LECTURE 6: CLUSTERING METHODS

EKATERINA MURAVLEVA

OVERVIEW OF THE COURSE

Lecture 1: General course information, CRISP-DM methodology

Lecture 2: Supervised learning/unsupervised learning. Classification/regression problems. Accuracy metrics (precision, recall, ROC-AUC scores). Concept of loss functions, overfitting / underfitting.

Lecture 3: Classical ML: Linear regression, logistic regression, support vector machine

Lecture 4: Classical ML: Decision trees, random forests, boosting.

Lecture 5: Classical ML: Dimensionality reduction: linear, non-linear methods.

Lecture 6: Classical ML: Clustering methods

Lecture 7: Basic neural networks

Lecture 8: Scalable algorithms

RECAP OF LECTURE 5

- Dimensionality reduction
- Linear methods: PCA, LDA
- Non-linear methods (Graph Laplacian and IsoMap).

PLAN OF LECTURE 6

- Definition of clustering
- K-Means
- Cluster validity measures
- Hierarchical clustering
- Graph cuts
- DBScan

DEFINITION OF CLUSTERING

— Clustering is a technique used in data analysis to group similar data points together based on certain characteristics or features.

— The purpose of clustering is to identify patterns and relationships within a dataset, which can help in understanding the underlying structure and organization of the data.

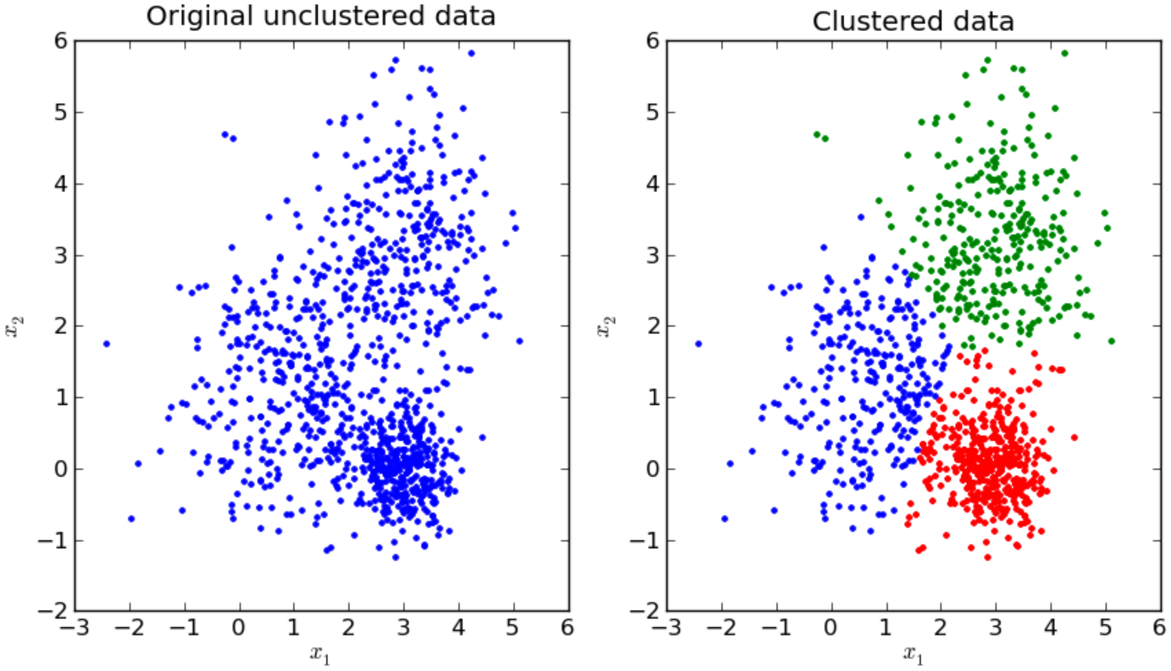
— In simpler terms, clustering is like organizing a messy room by grouping similar items together. It helps in making sense of large and complex datasets by dividing them into smaller, more manageable groups.

DEFINITION OF CLUSTERING (2)

— Clustering is an **unsupervised learning** technique, meaning that it does not require any predefined labels or categories.

— It relies on the **inherent structure** of the data to group them together.

— It is a useful tool for exploratory data analysis, as it can reveal hidden insights and patterns in the data.



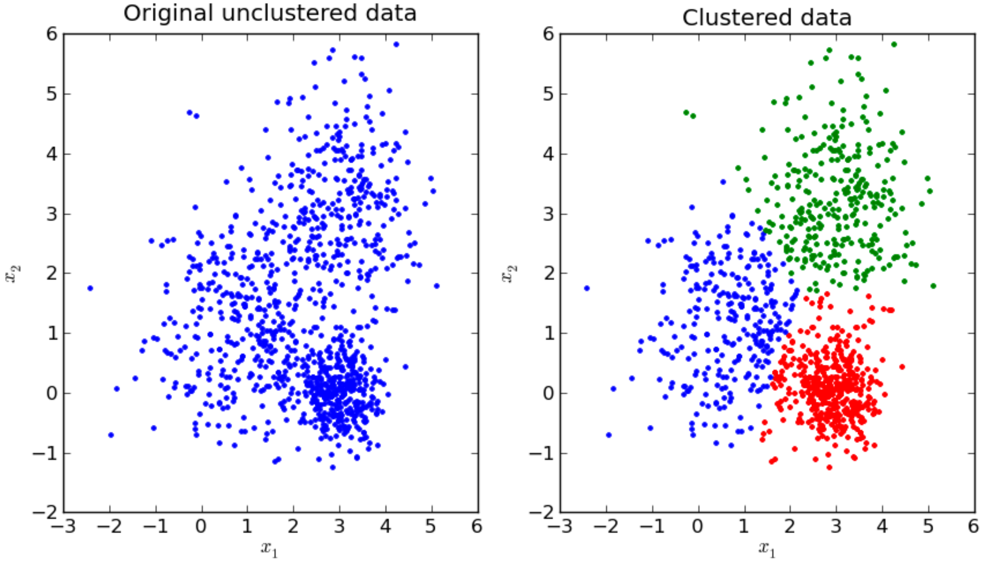
HOW THE CLUSTERING IS DEFINED

Suppose we have a dataset $X = \{x_1, \ldots, x_m\}$.

Clustering returns the partition of this dataset into K subsets $C_1, \ldots, C_k$ such that

$\bigcup_{i=1}^{K} C_i = X$ and C_i do not intersect.

In order to measure, how good is the clustering, we need to a have a metric between data points.



SIMPLEST CASE: EUCLIDEAN DISTANCE

Suppose the we measure the distance between two points as $d(x, y) = \|x - y\|^2$

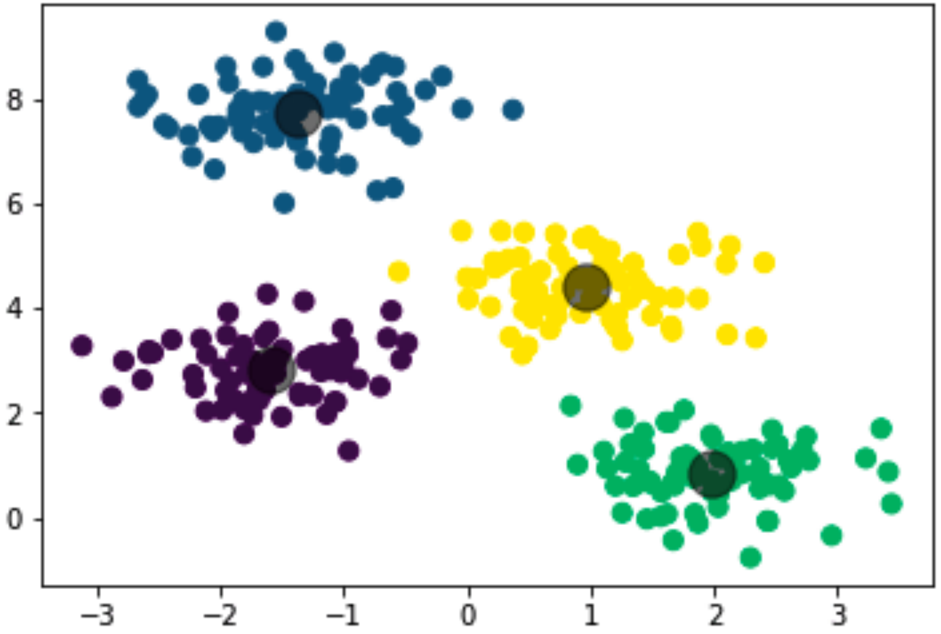
Then, we can specify the cluster by **centroid of it.** Let $\#C_i$ be the number of points in a cluster,

then we can define the centroid of the cluster as the average of all points in the corresponding cluster:

$$c_i = \frac{1}{\#C_i} \sum_{j:x_j \in C_i} x_j$$

Then, the point x belongs to the cluster C_i if it is close to the centroid of the cluster:

$x \in C_i$, iff, $d(x, c_i) < d(x, c_j), \forall j \neq i$.



PARAMETRIZATION OF CLUSTERING

Thus we can parametrize the clustering with euclidean distance as the collection of K centroids.

The centroids are averages of all elements in the cluster.

How we measure the ‘**goodness’** of clustering?

GOODNESS OF CLUSTERING

Members of each cluster should be similar to each other, thus the distance to the centroid should be small (**inter-cluster** compactness)

Members of two different clusters should be as far as possible from each other (**intra-class** distance).

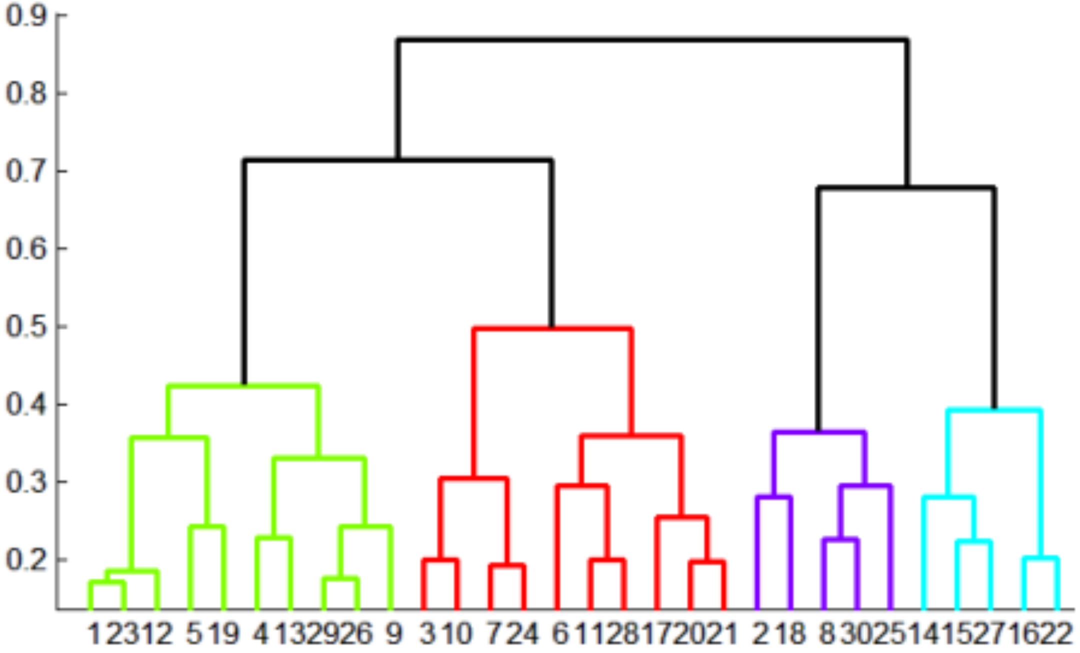
We can measure intra-class distance just by the distance between centroids.

HIERARCHICAL CLUSTERING

— One of the simple approaches to build clusters is **hierarchical clustering**

— It is a bottom-up approach that starts by treating each data point as a separate cluster and then merges the two most similar clusters iteratively until all data points are in a single cluster.

— This results in a hierarchical structure, also known as a **dendrogram**, which can be visually represented.



LINKAGE: MEASURING THE DISTANCE BETWEEN CLUSTERS

— **Single linkage**: It measures the distance between the closest pair of data points from two different clusters.

— **Complete linkage**: It measures the distance between the farthest pair of data points from two different clusters.

— **Average linkage**: It measures the average distance between all data points from two different clusters.

— **Ward’s method**: It minimizes the total within-cluster variance when merging two clusters.

WARD’S METHOD

Wards method says that the distance between two clusters, A and B is how much the sum of squares will increase when we merge them:

$$\begin{aligned}\Delta(A,B) &= \sum_{i\in A\cup B} \| x_i - c_{A\cup B} \|^2 - \sum_{i\in A} \| x_i - c_A \|^2 - \sum_{i\in B} \| x_i - c_B \|^2 \\ &= \frac{n_A n_B}{n_A + n_B} \| c_A - c_B \|^2\end{aligned}$$

where c_j is the center of cluster j, and n_j is the number of points in it. Δ is called the merging cost of combining the clusters A and B

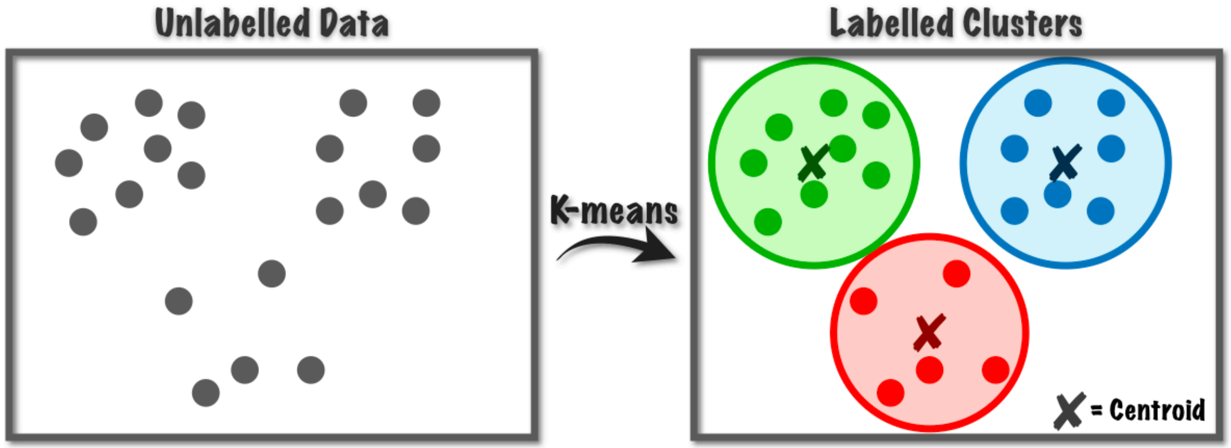
SUMMARY ON HIERARHICAL CLUSTERING

- Do not need to specify the number of classes
- Can be computationally intensive and not suitable for large datasets
- Errors at the early stage propagate and affect subsequent merging steps
- It is important to choose the linkage method

K-MEANS CLUSTERING

One of the most widely used algorithms for clustering is **K-Means** algorithm

We iteratively update the centroids of the clusters



K-MEANS CLUSTERING: ALGORITHM

— We randomly initialize k centroids of the clusters: $\mu^0 = (\mu_1^0, \ldots, \mu_K^0)$ (here the upper index refers to the number of iterations

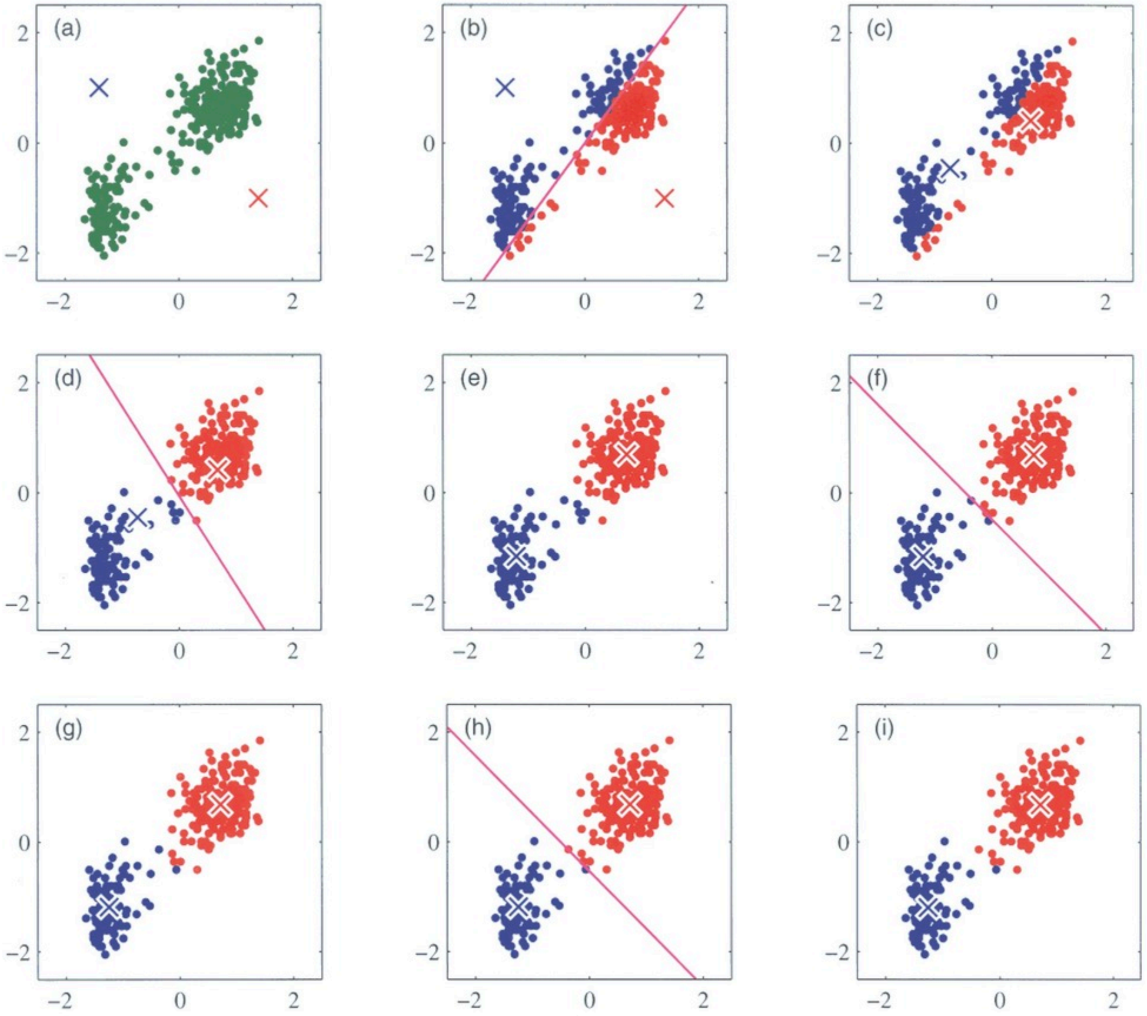
— **Step 1**, **Classify**: Assign each point $x_j, \quad j = 1,\ldots,m$ to the nearest center:

$$C_i^{k+1} = \{x_j : i = \arg\min d(x_j, \mu_i^k)\}$$

— **Step 2**: **Recenter**: At step 1, the clusters have been modified, thus we need to recompute the centroid as

$$\mu_i^{k+1} = \frac{1}{\#C_i^{k+1}} \sum_{j: x_j \in C_i^{k+1}} x_j$$

K-MEANS ILLUSTRATION



K-MEANS: SUMMARY

— We need to know number of clusters

— Clusters should be approximately the same size (distance of the points in the cluster to the center) and density (number of points)

CLUSTER VALIDITY

- For supervised models we have a lot of measures how to determine how good the model is (accuracy, precision, recall, F1-score, ROC-AUC, …)
- For cluster analysis the analogous question is how to evaluate the goodness of the clustering result?
- We need to compare:
  - Different clustering algorithms
  - Two sets of clusters

MEASURES OF CLUSTER VALIDITY

- **External measure**: used to measure the extent to which cluster labels match externally supplied class labels
  - Example: **Jaccard Index**
- **Internal measure**: used to measure goodness clustering structure without respect to external information
  - Example: **Sum of Squared Errors (SSE)**

EXTERNAL MEASURES OF CLUSTERING

Given a set $X = \{x_1, \ldots, x_m\}$ and two partitions $C = (C_1, \ldots, C_r)$ and $P = (P_1, \ldots, P_s)$

The first partition is obtained from the clustering algorithms,

The second partition comes from data labels

define:

— **a**, the number of pairs of elements in X that are in the **same** subset of C and are in the **same** subset of P

— **b**, the number of pairs of elements in X that are in the **different** subsets of C and are in the **different** subset s of P

— **c**, the number of pairs of elements in X that are in the **same** subset of C and are in the **different** subsets of P

— **d**, the number of pairs of elements in X that are in the **different** subsets of C and are in the **same** subset of P

JACCARD INDEX

Using a, b, c, d defined as before, the Jaccard Index is defined as

$$J = \frac{a}{a + c + d}$$

It measures overlap between sets of pairs.

RAND INDEX

Using a, b, c, d defined as before, the Rand Index (named after William Rand) is defined as

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

It represents frequency of agreement between two algorithms: a is ‘same’ and b is ‘different’.

Rand Index takes values from 0 to 1.

High value means high level of agreement

INTERNAL MEASURES

Both Jaccard Index and Rand Index require the knowledge of a reasonable partition of the dataset, so these are **external measures**

As an internal measure, **Silhouette coefficient can be used**

SILHOUETTE COEFFICENT

Consider the I-th point in the dataset.

– Let a_i be the average distance of the point x_i to the point of its cluster.

– Let b_i be minimum (or average) distance of the point x_i to the points in other clusters.

– Then, $s_i = \dfrac{b_i - a_i}{\max(a_i, b_i)}$

SILHOUETTE COEFFICENT (2)

$s_i = \dfrac{b_i - a_i}{\max(a_i, b_i)}$, is defined for every point in the dataset.

— $-1 \leq s_i \leq 1$

— If s_i is close to 1, the point is assigned to the cluster very well.

— If s_i is close to 0, the sample lies equally away from several clusters (can be assigned to another cluster)

— If s_i is close to -1, the point is missclassified

— We can compute the validity of the cluster by averaging s_i over the dataset.

WITHIN-CLUSTER-SUM OF SQUARED ERRORS

Another metric is WSS (Within Cluster Sum of Squared Errors):

Just sum the squares of distances to the centroid!

WSS can be used to determine the optimal number of clusters using the **elbow method**

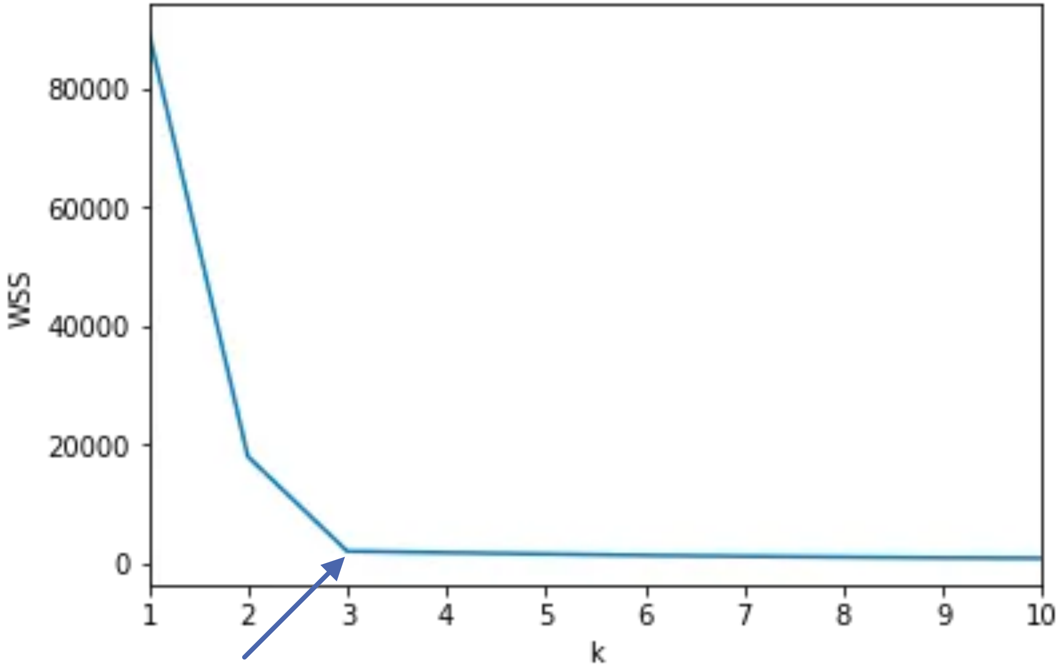
ELBOW METHOD

— In the Elbow methods, we plot WSS versus the number of clusters

— We have to learn many clusterisations

— Typically, we get the picture similar to what has been shown on the picture on the right

— Here the elbow is at k = 3.



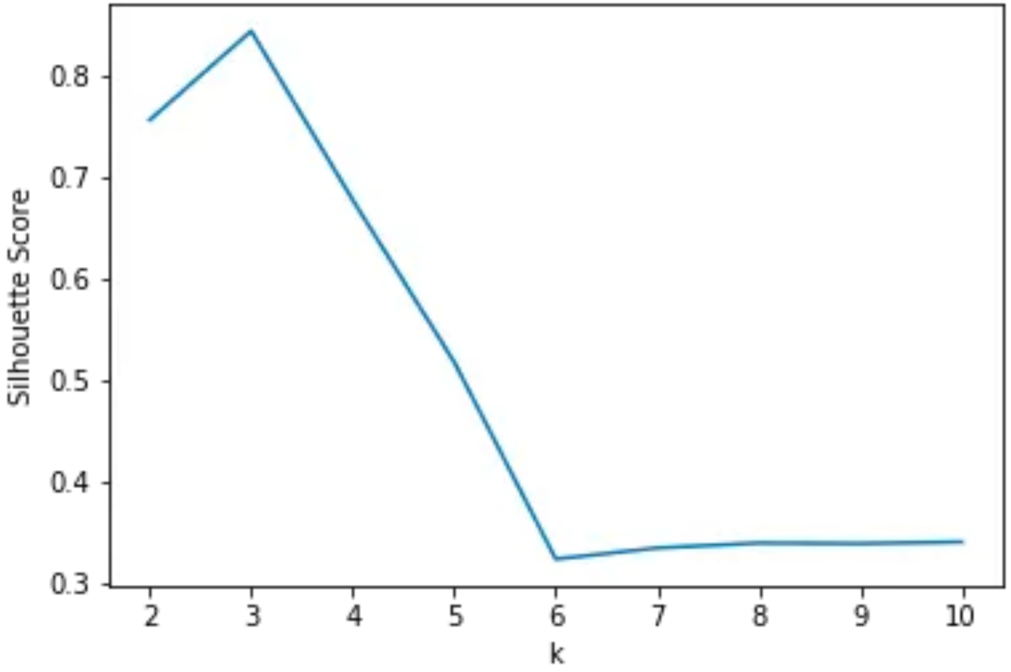
Elbow

SILHOUETTE METHOD

— We compute Silhouette score for the dataset (averaging over samples).

— It is maximum at the optimal number of clusters!

— For this example, obviously k=3 is optimal.



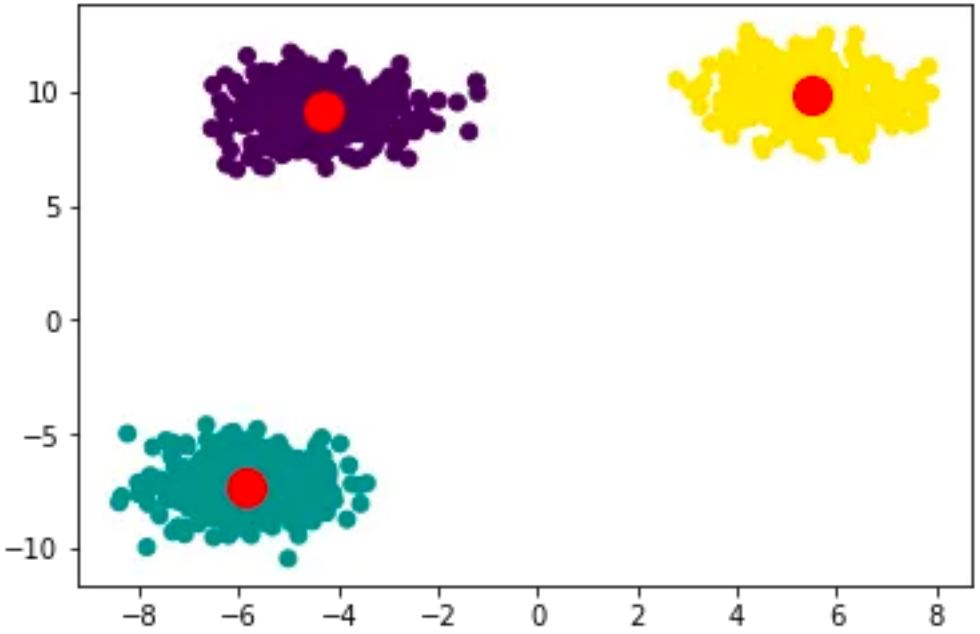
SILHOUETTE & ELBOW

— Elbow is more a decision rule (we need to formalize what ‘elbow’ means

— The silhouette score can be used during the algorithm as a metric.

— They do not replace each other, can be used together.

— The picture on the right was the dataset used for computing elbow and silhouette score. Clearly, nothing very complicated.



CLUSTERING USING GRAPHS

From the dataset, we can build a graph.

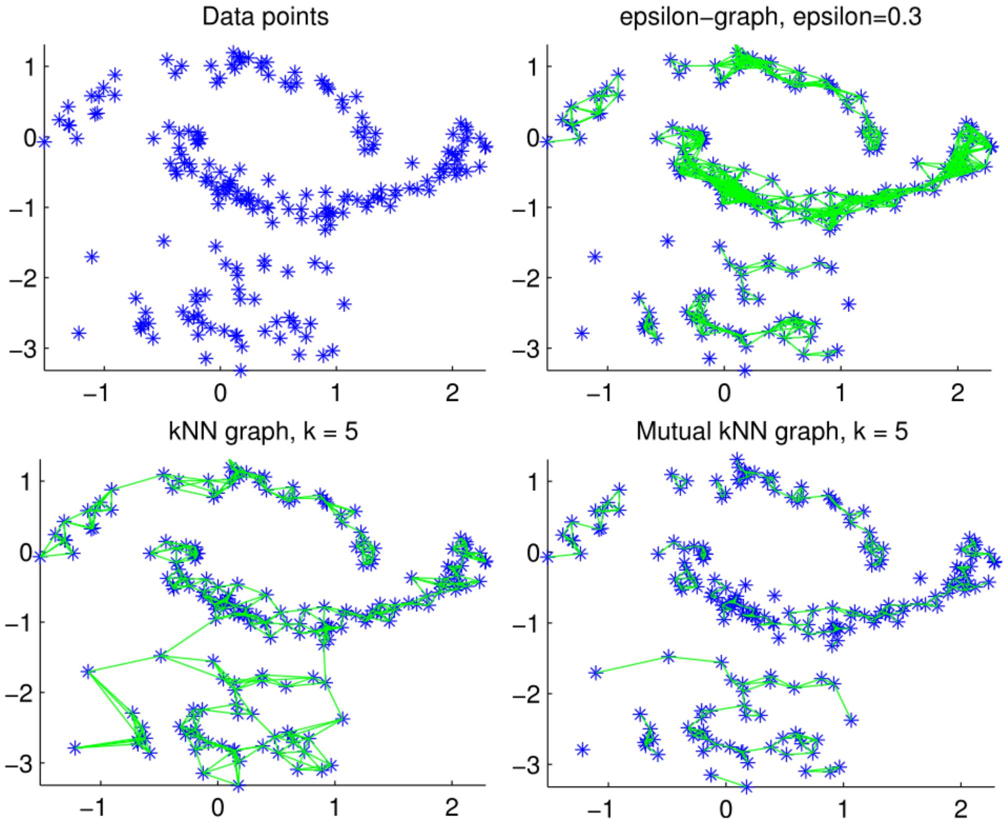
Similar idea has been discussed in dimensionality reduction.

Types of graph:

— ε-neighborhood

— nearest neighbors graph (kNN)

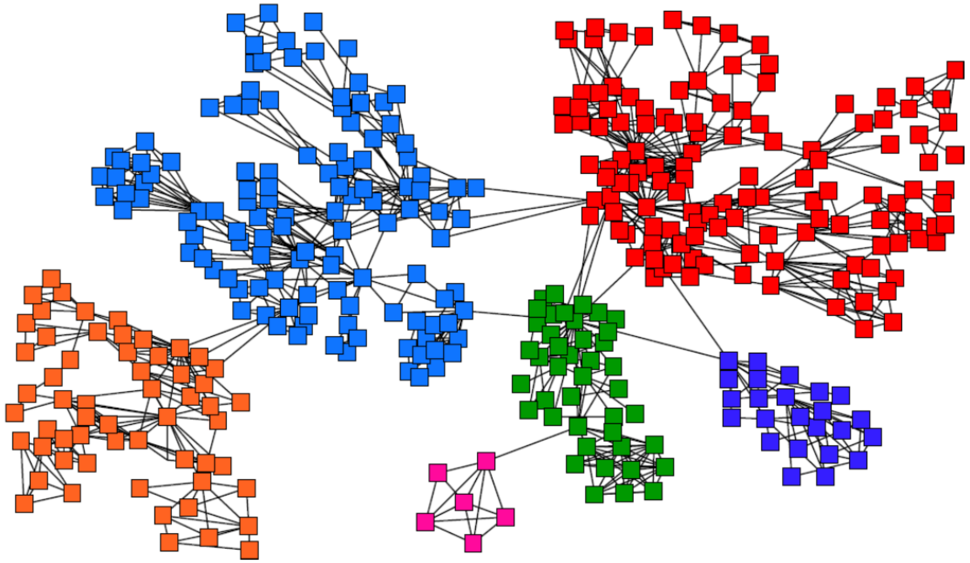
Some data is already given as a graph, for example social networks, or citation graphs.



CLUSTERING FOR THE GRAPHS

Given the (weighted) graph we want to partition it such that edges between the groups have low weight

This can be achieved using **graph cuts**



GRAPH CUTS

– **Problem:** Partition graph such that edges between groups have low weights

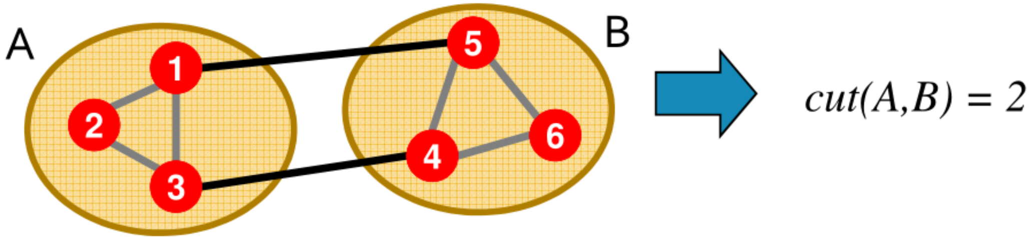
– $W(A,B) = \sum_{i \in A, j \in B} w_{ij}$

– MinCut problem:

$$\mathrm{Cut}\left(A_1, \ldots, A_k\right) = \sum_{i=1}^{k} W\left(A_i, \bar{A}_i\right)$$

– Choose

$$A_1, \ldots, A_k = \arg \min_{A_1, \ldots, A_k} \mathrm{Cut}\left(A_1, \ldots, A_k\right).$$

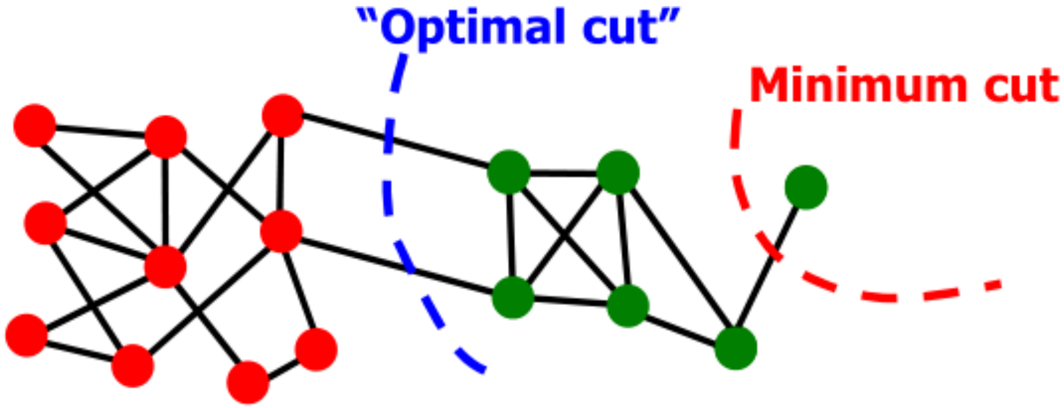


CLUSTERING USING GRAPH CUTS

Problem: MinCut favors isolated clusters

Solutions: Normalized Cuts (divide by the sizes of the clusters).

And spectral clustering (related to Laplacian Eigenmaps, only second eigenvector of the graph Laplacian is used).



DBSCAN

One of the popular algorithms for clustering is DBScan (Density-Based Spatial Clustering of Applications with Noise)

— The main idea behind DBScan is to find dense regions in the data and consider them as clusters, while points that are not part of any dense region are considered as noise.

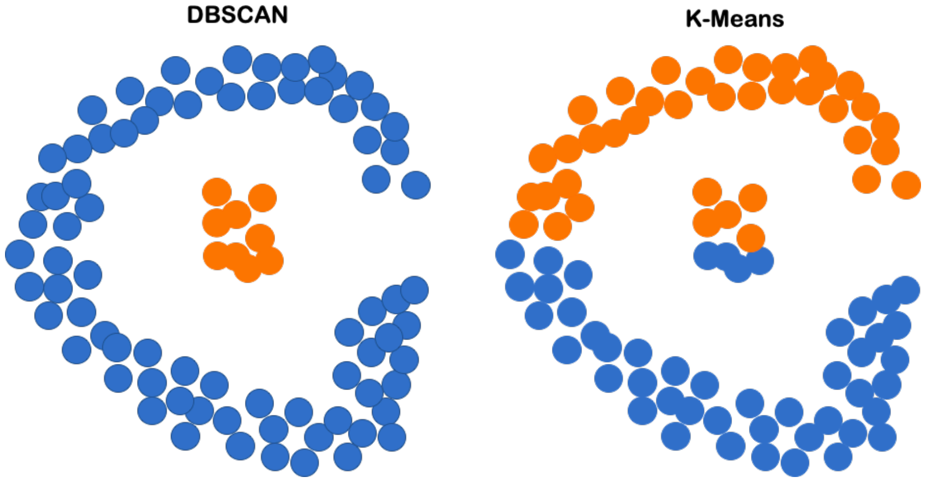
— This is achieved by defining two important parameters: ε and minimum points (minPts).

— The algorithm starts by randomly selecting a point from the dataset and finding its neighbors within a distance ε.

— If the number of neighbors is greater than or equal to minPts, then this point is labeled as a core point and becomes the center of a new cluster.

— If the number of neighbors is less than minPts, then the point is labeled as a border point and is added to an existing cluster.

— If the point has no neighbors, then it is labeled as noise.



RECAP OF LECTURE 6

- Definition of clustering
- K-Means
- Cluster validity measures
- Hierarchical clustering
- Graph cuts
- DBScan

NEXT LECTURE

Basic neural network models