# Machine Learning 520
# Advanced Machine Learning

## Lesson 6: Unsupervised Learning

UNIVERSITY of WASHINGTON

# Today's Agenda

- Clustering
  - Concept of Clustering
  - Distance Measures
  - Hierarchical Clustering
  - K-means Clustering
  - Clustering Evaluation

# Learning Objectives

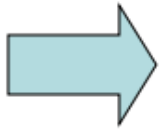
By the end of this session, you should be able to:

- Identify use cases for unsupervised learning.
- Explain the effects of applying different distance metrics.
- Apply K-means to benchmark data and evaluate the quality of the clusters.
- Apply hierarchical clustering to benchmark data and evaluate the quality of the clusters.

# Unsupervised Learning

So far, our data has been in this form:

| | |
|---|---|
| $x_1^1, x_2^1, x_3^1, \ldots, x_m^1$ | $y^1$ |
| $x_1^2, x_2^2, x_3^2, \ldots, x_m^2$ | $y^2$ |
| … | |
| … | |
| $x_1^n, x_2^n, x_3^n, \ldots, x_m^n$ | $y^n$ |



We will be looking at ***unlabeled data:***

| |
|---|
| $x_1^1, x_2^1, x_3^1, \ldots, x_m^1$ |
| $x_1^2, x_2^2, x_3^2, \ldots, x_m^2$ |
| … |
| … |
| $x_1^n, x_2^n, x_3^n, \ldots, x_m^n$ |

- What do we expect to learn from such data?
- I have tons of data (web documents, gene data, etc) and need to:
  - organize it better – e.g., find subgroups
  - understand it better – e.g., understand interrelationships
  - find regular trends in it – e.g., If A and B, then C

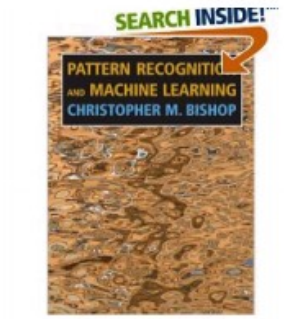
# Clustering

- **Clustering:** Organization of unlabeled data into similar groups (subsets).

- **Cluster:** A collection of data where the items in the collection are *similar* to one another and are *dissimilar* to other collections of data.

- **What are good clusters?**
  - Maximize the similarity within objects in the same group.
  - Maximize the difference between objects in different groups.

# Recommendations




**Pattern Recognition and Machine Learning (Information Science and Statistics) (Hardcover)**
by Christopher M. Bishop (Author)
★★★★☆ (22 customer reviews)

**List Price:** ~~$74.95~~
**Price:** **$50.11** & this item ships for **FREE with Super Saver Shipping**. Details
**You Save:** $24.84 (33%)

**Upgrade this book** for $14.99 more, and you can read, search, and annotate every page online. See details
**Availability:** In Stock. Ships from and sold by **Amazon.com**. Gift-wrap available.

**Want it delivered Monday, October 29?** Order it in the next 0 hours and 33 minutes, and choose **One-Day Shipping** at checkout. See details

Share your own customer images
Search inside this book

**58 used & new** available from $43.59

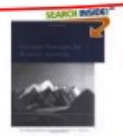
**Better Together**

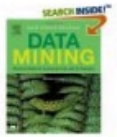
Buy this book with The Elements of Statistical Learning by T. Hastie today!

**Buy Together Today: $118.84**
Buy both now!

**Customers Who Bought This Item Also Bought**


Pattern Classification (2nd Edition) by


Gaussian Processes for Machine Learning by



Data Mining by Ian H. Witten


Machine Learning by Tom M. Mitchell


Computer Manual in MATLAB to Accompany

# Image compression: Vector quantization

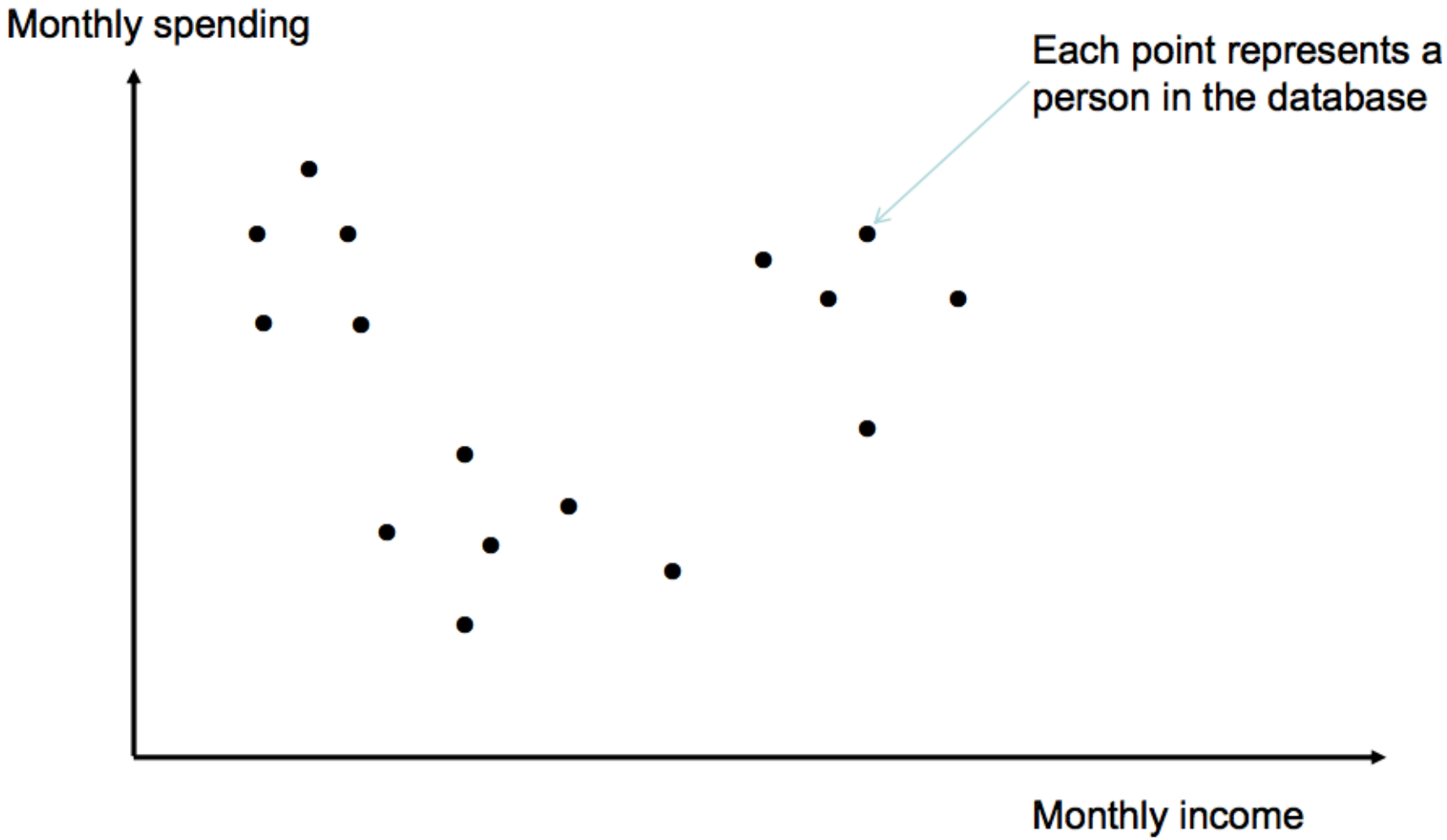
Group all pixels into self-similar groups, instead of storing all pixel values, store the means of each group

701,554 bytes → 127,292 bytes

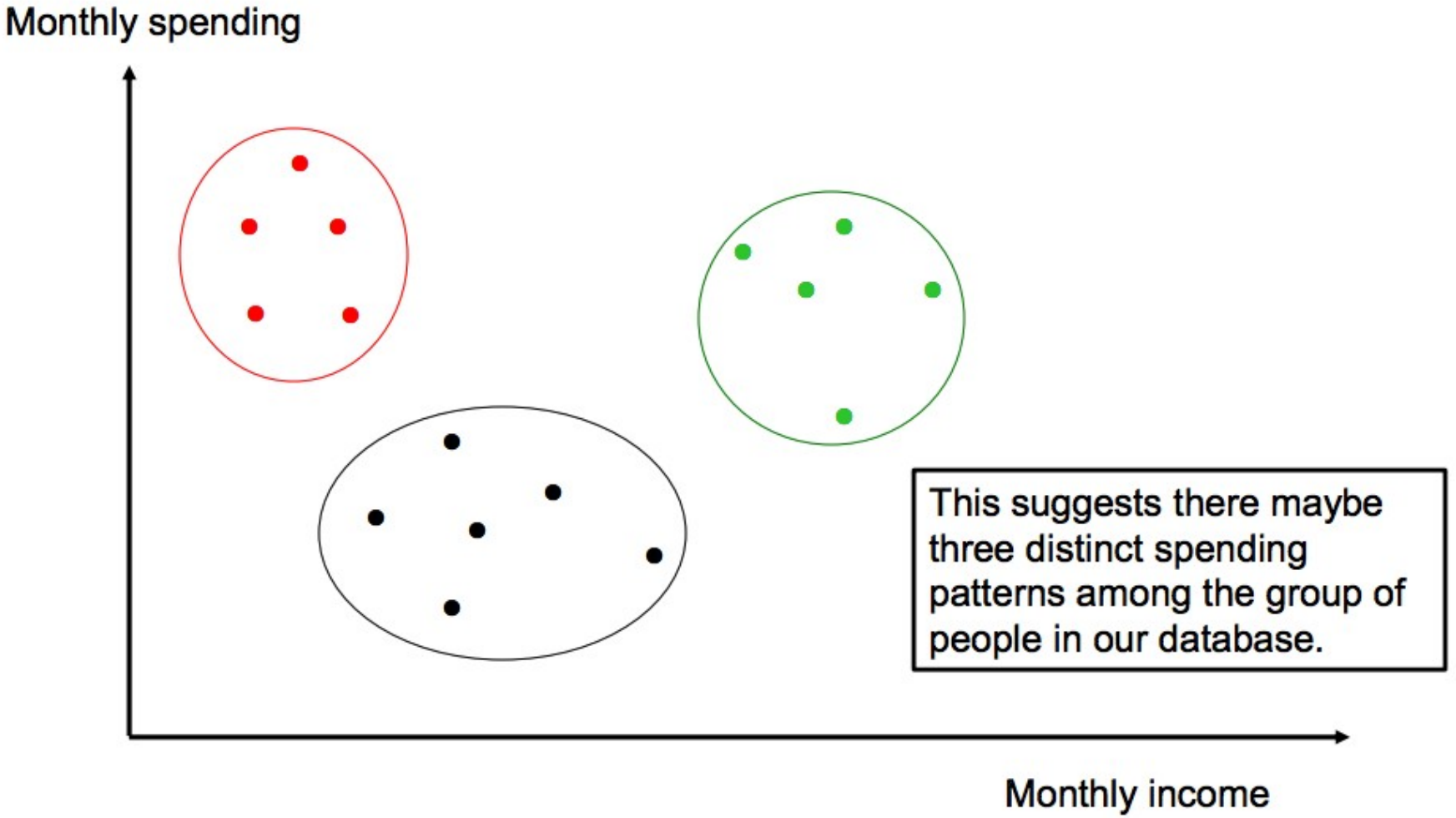
# Applications

- Information retrieval: cluster retrieved documents to present more organized and understandable results
- Consumer market analysis: cluster consumers into different interest groups so that marketing plans can be specifically designed for each individual group
- Image segmentation: decompose an image into regions with coherent color and texture
- Vector quantization for data (i.e., image) compression: group vectors into similar groups, and use group mean to represent group members
- Computational biology: group gene into co-expressed families based on their expression profile using different tissue samples and different experimental conditions

# A Hypothetical Clustering Example

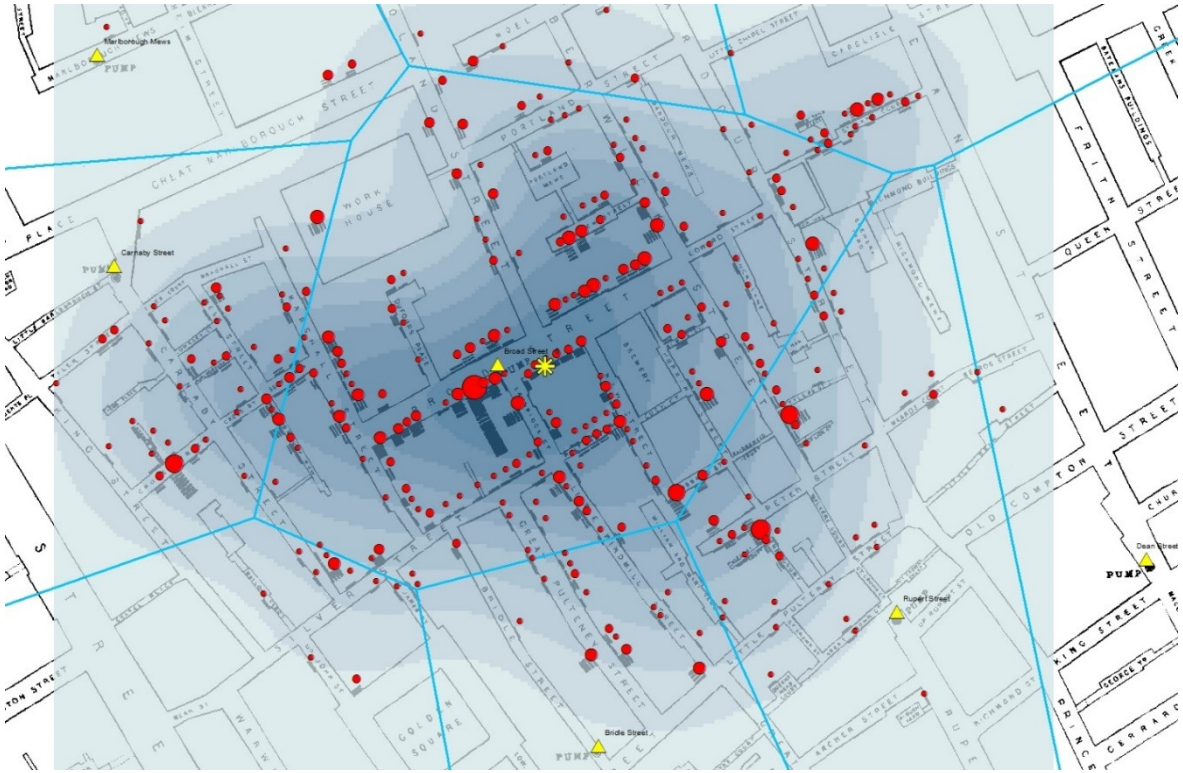


# A Hypothetical Clustering Example



# John Snow’s Cholera Study (1854)

- One of the earliest known examples of using clustering for data analysis is the pioneering study of cholera outbreak in Soho, London in 1854.
- Collected data on cholera instances and their proximity to water pumps.
- Outbreaks were clustered around water pumps with some exceptions.



# Important Components in Clustering

- Distance/similarity measures (aka proximity measures)
  - Measures to determine how similar or different are instances to one another.

- Clustering algorithm
  - Algorithm to find the clusters based on the distance/similarity measures.

- Evaluation Criteria
  - Metrics to tell if one form of clustering is better than another form.

# Distance/similarity Measures

- One of the most important question in unsupervised learning, often more important than the choice of clustering algorithms.

- What is similarity?
  - Similarity is hard to define, but we know it when we see it.
  - More pragmatically, there are many mathematical definitions of distance/similarity

# Distance/Similarity Measures

> Manhattan Distance:

$$d_1(\mathbf{p}, \mathbf{q}) = \|\mathbf{p} - \mathbf{q}\|_1 = \sum_{i=1}^{n} |p_i - q_i|,$$

> Euclidean Distance:

$$d(\mathbf{p}, \mathbf{q}) = d(\mathbf{q}, \mathbf{p}) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2}$$

$$= \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}.$$

> Minkowski Distance

$$D(X, Y) = \left( \sum_{i=1}^{n} |x_i - y_i|^p \right)^{1/p}$$

# Distance/Similarity Measures

- **Inner Product Space:** The angle between two vectors as a distance metric. It is especially useful when clustering high dimens

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}$$

- **Chebychev Distance:** Two vectors are different if they are different on any one of the attributes.

$$dist(\mathbf{x}_i, \mathbf{x}_j) = \max(|x_{i1} - x_{j1}|, |x_{i2} - x_{j2}|, \ldots, |x_{ir} - x_{jr}|)$$

We can always transform between distance and similarity using a monotonically decreasing function. For example. $e^{-\alpha D(x_1, x_2)^2}$

# Distance Metrics on Nominal Values

- **Hamming Distance:**
  The minimum number of *substitutions* required to change one string into the other
  $d_{ham}$(ka**rol**in,ka**thr**in) = 3, $d_{ham}$(10**1**0, 10**0**0) = 1

- **Convert to Numeric using one-hot encoding:**

| ID | Fruit |
|---|---|
| 1 | Apple |
| 2 | Apple |
| 3 | Orange |
| 4 | Banana |
| 5 | Orange |
| 6 | Apple |
| 7 | Orange |

| ID | Apple | Orange | Banana |
|---|---|---|---|
| 1 | 1 | 0 | 0 |
| 2 | 1 | 0 | 0 |
| 3 | 0 | 1 | 0 |
| 4 | 0 | 0 | 1 |
| 5 | 0 | 1 | 0 |
| 6 | 1 | 0 | 0 |
| 7 | 0 | 1 | 0 |

# How to decide which to use?

- It is application dependent.
- Consider your application domain, you need to ask questions such as:
  - What does it mean for two consumers to be similar to each other? Or for two genes to be similar to each other?
- For example, for text domain, we typically use cosine similarity.
- This may or may not give you the answer you want depends on your existing knowledge of the domain.

- Clustering is an exploratory procedure and it is important to explore – i.e., try different options and see what makes sense.

# Clustering Algorithms

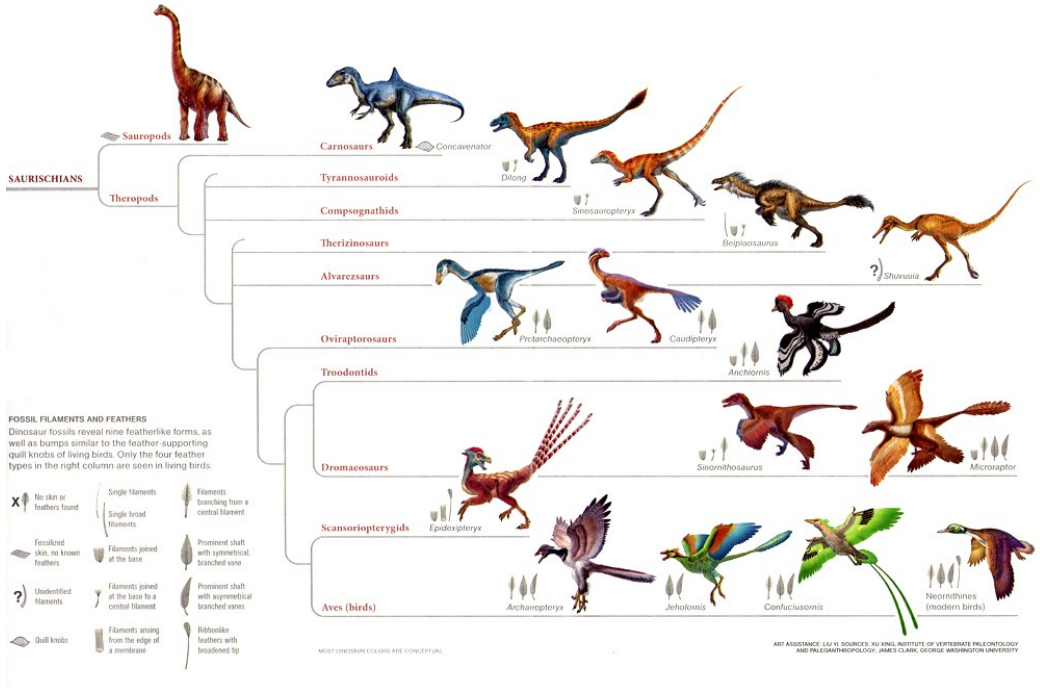
- **Hierarchical algorithms** build a tree-based hierarchical taxonomy (dendrogram).
  - Bottom up
  - Top down

- **Partition (Flat) algorithms** produce a single partition of the unlabeled data.
  - K-means
  - Mixture of Gaussian

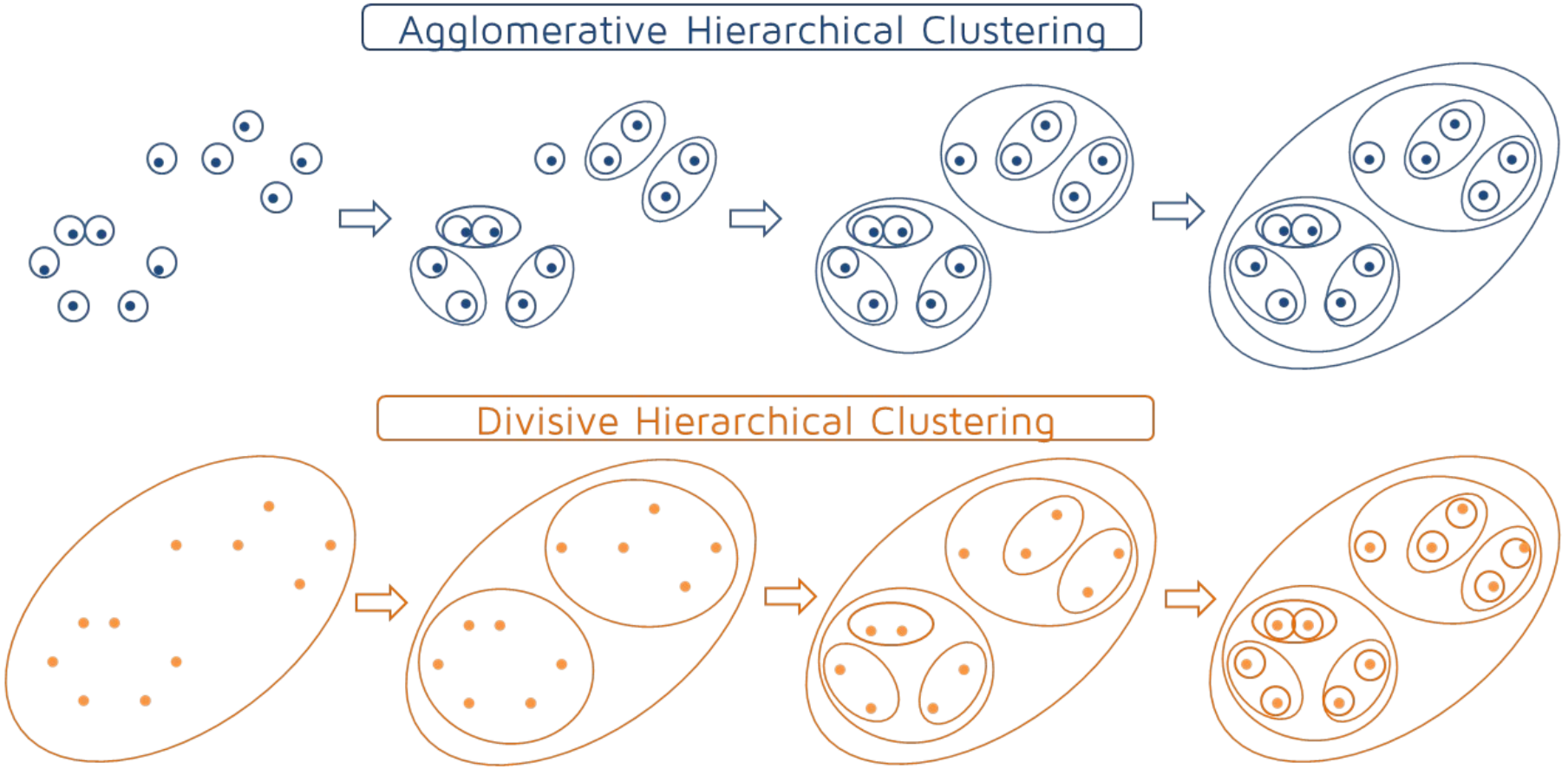
# Hierarchical Clustering

# Hierarchical Clustering

- **Goal:** Build a hierarchy of clusters
- **Agglomerative Clustering**: (Bottom up approach)
  - Each instance is its own cluster
  - Pairs of clusters are merged as one moves up the hierarchy until there is only one cluster
- **Divisive Clustering**: (Top Down approach)
  - All observations start as one cluster
  - Recursively split the data as one moves down the hierarchy until each instance is a cluster



# Types of Hierarchical Clustering

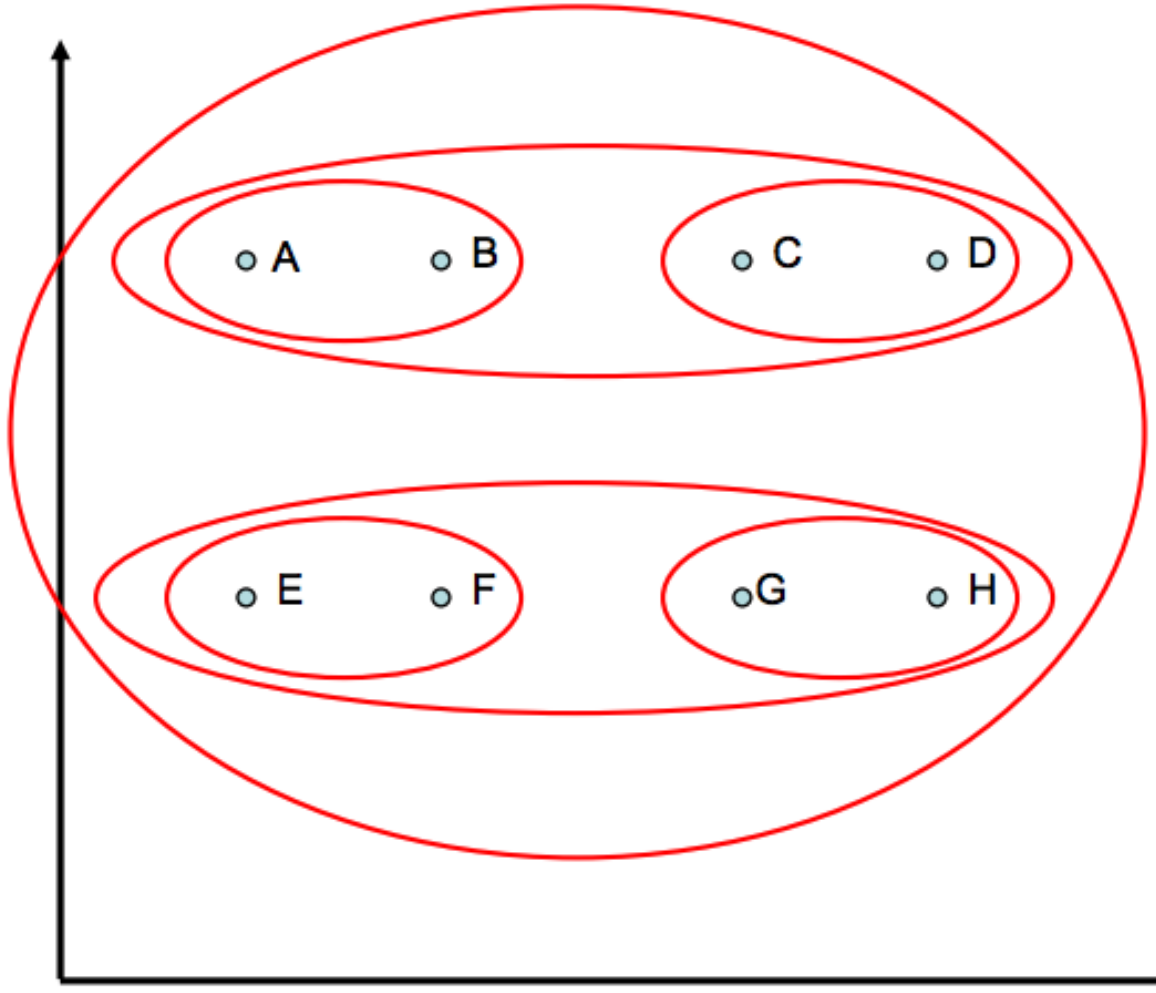


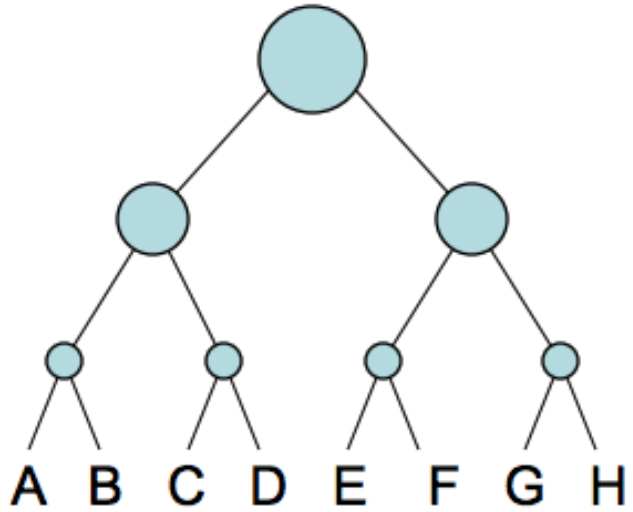
# Hierarchical Agglomerative Clustering Algorithm

Start with all objects in their own cluster.
Repeat until there is only one cluster:
    Among the current clusters, determine the two clusters, $c_i$ and $c_j$, that are closest
    Replace $c_i$ and $c_j$ with a single cluster $c_i \cup c_j$

**Problem**: we assume a distance/similarity function that computes distance/similarity between examples, but here we also need to compute distance between clusters.

# HAC Example



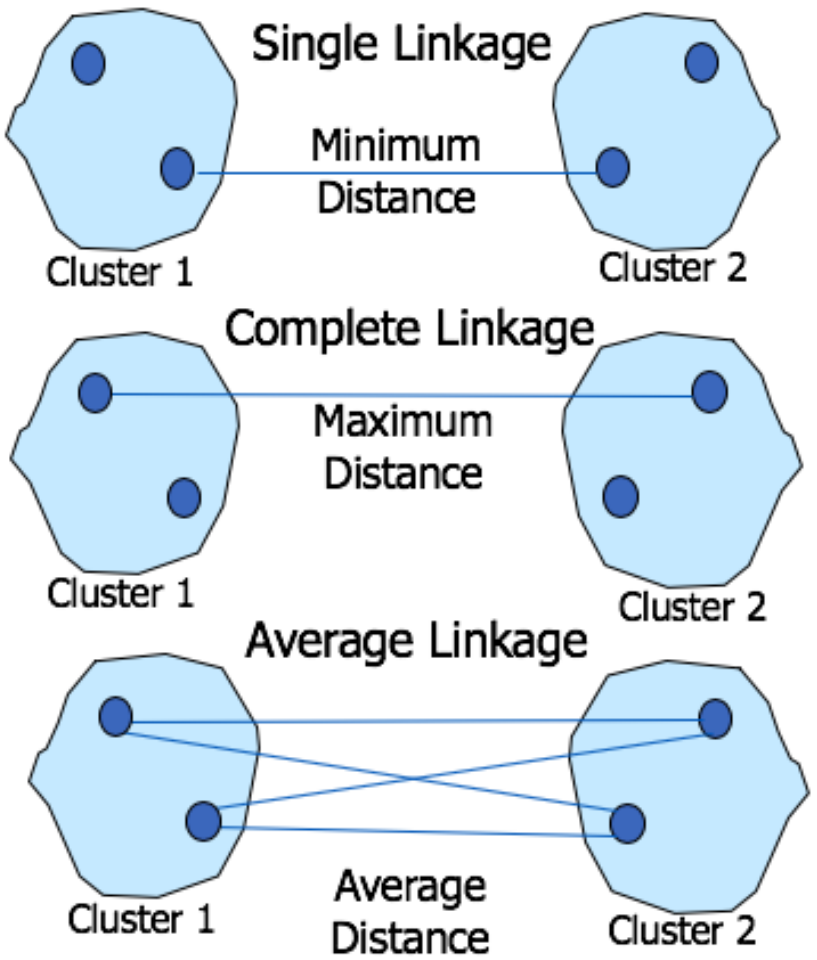



# Distance Between Clusters

- **Single Link:** The distance between two clusters is the distance between two closest points in the two clusters, one from each cluster. Can result in long chains of clusters

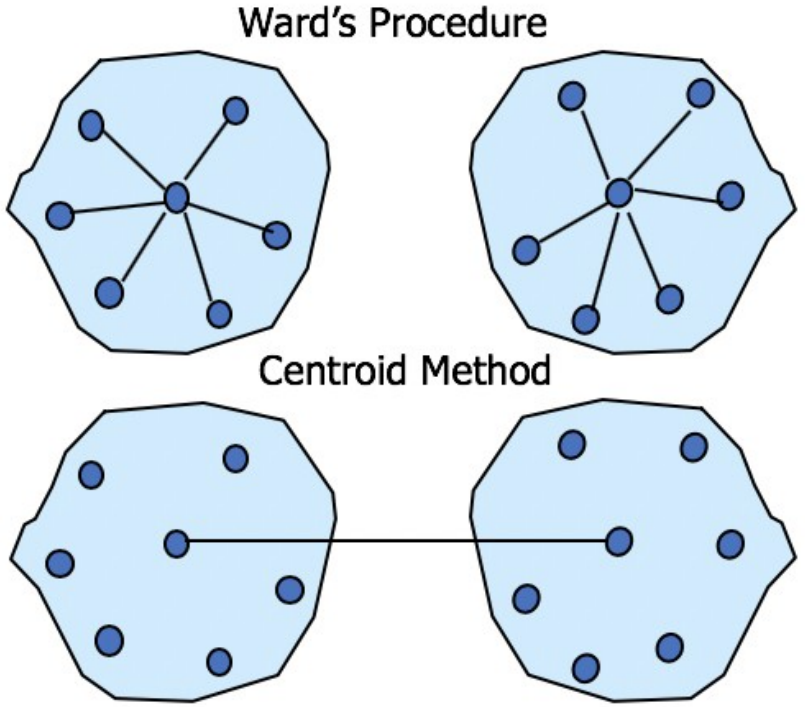
- **Complete Link:** The distance between two clusters is the distance of two furthest data points in the two clusters. Sensitive to outliers

- **Average Link:** the distance between two clusters is the average distance of all pair-wise distances between the data



# Distance Between Clusters

- **Ward's Procedure:** For each cluster calculate the sum of squares. The two clusters with the smallest increase in the overall sum of squares within cluster distances are combined.
- **Centroids Method:** The distance between two clusters is the distance between their centroids.




In general, the average linkage and Ward's methods have been shown to perform better than others.

# Updating the Cluster Distances

- After merging $c_i$ and $c_j$, the distance of the resulting cluster to any other cluster, $c_k$, can be computed by:
  - Single Link:

$$D((c_i \cup c_j), c_k) = \min(D(c_i, c_k), D(c_j, c_k))$$

  - Complete Link:

$$D((c_i \cup c_j), c_k) = \max(D(c_i, c_k), D(c_j, c_k))$$

# Updating the Cluster Distances con't

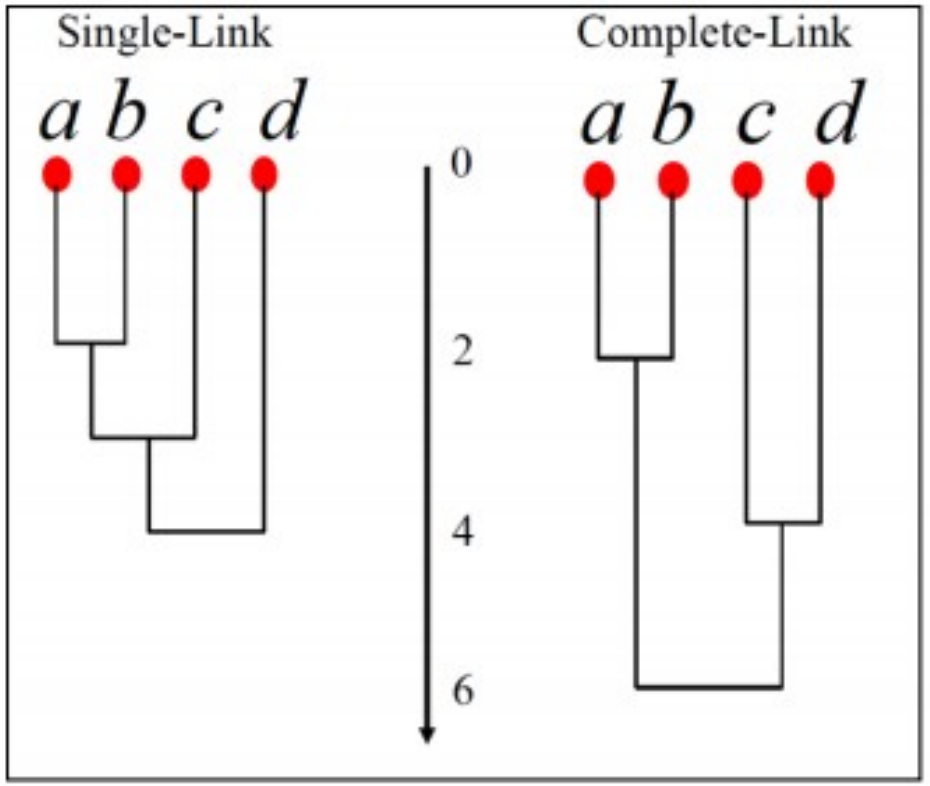
- Average Link: average similarity between members of the two clusters.
  - Averaged across all pairs between the original two clusters

$$D(c_i, c_j) = \frac{1}{|c_i||c_j|} \sum_{\mathbf{x} \in c_i} \sum_{\mathbf{x}' \in c_j} D(\mathbf{x}, \mathbf{x}')$$

- Compared to single link and complete link:
  - Computationally more expensive – $O(n_1 n_2)$
  - Achieves a compromise between single and complete link

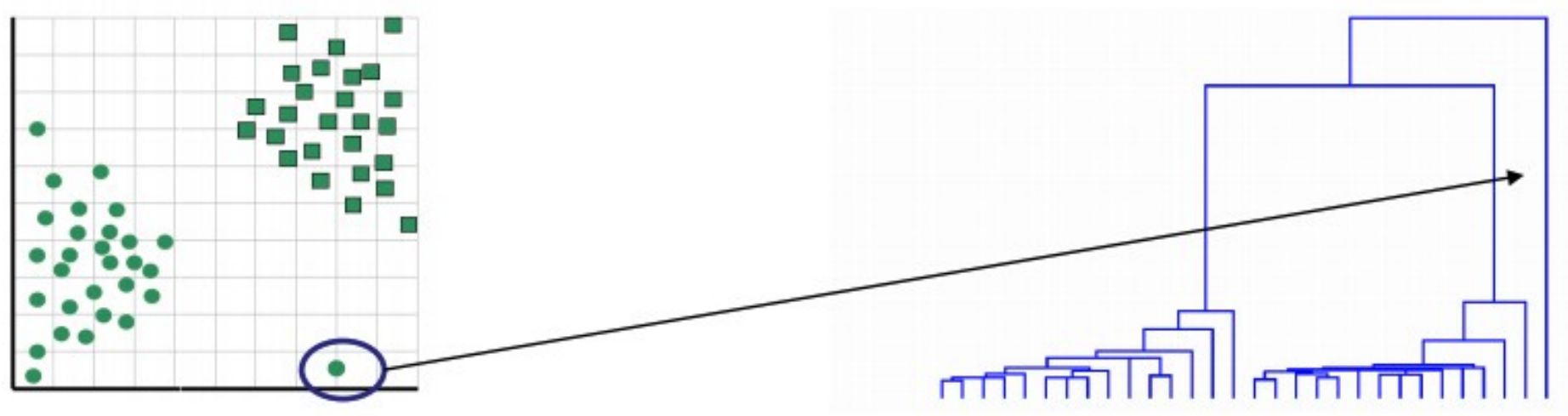
# Dendrogram: Cluster Visualization

- Height of the joint = the distance between the two merge clusters.
- The merge distance monotonically increases as we merge more and more for
  - Single, complete and average linkage methods
  - Not for the centroid method
- This can provide some understanding about how many natural groups there are in the data.
- A drastic height change indicates that we are merging two very different clusters together




This example is shown upside down.

# Interpreting Dendrogram



> Dendrogram can be used to identify
  - The number of clusters in data
  - Well-formed clusters
  - Outliers

# K-Means Clustering

# Goal of Clustering

- Given a data set $D = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$

- We need to partition $D$ into $k$ disjoint clusters, such that
  - Examples are self-similar in the same cluster
  - Examples are dissimilar across different clusters

- How do we quantify this?

# Objective: Sum of Squared Errors

> Given a partition of the data into k clusters, we can compute the **center** (i.e., mean, center of mass) of each cluster.

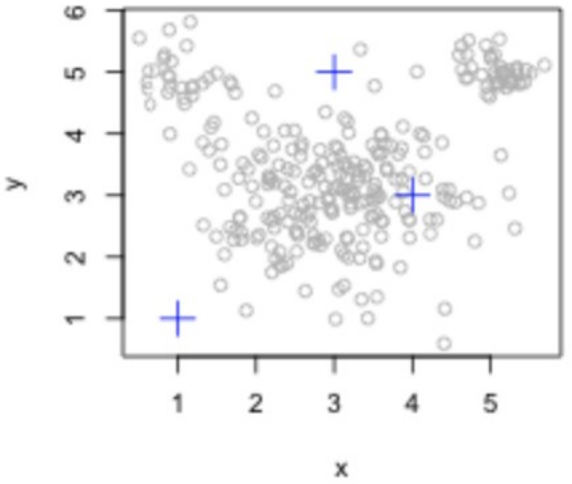
$$\mu_i = \frac{1}{n_i} \sum_{x \in C_i} x$$

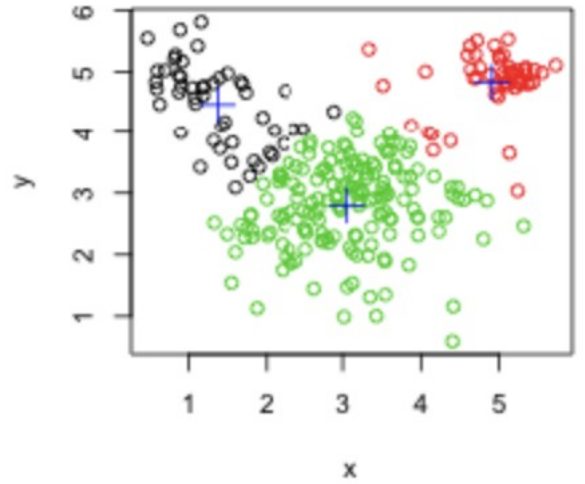
> For a well-formed cluster, its points should be close to its center. We measure this with sum of squared error (SSE), and formulate our objective to find a partition $\mathbb{C}^*$ that minimizes sum of squared error:

$$\mathbb{C}^* = \underset{\mathbb{C}=\{C_1,\ldots,C_k\}}{\operatorname{argmin}} \sum_{i=1}^{k} \sum_{x \in C_i} \|x - \mu_i\|^2$$

# Basic Idea

- We assume the number of desired clusters, k, is given.
- Randomly choose k examples as seeds, one for each cluster.
- Form initial clusters based on these seeds.
- Iterate by repeatedly re-allocating instances to different clusters to improve the overall clustering.
- Stop when clustering converges or after a fixed number of iterations.



# The K-means algorithm

**Input:** $D = \{x_1\ x_2\ \ldots x_n\}$ and desired number of clusters $k$

**Output:** a partition of D into $k$ disjoint clusters $c_1\ \ldots\ c_k$ (s.t. D = $c_1 \cup c_2 \cup \cdots \cup c_k$)

Let $\boldsymbol{d}$ be the distance function between examples

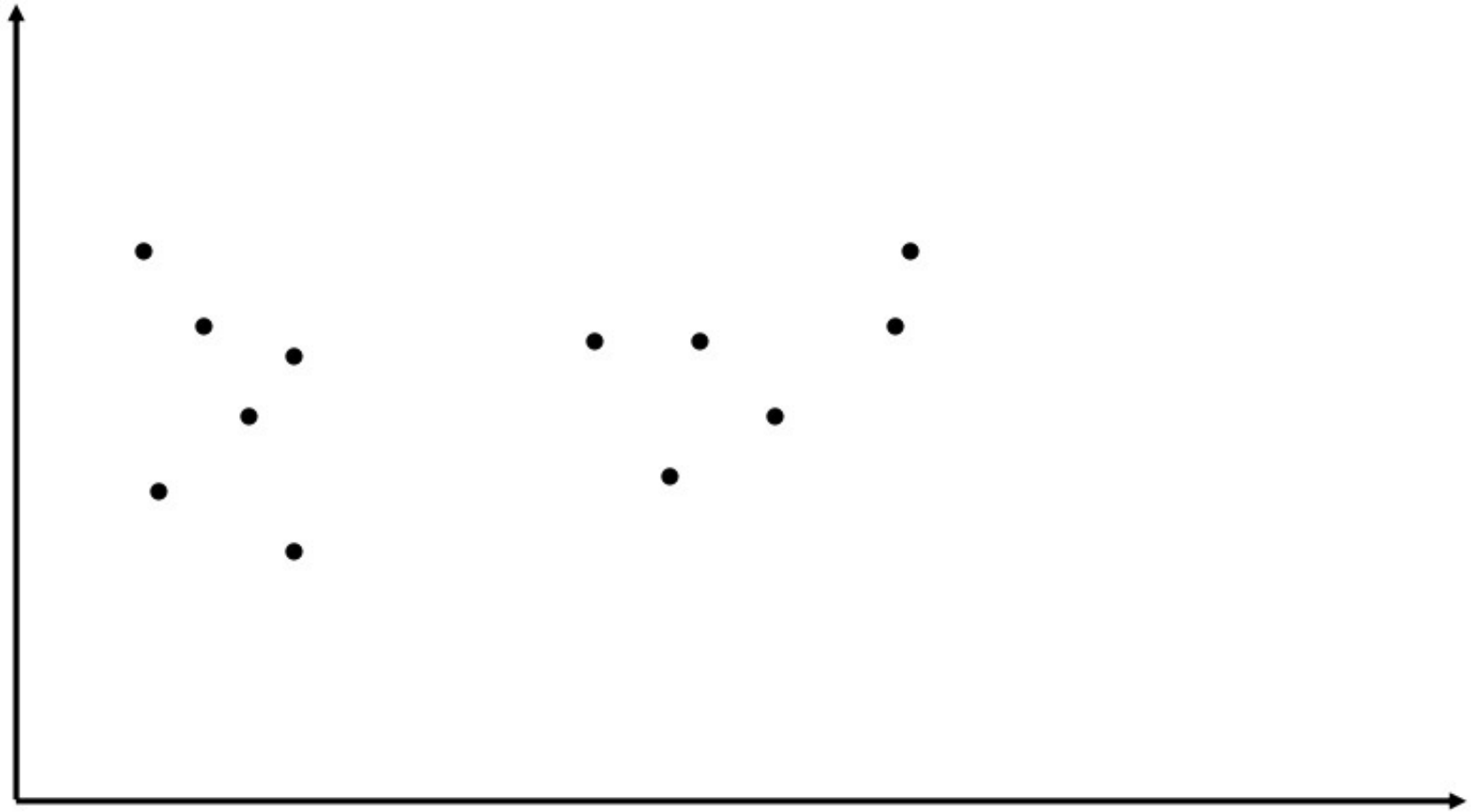
1. Select $k$ random samples from $D$ as centers $\{\mu_1\ \ldots \mu_k\}$ **//Initialization**
2. Do
3. for each example $\boldsymbol{x_i}$,
4. assign $\boldsymbol{x_i}$ to $\boldsymbol{c_j}$ such that $\boldsymbol{d}(\boldsymbol{\mu_j}, \boldsymbol{x_1})$ is minimized **// the Assignment step**
5. for each cluster $j$, update cluster center
6. $$\mu_j = \frac{1}{|c_j|}\sum_{\mathbf{x}\in c_j}\mathbf{x}$$ **// the update step**
7. Until convergence

# Convergence Criterions

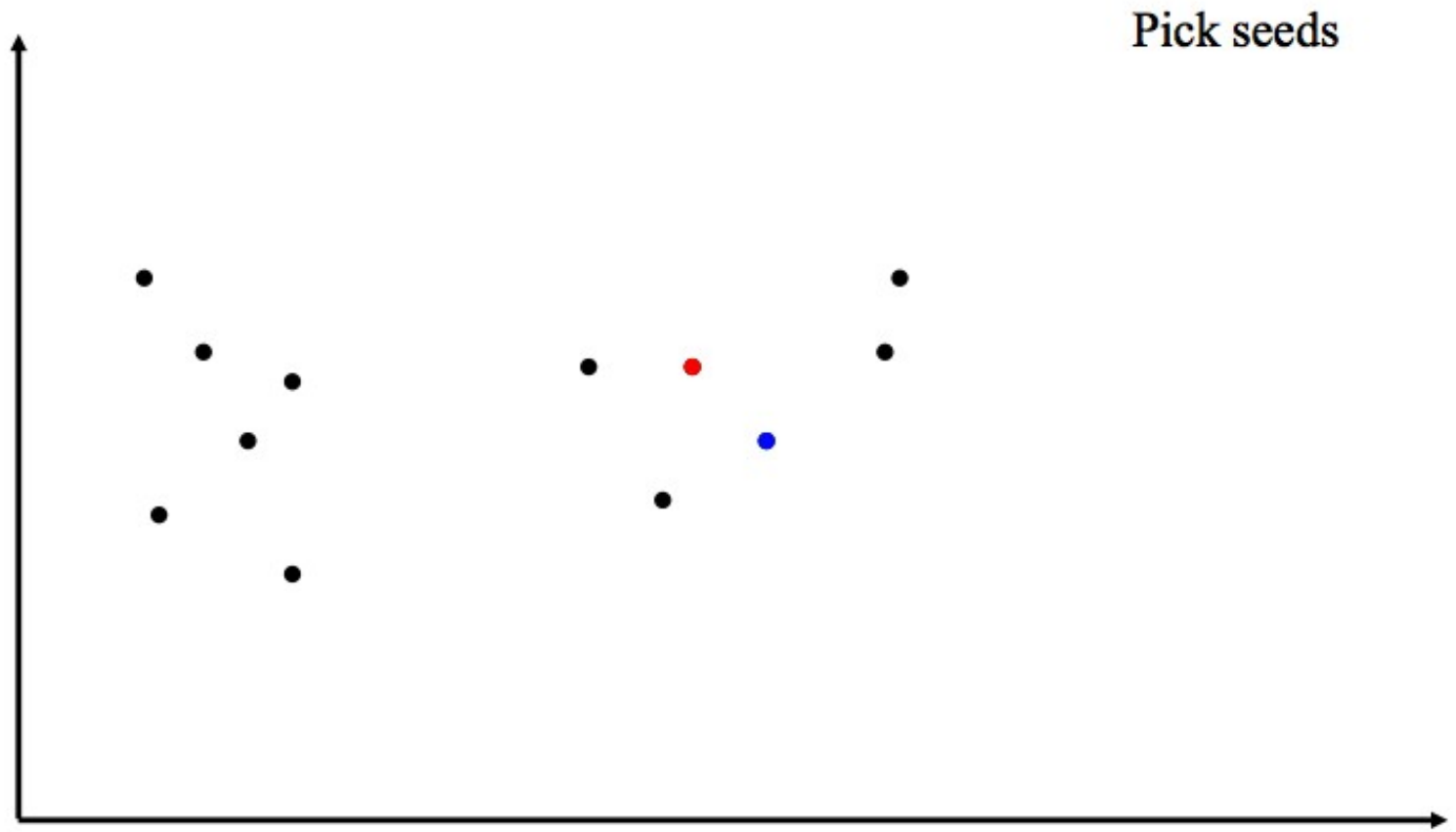
- No (or minimum) reassignments of data to different clusters
- No (or minimum) change of centroids
- Minimum decrease in the sum of squared error (Optimization Criteria)

$$J = \sum_{j=1}^{K} \sum_{n \in S_j} |x_n - \mu_j|^2$$

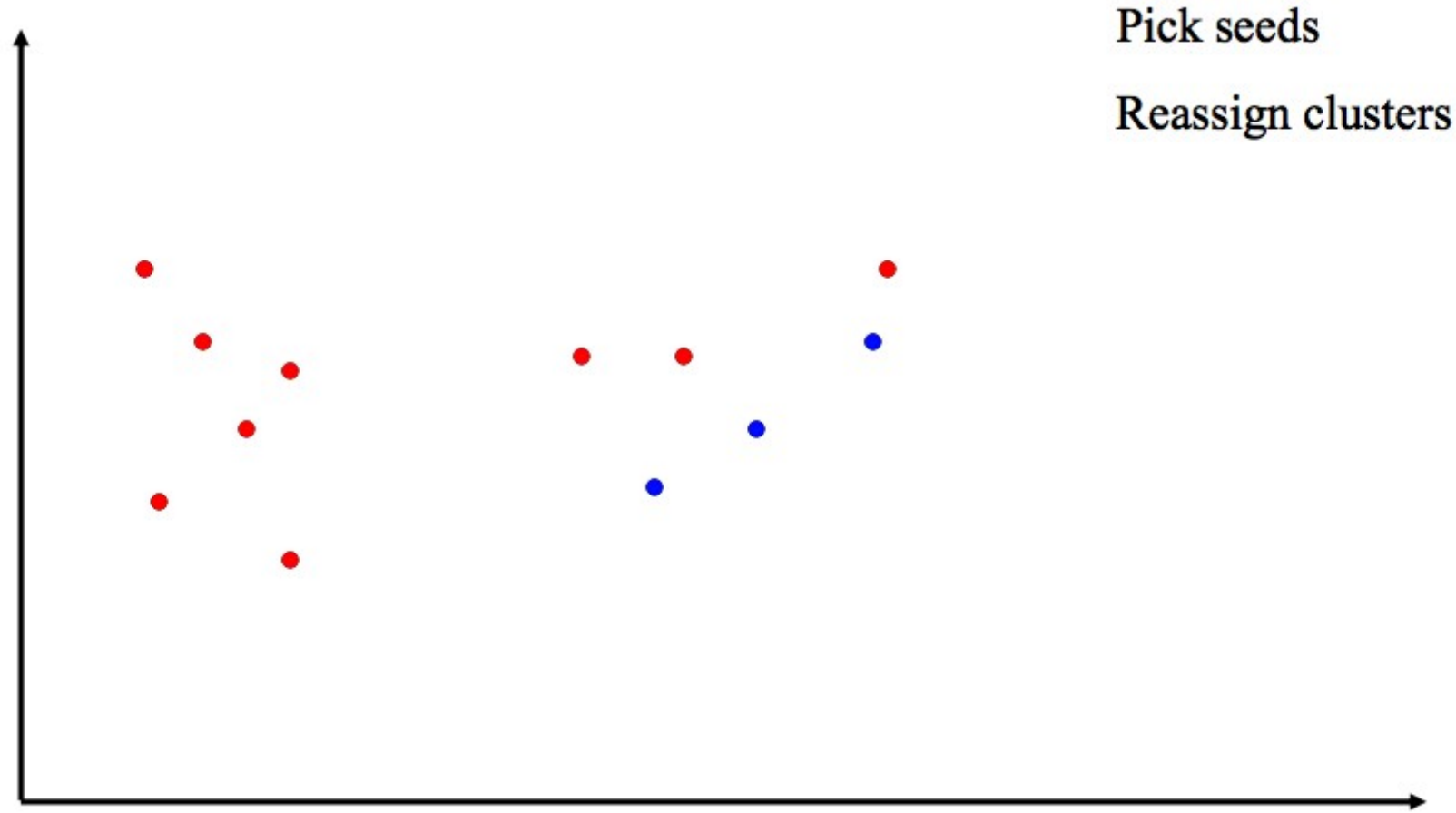
# K-Means Example



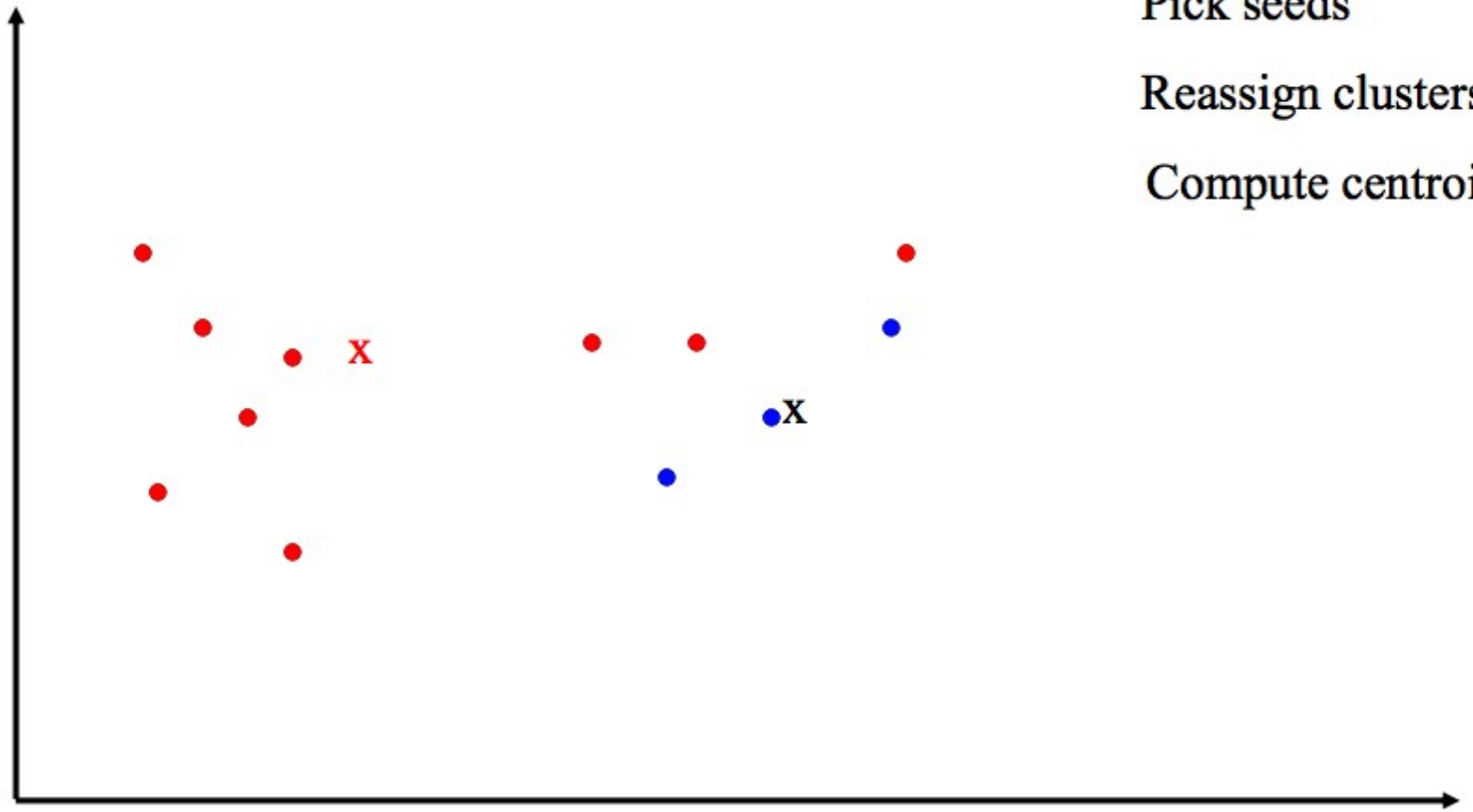
# K-Means Example



# K-Means Example



# K-Means Example

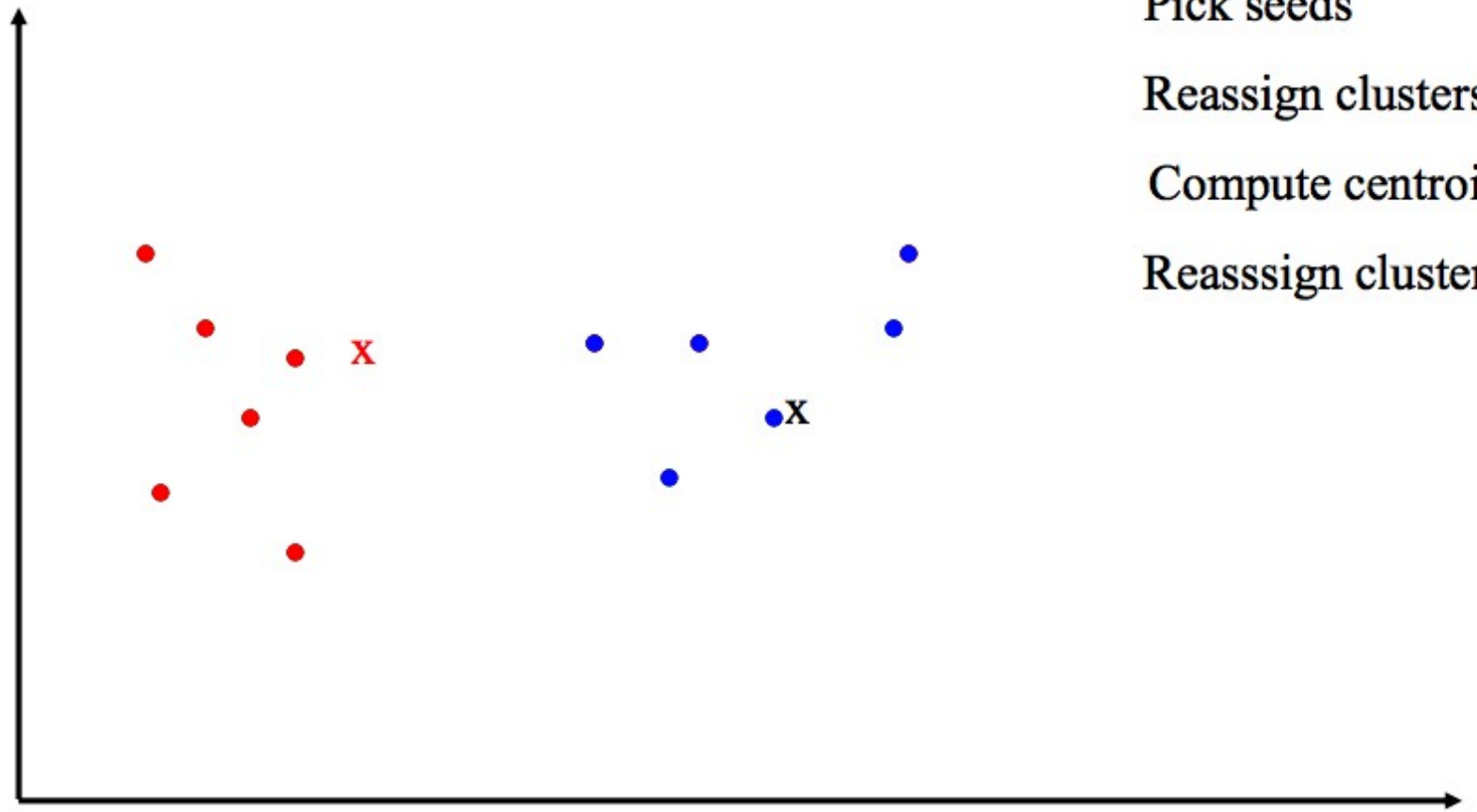


Pick seeds

Reassign clusters

Compute centroids

# K-Means Example



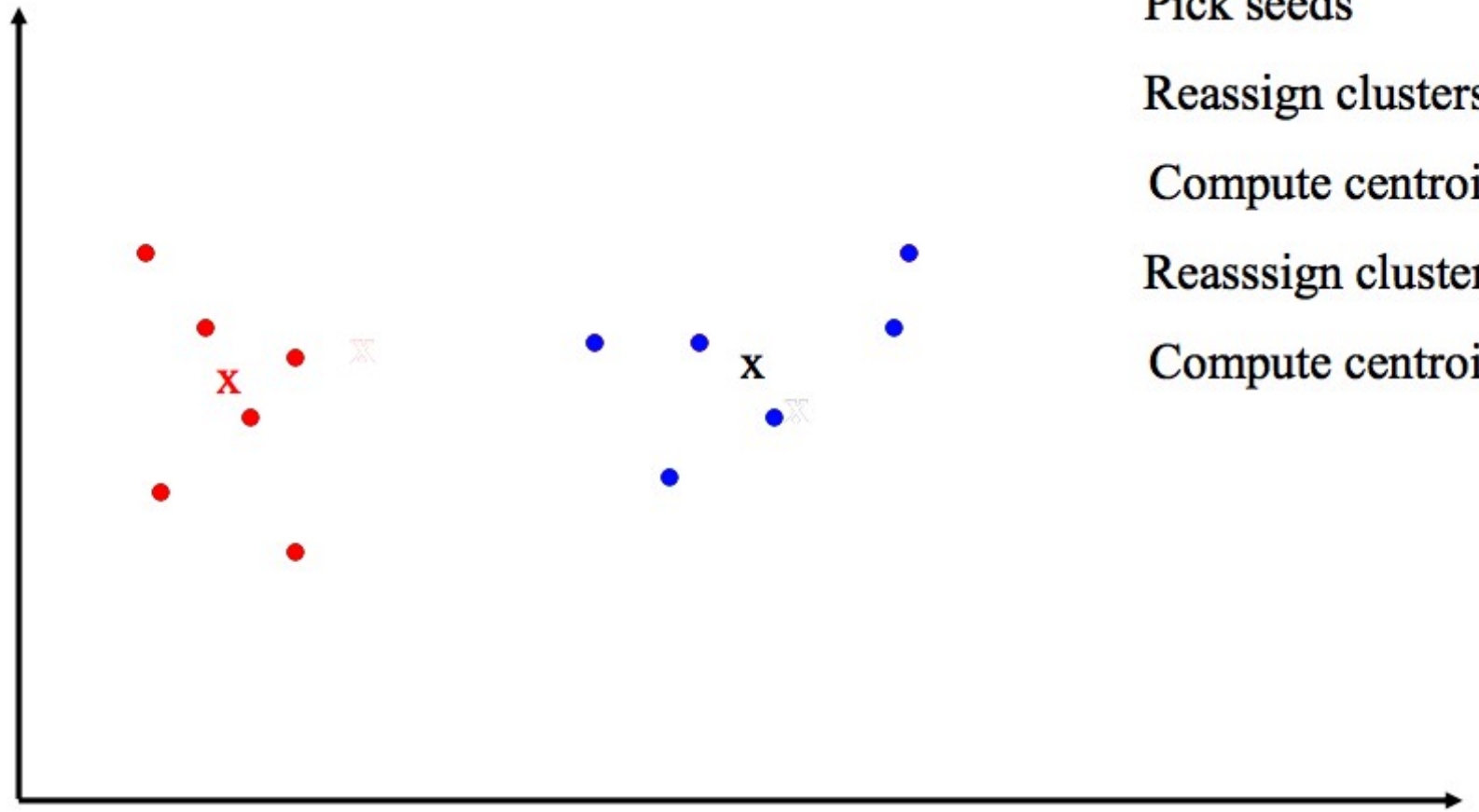
Pick seeds

Reassign clusters

Compute centroids

Reasssign clusters

# K-Means Example



Pick seeds

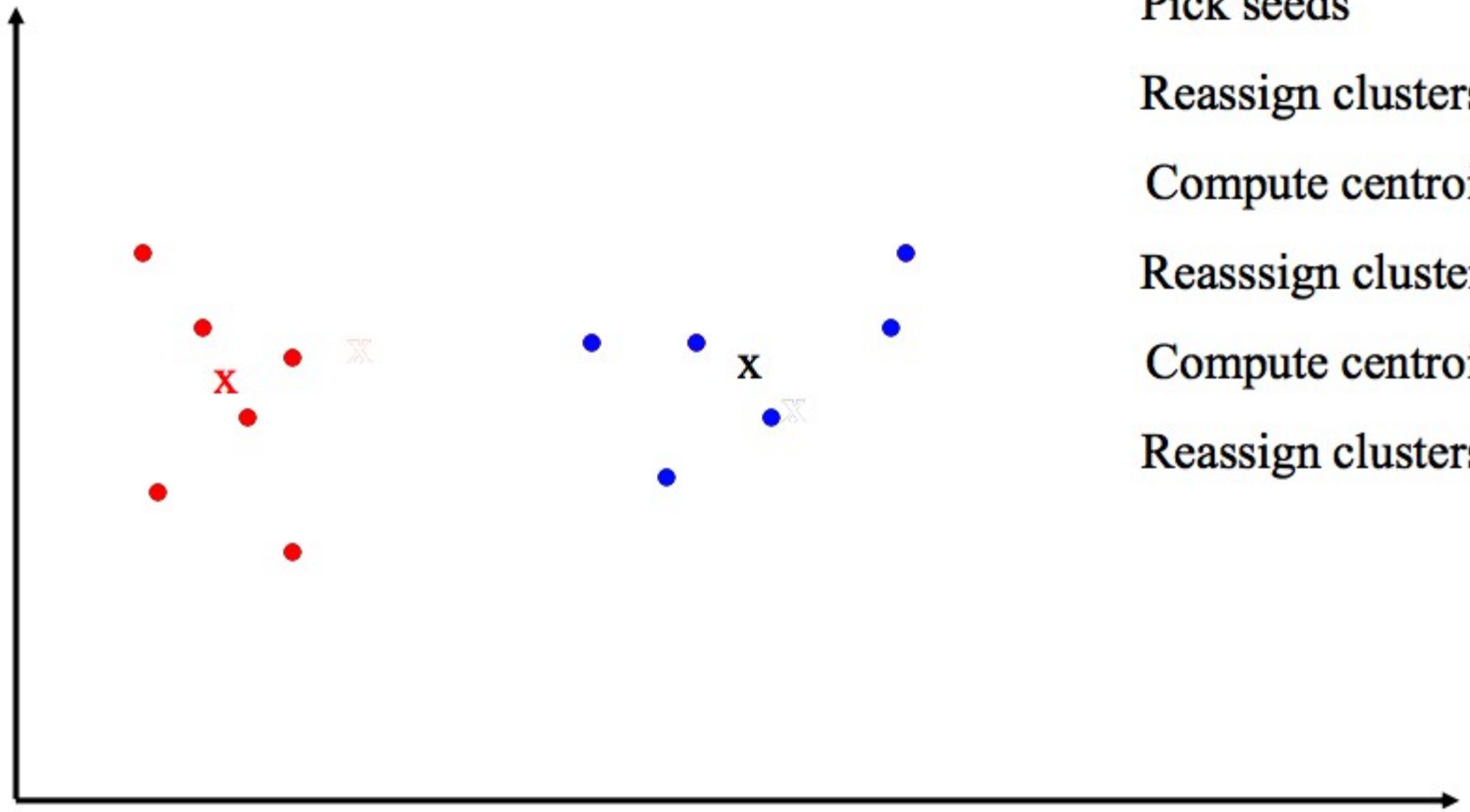
Reassign clusters

Compute centroids

Reasssign clusters

Compute centroids

# K-Means Example



Pick seeds

Reassign clusters

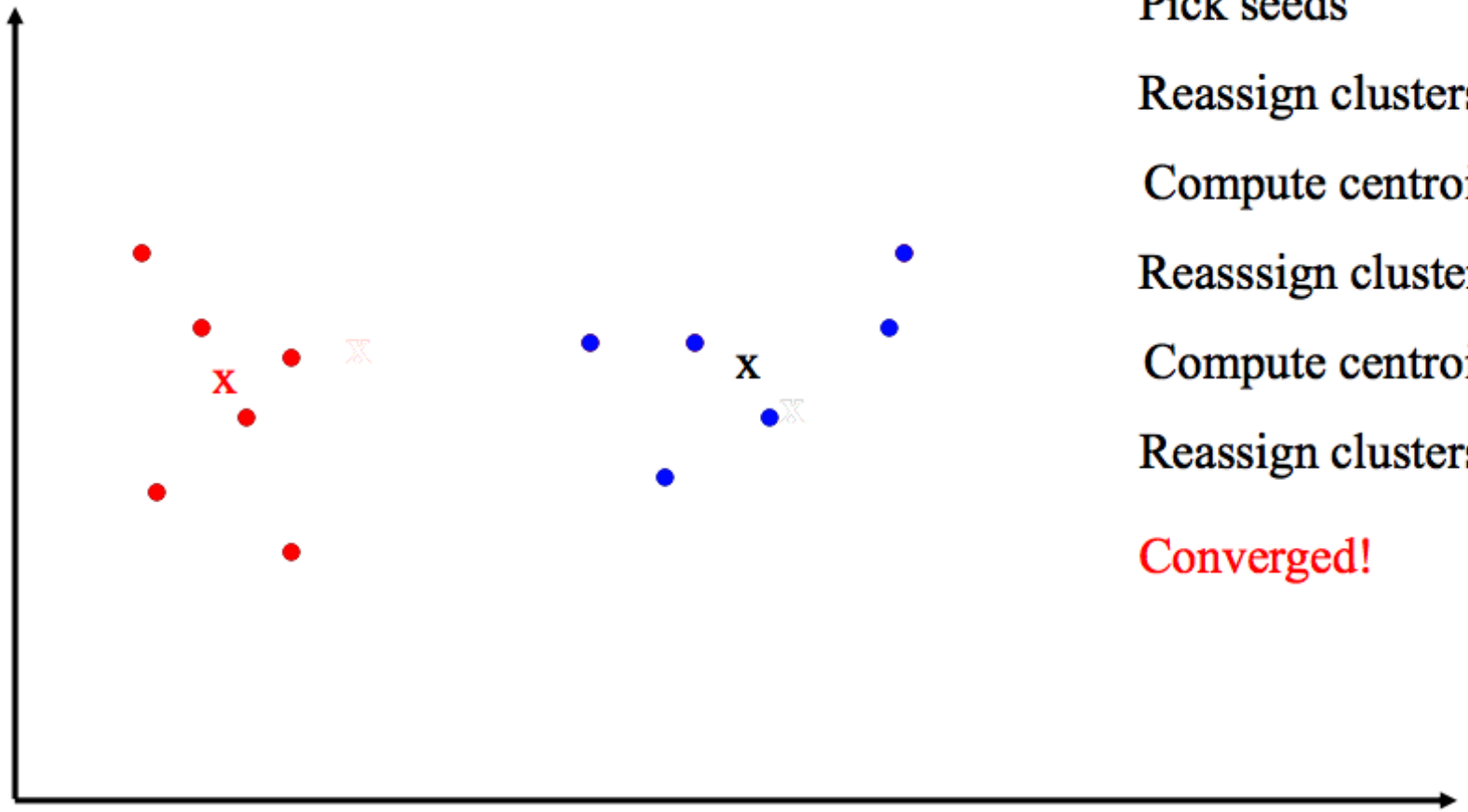
Compute centroids

Reasssign clusters

Compute centroids

Reassign clusters

# K-Means Example



Pick seeds

Reassign clusters

Compute centroids

Reasssign clusters

Compute centroids

Reassign clusters

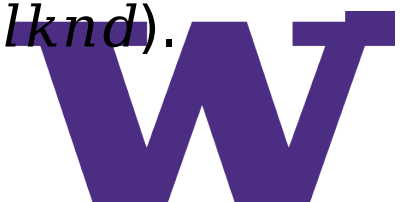
Converged!

# Convergence of K-means

- Monotonicity Property: Each iteration of K-means strictly decreases the SSE until convergence.

- K-means always converges in a finite number of steps. Typically converges very fast.

- Time complexity:
  - Assume computing distance between two instances is $O(d)$ where $d$ is the dimensionality of the vectors.
  - Reassigning clusters: $O(kn)$ distance computations, or $O(knd)$.
  - Computing centers: Each instance vector gets added once to some center: $O(nd)$.
  - Assume these two steps are each done once for $l$ iterations: $O(lknd)$.



# Weaknesses of K-Means: Local Minimal

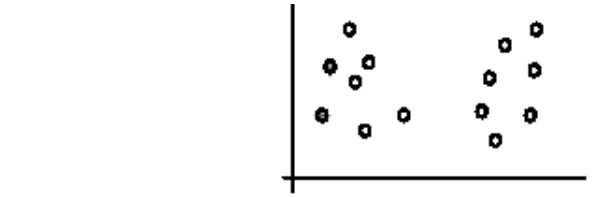
> K-means is very sensitive to initial conditions i.e., the initial assignment will determine the final set of clusters that one will get.

> This is because SSE has many local minimal solutions.

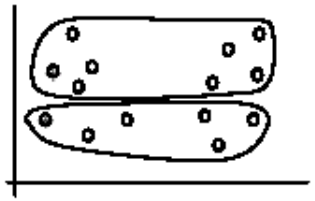
**Solutions**

> Run multiple trials and choose the one with the best SSE.

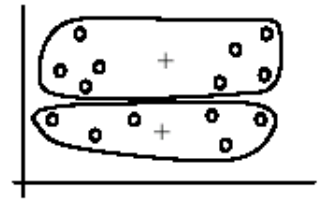
> Heuristics. Try to choose initial centers to be far apart (e.g. K-means++).



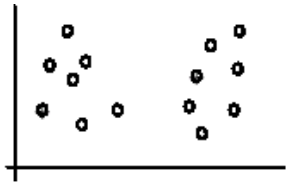
(A). Random selection of seeds (centroids)



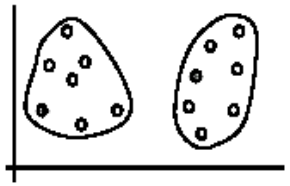
(B). Iteration 1



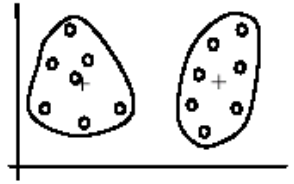
(C). Iteration 2



(A). Random selection of $k$ seeds (centroids)



(B). Iteration 1

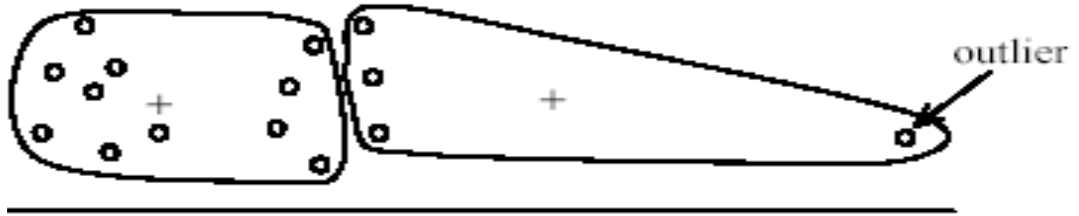


(C). Iteration 2

K-means++: http://ilpubs.stanford.edu:8090/778/1/2006-13.pdf

# Weaknesses of K-Means: Outliers

- K-means is very sensitive to outliers.
- The presence of outliers can lead to very *unreasonable* clusters.
- *Solution:*
  - Remove possible outliers before running K-means.
  - K-medoids




(A): Undesirable clusters


(B): Ideal clusters

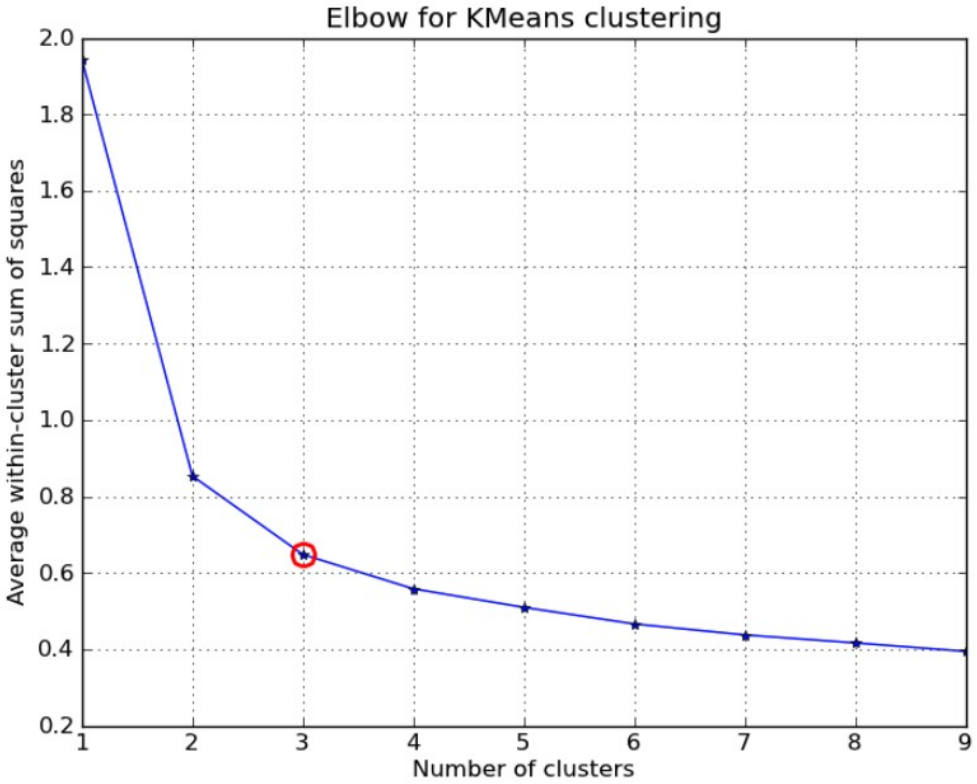
# K-medoids

> K-medoids – use the medoid for each cluster, i.e., the data point that is closest to other points in the cluster.


> K-medoids is computationally more expensive but more robust to outliers

# Selection of K

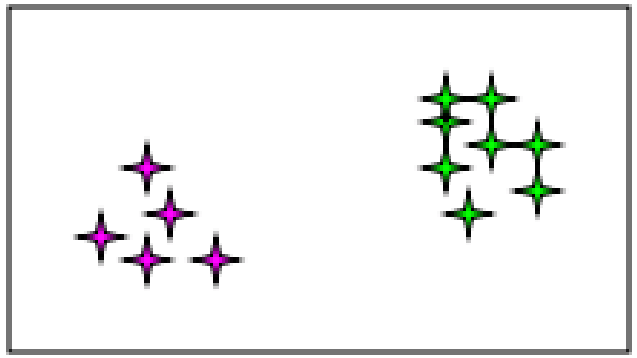
- What if we don’t know how many clusters there are in the data?
- Can we use SSE to decide k by choosing k that gives the smallest SSE?
  - We will always favor larger k values
- Heuristic: find the knee

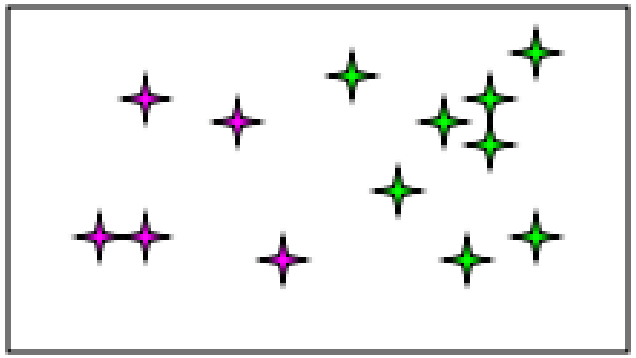


# Evaluating clustering

# Evaluating Clusters: Internal Evaluation

- By user interpretation
  - does a document cluster seem to correspond to a specific topic?
- Internal criterion: a good clustering will produce high quality clusters
  - high within-cluster similarity:
  - Low between-cluster similarity:





# External Indexes: Rand Index

> If true class labels (ground truth) are known, the validity of a clustering can be verified by comparing the class labels and clustering labels.

> Given partition (P) and ground truth (G), measure the number of vector pairs that are:

- a: in the same class both in P and G.
- b: in the same class in P, but in the different class in G.
- c: in different classes in P, but in the same class in G.
- d: in different classes both in P and G.

$$RI = \frac{a+d}{a+b+c+d}$$

# Normalized Mutual Information

$$NMI(Y, C) = \frac{2 \times I(Y; C)}{[H(Y) + H(C)]}$$

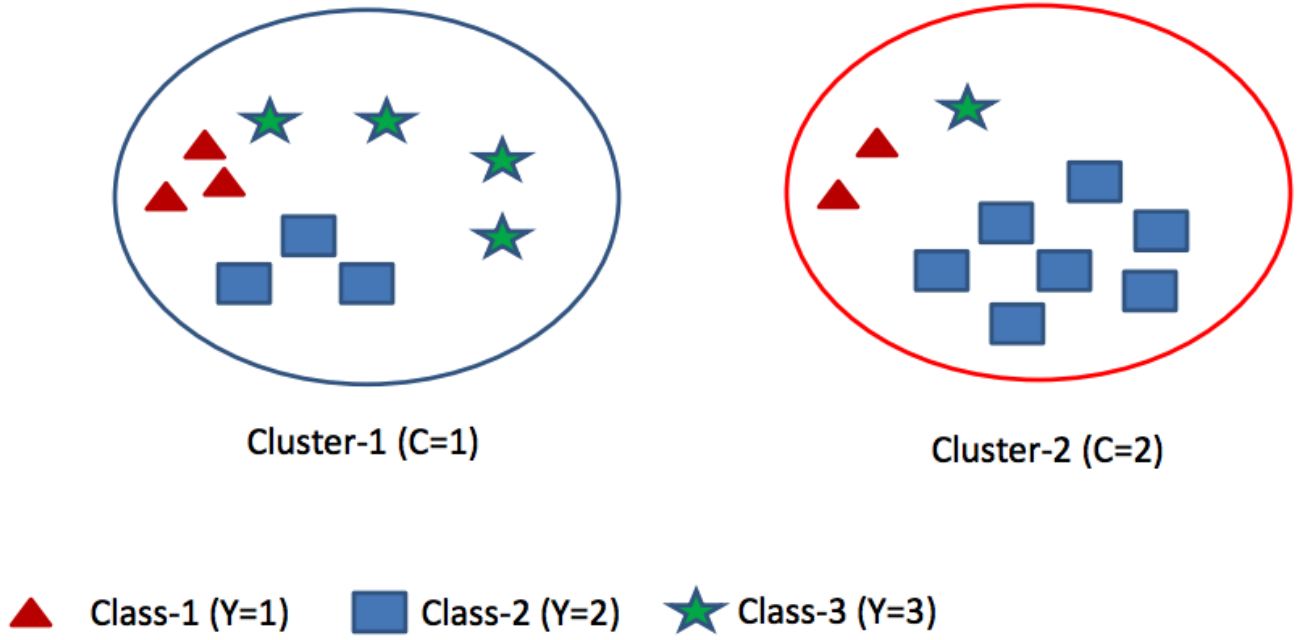
where,

1) Y = class labels
2) C = cluster labels
3) H(.) = Entropy
4) I(Y;C) = Mutual Information b/w Y and C

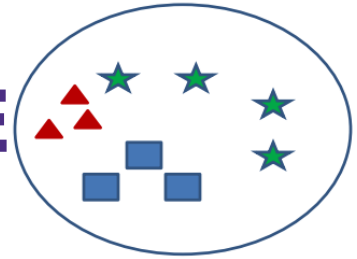
Note: All logs are base-2.

# Normalized Mutual Information Example

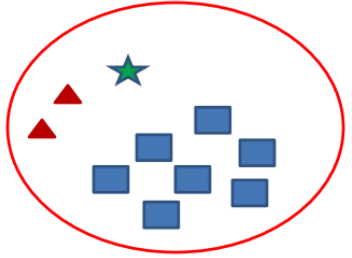
- Assume m=3 classes and k=2 clusters



# Normalized Mutual Information E



Cluster-1 (C=1)

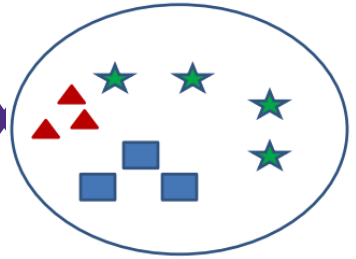


Cluster-2 (C=2)

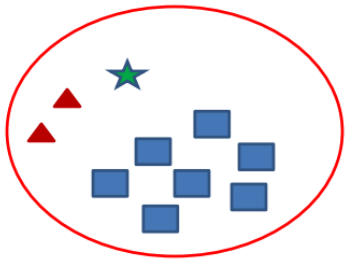
## H(Y) = Entropy of Class Labels

- P(Y=1) = 5/20 = ¼
- P(Y=2) = 5/20 = ¼
- P(Y=3) = 10/20 = ½
- $H(Y) = -\frac{1}{4}\log\left(\frac{1}{4}\right) - \frac{1}{4}\log\left(\frac{1}{4}\right) - \frac{1}{2}\log\left(\frac{1}{2}\right) = 1.5$

# Normalized Mutual Information Ex



Cluster-1 (C=1)



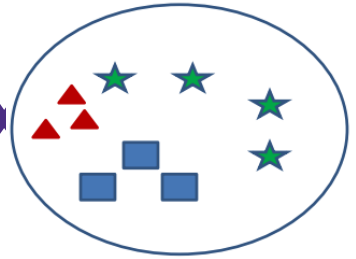
Cluster-2 (C=2)

## H(C) = Entropy of Cluster Labels

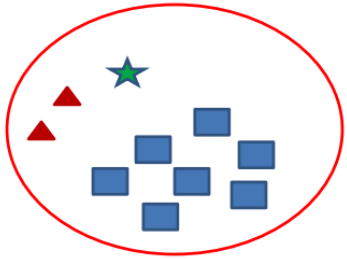
- P(C=1) = 10/20 = 1/2
- P(C=2) = 10/20 = ½
- $H(Y) = -\frac{1}{2}\log\left(\frac{1}{2}\right) - \frac{1}{2}\log\left(\frac{1}{2}\right) = 1$

This will be calculated every time the clustering changes. You can see from the figure that the clusters are balanced (have equal number of instances).

# Normalized Mutual Information Ex



Cluster-1 (C=1)



Cluster-2 (C=2)

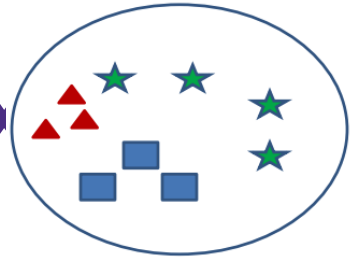
## H(Y|C): conditional entropy of class labels for clustering C

- Consider Cluster-1:
  - P(Y=1|C=1)=3/10 (three triangles in cluster-1)
  - P(Y=2|C=1)=3/10 (three rectangles in cluster-1)
  - P(Y=3|C=1)=4/10 (four stars in cluster-1)
  - Calculate conditional entropy as:

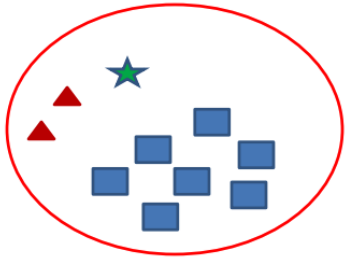
$$H(Y|C=1) = -P(C=1) \sum_{y\in\{1,2,3\}} P(Y=y|C=1)\log(P(Y=y|C=1))$$

$$= -\frac{1}{2} \times \left[\frac{3}{10}\log\left(\frac{3}{10}\right) + \frac{3}{10}\log\left(\frac{3}{10}\right) + \frac{4}{10}\log\left(\frac{4}{10}\right)\right] = 0.7855$$

# Normalized Mutual Information Ex



Cluster-1 (C=1)



Cluster-2 (C=2)

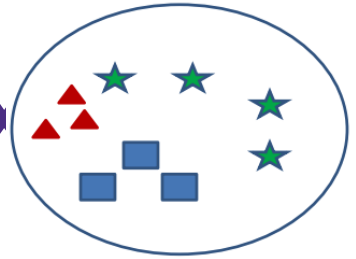
## H(Y|C): conditional entropy of class labels for clustering C

- Now, consider Cluster-2:
  - P(Y=1|C=2)=2/10 (two triangles in cluster-1)
  - P(Y=2|C=2)=7/10 (seven rectangles in cluster-1)
  - P(Y=3|C=2)=1/10 (one star in cluster-1)
  - Calculate conditional entropy as:

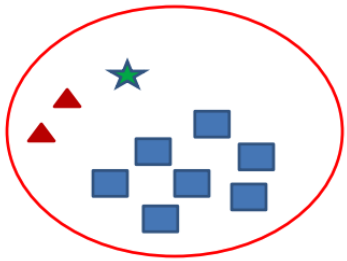
$$H(Y|C=2) = -P(C=2) \sum_{y\in\{1,2,3\}} P(Y=y|C=2)\log(P(Y=y|C=2))$$

$$= -\frac{1}{2} \times \left[\frac{2}{10}\log\left(\frac{2}{10}\right) + \frac{7}{10}\log\left(\frac{7}{10}\right) + \frac{1}{10}\log\left(\frac{1}{10}\right)\right] = 0.5784$$

# Normalized Mutual Information Ex



Cluster-1 (C=1)



Cluster-2 (C=2)

I(Y;C)

- Finally the mutual information is:

$$I(Y;C) = H(Y) - H(Y|C)$$
$$= 1.5 - [0.7855 + 0.5784]$$
$$= 0.1361$$

The NMI is therefore,

$$NMI(Y,C) = \frac{2 \times I(Y;C)}{[H(Y) + H(C)]}$$

$$NMI(Y,C) = \frac{2 \times 0.1361}{[1.5 + 1]} = 0.1089$$

# Summary on Clustering Evaluation

- Internal evaluation
  - Within-cluster similarity, larger the better
  - Between-cluster similarity, smaller the better
  - Potential issue: Dependent on how similarity is measured

- External evaluation (compare against a ground truth clustering)
  - Rand Index (very sensitive to the number of clusters)
  - Normalized mutual information (less sensitive to the number of clusters)
  - Potential issue: there may not be only one way to cluster the data

# Conclusion

- There are a massive number of clustering algorithms.
- Clustering is hard to evaluate, but very useful in practice.
- Clustering is highly application dependent and to some extent subjective.
- There is no one universal recipe for choosing a clustering technique and its associated parameters.

# Appendix