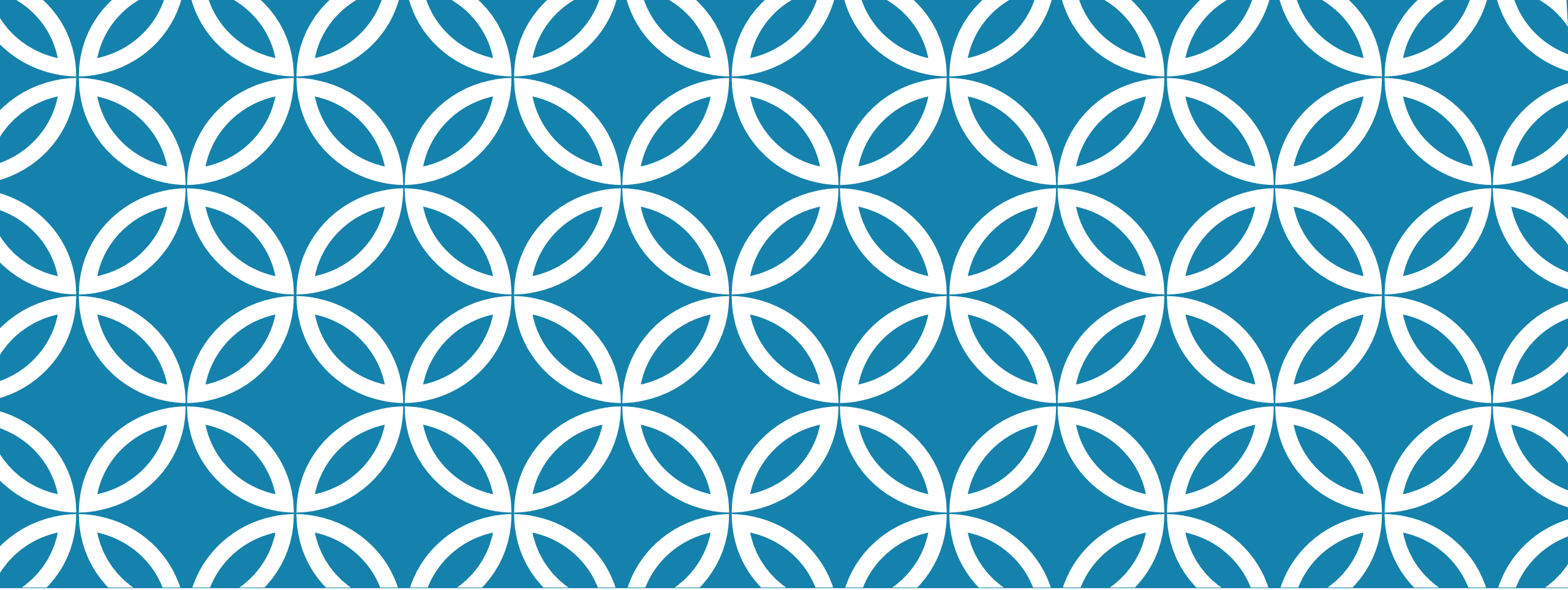


CS5228 LECTURE 2: CLUSTERING

Bryan Hooi

School of Computing

National University of Singapore

Slide credits: Chris von der Weth, Wang Wei, Ng See Kiong, Wynne Hsu

ANNOUNCEMENTS

Final Project Information

- Full details will be released next week
- 2 choices: 1) Kaggle contest and 2) open-ended data-related project on a topic of your choice
- Group size: 3-4
- There will be a survey for those who want the course staff to group you into groups. The survey will have some (optional) questions about your preferred group size, choice of project, preferred working time / style etc., to help us group you more appropriately.
- Let the course staff know if you have any more questions / issues we can help with.

REVIEW: DATA PREPROCESSING

Data Quality ("Cleaning")

- Outliers
- Missing Values
- Duplicates

Aggregation

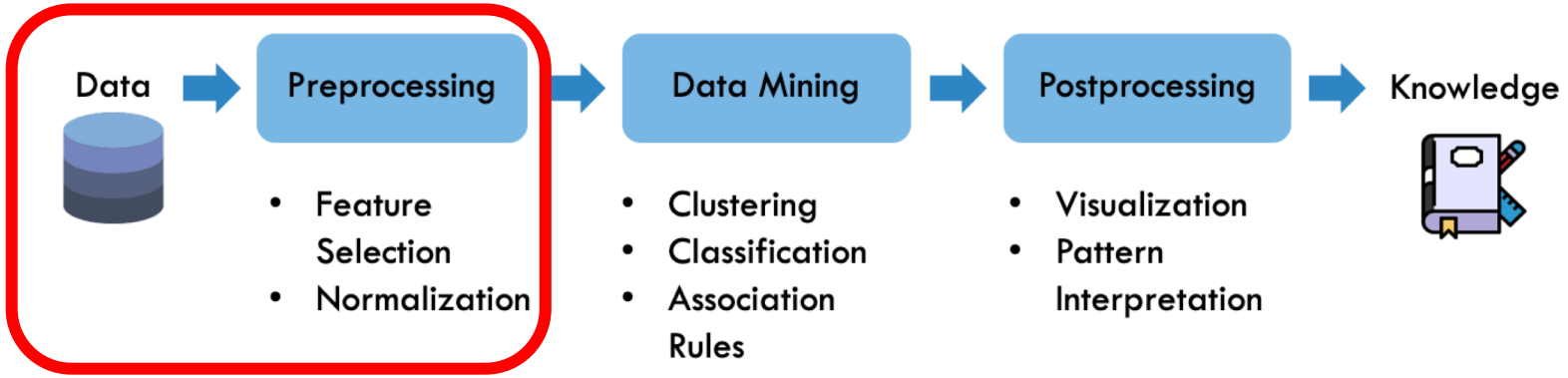
Dimensionality Reduction

- PCA

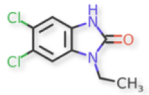
Feature Creation

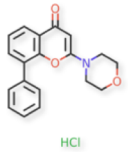
Discretization

One-Hot Encoding



REVIEW: THE DATA MINING PROCESS





(2, 0, 0, 1, …)

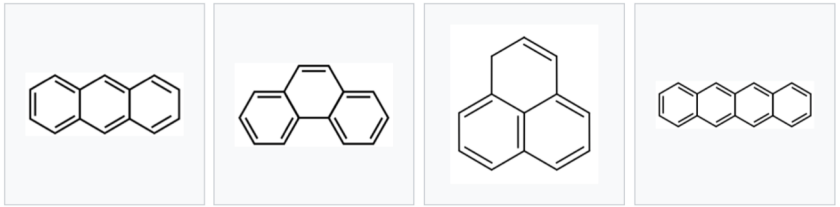
(4, 1, 0, 1, …)

if  > 1:
    toxic
else
    nontoxic

Polycyclic aromatic hydrocarbon

From Wikipedia, the free encyclopedia

Principal PAH Compounds



Data

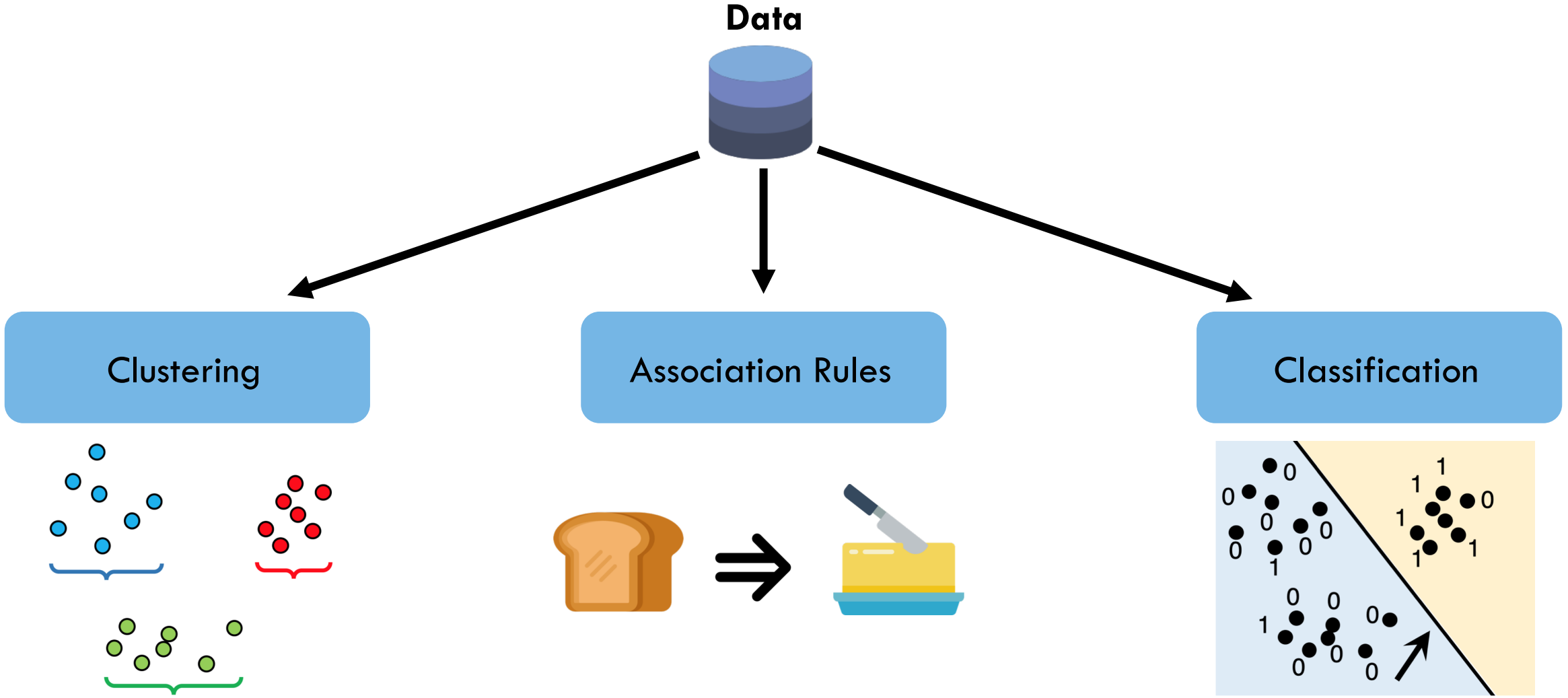
→ **Preprocessing** → **Data Mining** → **Postprocessing** → Knowledge

- Feature Selection
- Normalization

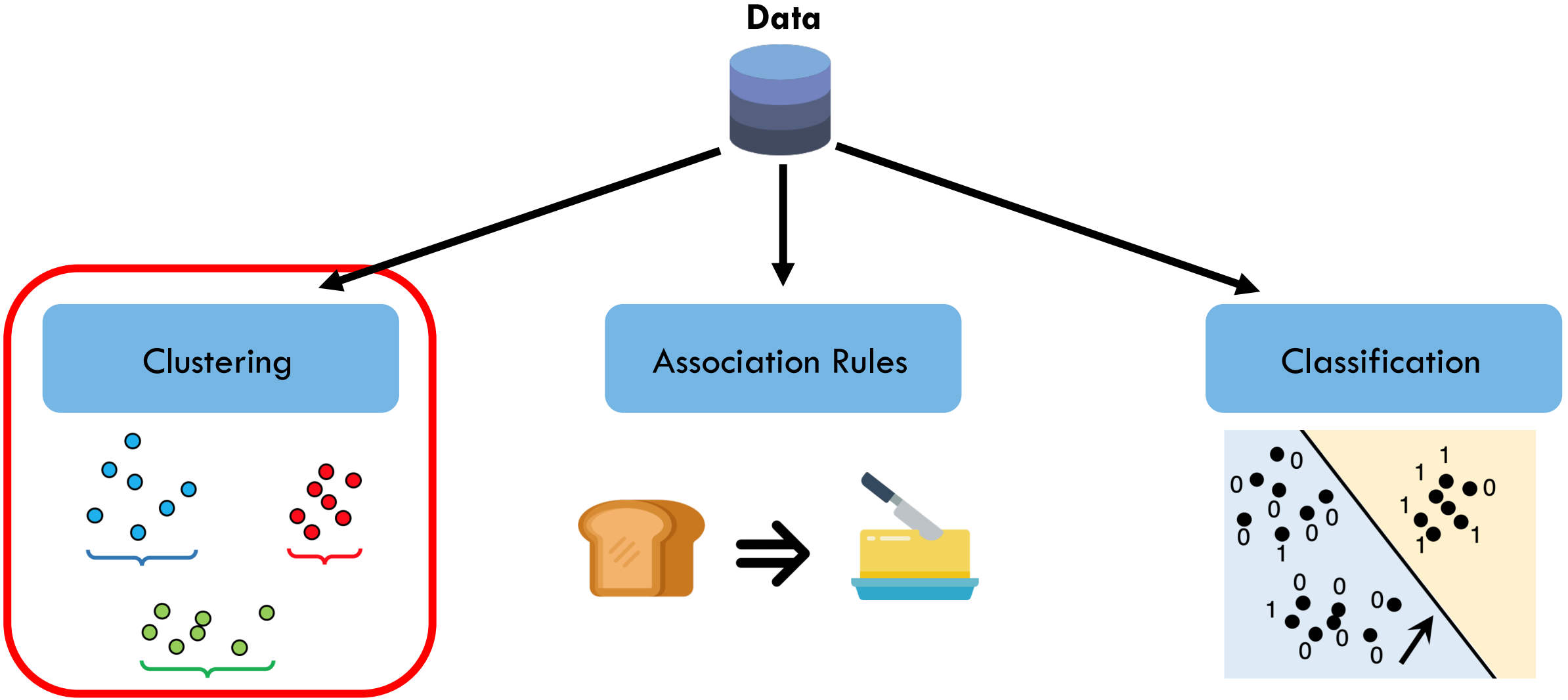
- Clustering
- Classification
- Association Rules

- Visualization
- Pattern Interpretation

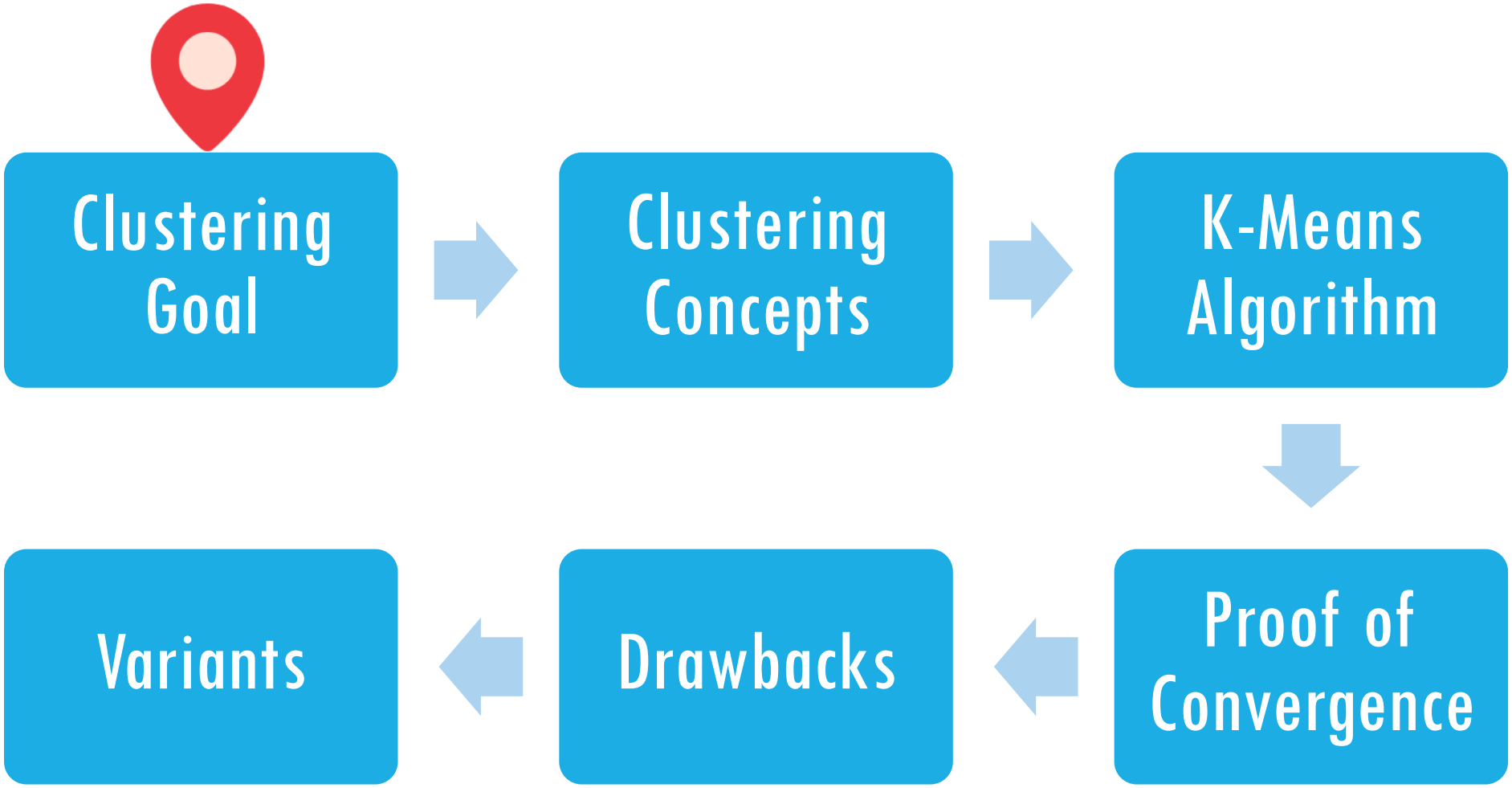
REVIEW: DATA MINING APPROACHES



REVIEW: DATA MINING APPROACHES

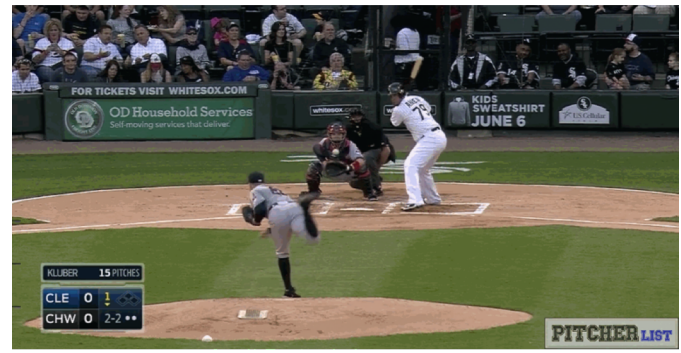


OUTLINE


Clustering Goal
Clustering Concepts
K-Means Algorithm
Proof of Convergence
Drawbacks
Variants


SFGIANTS.COM/VO
at&t
FRA
TEM

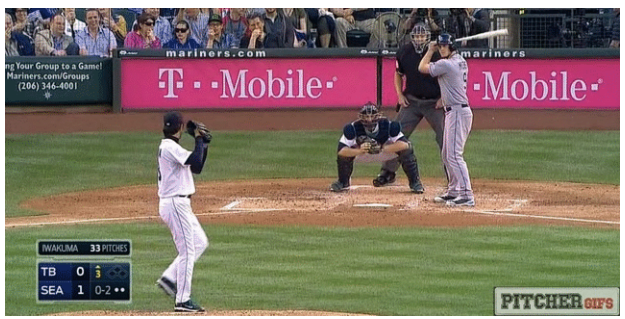

mariners.com
T-Mobile
T-Mobile
IWAKUMA 33 PITCHES
TB 0
SEA 1 0-2
PITCHER GIFS

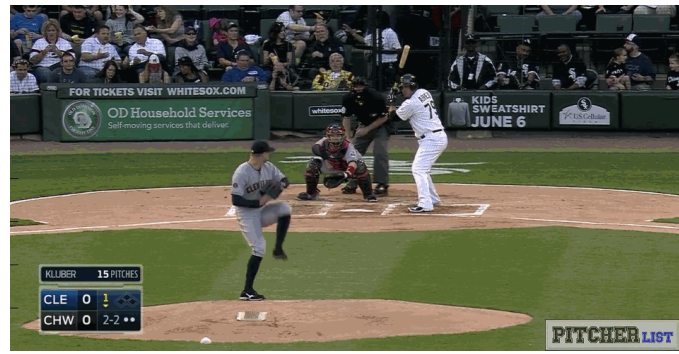

FOR TICKETS VISIT WHITESOX.COM
OD Household Services
KIDS SWEATSHIRT JUNE 6
KLUBER 15 PITCHES
CLE 0
CHW 0 2-2
PITCHER LIST


KIA 4 5 삼성
삼성카드
삼성증권
KCW
대구은행
치킨더홈
SPOTV


FOR TICKETS VISIT WHITESOX.COM
U.S. Cellular
CROSSTOWN CUP
MAY 7-8
TICKETS AVAILABLE
SCHERZER 24 PITCHES
DET 0
CHW 0 0-2
PITCHER GIFS


SFGIANTS.COM/VOT
at&t
FRA
TEMI

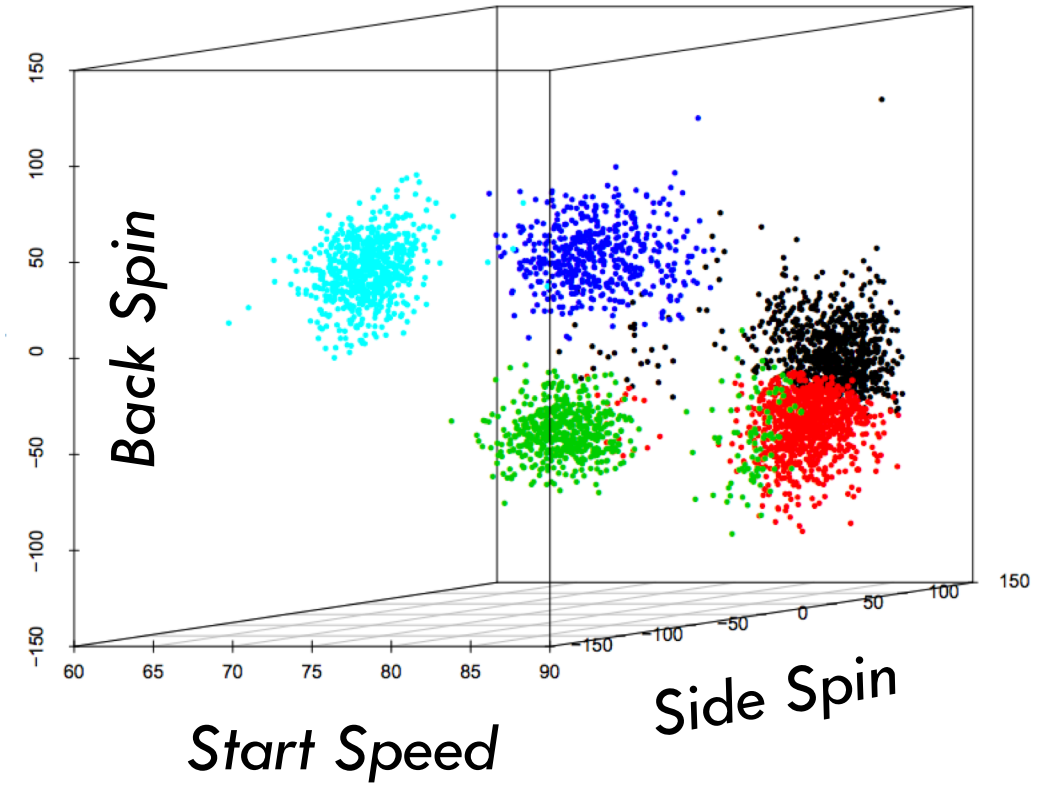

mariners.com
T-Mobile
IWAKUMA 33 PITCHES
TB 0
SEA 1 0-2
PITCHER GIFS

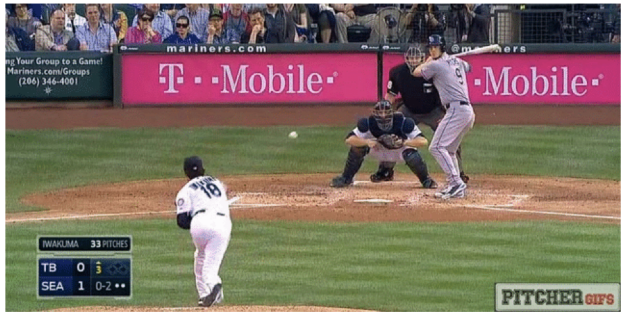

FOR TICKETS VISIT WHITESOX.COM
OD Household Services
KIDS SWEATSHIRT JUNE 6
KLUBER 15 PITCHES
CLE 0
CHW 0 2-2
PITCHER LIST

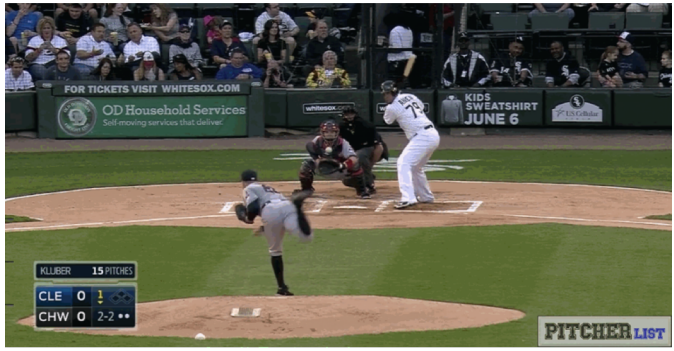

KIA 4 5 삼성
삼성카드 삼성증권
SPOTV
KCW
대구은행
치킨더훅

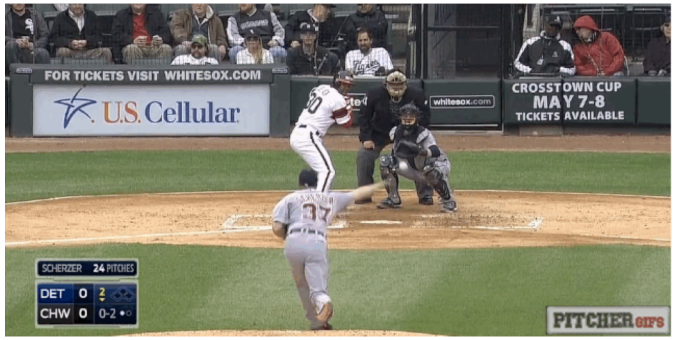

FOR TICKETS VISIT WHITESOX.COM
U.S. Cellular
CROSSTOWN CUP
MAY 7-8
TICKETS AVAILABLE
SCHERZER 24 PITCHES
DET 0
CHW 0 0-2
PITCHER GIFS

Pitches by Barry Zito



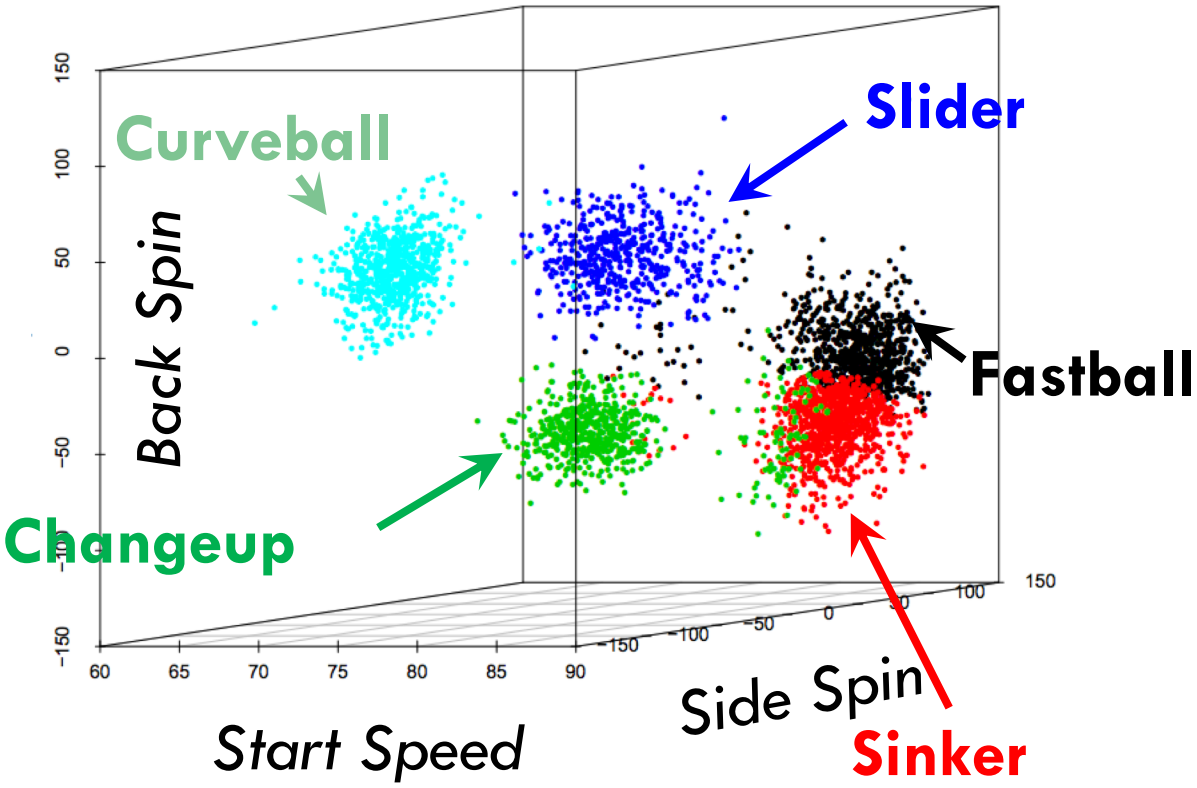


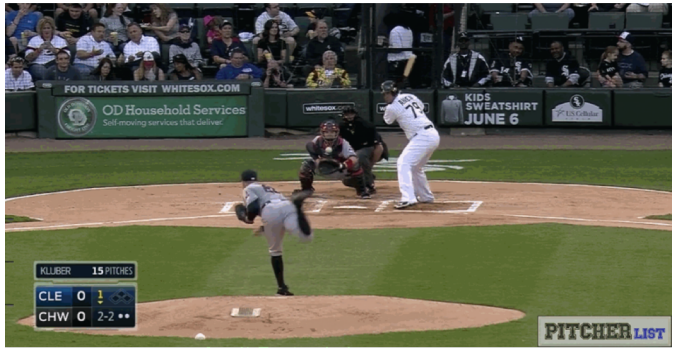


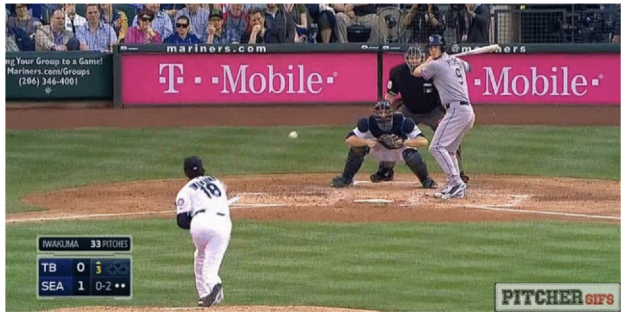


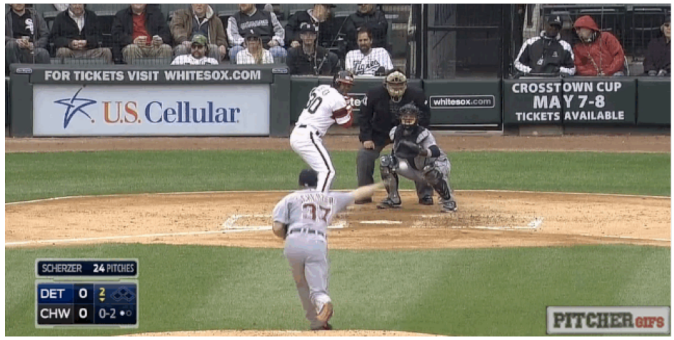
Pane, Michael A. "Trouble with the curve: identifying clusters of MLB pitchers using improved pitch classification techniques." (2013).

Pitches by Barry Zito







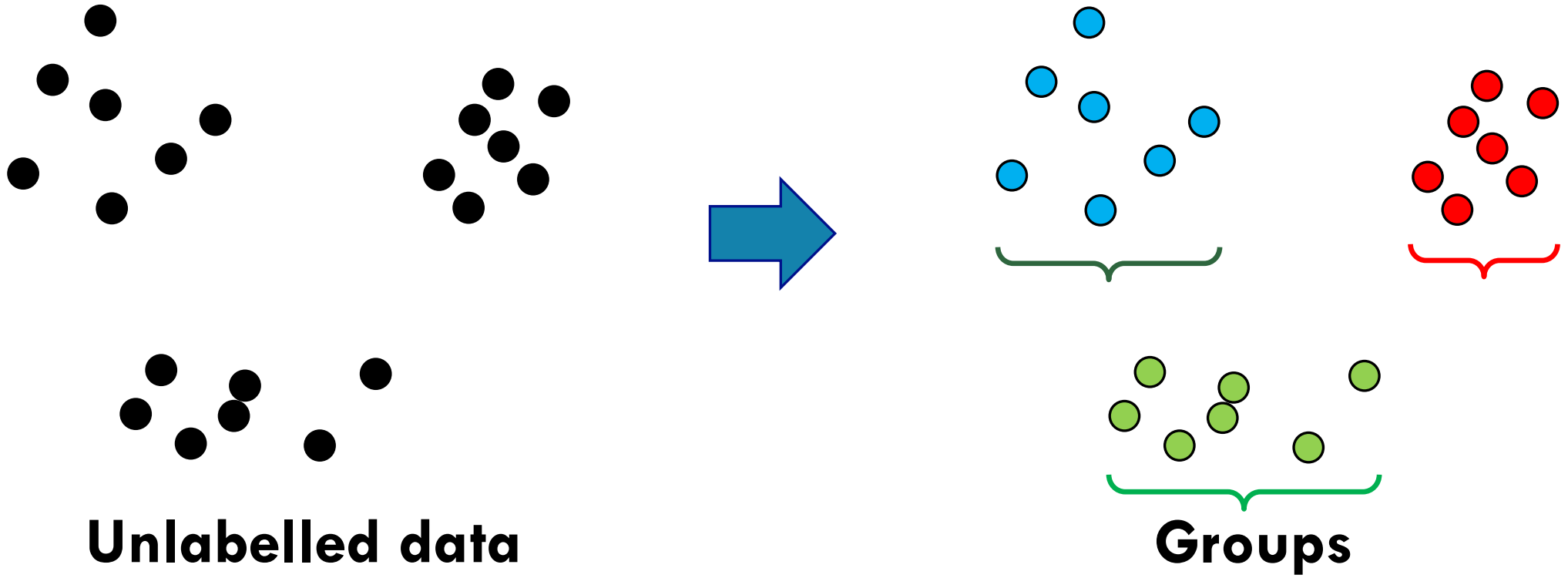


Pane, Michael A. "Trouble with the curve: identifying clusters of MLB pitchers using improved pitch classification techniques." (2013).

GOAL OF CLUSTERING

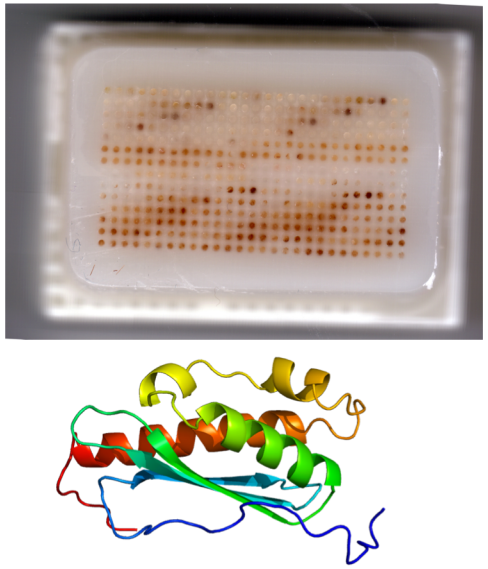
Clustering separates **unlabelled** data into **groups** of similar points.

Clusters should have high **intra-cluster similarity**, and low **inter-cluster similarity**.

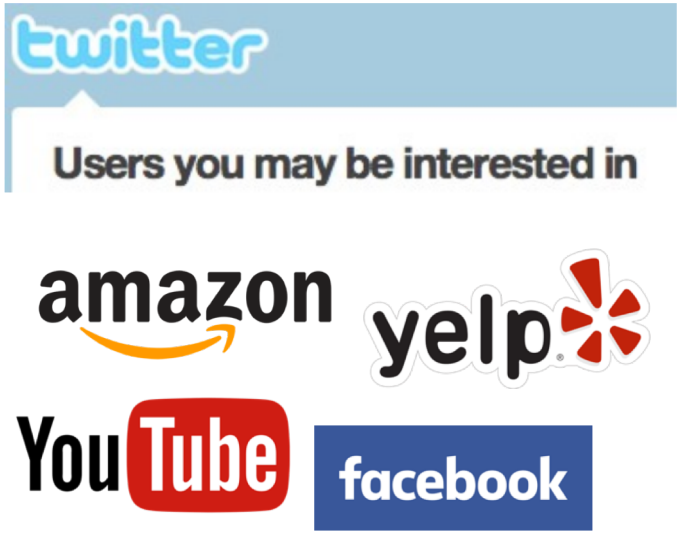


APPLICATIONS OF CLUSTERING

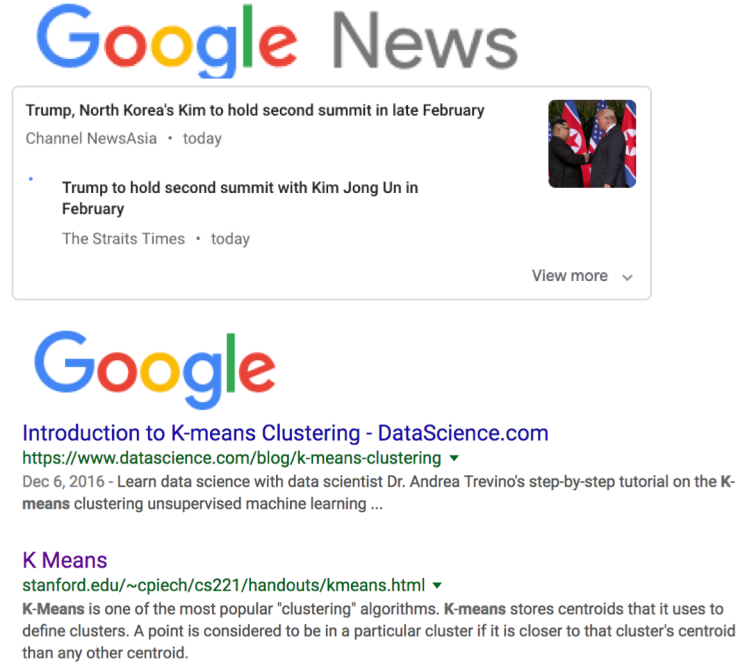
Many applications:



Microbiology: find groups of related genes (or proteins etc.)

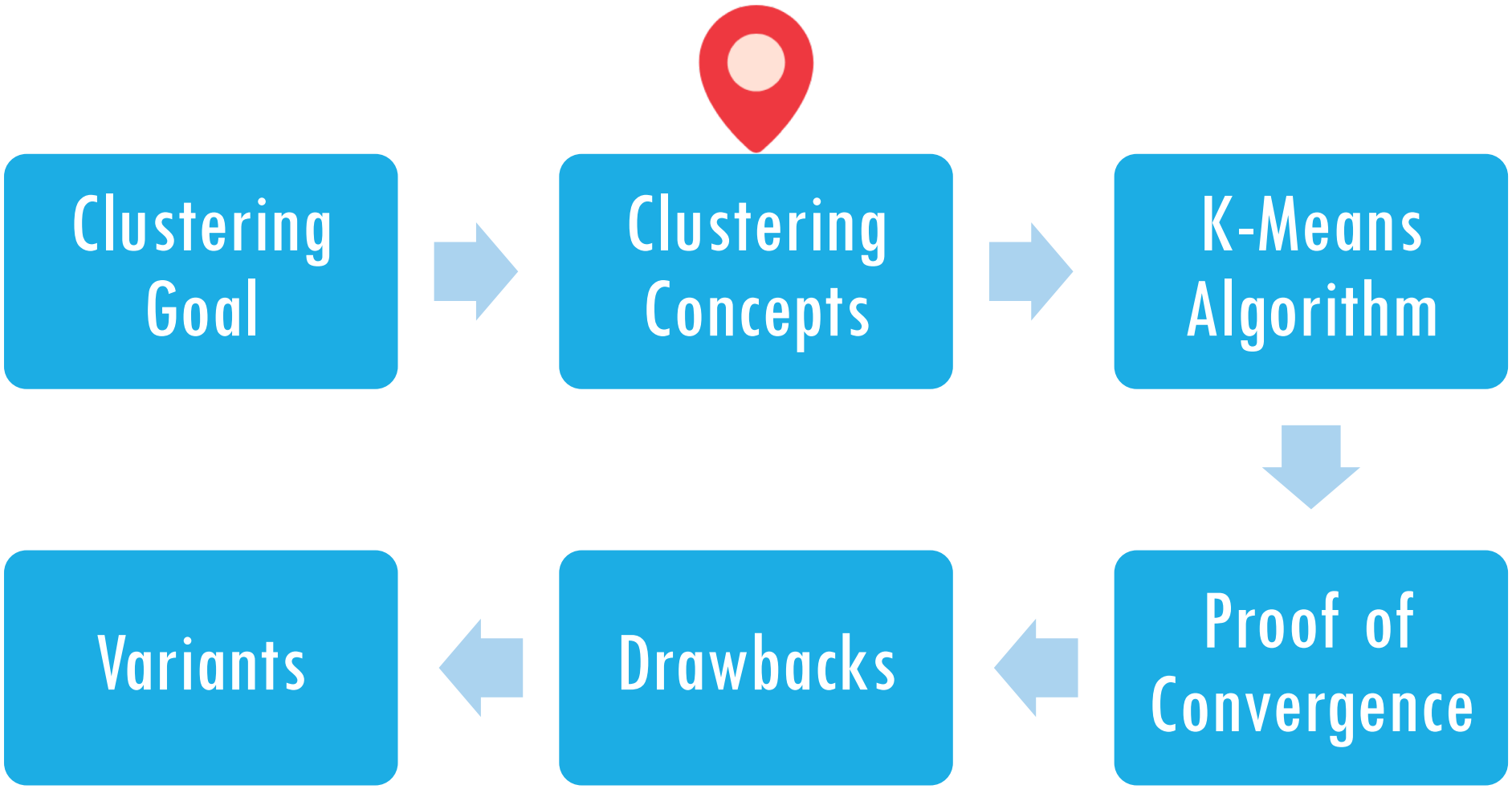


Recommendation & Social Networks: find groups of similar users



Search & Information Retrieval: grouping similar search (or news etc.) results

OUTLINE


Clustering Goal
Clustering Concepts
K-Means Algorithm
Proof of Convergence
Drawbacks
Variants

WHAT DOES SIMILARITY MEAN?

(E.g. two images may be quite similar at the **pixel level**, but not **semantically**)

(In terms of their "meaning")

DEFINITION OF A DISTANCE METRIC

Given a set S, a **distance metric** is a **nonnegative** function $d : S \times S \rightarrow \mathbb{R}^{\geq 0}$ satisfying the properties (for any a, b and c):

Set of pairs of objects from S

Nonnegative real numbers

- Uniqueness: $d(a,b) = 0 \Leftrightarrow a = b$

  Equivalent to ("if and only if")

  (We don't want there to be objects that we cannot tell apart)

- Symmetry: $d(a,b) = d(b,a)$

  (If Alice is like Bob, then Bob is like Alice)

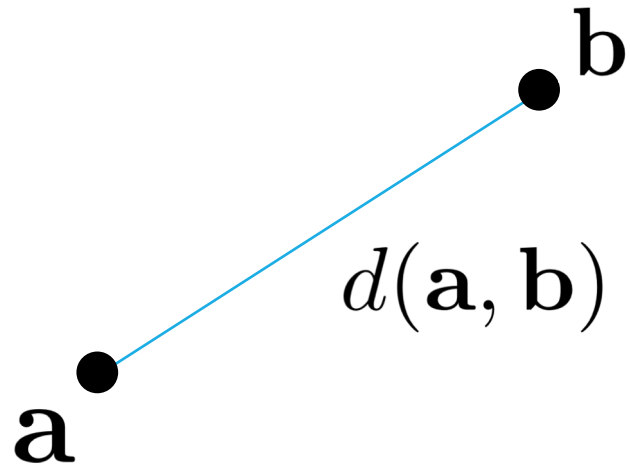
- Triangle Inequality: $d(a,b) \leq d(a,c) + d(c,b)$

  (Otherwise, Alice could be very like Carol, and Carol very like Bob, but Alice very unlike Bob)

COMMON DISTANCE / SIMILARITY METRICS

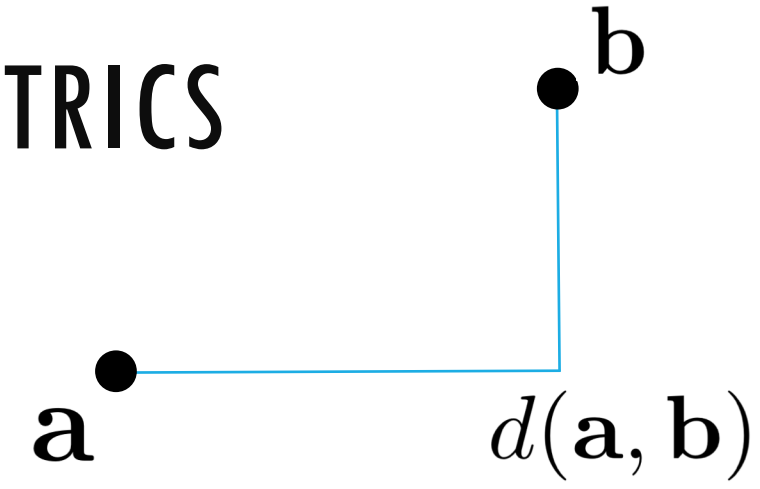
- **Euclidean distance**

$$d(\mathbf{a}, \mathbf{b}) = \|\mathbf{a} - \mathbf{b}\|_2 = \sqrt{\sum_{i=1}^{D} (a_i - b_i)^2}$$

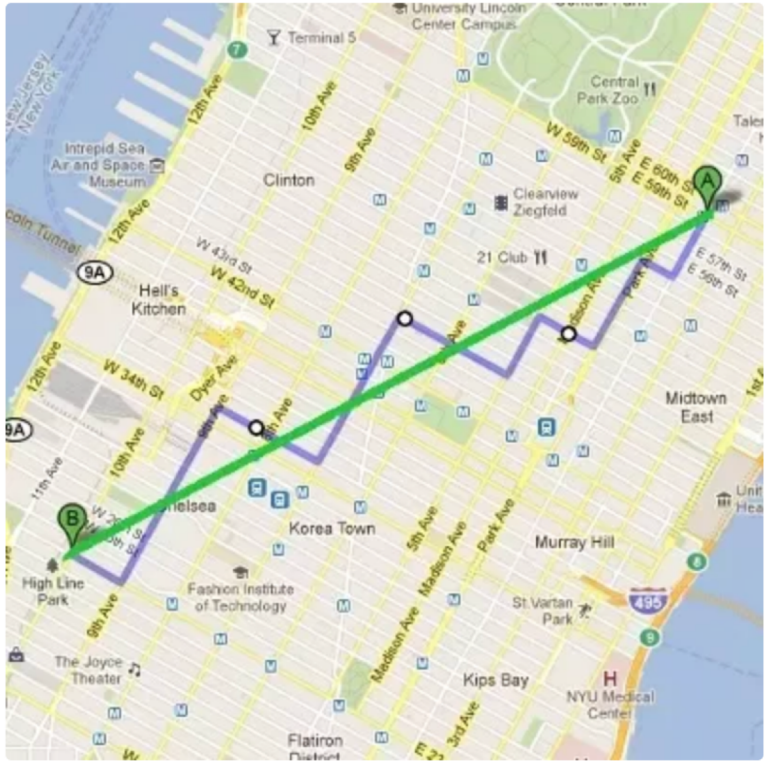


COMMON DISTANCE / SIMILARITY METRICS

- **Manhattan distance**



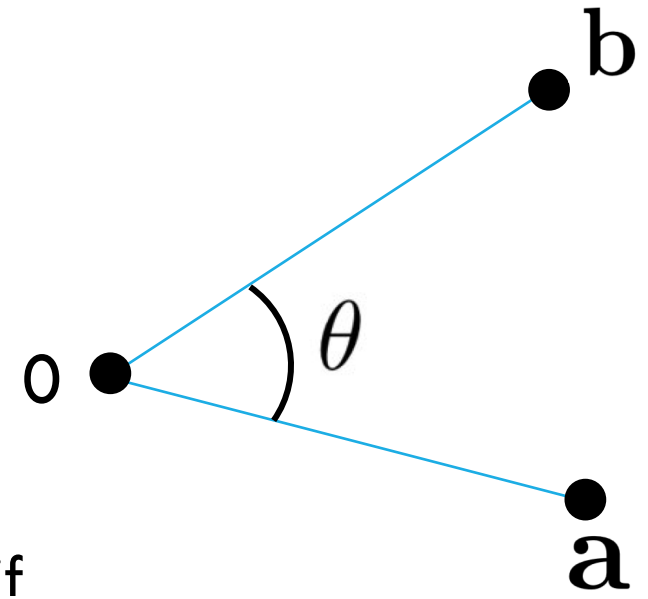
$$d(\mathbf{a}, \mathbf{b}) = \|\mathbf{a} - \mathbf{b}\|_1 = \sum_{i=1}^{D} |a_i - b_i|$$



COMMON DISTANCE / SIMILARITY METRICS

- **Cosine similarity**

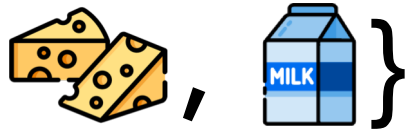
$$s(\mathbf{a}, \mathbf{b}) = \cos\theta = \frac{\mathbf{a} \cdot \mathbf{b}}{\|\mathbf{a}\| \cdot \|\mathbf{b}\|}$$



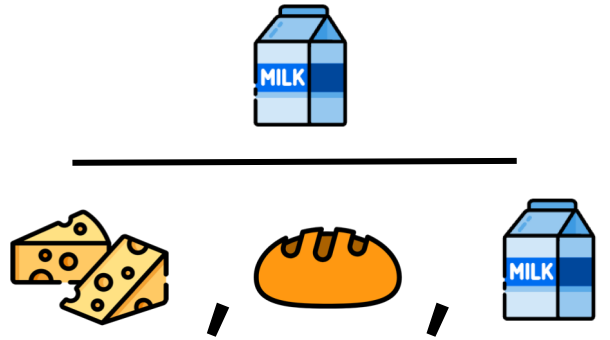
Only considers **direction:** cosine similarity doesn't change if we scale **a** or **b** (i.e. multiplying them by a constant)

COMMON DISTANCE / SIMILARITY METRICS

- **Jaccard Similarity**
  (between **sets** A and B)

A = {  } B = {  }

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

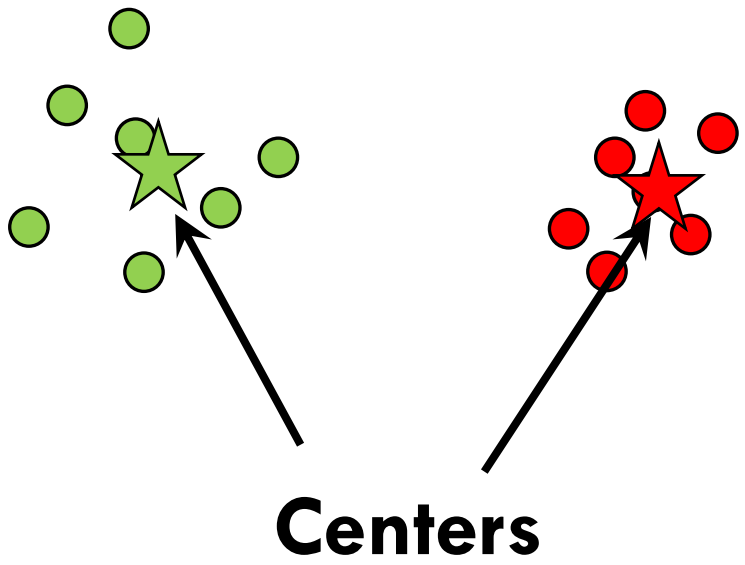
s_{Jaccard} =  = 1/3

- **Jaccard Distance**

$$d_{\text{Jaccard}}(A, B) = 1 - s_{\text{Jaccard}}(A, B)$$

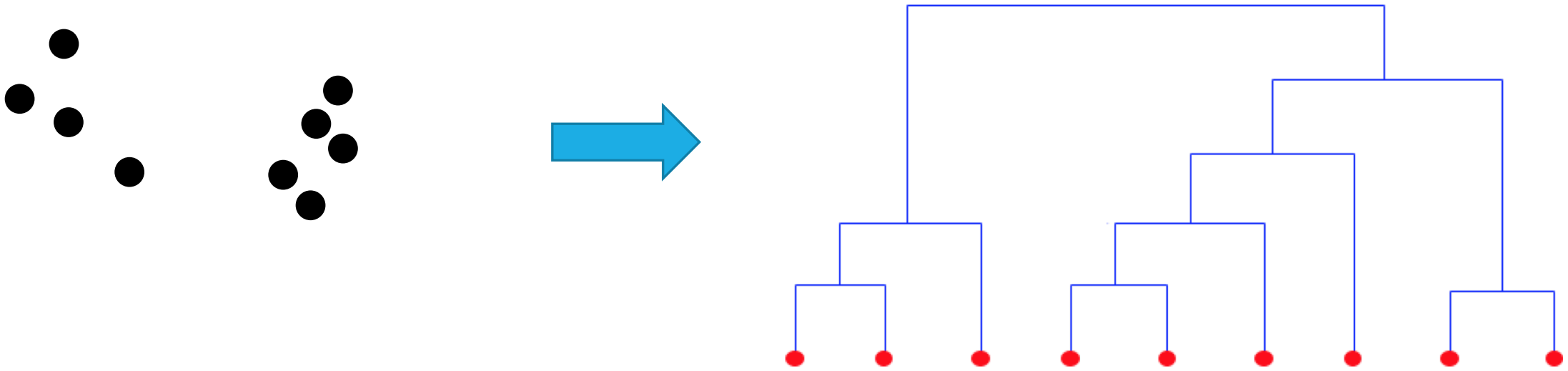
OVERVIEW OF CLUSTERING APPROACHES

- **Center-based:** each cluster is characterized by its center



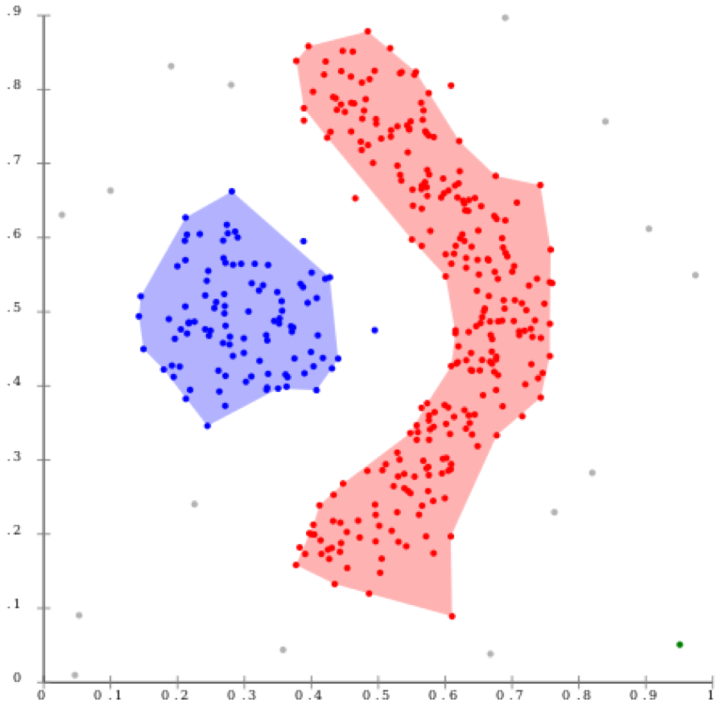
OVERVIEW OF CLUSTERING APPROACHES

- **Hierarchical:** points are organized according to a hierarchy (or tree structure)

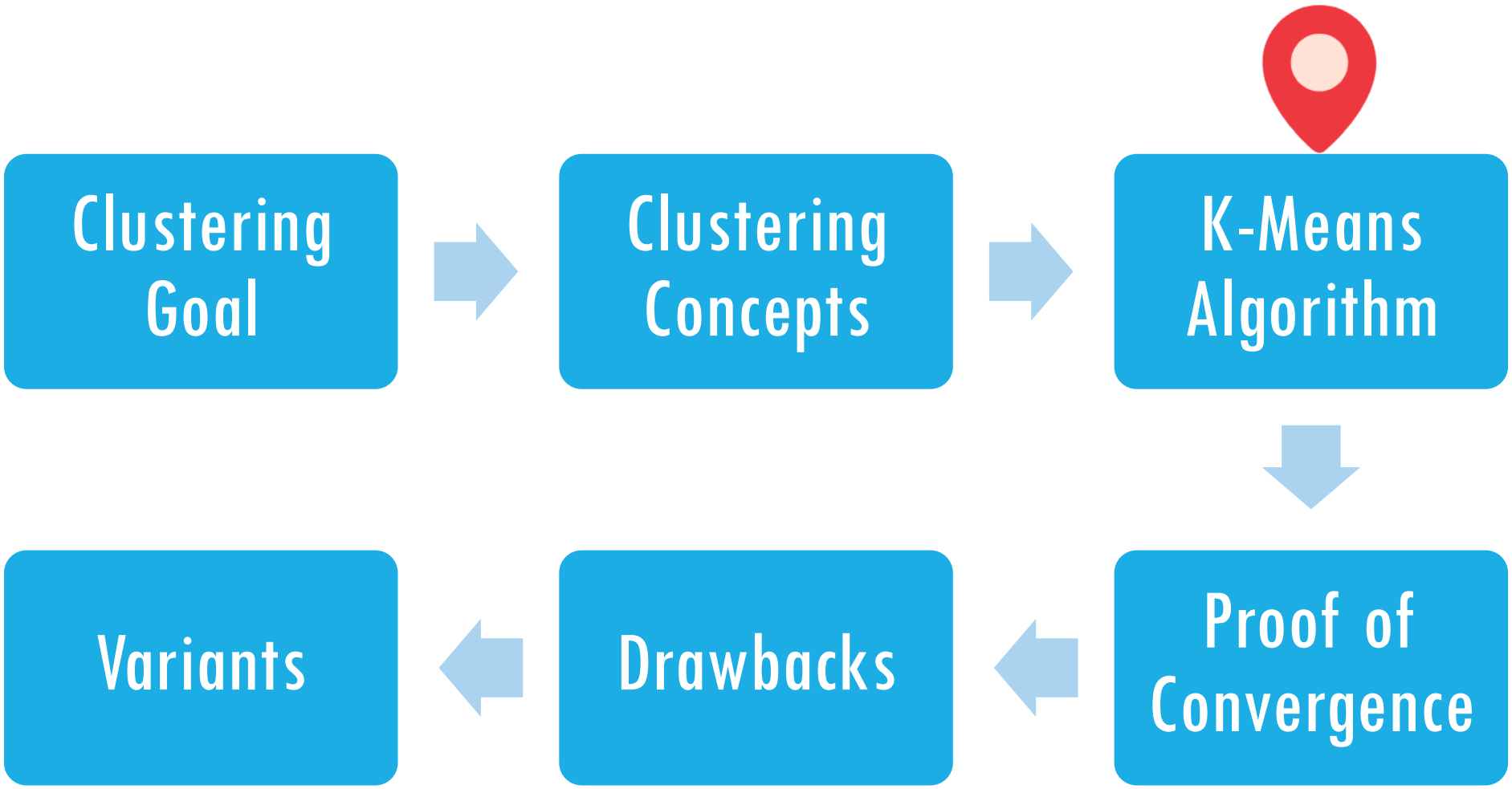


OVERVIEW OF CLUSTERING APPROACHES

- **Density-based:** clusters are high-density regions surrounded by low-density regions



OUTLINE


Clustering Goal
Clustering Concepts
K-Means Algorithm
Proof of Convergence
Drawbacks
Variants

K-MEANS ALGORITHM: STEPS

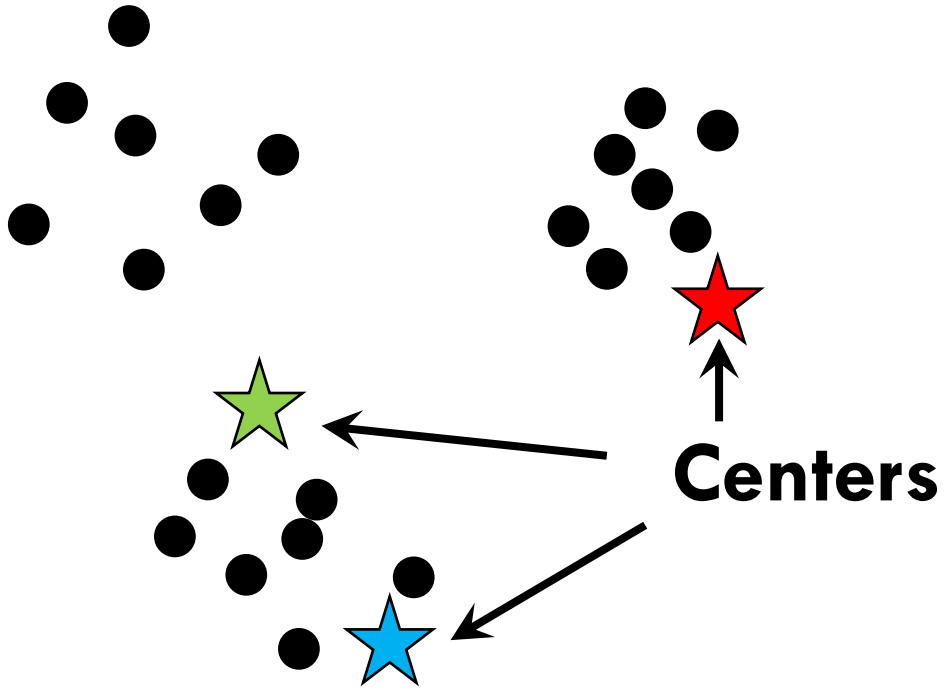
1. Initialization: Pick K random points as centers

2. Repeat:

a) **Assignment:** assign each point to nearest cluster

b) **Update:** move each cluster center to average of its assigned points

Stop if no assignments change



K-MEANS ALGORITHM: STEPS

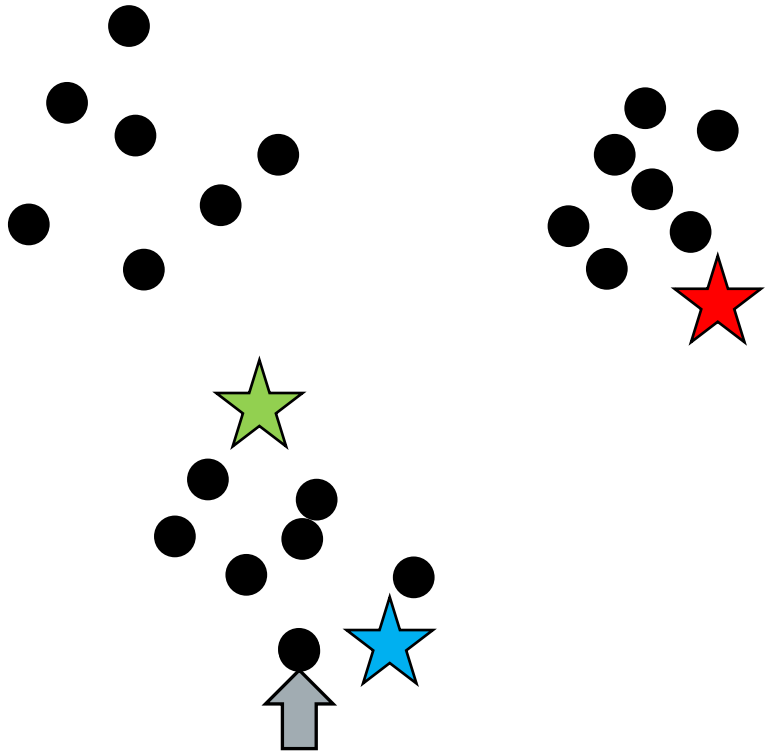
1. Initialization: Pick K random points as centers

2. Repeat:

a) **Assignment:** assign each point to nearest cluster

b) **Update:** move each cluster center to average of its assigned points

Stop if no assignments change



K-MEANS ALGORITHM: STEPS

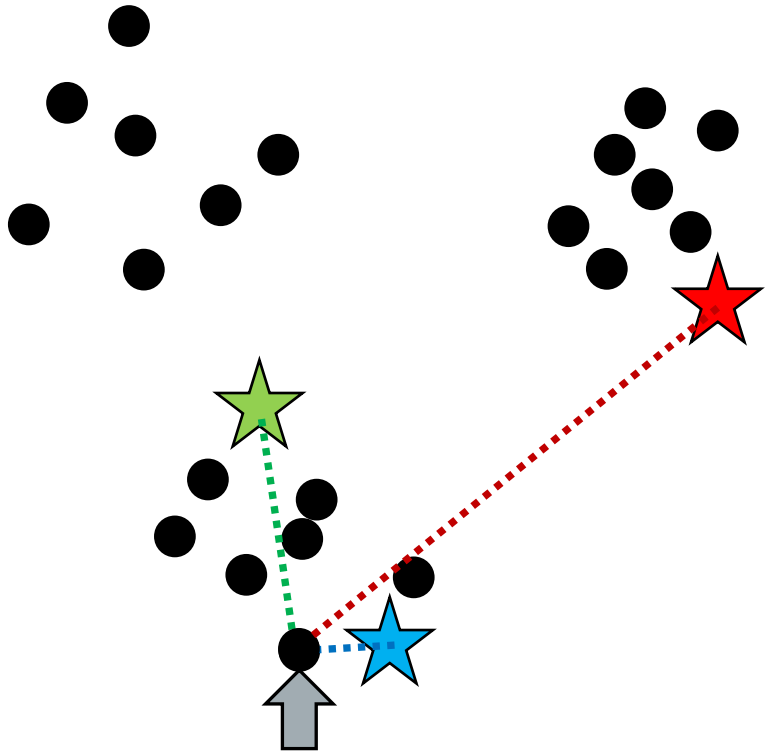
1. Initialization: Pick K random points as centers

2. Repeat:

a) **Assignment:** assign each point to nearest cluster

b) **Update:** move each cluster center to average of its assigned points

Stop if no assignments change



K-MEANS ALGORITHM: STEPS

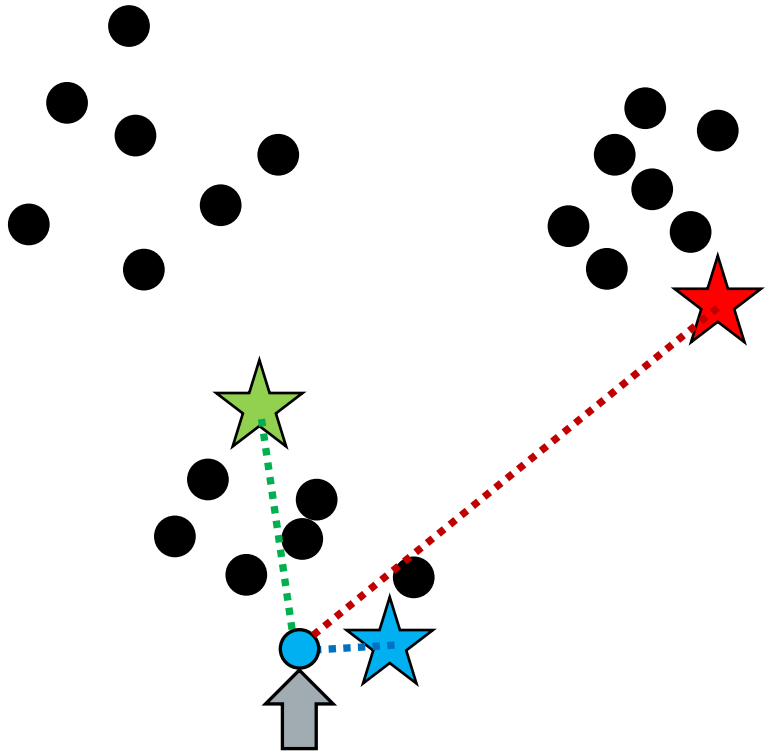
1. Initialization: Pick K random points as centers

2. Repeat:

a) **Assignment:** assign each point to nearest cluster

b) **Update:** move each cluster center to average of its assigned points

Stop if no assignments change



K-MEANS ALGORITHM: STEPS

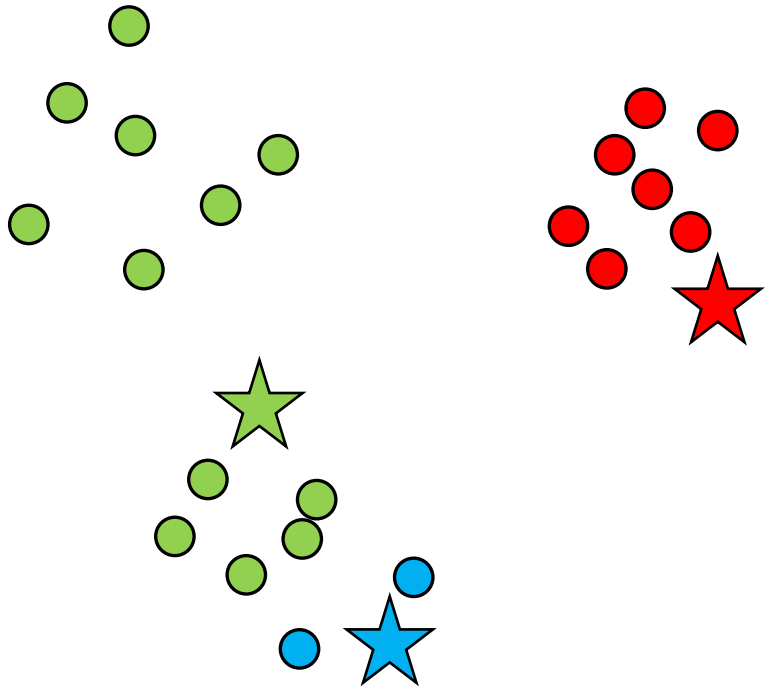
1. Initialization: Pick K random points as centers

2. Repeat:

a) **Assignment:** assign each point to nearest cluster

b) **Update:** move each cluster center to average of its assigned points

Stop if no assignments change



K-MEANS ALGORITHM: STEPS

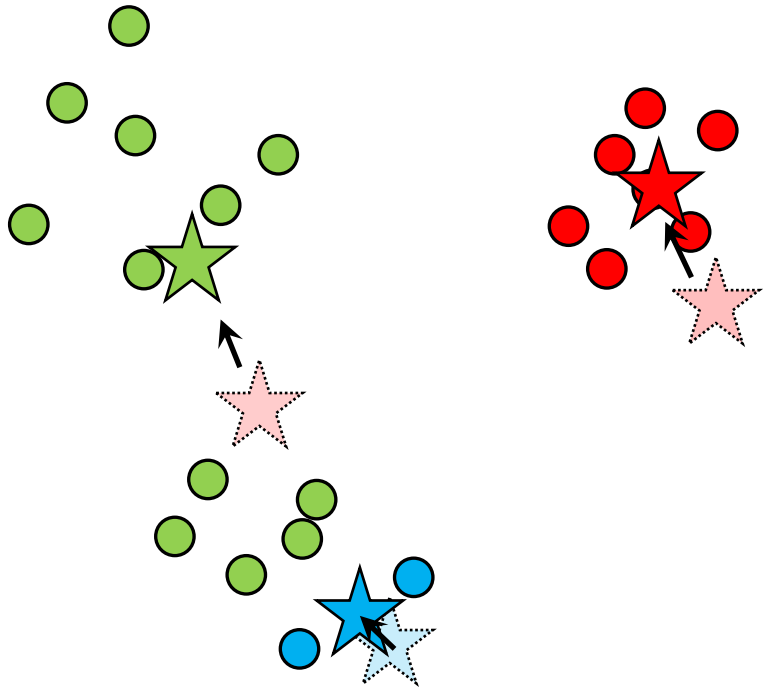
1. Initialization: Pick K random points as centers

2. Repeat:

a) **Assignment:** assign each point to nearest cluster

b) **Update:** move each cluster center to average of its assigned points

Stop if no assignments change



K-MEANS ALGORITHM: STEPS

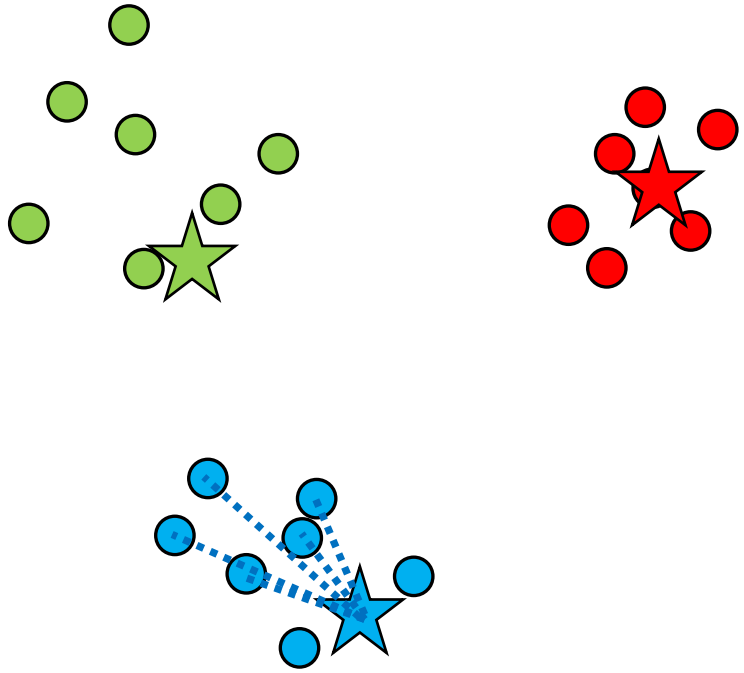
1. **Initialization**: Pick K random points as centers

2. Repeat:

a) **Assignment:** assign each point to nearest cluster

b) **Update:** move each cluster center to average of its assigned points

Stop if no assignments change



K-MEANS ALGORITHM: STEPS

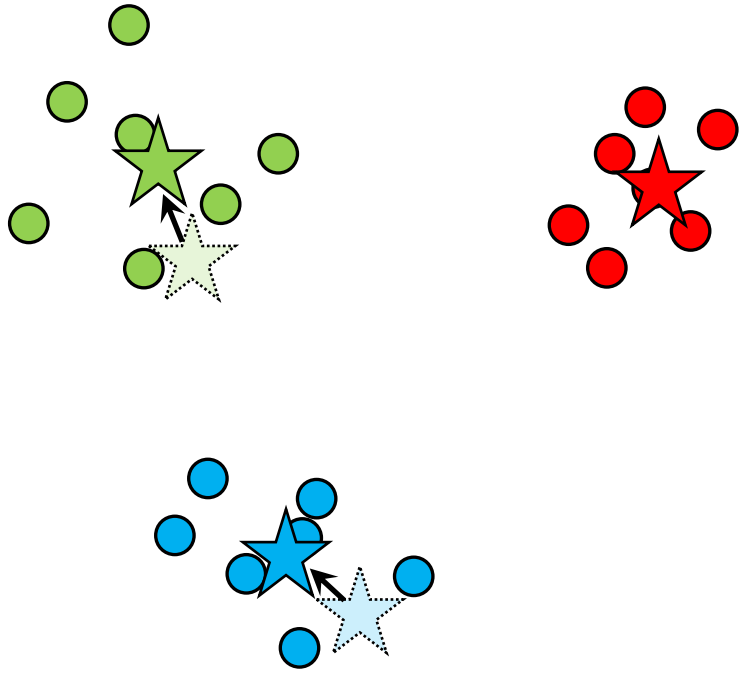
1. **Initialization**: Pick K random points as centers

2. Repeat:

a) **Assignment:** assign each point to nearest cluster

b) Update: move each cluster center to average of its assigned points

Stop if no assignments change



K-MEANS ALGORITHM: STEPS

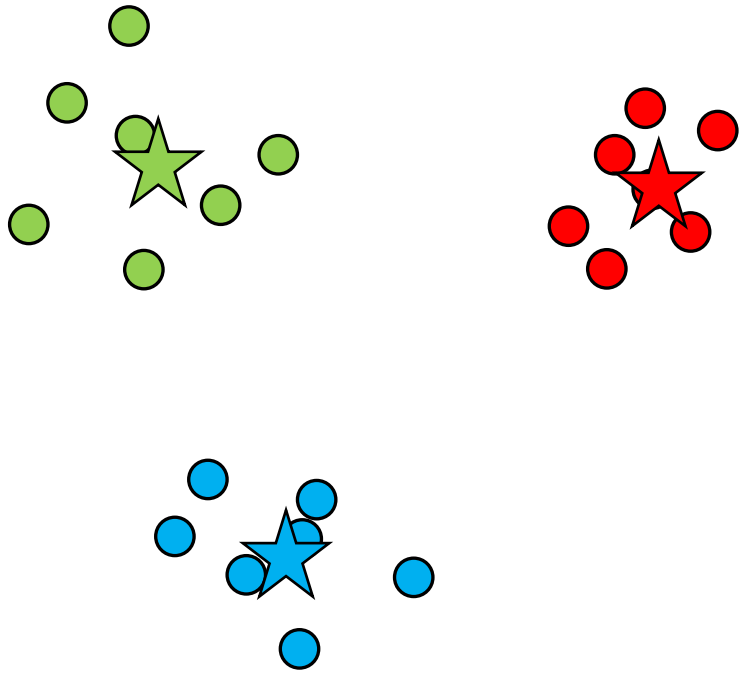
1. Initialization: Pick K random points as centers

2. Repeat:

a) **Assignment:** assign each point to nearest cluster

b) **Update:** move each cluster center to average of its assigned points

Stop if no assignments change



K-MEANS ALGORITHM: STEPS

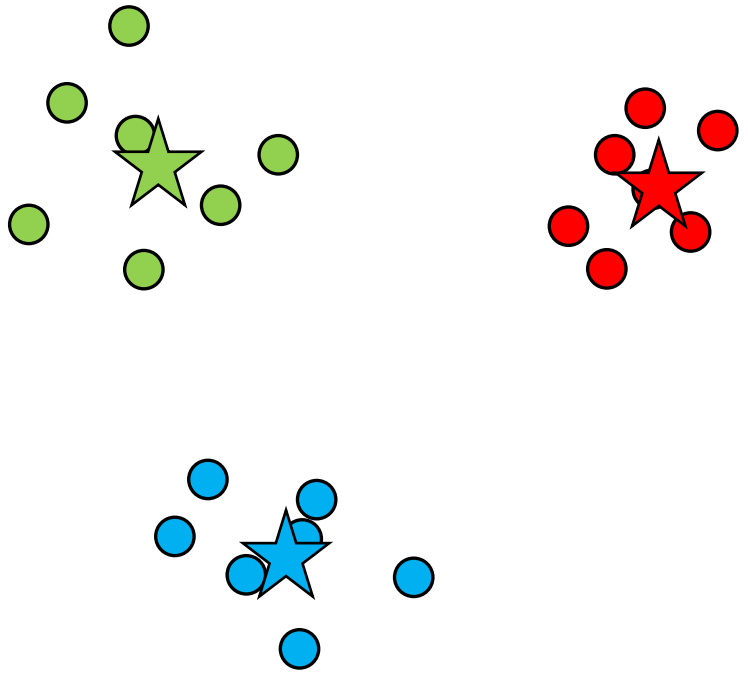
1. Initialization: Pick K random points as centers

2. Repeat:

a) **Assignment:** assign each point to nearest cluster

b) **Update:** move each cluster center to average of its assigned points

Stop if no assignments change



K-MEANS ALGORITHM: TIME COMPLEXITY

Notation

n = number of points
K = number of clusters
d = dimensionality of data
T = number of iterations

During the Assignment step, we measure the distance of each point to each cluster center.

Q: How many distances do we need to compute (during an Assignment step)?

- (a) n
- (b) nd
- (c) nK

1. **Initialization**: Pick K random points as centers

2. Repeat:

a) **Assignment:** assign each point to nearest cluster

b) **Update:** move each cluster center to average of its assigned points

Stop if no assignments change

K-MEANS ALGORITHM: TIME COMPLEXITY

Notation

n = number of points
K = number of clusters
d = dimensionality of data
T = number of iterations

During the Assignment step, we measure the distance of each point to each cluster center.

Q: How many distances do we need to compute (during an Assignment step)?

- (a) n
- (b) nd
- (c) nK

1. **Initialization**: Pick K random points as centers

2. **Repeat:**

   a) **Assignment:** assign each point to nearest cluster

   b) **Update:** move each cluster center to average of its assigned points

   Stop if no assignments change

K-MEANS ALGORITHM: TIME COMPLEXITY

Notation

n = number of points

K = number of clusters

d = dimensionality of data

T = number of iterations

During the Assignment step, we measure the distance of each point to each cluster center

This is nK distances, and computing each is $O(d)$

Thus, each iteration takes $O(dnK)$, and the whole algorithm takes $O(dnKT)$

1. **Initialization**: Pick K random points as centers

2. **Repeat:**

   a) **Assignment:** assign each point to nearest cluster

   b) **Update:** move each cluster center to average of its assigned points

   Stop if no assignments change

Optional

K-MEANS ALGORITHM: TIME COMPLEXITY

K-means takes $O(dnKT)$ for T iterations.

In theory: in the worst case, K-means can take exponential no. of iterations to converge.

In practice: 20-50 iterations are enough for most practical situations. Thus, K-means is seen as **linear time in practice,** which is very fast (e.g. compared to our later algorithms)

***k*-means Requires Exponentially Many Iterations Even in the Plane**

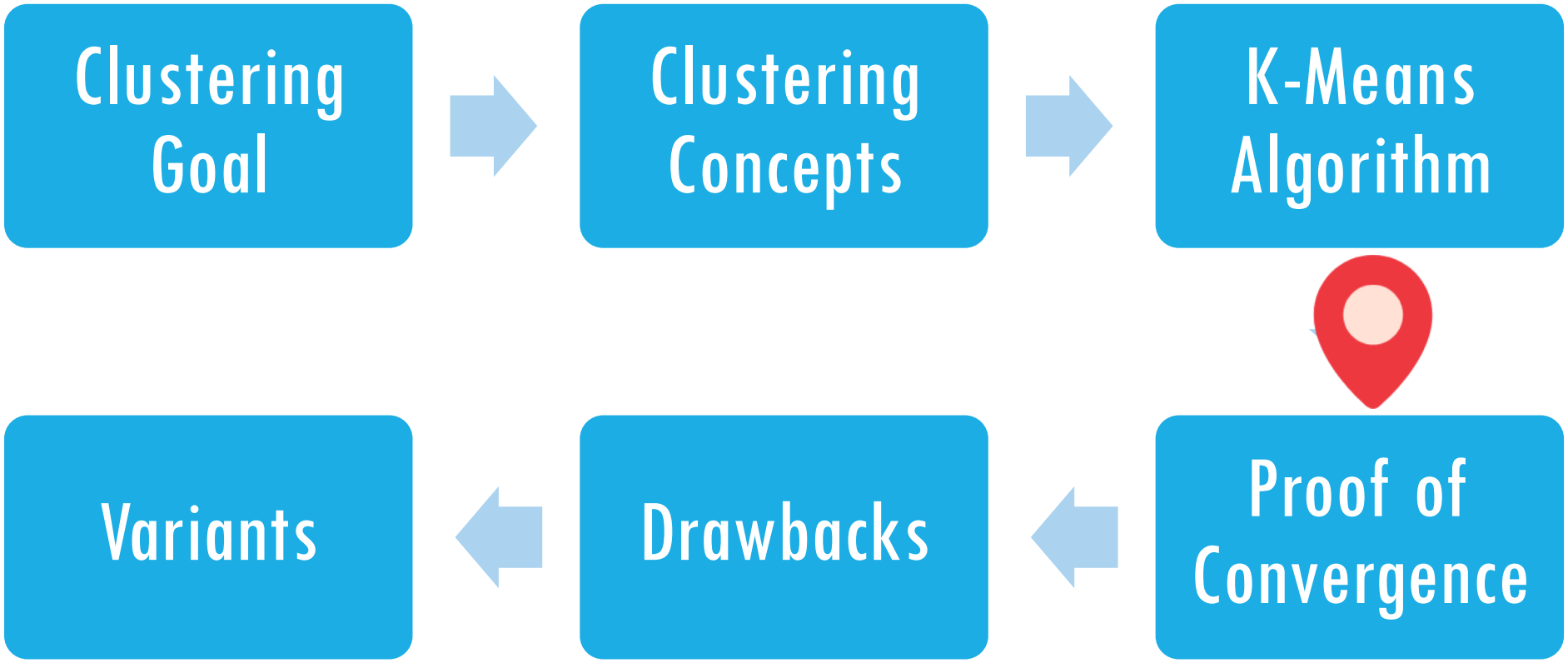
Andrea Vattani

Received: 30 June 2009 / Revised: 29 November 2009 / Accepted: 7 December 2009 /
Published online: 23 March 2011

Scalable K-Means by Ranked Retrieval·

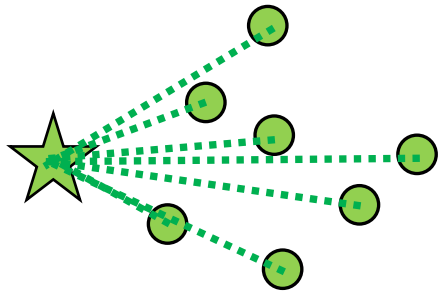
in practice. Although it can take an exponential number of steps to converge to a local optimum [34], in all practical situations it converges after 20-50 iterations (a fact confirmed by the experiments in Section 4). The latter has been partially explained using smoothed analysis [3, 5] to show that worst case instances are unlikely to happen.

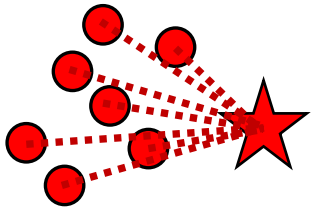
OUTLINE

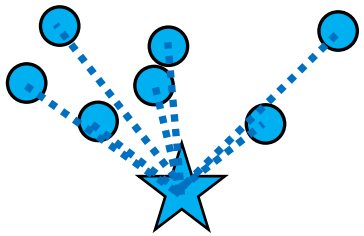

Clustering Goal
Clustering Concepts
K-Means Algorithm
Proof of Convergence
Drawbacks
Variants

OPTIMIZATION OBJECTIVE (OR LOSS FUNCTION)

Within-Cluster Sum of Squares (WCSS): sum of squared distances between each point and its cluster center







$$\text{WCSS} = \sum_{i=1}^{K} \sum_{x \in C_i} \|x - c_i\|_2^2$$

ith cluster center

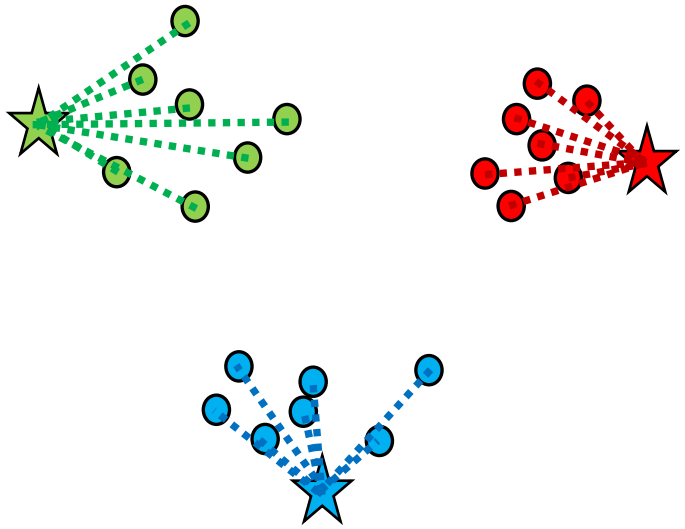
Sum over clusters

Set of points in ith cluster

Squared distance between point and center

OPTIMIZATION OBJECTIVE (OR LOSS FUNCTION)

Within-Cluster Sum of Squares (WCSS): sum of squared distances between each point and its cluster center



$$\text{WCSS} = \sum_{i=1}^{K} \sum_{x \in C_i} \|x - c_i\|_2^2$$

ith cluster center

Sum over clusters

Set of points in ith cluster

Squared distance between point and center

Different view of K-means: the K-means steps (**Assignment** and **Update**) can be understood as steps for minimizing the WCSS objective.

K-MEANS AS ALTERNATING MINIMIZATION

1. **Initialization**: Pick K random points as centers

2. Repeat:

a) **Assignment:** assign each point to nearest cluster

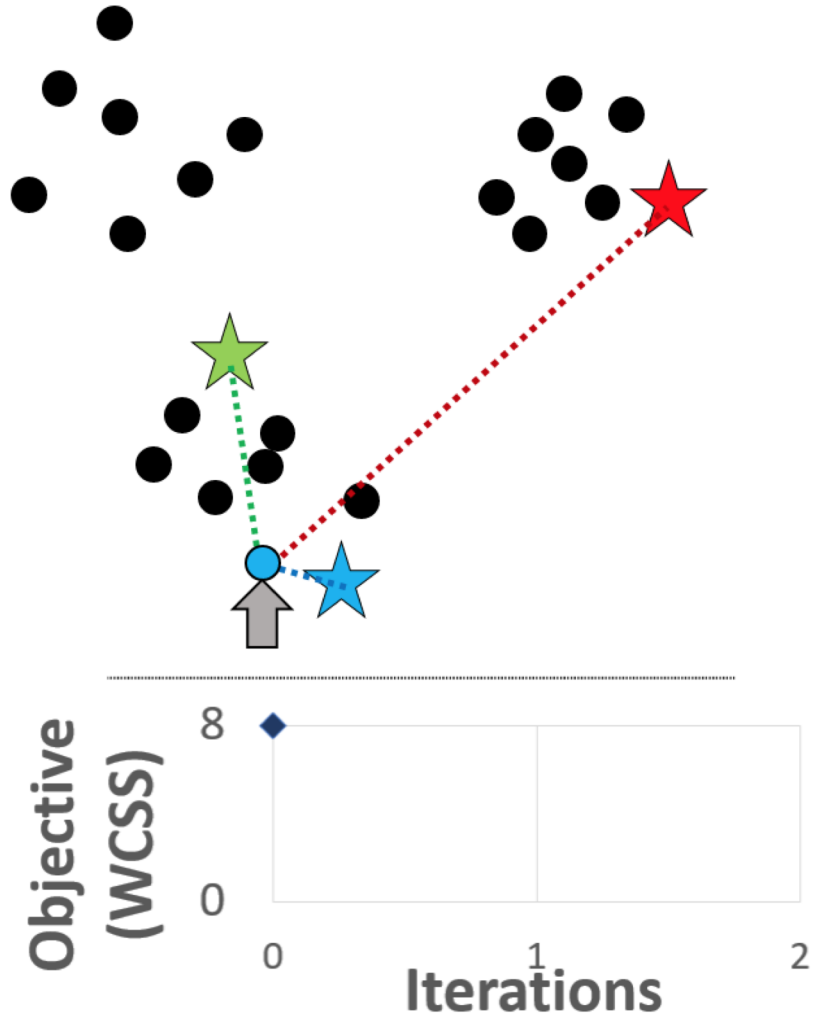
Cluster Assignments →

$$\underset{C_1,\cdots,C_K}{\text{minimize}} \sum_{i=1}^{K} \sum_{x \in C_i} \|x - c_i\|_2^2$$

b) **Update:** move each cluster center to average of its assigned points

$$\underset{c_1,\cdots,c_K}{\text{minimize}} \sum_{i=1}^{K} \sum_{x \in C_i} \|x - c_i\|_2^2$$

Stop if no assignments change



K-MEANS AS ALTERNATING MINIMIZATION

1. **Initialization**: Pick K random points as centers

2. Repeat:

a) **Assignment:** assign each point to nearest cluster

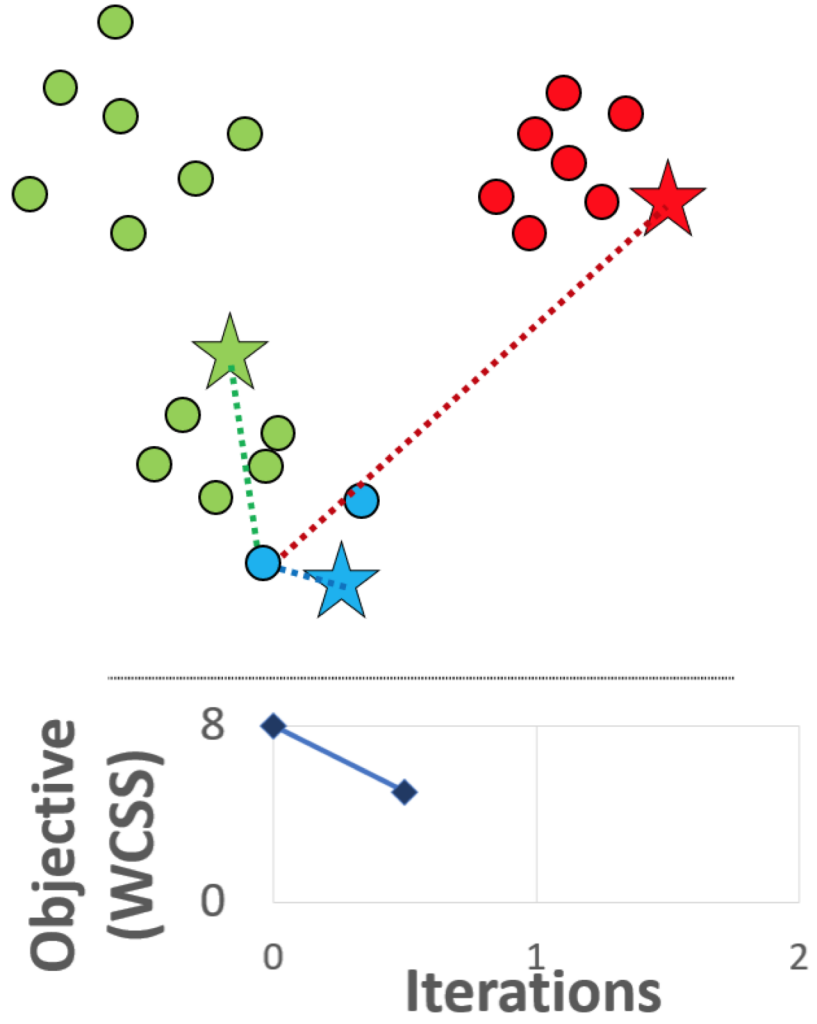
$$\underset{C_1,\cdots,C_K}{\text{minimize}} \sum_{i=1}^{K} \sum_{x \in C_i} \|x - c_i\|_2^2$$

b) **Update:** move each cluster center to average of its assigned points

Cluster Centers →

$$\underset{c_1,\cdots,c_K}{\text{minimize}} \sum_{i=1}^{K} \sum_{x \in C_i} \|x - c_i\|_2^2$$

Stop if no assignments change



K-MEANS AS ALTERNATING MINIMIZATION

1. **Initialization**: Pick K random points as centers

2. Repeat:

a) **Assignment:** assign each point to nearest cluster

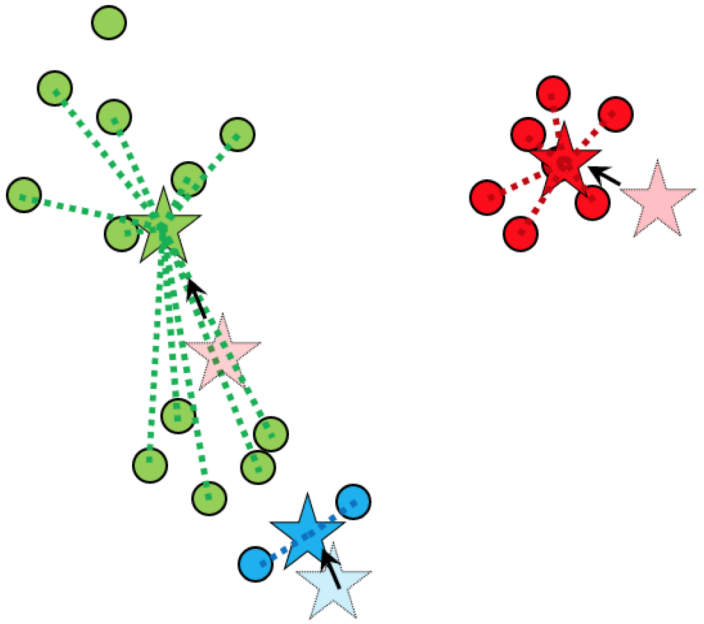
$$\underset{C_1,\cdots,C_K}{\text{minimize}} \sum_{i=1}^{K} \sum_{x \in C_i} \|x - c_i\|_2^2$$

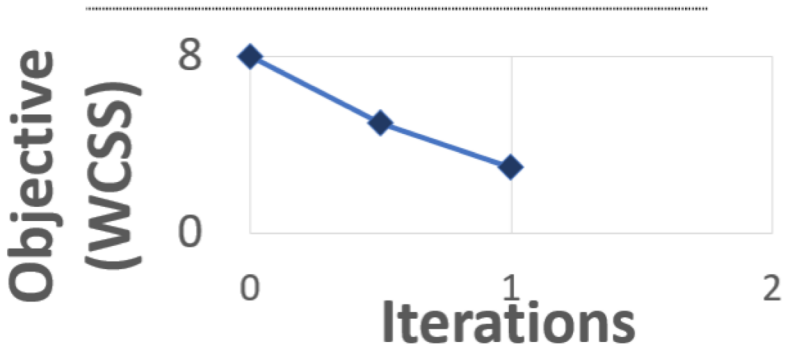
b) **Update:** move each cluster center to average of its assigned points

Cluster Centers →

$$\underset{c_1,\cdots,c_K}{\text{minimize}} \sum_{i=1}^{K} \sum_{x \in C_i} \|x - c_i\|_2^2$$

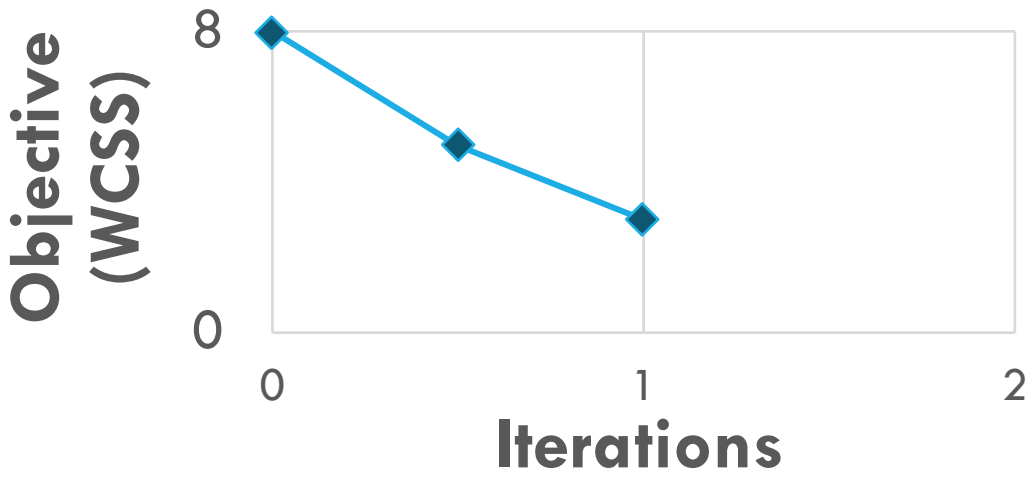
Stop if no assignments change



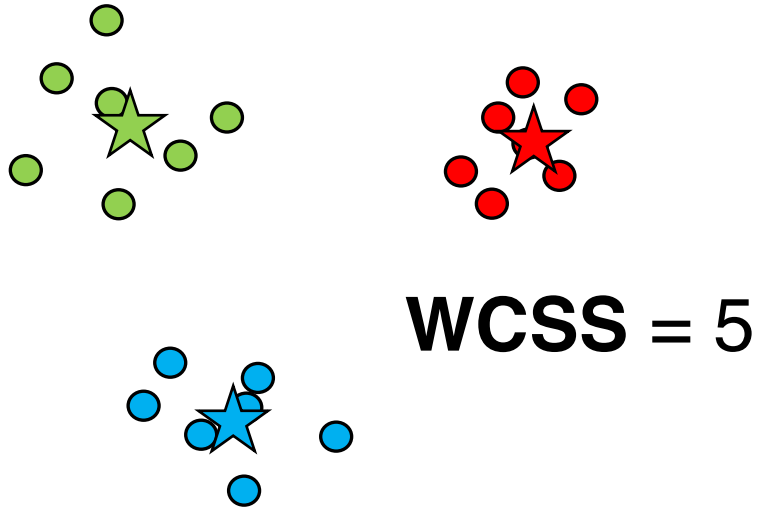


PROOF OF CONVERGENCE

1. The WCSS objective is strictly decreasing

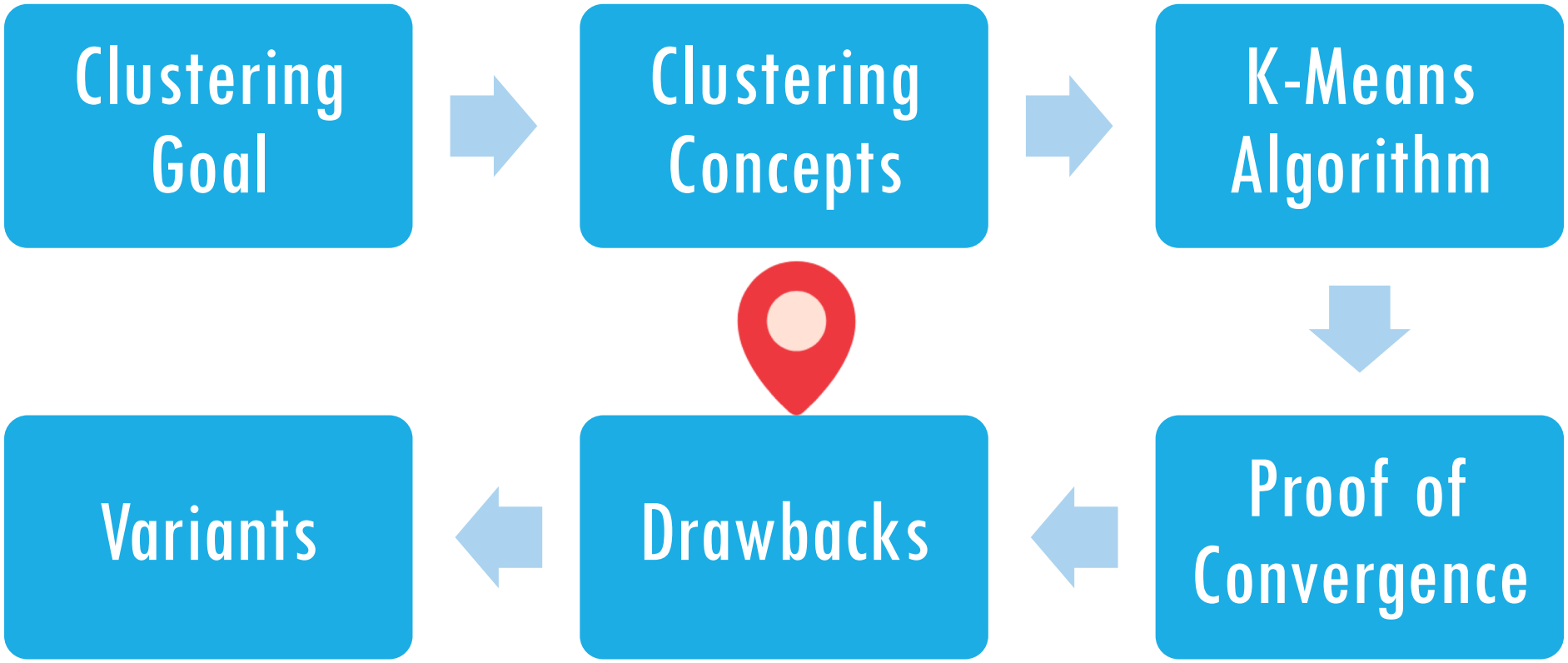


2. There are a finite number of possible clusterings



➡ **The algorithm must eventually stop!**

OUTLINE


Clustering Goal
Clustering Concepts
K-Means Algorithm
Proof of Convergence
Drawbacks
Variants

LOCAL, NOT GLOBAL OPTIMUM

- The algorithm only returns a **local**, not a **global** optimum!
  - (Finding the global optimum is NP-hard)

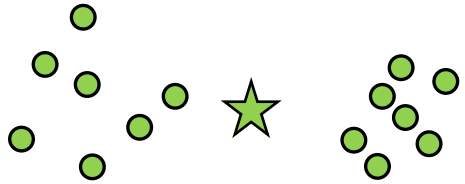
The hardness of k-means clustering in the plane

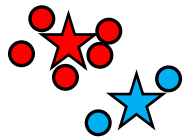
Andrea Vattani
University of California, San Diego
avattani@ucsd.edu

Abstract

We show that k-means clustering is an NP-hard optimization problem, even for instances in the plane. Specifically, the hardness holds for $k = \Theta(n^\epsilon)$, for any $\epsilon > 0$, where n is the number of points in the instance, and k is the number of clusters.

Example of a local minima

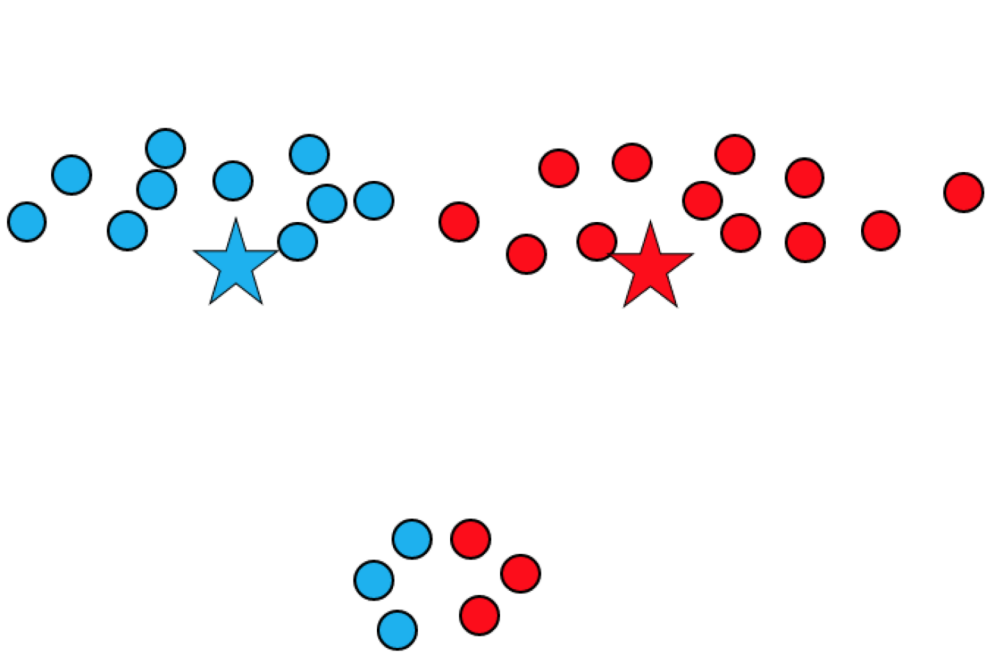


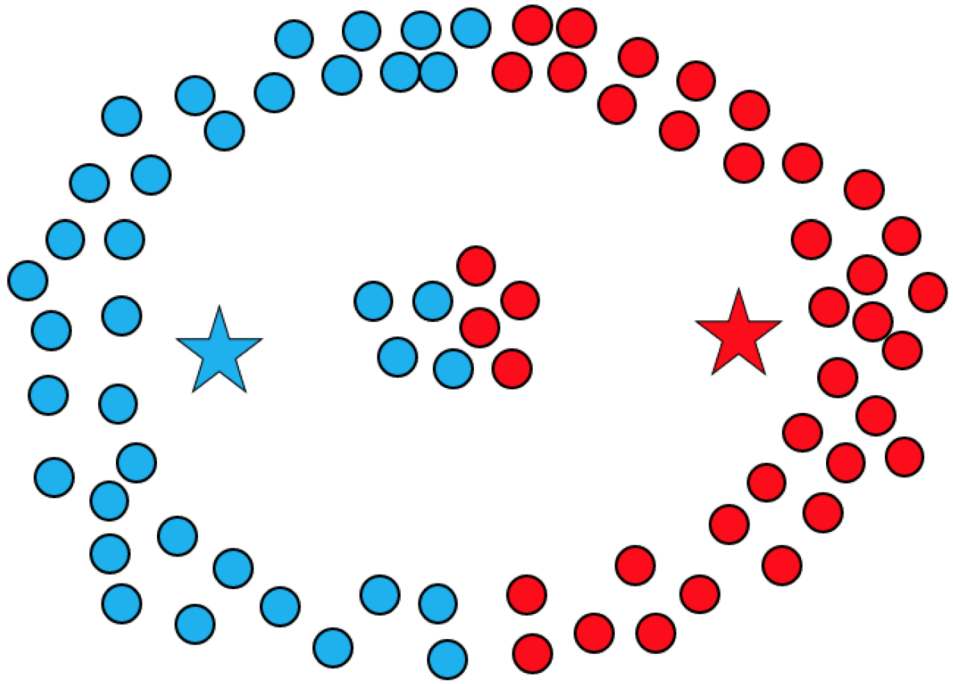


- Initialization is important

NON-SPHERE-LIKE CLUSTERS

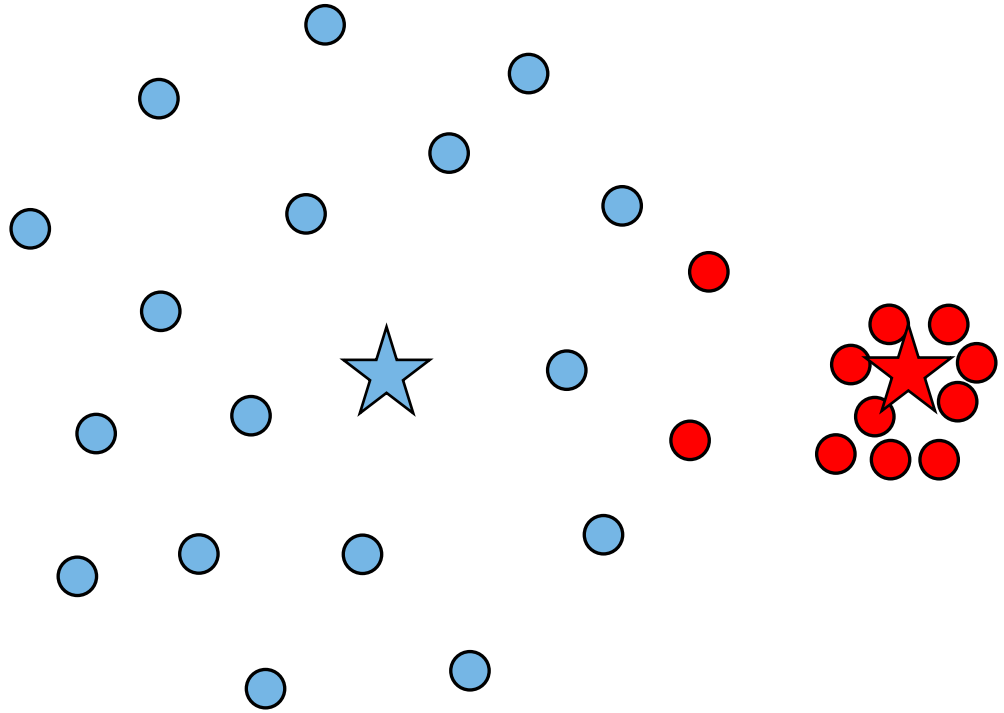
- Optimization objective results in roughly sphere shaped clusters





CLUSTERS OF DIFFERENT DENSITY

- Optimization objective does not adjust for different cluster densities



SUMMARY: PROS AND CONS

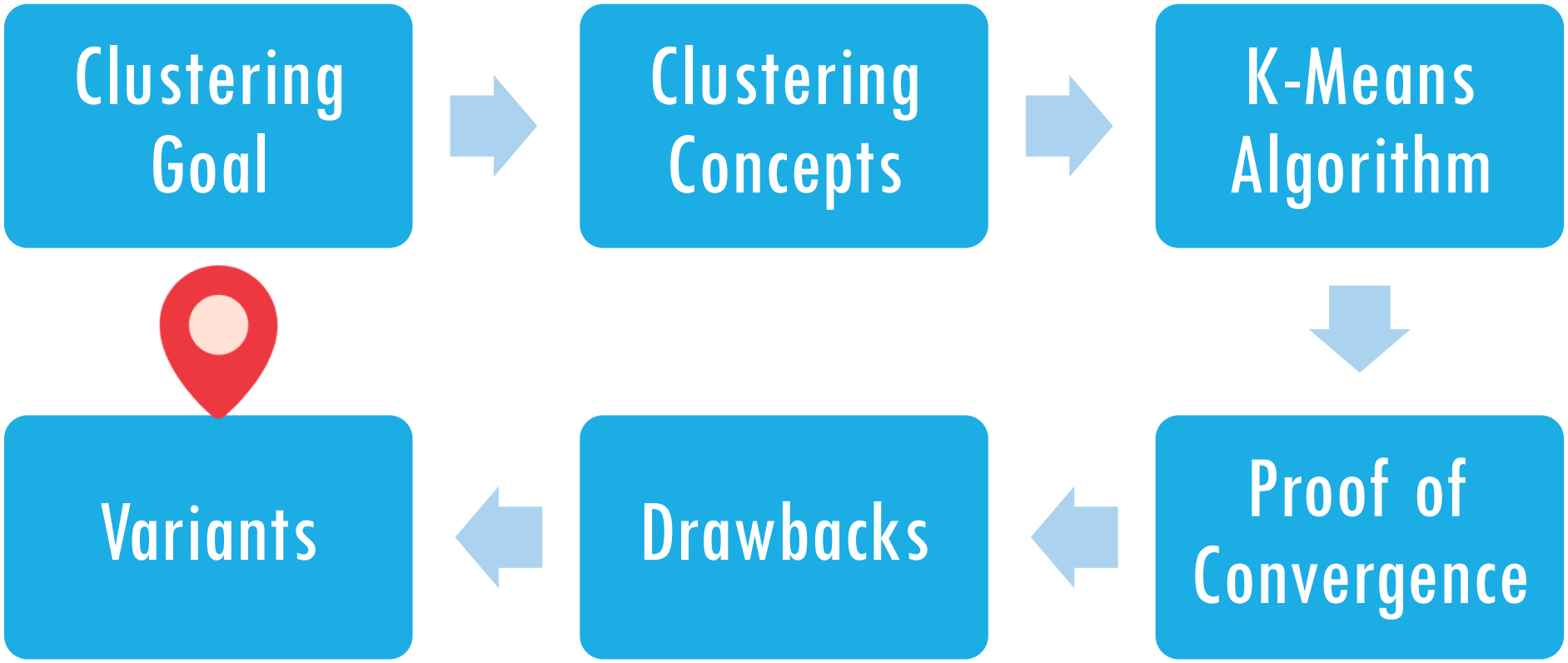
Pros

- Simple to implement and understand
- Clusters are easy to interpret: they can be fully understood based on their cluster centers
- Fast (linear time in practice)

Cons

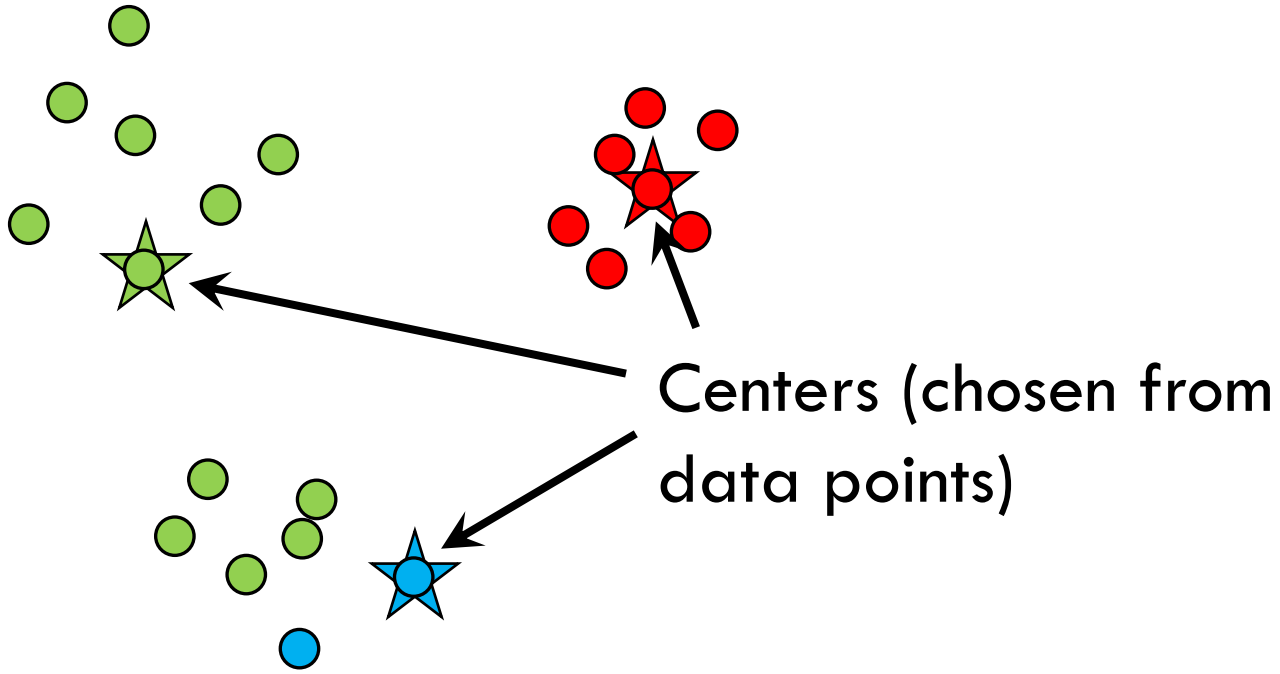
- Local, not global optimum
- Only works well for roughly spherical clusters of the same density

OUTLINE


Clustering Goal
Clustering Concepts
K-Means Algorithm
Proof of Convergence
Drawbacks
Variants

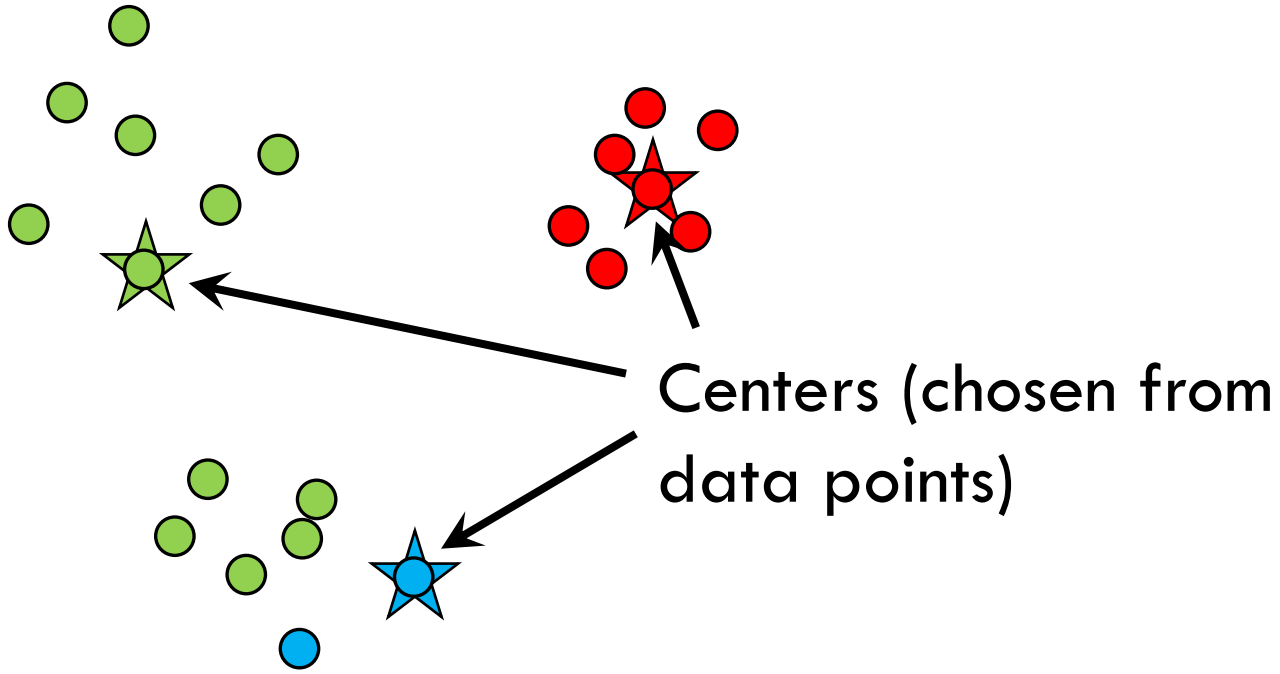
1. K-MEDOIDS ALGORITHM

- **K-Medoids:** like K-Means, but centers are chosen from data points
  - In **Update** step, we choose each cluster's center as the *data point* minimizing the sum of squared distances to the points in the cluster
- Useful when:
  - We want data points as cluster representatives
  - Complex data types – we can only measure distances between data points



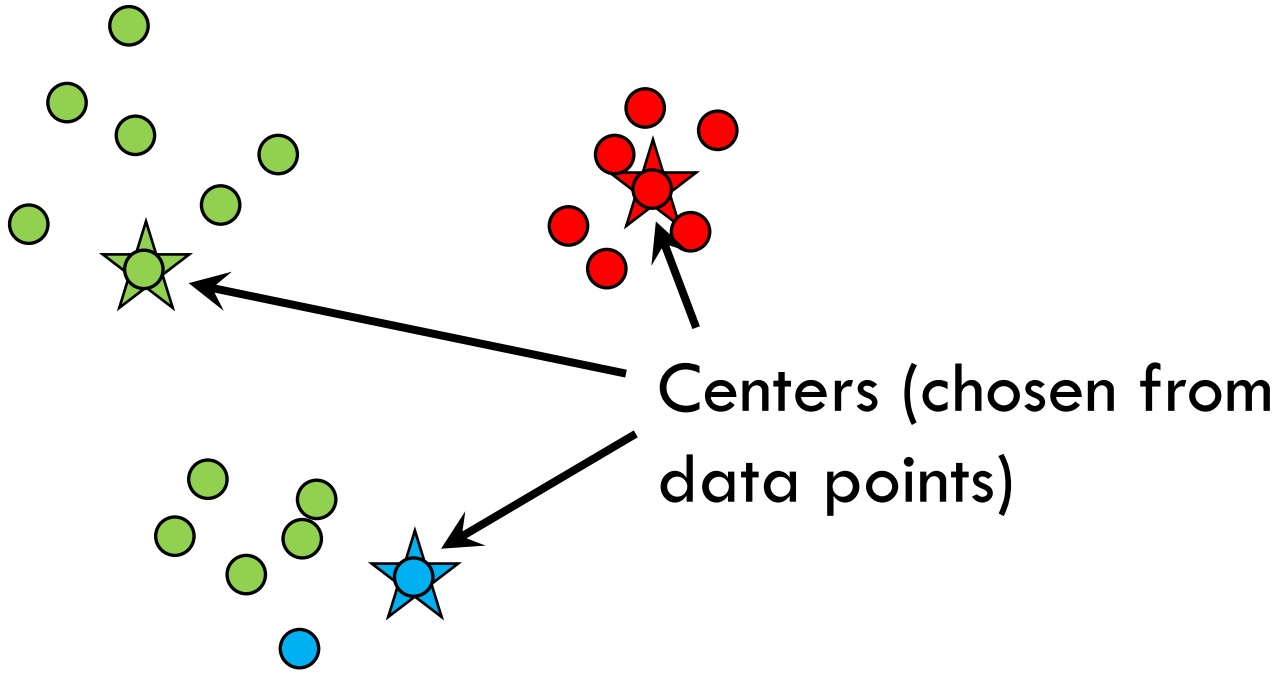
1. K-MEDOIDS ALGORITHM

- **K-Medoids:** like K-Means, but centers are chosen from data points
  - In **Update** step, we choose each cluster's center as the *data point* minimizing the sum of squared distances to the points in the cluster
- **Q:** the time complexity required for an **Update** is _____ in n.
  - (a) Linear
  - (b) Quadratic



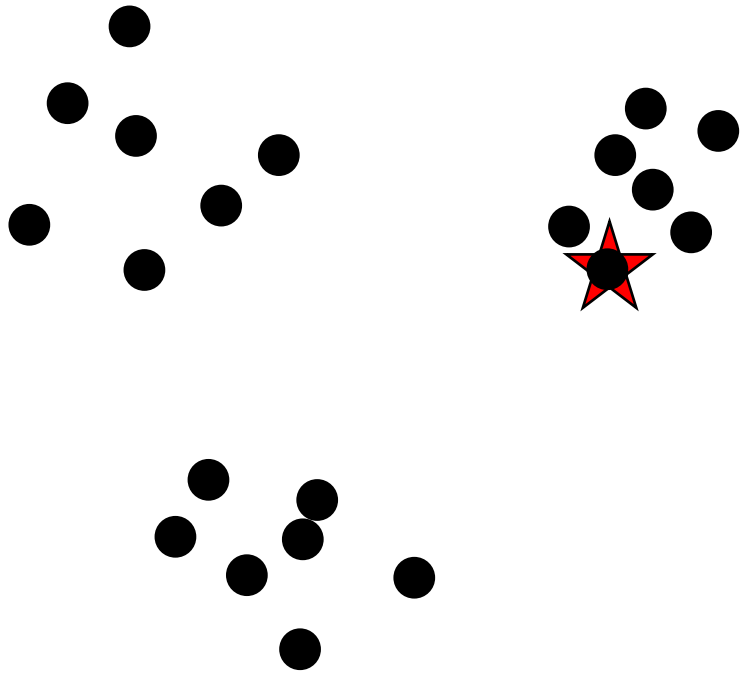
1. K-MEDOIDS ALGORITHM

- **K-Medoids:** like K-Means, but centers are chosen from data points
  - In **Update** step, we choose each cluster's center as the *data point* minimizing the sum of squared distances to the points in the cluster
- **Q:** the time complexity required for an **Update** is _____ in n.
  - (a) Linear
  - (b) Quadratic



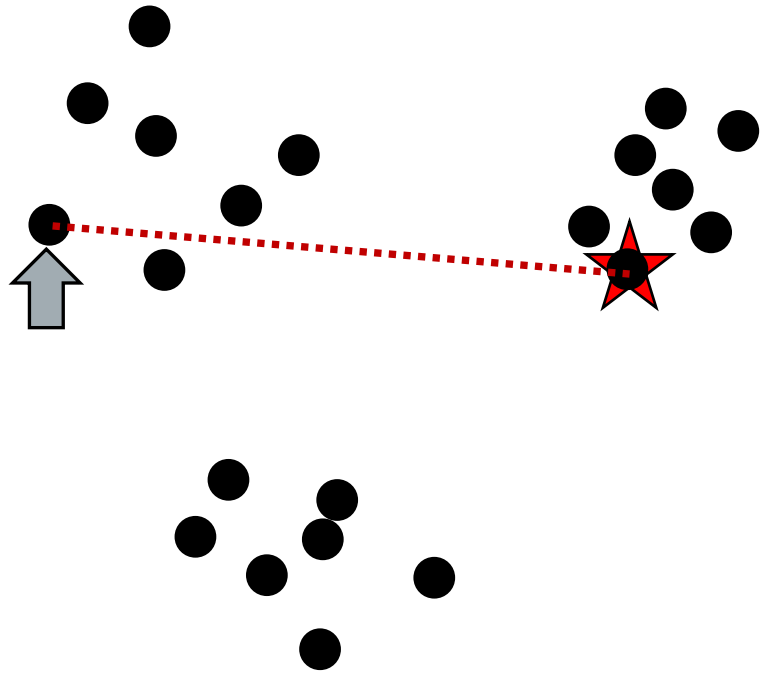
2. K-MEANS++ ALGORITHM

- **K-Means++:** only changes the initialization step
- **"Spread out centers":**
  - First center is a uniformly random point



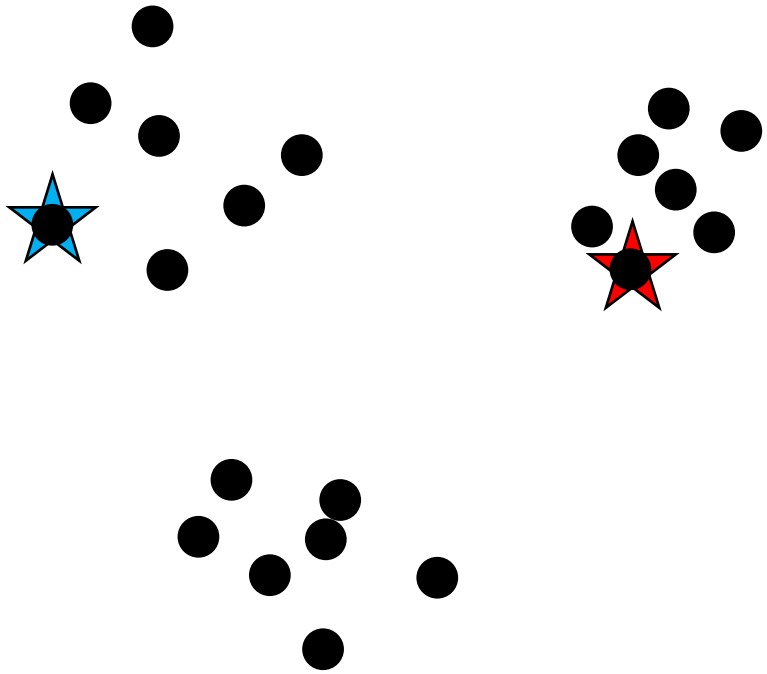
2. K-MEANS++ ALGORITHM

- **K-Means++:** only changes the initialization step
- **"Spread out centers":**
  - First center is a uniformly random point
  - Next centers: each point chosen with probability proportional to square of distance to its closest center



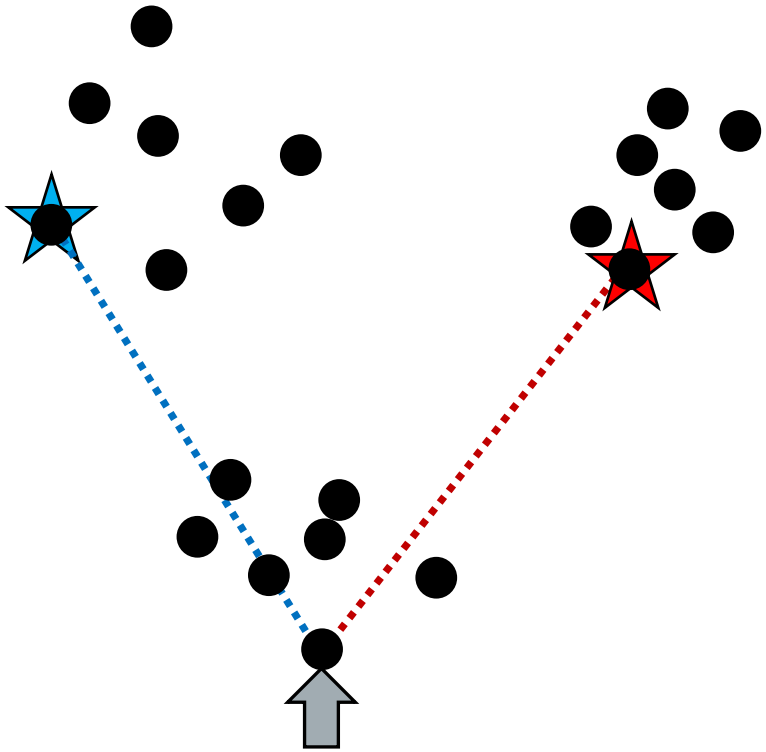
2. K-MEANS++ ALGORITHM

- **K-Means++:** only changes the initialization step
- **"Spread out centers":**
  - First center is a uniformly random point
  - Next centers: each point chosen with probability proportional to square of distance to its closest center



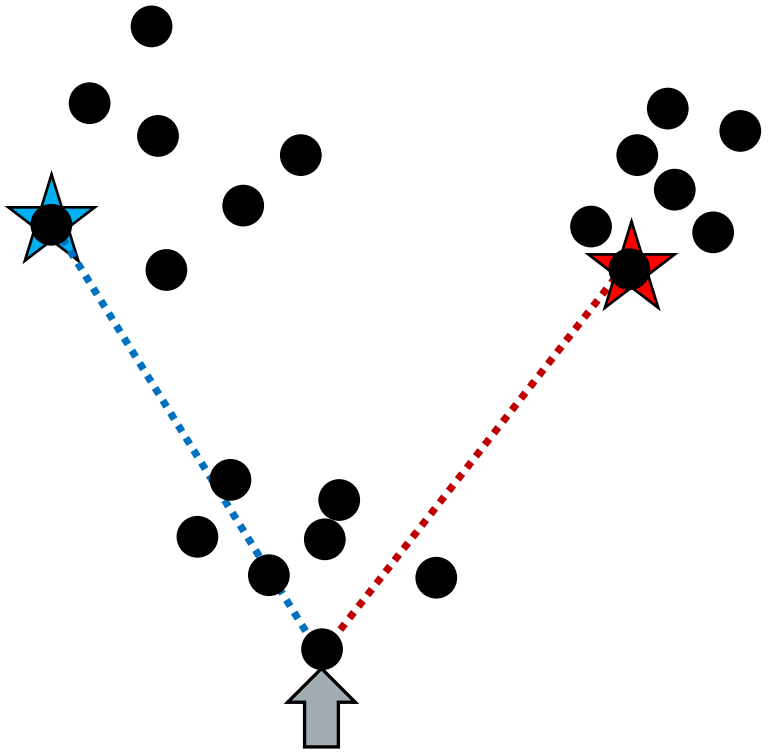
2. K-MEANS++ ALGORITHM

- **K-Means++:** only changes the initialization step
- **"Spread out centers":**
  - First center is a uniformly random point
  - Next centers: each point chosen with probability proportional to square of distance to its closest center



2. K-MEANS++ ALGORITHM

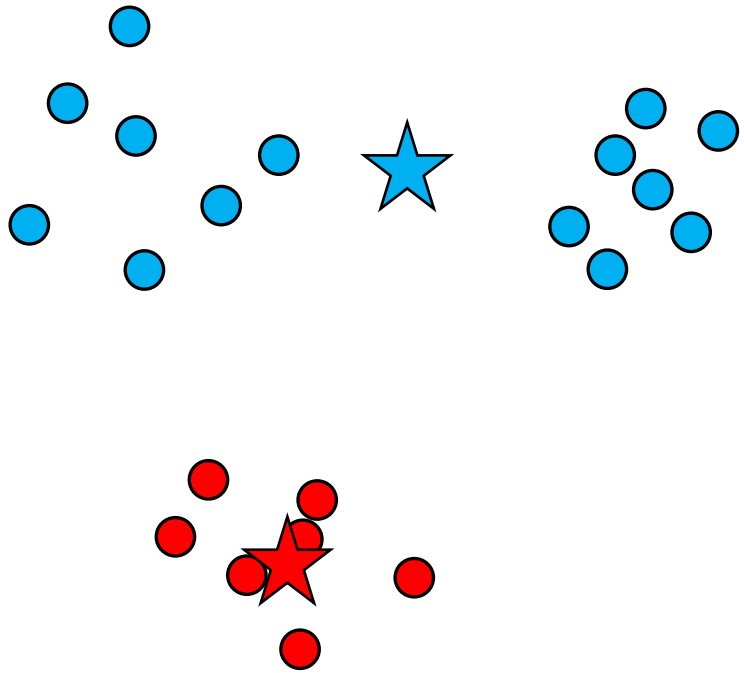
- **K-Means++:** only changes the initialization step
- Better practical performance
- Theoretical guarantee: O(log k) approximation ratio in expectation
- Note: K-Means++ is the default initialization style in sci-kit learn



3. X-MEANS ALGORITHM

- **Automatic way to choose K**

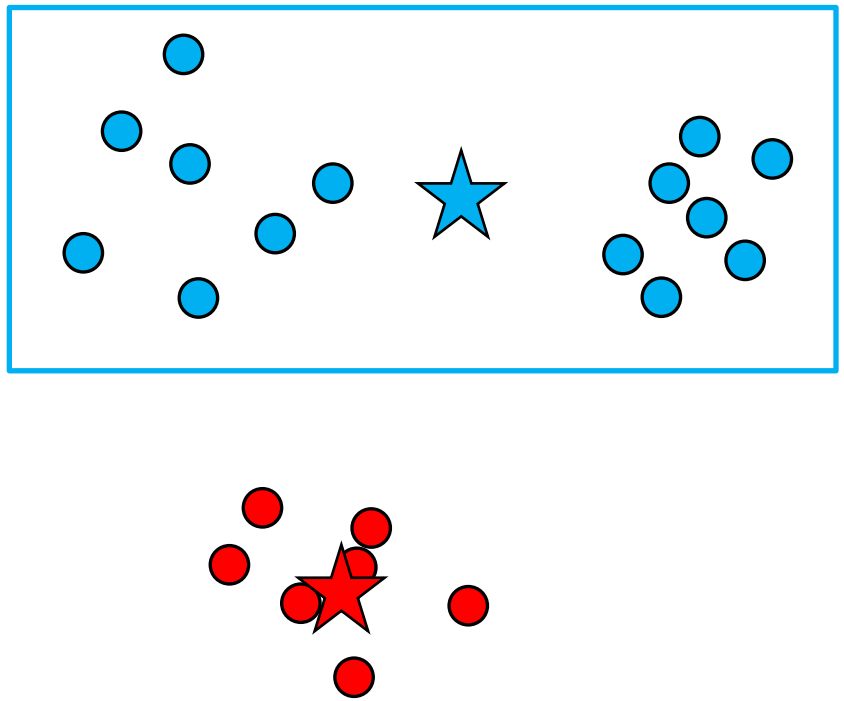
1. Run usual K-Means with K=2



3. X-MEANS ALGORITHM

- **Automatic way to choose K**

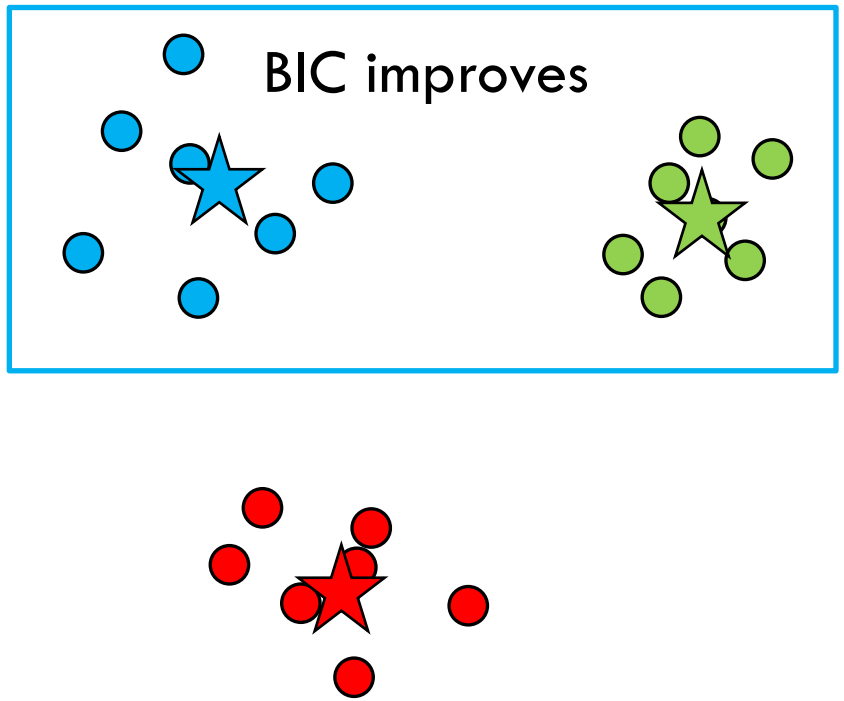
1. Run usual K-Means with K=2
2. Attempt to split each cluster by running K-Means with K=2 only within that cluster



3. X-MEANS ALGORITHM

- **Automatic way to choose K**

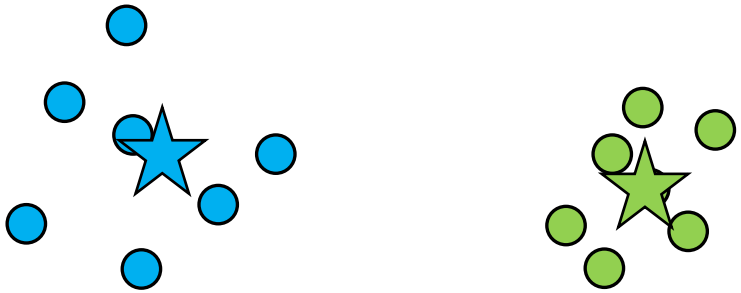
1. Run usual K-Means with K=2
2. Attempt to split each cluster by running K-Means with K=2 only within that cluster
   - Use “Bayesian Information Criterion” (BIC) or "Akaike Information Criterion" (AIC) to decide whether to split (these add a penalty based on no. of clusters)



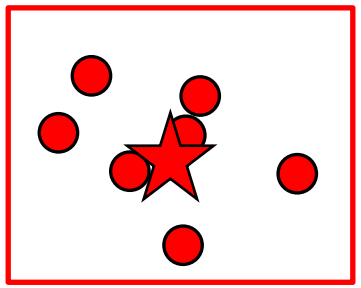
3. X-MEANS ALGORITHM

- **Automatic way to choose K**

1. Run usual K-Means with K=2
2. Attempt to split each cluster by running K-Means with K=2 only within that cluster
   - Use “Bayesian Information Criterion” (BIC) or "Akaike Information Criterion" (AIC) to decide whether to split (these add a penalty based on no. of clusters)



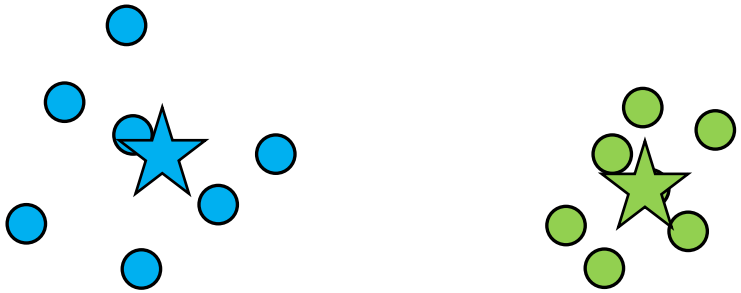
BIC does not improve



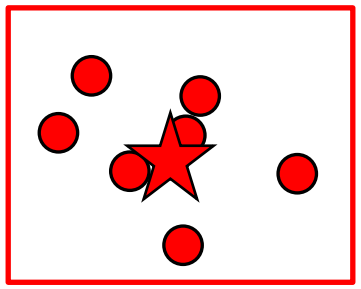
3. X-MEANS ALGORITHM

- **Automatic way to choose K**

1. Run usual K-Means with K=2
2. Attempt to split each cluster by running K-Means with K=2 only within that cluster
   - **Q**: Assume we decide whether to split clusters or not using the WCSS objective (instead of BIC). Would this work?
     - (a) Yes
     - (b) No; it would always split
     - (c) No; it would never split



BIC does not improve



APPLICATION: IMAGE SEGMENTATION

Original

k=5

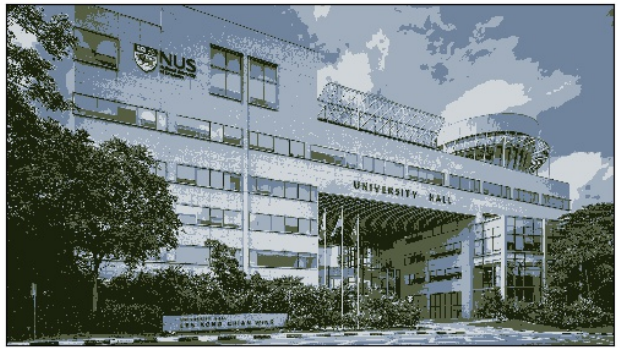
k=10

k=15

APPLICATION: IMAGE SEGMENTATION

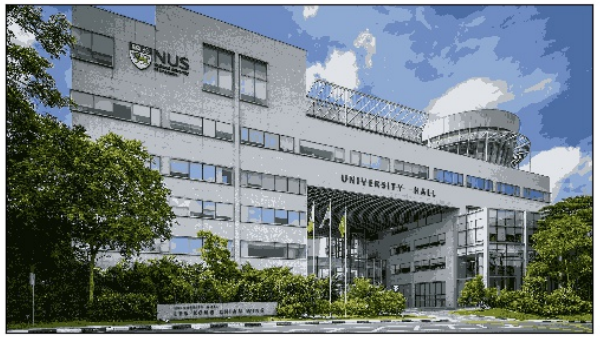
Original

k=5



k=10

k=15



- Note (out of syllabus): modern image segmentation methods generally use deep learning.
  - However, it is still quite common for them to use some kind of clustering internally (e.g. in a neural network's "feature space" instead of clustering the raw pixels)