CPE/EE 695: Applied Machine Learning

Module 10-2: Unsupervised Learning

Dr. Shucheng Yu, Associate Professor
Department of Electrical and Computer Engineering
Stevens Institute of Technology

Unsupervised Learning

In real world, the vast majority of data is unlabeled.

- Supervised learning is not applicable.
- Unsupervised learning is needed.

Examples of unsupervised learning tasks:

- **Clustering** – to group similar instances into clusters
- **Anomaly detection** – to detect abnormal instances
- **Density estimation** – to estimate the probability density function (PDF) of the random process generating the data
- **Dimensionality reduction** – to reduce high-dimension data to low-dimension data.

Clustering

Clustering can be used in different applications, such as

- Customer segmentation
- Data analysis
- Dimensionality reduction
- Anomaly detection
- Semi-supervised learning
- Image searching
- Image segmentation

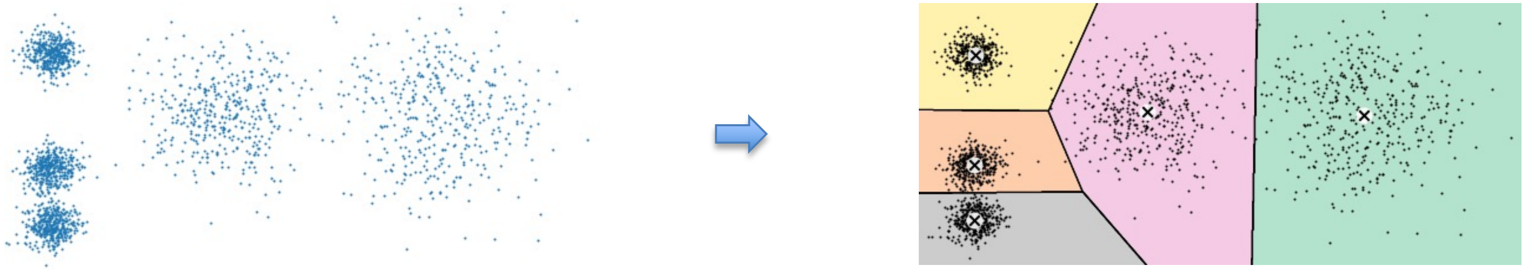
Popular clustering algorithms:

- K-Means
- DBSCAN
- Others: agglomerative clustering, BIRCH, mean-shift, affinity propagation, spectral clustering, etc.

K-Means

Five blobs of instances exists in below example. How to cluster the blobs efficiently?

K-Means (by Stuart Lloyd, Bell Labs, 1957) can do this in just few iterations.

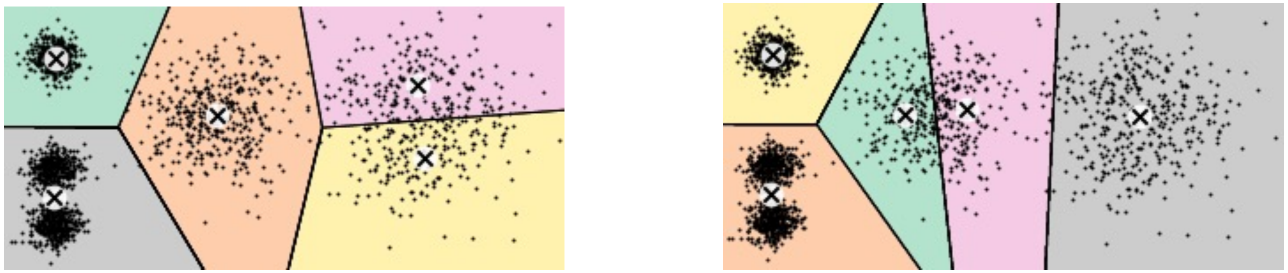


Basic idea of K-Means:

1) Select the number *k.*
2) Randomly select *k* instances from the data. Use their locations as *k* centroids.
3) Repeat the following steps until the centroids are not moving (convergence):
    a) Assign each instance to a closest centroid based on Euclidean distance.
    b) Update each centroid with the mean of the instances assigned to it.

Problems with K-Means

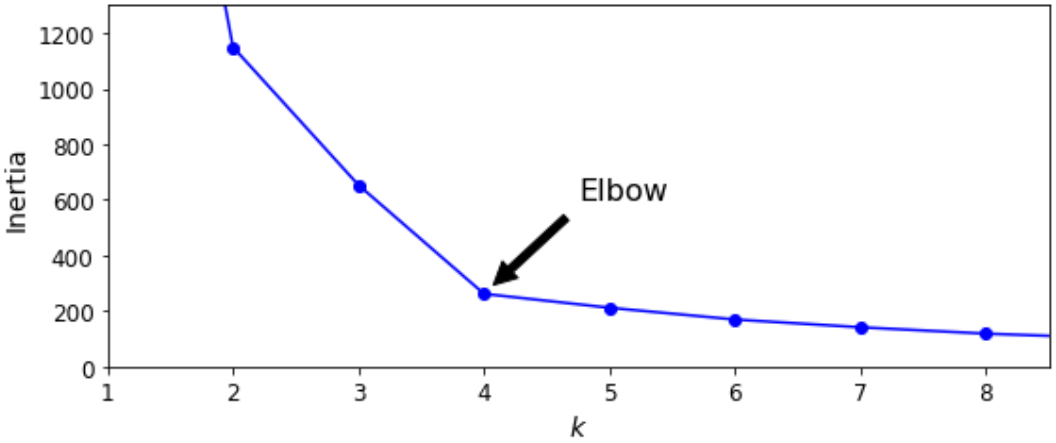
- Variability: Running K-Means multiple times result in different clustering



  - Problem: how to know which one is the best clustering?
  - Performance measure one:
    - **Inertia** - the sum of distances between instances to their respective centroids.
  - Run K-Means multiple times with different initial centroids, respectively. The clustering with the smallest inertia is picked as final solution.

Problems with K-Means

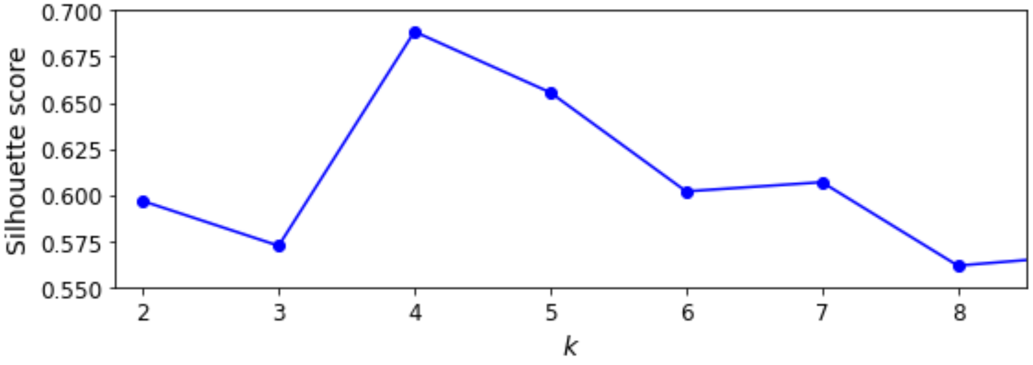
- How to determine the number of clusters *k*?
  - The K-Means algorithm can return a clustering for any number *k*.
  - The optimal choice of *k* shall have a good balance of inertia vs. efficiency.
  - The commonly used method is so-called **the "elbow" rule**
    - Plot the curve of Inertia vs. *k*
    - The curve drops fast before the "Elbow" but slow after.



(Image from Geron textbook)

Problems with K-Means

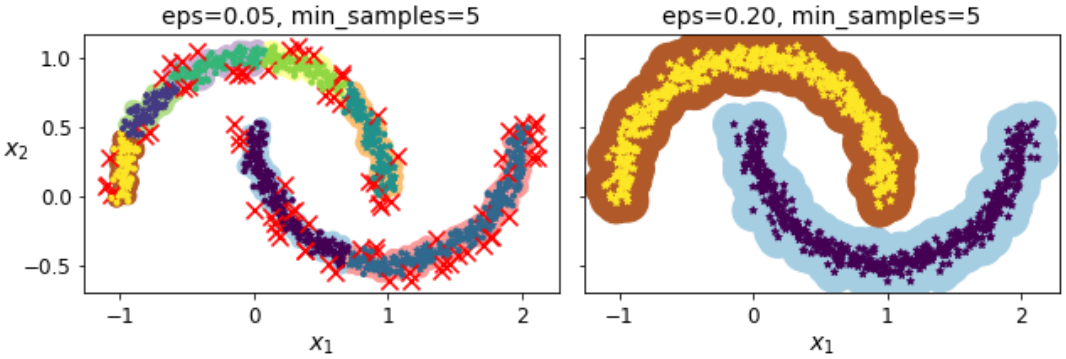
- How to determine the number of clusters *k*?
  - Another approach is to use so-called "***silhouette score***", which is the mean of so-called "*silhouette coefficient*" of each instance.
    - $\textit{silhouette coefficient} = \frac{b-a}{\max(a,b)}, -1 \le \textit{silhouette coefficient} \le 1$
    - where a: the mean distance to other instances; b: the mean of distance to instances in the closest cluster.
    - *silhouette coefficient* = +1: the instance is well inside its own cluster.
    - *silhouette coefficient* = 0: the instance is on the boundary of cluster.
    - *silhouette coefficient* = -1: the instance is misplaced in wrong cluster.



(Image from Geron textbook)

DBSCAN

- A density-based clustering algorithm
  - It classifies instances (points) into three types – **core**, **reachable** and **outliers**, based on a parameter ***ε*** that specifies the **radius** of a neighborhood of a point.
    - A point *p* is a **core point** only if there are at least **#Min points** within ***ε*** distance of *p* including *p*.
    - A point *q* is **directly reachable** from *p* if *q* is within ***ε*** distance of *p*.
    - A point *q* is **reachable** from *p* if there exists a path *p -> … -> q,* with each pair of neighboring nodes in the path are directly reachable.
    - Any nodes not reachable from any core point are **outliers**.



(Image from Geron textbook)

Gaussian Mixtures Model (GMM)

- GMM is a probabilistic model assuming that instances were generated from a mixture of several Gaussian distributions whose parameters are unknown. [see page 260 -261 of Geron textbook]
  - GMM works very well with clusters with different ellipsoidal shapes, sizes, density and orientation, while K-Means does not.
  - GMM can be used for clustering, density estimation and anomaly detection.
  - When observing an instance, you know it was generated from one of the Gaussian distributions but not exactly which one.
  - Each instance x_i is therefore can be interpreted as:

    $$(x_i, z_{i1}, z_{i2}, \ldots, z_{ik})$$

    - where z_{ij} is a latent variable (whose value is not observable) representing Gaussian distribution j, assuming there are k Gaussian distributions.
    - z_{ij} =1 if x_i was generated from Gaussian distribution j; $z_{im} = 0$ if $m \neq j$.

Gaussian Mixtures Model (GMM)

- The ***EM algorithm*** can be used to estimate the parameters of GMM. [see page 191 -193 of Mitchell textbook]
  - Initialize hypothesis to $h =< u_1, u_2, \dots, u_k >$, assuming only means of the *k* distributions are unknown.
  - **Step 1**: calculate the expected value $E(z_{ij})$ for each latent variable z_{ij}, assuming the current hypothesis $h =< u_1, u_2, \dots, u_k >$ holds.
  - **Step 2**: Calculate a new maximum likelihood hypothesis $h' =< u_1', u_2', \dots, u_k' >$, assuming each each latent variable z_{ij} takes the value $E(z_{ij})$ calculated in Step 1. Then replace h with h' and iterate.
- Determining the number of clusters
  - To find a model that minimizes a ***theoretical information criterion***:
    - Bayesian Information Criterion (BIC)

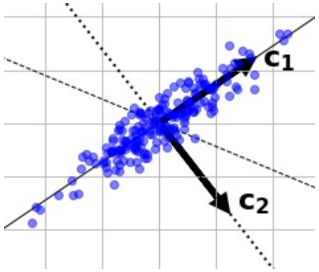
      $\boldsymbol{BIC} = \log(m)\, p \; - 2\log(\hat{L})$, or
    - Akaike Information Criterion (AIC)

      $\boldsymbol{AIC} = 2p \; - 2\log(\hat{L})$

    where m is the number of instances, p is the number of parameters learned by the model, $\hat{L}$ is the maximized value of the likelihood function of the model.

Principle Component Analysis (PCA)

- PCA is a popular technique for dimensionality reduction
  - PCA identifies the axis accounting for the largest amount of variance of training data, as shown by C1 in the figure.
  - Then it finds the second axis for the second largest amount of variance, so on and so forth.
  - The *i*th axis is called the *i*th **principal component (PC)** of data.
  - The axes are orthogonal to each other.
  - Actually, principal components are **eigenvectors** of the data's covariance matrix.
  - PC can be computed in two ways:
    1) Eigen-decomposition of the data covariance matrix or
    2) Singular value decomposition (SVD) of the data matrix.



(Image from Geron textbook)