



Introduction to

# **Machine Learning and Data Mining**

(Học máy và Khai phá dữ liệu)

**Khoat Than**

School of Information and Communication Technology  
Hanoi University of Science and Technology

2020

# Contents

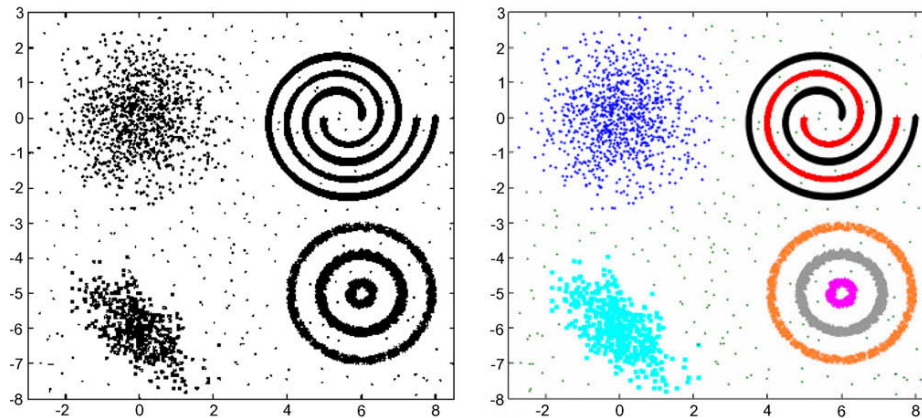
---

- Introduction to Machine Learning & Data Mining
- Supervised learning
- **Unsupervised learning**
  - **Clustering**
- Practical advice

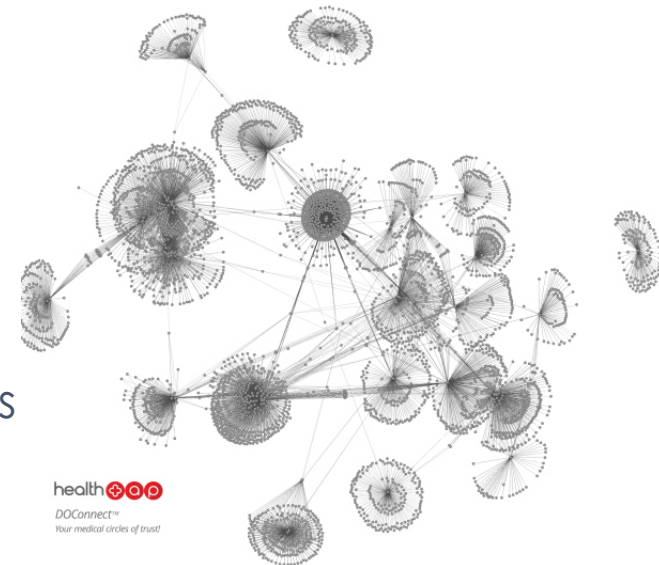
-

# Unsupervised learning: examples (1)

- Clustering data into clusters
  - Discover the data groups/clusters



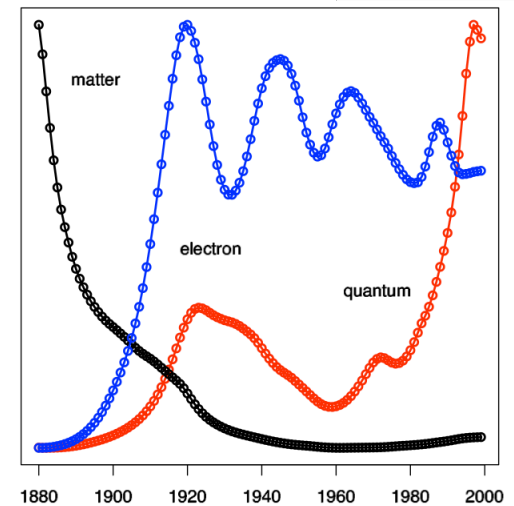
- Community detection
  - Detect communities in online social networks



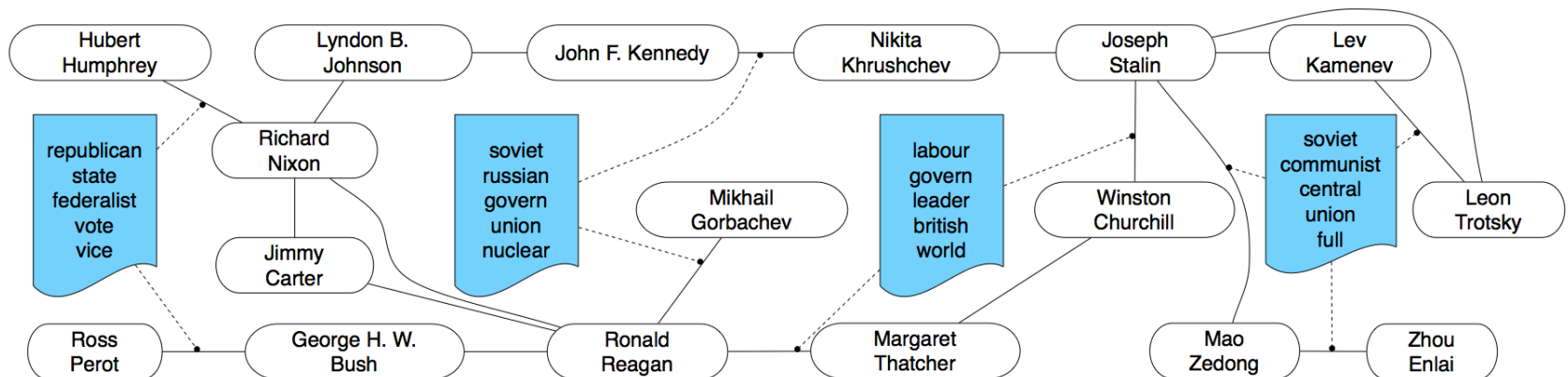
# Unsupervised learning: examples (2)

## ■ Trends detection

- Discover the trends, demands, future needs of online users



## ■ Entity-interaction analysis



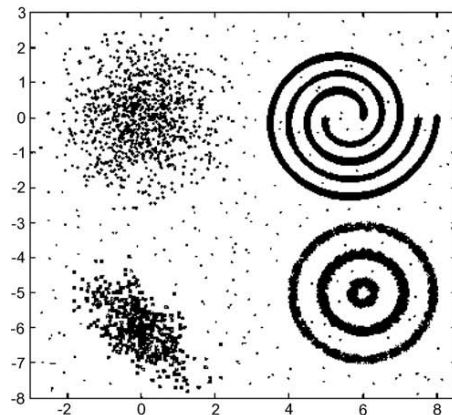
## 2. Clustering

- Clustering problem:

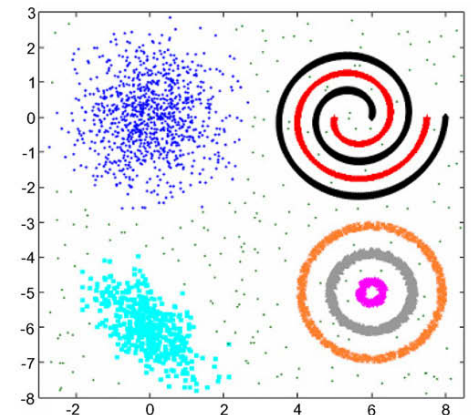
- Input: a training set without any label.
- Output: clusters of the training instances

- A cluster:

- Consists of similar instances in some senses.
- Two clusters should be different from each other.



After clustering



# Clustering

---

## ■ Approaches to clustering

- Partition-based clustering
- Hierarchical clustering
- Mixture models
- Deep clustering
- ...

## ■ Evaluation of clustering quality

- Distance/difference between any two clusters should be large.  
(inter-cluster distance)
- Difference between instances inside a cluster should be small.

### 3. K-means for clustering

---

- K-means was first introduced by Lloyd in 1957.
- K-means is the most popular method for clustering, which is partition-based.
- Data representation:  $D = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_r\}$ , each  $\mathbf{x}_i$  is a vector in the  $n$ -dimensional Euclidean space.
- K-means partitions  $D$  into  $K$  clusters:
  - Each cluster has a central point which is called **centroid**.
  - $K$  is a pre-specified constant.

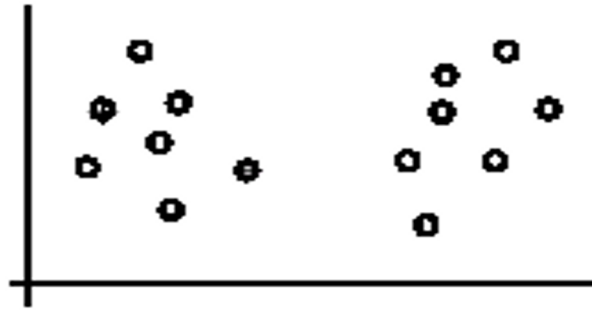


# K-means: main steps

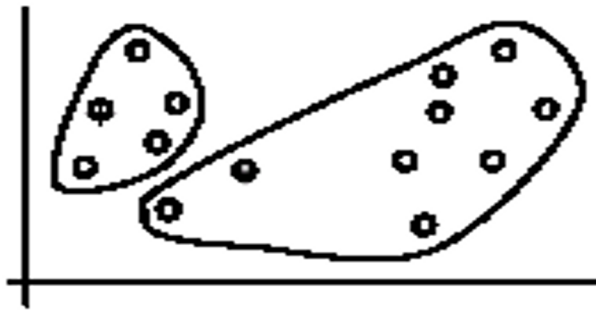
---

- *Input*: training data **D**, number K of clusters, and distance measure  $d(x,y)$ .
- *Initialization*: select randomly K instances in **D** as the initial centroids.
- Repeat the following two steps until convergence
  - *Step 1*: for each instance, assign it to the cluster with nearest centroid.
  - *Step 2*: for each cluster, recompute its centroid from all the instances assigned to that cluster.

# K-means: example (1)



(A). Random selection of  $k$  centers



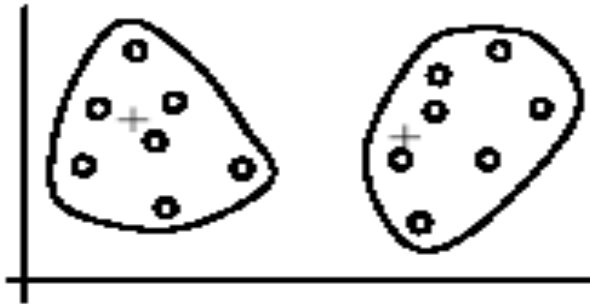
Iteration 1: (B). Cluster assignment



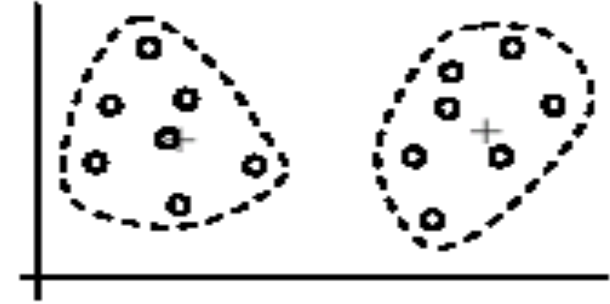
(C). Re-compute centroids

[Liu, 2006]

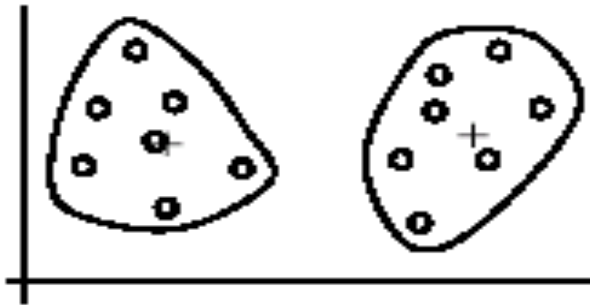
## K-means: example (2)



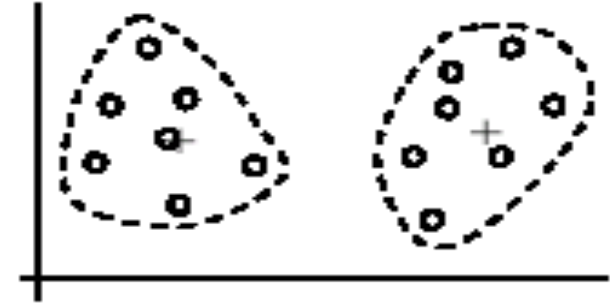
Iteration 2: (D). Cluster assignment



(E). Re-compute centroids



Iteration 3: (F). Cluster assignment



(G). Re-compute centroids

# K-means: convergence

---

- The algorithm converges if:
  - Very few instances are reassigned to new clusters, or
  - The centroids do not change significantly, or
  - The following sum does not change significantly

$$Error = \sum_{i=1}^k \sum_{\mathbf{x} \in C_i} d(\mathbf{x}, \mathbf{m}_i)^2$$

- Where  $C_i$  is the  $i^{\text{th}}$  cluster;  $\mathbf{m}_i$  is the centroid of cluster  $C_i$ .

# K-means: centroid, distance

---

- Re-computation of the centroids:

$$\mathbf{m}_i = \frac{1}{|C_i|} \sum_{\mathbf{x} \in C_i} \mathbf{x}$$

- $\mathbf{m}_i$  is the centroid of cluster  $C_i$ .  $|C_i|$  denotes the size of  $C_i$ .

- Distance measure:

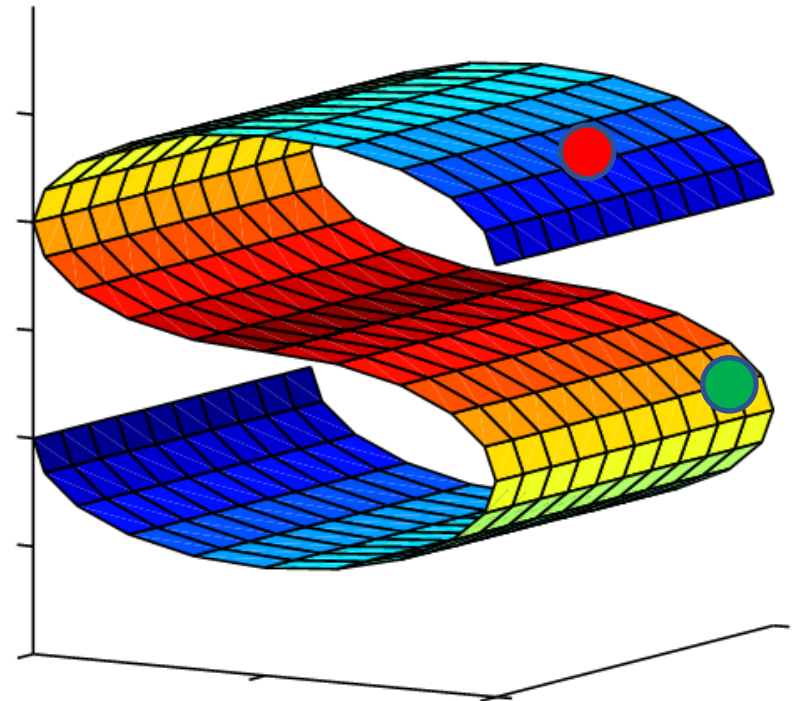
- Euclidean

$$d(\mathbf{x}, \mathbf{m}_i) = \|\mathbf{x} - \mathbf{m}_i\| = \sqrt{(x_1 - m_{i1})^2 + (x_2 - m_{i2})^2 + \dots + (x_n - m_{in})^2}$$

- Other measures are possible.

# K-means: about distance

- Distance measure
  - Each measure provides a view on data
  - There are infinite number of distance measures
  - Which distance is good?
- Similarity measures can be used
  - Similarity between two objects

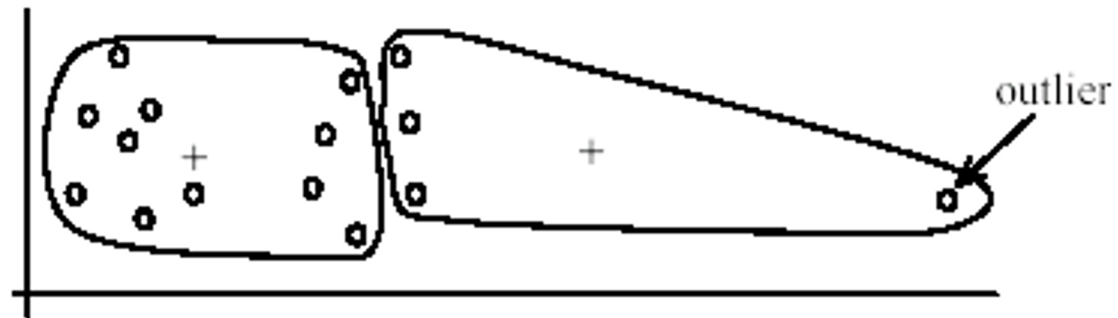


# K-means: affects of outliers

---

- *K-means is sensitive with outliers, i.e., outliers might affect significantly on clustering results.*
  - Outliers are instances that significantly differ from the normal instances.
  - The attribute distributions of outliers are very different from those of normal points.
  - Noises or errors in data can result in outliers.

# K-means: outlier example



(A): Undesirable clusters



(B): Ideal clusters



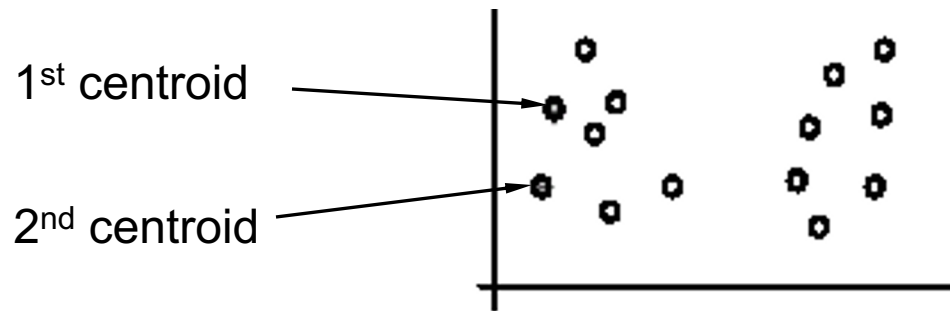
# K-means: outlier solutions

---

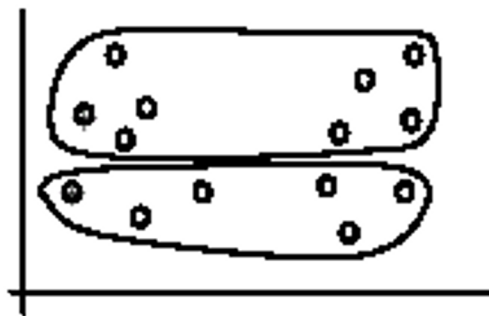
- *Outlier removal*: we may remove some instances that are significantly far from the centroids, compared with other instances.
  - Removal can be done a priori or when learning clusters.
- *Random sampling*: instead of clustering all data, we take a random sample  $S$  from the whole training data.
  - $S$  will be used to learn  $K$  clusters. Note that  $S$  often contains fewer noises/outliers than the original training data.
  - After learning, the remaining data will be assigned to the learned clusters.

# K-means: initialization

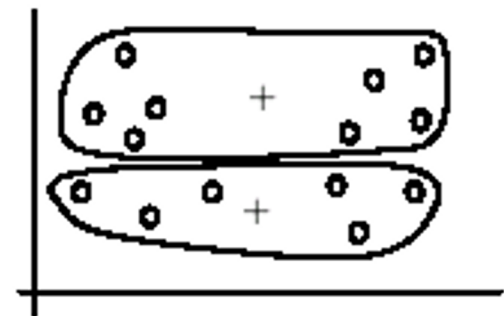
- Quality of K-means depends much on the initial centroids.



(A). Random selection of seeds (centroids)



(B). Iteration 1



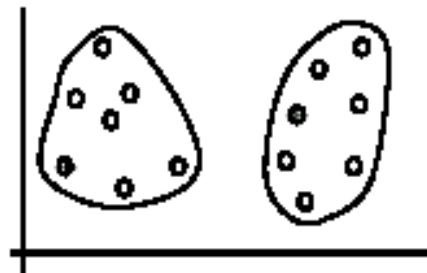
(C). Iteration 2

# K-means: initialization solution (1)

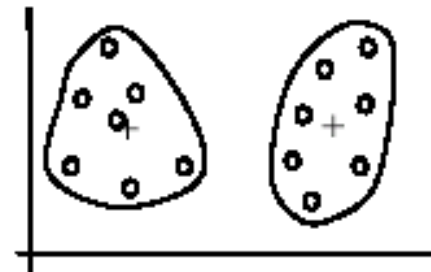
- We repeat K-means many times
  - Each time we initialize a different set of centroids.
  - After learning, we combine results from those runs to obtain a unified clustering.



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



(B). Iteration 1



(C). Iteration 2

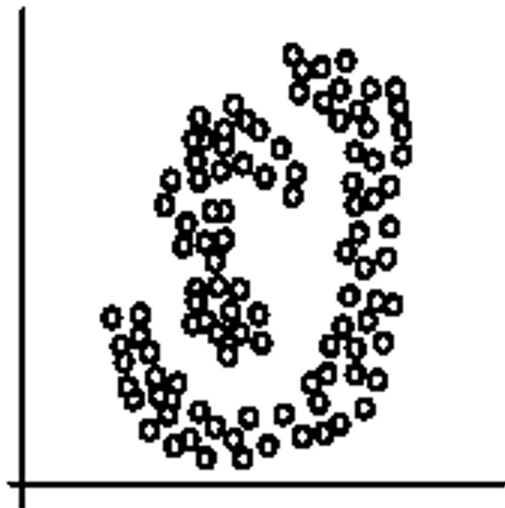
## K-means: initialization solution (2)

---

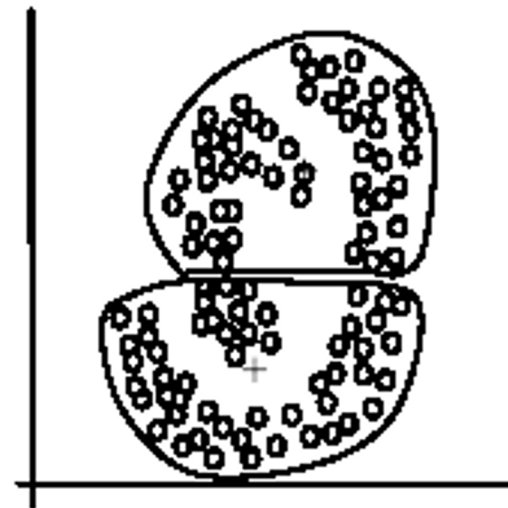
- **K-means++**: to obtain a good clustering, we can initialize the centroids from **D** in sequence as follows
  - Select randomly the first centroid  $\mathbf{m}_1$ .
  - Select the second centroid which are farthest to  $\mathbf{m}_1$ .
  - ...
  - Select  $i^{\text{th}}$  centroid which are farthest from  $\{\mathbf{m}_1, \dots, \mathbf{m}_{i-1}\}$ .
  - ...
- By using this initialization scheme, K-means can converge to a near optimal solution [Arthur, D.; Vassilvitskii, 2007]

# K-means: curved clusters

- When using Euclidean distance, K-means cannot detect non-spherical clusters.
  - How to deal with those cases?



(A): Two natural clusters



(B):  $k$ -means clusters

[Liu, 2006]

# K-means: summary

---

## ■ Advantages:

- Be very simple,
- Be efficient in practice,
- Converges in expected polynomial time [Manthey & Röglin, JACM, 2011]
- Be flexible in choosing the distance measures.

## ■ Limitations:

- Choose a good similarity measure for a domain is not easy.
- Be sensitive with outliers.

## 4. Online K-means

---

- K-means:

- We need all training data for each iteration.
- Therefore, it cannot work with big datasets,
- And cannot work with stream data where data come in sequence.

- *Online K-means* helps us to cluster big/stream data.

- It is an online version of K-means [Bottou, 1998].
- It follows the methodology from online learning and stochastic gradient.
- At each iteration, one instance will be exploited to update the available clusters.

# Revisiting K-means

- Note that K-means finds K clusters from the training instances  $\{x_1, x_2, \dots, x_M\}$  by minimizing the following loss function:

$$Q_{k\text{-means}}(w) = \frac{1}{2} \sum_{i=1}^M \|x_i - w(x_i)\|_2^2$$

- Where  $w(x_i)$  is the nearest centroid to  $x_i$ .
- Using its gradient, we can minimize Q by repeating the following update until convergence:

$$w_{t+1} = w_t + \gamma_t \sum_{i=1}^M [x_i - w_t(x_i)]$$

- Where  $\gamma_t$  is a small constant, often called learning rate.
- This update will converge to a local minimum.



# Online K-means: idea

---

- Note that each iteration of K-means requires the full gradient:

$$Q'_t = \sum_{i=1}^M [x_i - w_t(x_i)]$$

- Which requires all training data.
- Online K-means minimizes Q stochastically:
  - At each iteration, we just use a little information from the whole gradient  $Q'$ .
  - Those information comes from the training instances at iteration  $t$ :

$$x_t - w_t(x_t)$$

# Online K-means: algorithm

---

- Initialize  $K$  centroids randomly.
- Update the centroids as an instance comes
  - At iteration  $t$ , take an instance  $x_t$ .
  - Find the nearest centroid  $w_t$  to  $x_t$ , and then update  $w_t$  as follows:

$$w_{t+1} = w_t + \gamma_t(x_t - w_t)$$

- **Note:** the learning rates  $\{\gamma_1, \gamma_2, \dots\}$  are positive constants, which should satisfy

$$\sum_{t=1}^{\infty} \gamma_t = \infty; \sum_{t=1}^{\infty} \gamma_t^2 < \infty$$

# Online K-means: learning rate

---

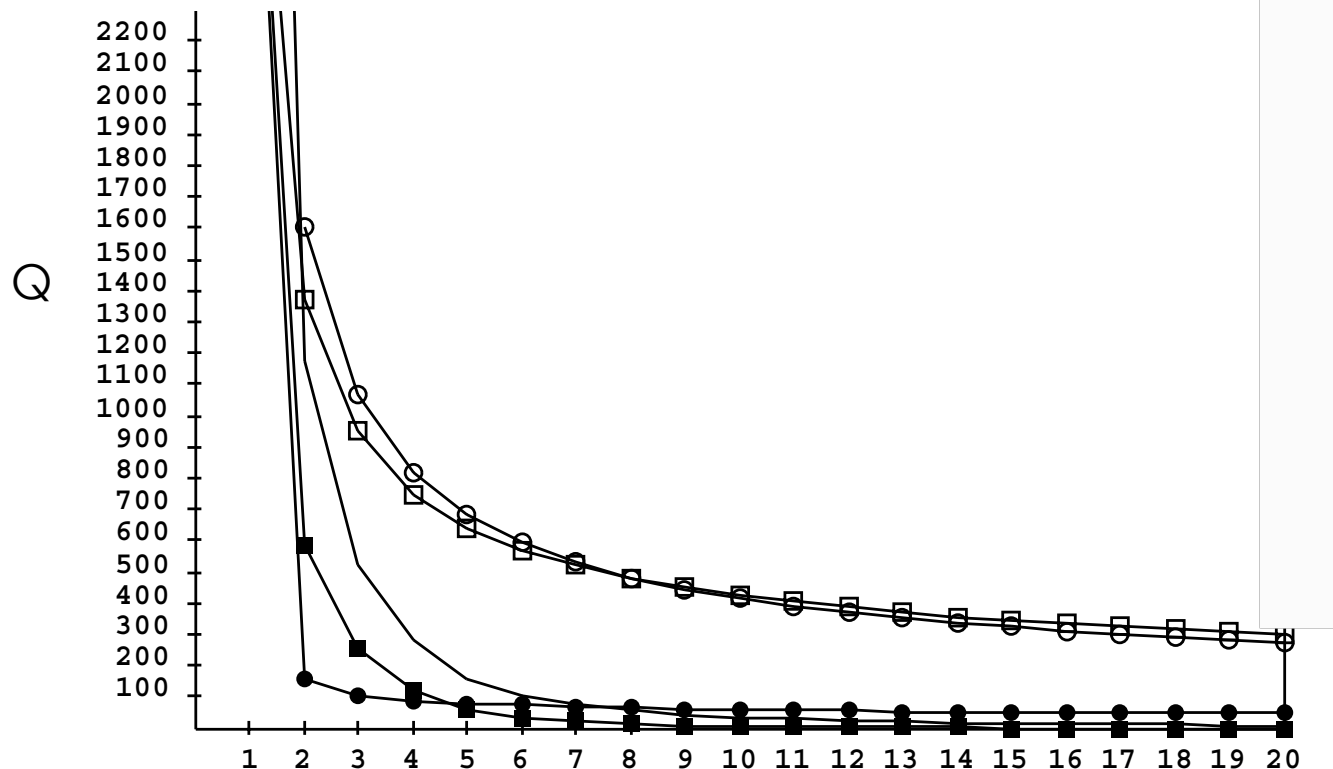
- A popular choice of learning rate:

$$\gamma_t = (t + \tau)^{-\kappa}$$

- $\tau, \kappa$  are positive constants.
- $\kappa \in (0.5, 1]$  is called *forgetting rate*. Large  $\kappa$  means that the algorithm remembers the past longer, and that new observations play less and less important role as  $t$  grows.

# Convergence of Online K-means

- Objective  $Q$  decreases as  $t$  increases.



- Online K-means (Black circles), K-means (Black squares)
- Partial gradient (empty circles), full gradient (empty squares)

# References

---

- Arthur, D.; Vassilvitskii, S. (2007). K-means++: the advantages of careful seeding. *Proceedings of the 18th annual ACM-SIAM symposium on Discrete algorithms*, pp. 1027–1035.
- Arthur, D., Manthey, B., & Röglin, H. (2011). Smoothed analysis of the k-means method. *Journal of the ACM (JACM)*, 58(5), 19.
- Bottou, Léon. Online learning and stochastic approximations. *On-line learning in neural networks* 17 (1998).
- B. Liu. *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data*. Springer, 2006.
- Lloyd, S., 1982. Least squares quantization in PCM. *IEEE Trans. Inform. Theory* 28, 129–137. Originally as an unpublished Bell laboratories Technical Note (1957).
- Jain, A. K. (2010). Data clustering: 50 years beyond K-means. *Pattern recognition letters*, 31(8), 651-666.

# Exercises

---

- Solutions to K-means when the data distributions are not spherical?
- How to decide a suitable cluster for a new instance?