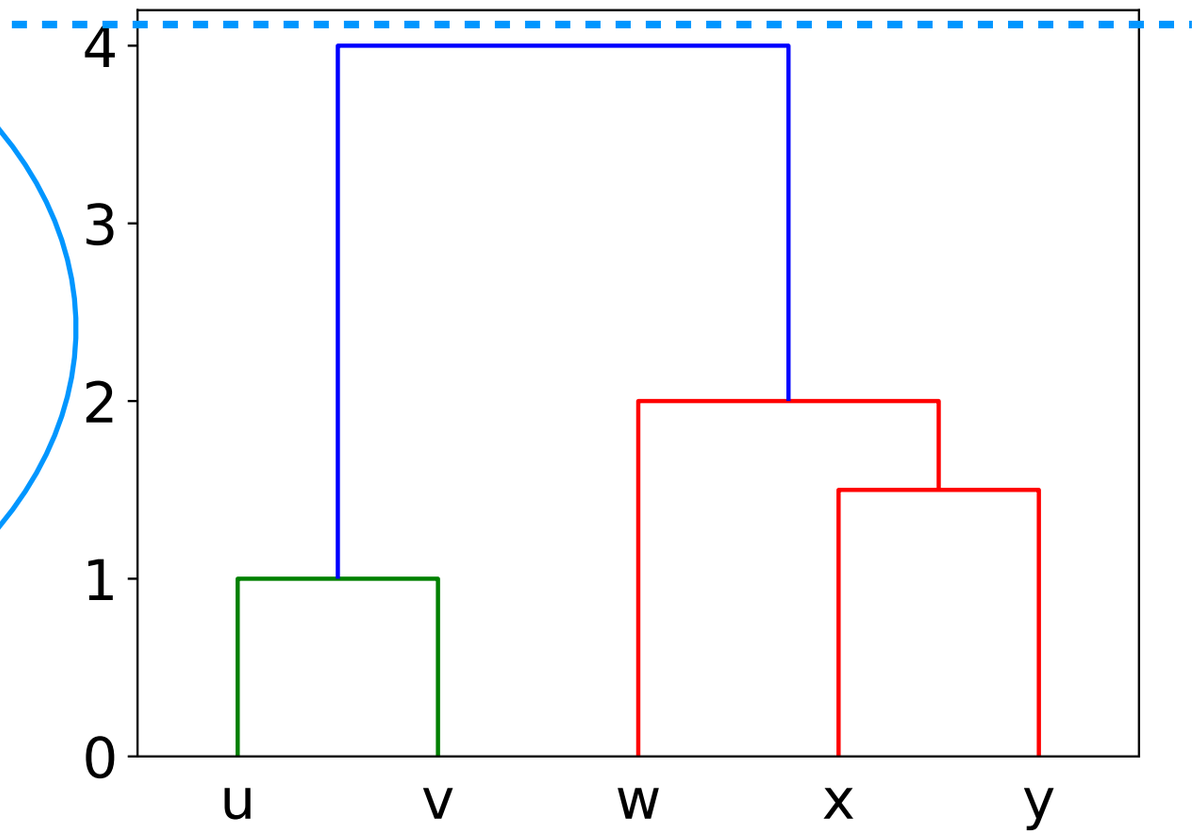
Dendrograms

Dendrogram: interpretable, binary tree depicting hierarchical structure

Height of a node indicates the dissimilarity between its two child nodes

Any horizontal cut in the dendrogram corresponds to a clustering





(based on single linkage HAC with Manhattan distance)

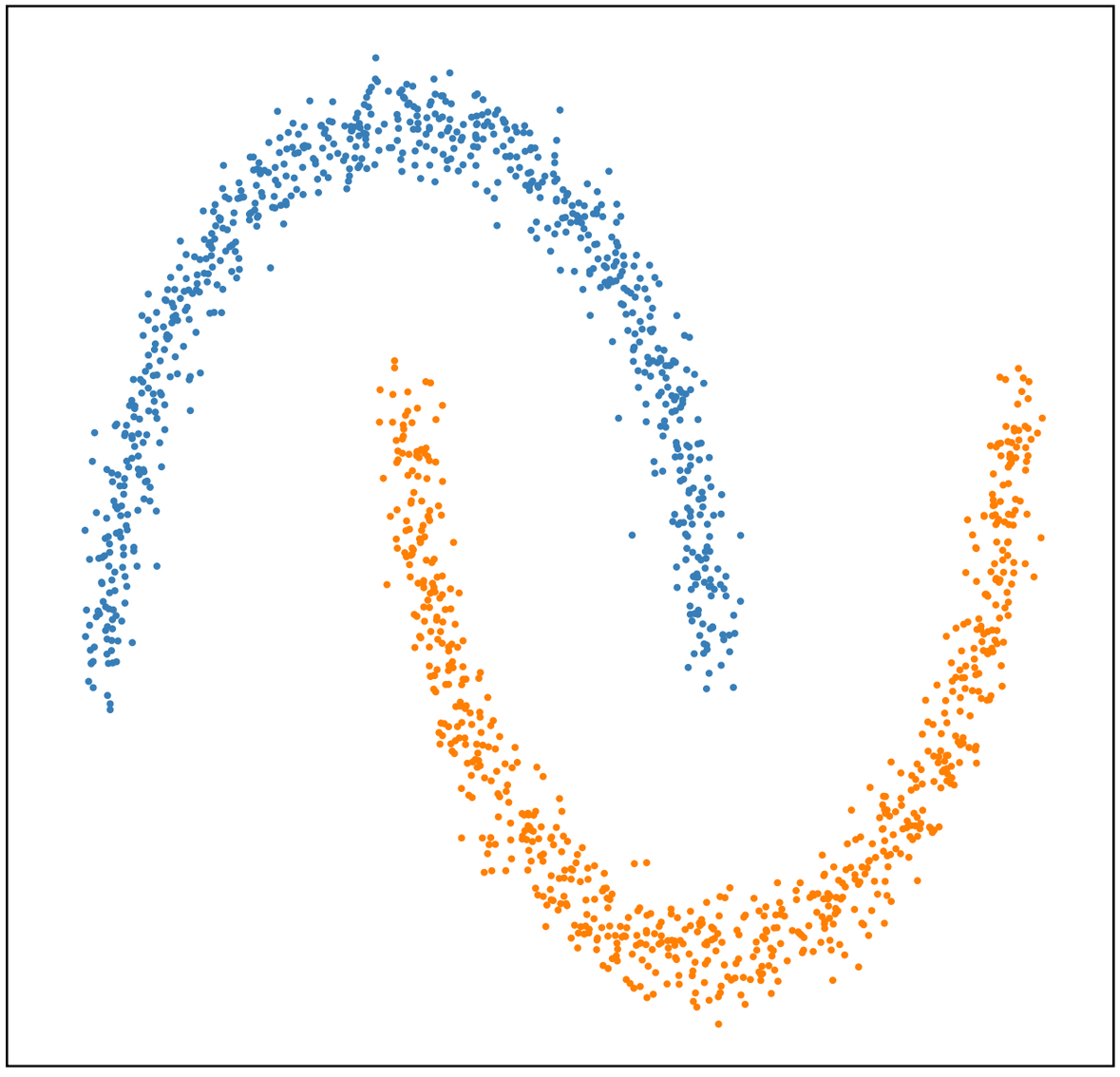
Different linkage methods

Each linkage method has its own advantages and disadvantages

Single linkage can handle non-globular clusters well

This clustering was produced using single linkage HAC

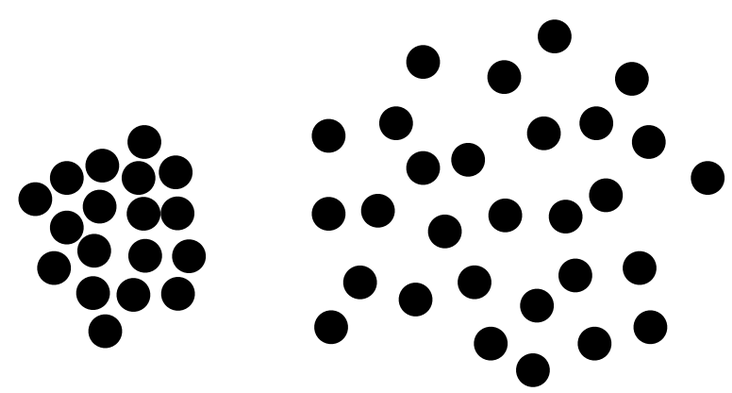
(…that said, try and think of examples where single linkage might give a horrible clustering)



Different linkage methods

Each linkage method has its own advantages and disadvantages

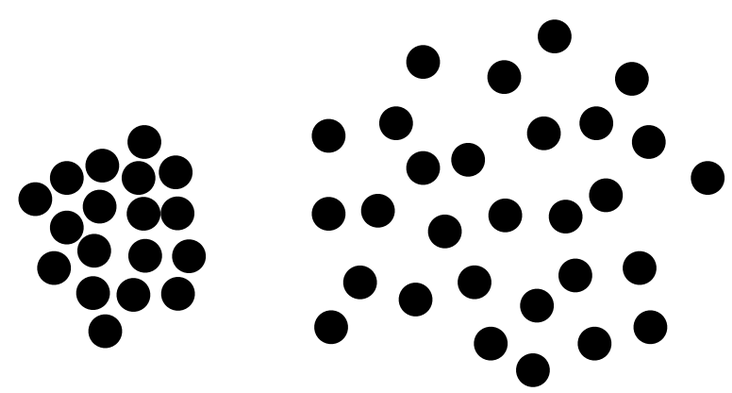
Complete linkage can break more-spread out clusters



Different linkage methods

Each linkage method has its own advantages and disadvantages

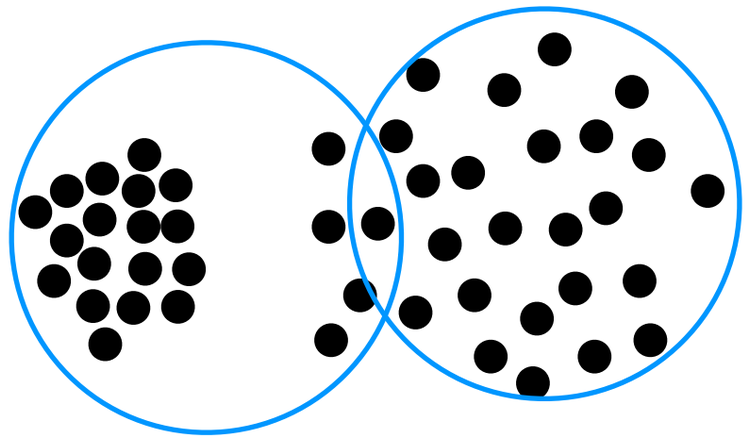
Complete linkage can break more-spread out clusters



Different linkage methods

Each linkage method has its own advantages and disadvantages

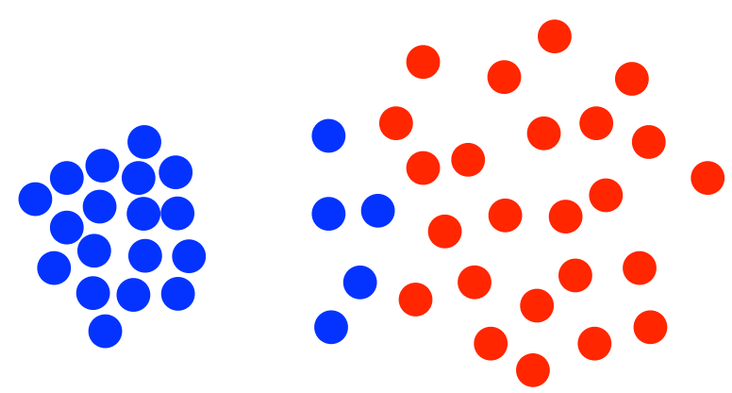
Complete linkage can break more-spread out clusters



Different linkage methods

Each linkage method has its own advantages and disadvantages

Complete linkage can break more-spread out clusters



Different linkage methods

Each linkage method has its own advantages and disadvantages

Average linkage is a compromise between the two (closer to Lloyd's algorithm, in terms of the objective)