Classification: K-means + LDA

Adil RASHEED, Frank WESTAD, Damiano VARAGNOLO

So far...

- **PCA** — Score, Loading, Outlier Detection, Clustering
- **Regression: Algorithm and examples** — Linear Methods: MLR, PCR, PLSR; Non-linear methods: Decision Tree, Random Forest, SVM, DNN
- **Feature Engineering**
- **Categorical Variables and One Hot Encoding**
- **Validation: Bias-Variance tradeoff**
- **Several real life examples: Unscrambler, Python notebook, matlab**
- **Classification** — Logistic Regression, PCA; K-mean clustering, Linear Discriminant Analysis

Reminder

5 minutes presentation of the data and plan regarding how each student wants to proceed (on 8th October) + 2 minutes question.

Then submit a 2 page writeup before 5th November. This will be graded say 25%.

Our minimal expectations from the project:

- Demonstrate the use of the linear methods (PCA, PCR, PLSR, MLR, IDLE, PARAFAC) on their own dataset
- Refection on why the methods worked or failed with demonstration. If the methods failed then
  - Demonstrate the use of non-linear methods (DT, RF, SVM, DNN)
  - Refection on why the methods worked or failed with demonstration
- Presentation of the results (data cleaning, outlier detection, regression and classification)
- What could be done to improve the modelling
- Oral exam based on the presentation

The worst 10% presentations will fail the exam 😝

Contents: Classification, Clustering

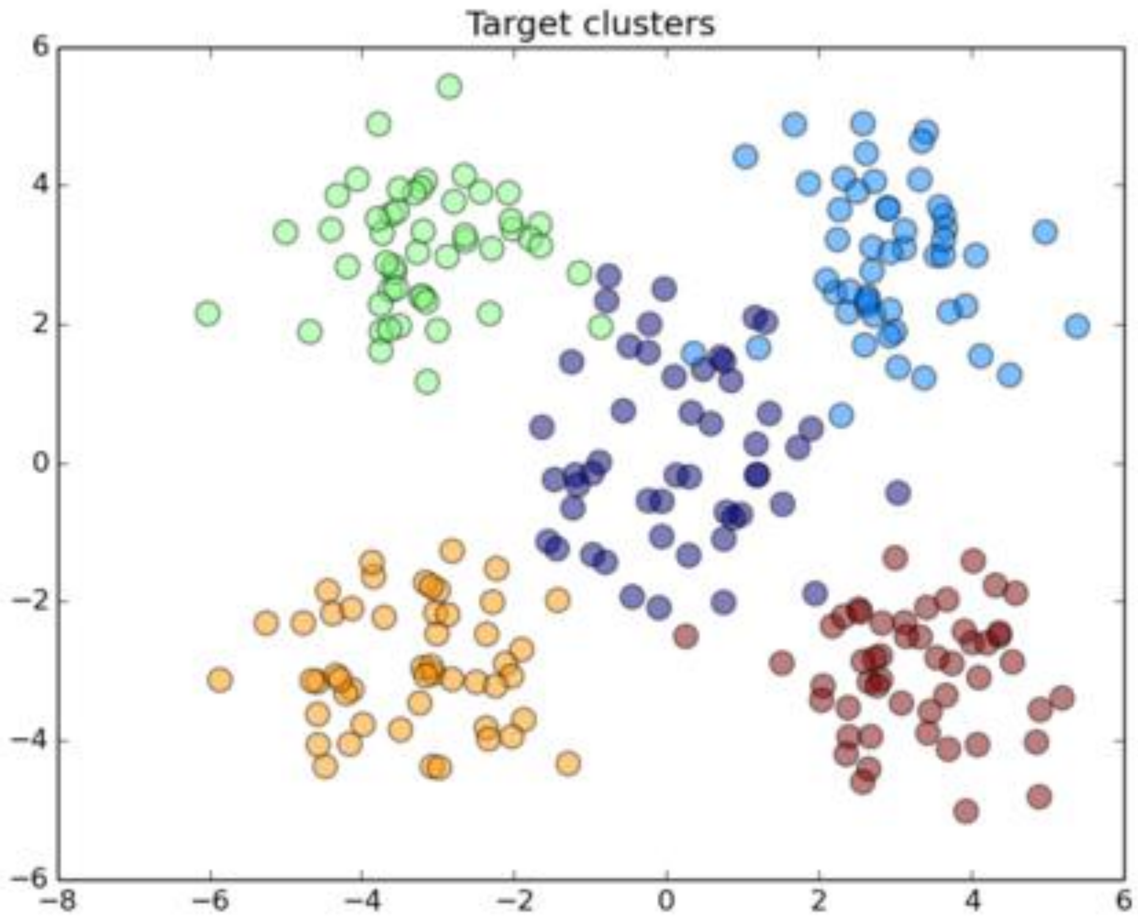
- Logistic Regression
- PCA
- Decision Trees and Random Forest
- Cluster analysis (k-means)
- Linear Discriminant Analysis (LDA)
- Support Vector Machine Classification
- Deep Learning

Covered today

Non-linear methods

Image compression / segmentation using K-means clustering
A few demonstrations in Unscambler (Housing data, Archeology: Classification, LDA, PCA-LDA, K-mean)
Vagina pressure data analysis in matlab continues

K-mean clustering



Steps for k-mean clustering

1. Specify number of clusters *K*.
2. Initialize centroids by first shuffling the dataset and then randomly selecting *K* data points for the centroids without replacement.
3. Keep iterating until there is no change to the centroids. i.e assignment of data points to clusters isn’t changing.
4. Compute the sum of the squared distance between data points and all centroids.
5. Assign each data point to the closest cluster (centroid).
6. Compute the centroids for the clusters by taking the average of the all data points that belong to each cluster.

Mathematically

$$J = \sum_{i=1}^{m}\sum_{k=1}^{K} w_{ik} \| x^i - \mu_k \|^2$$

Where $w_{ik} = 1$ for data point x^i if it belongs to cluster μ_k otherwise $w_{ik} = 0$

$$\frac{\partial J}{\partial w_{ik}} = \sum_{i=1}^{m}\sum_{k=1}^{K} \| x^i - \mu_k \|^2 \qquad \Rightarrow w_{ik} = \begin{cases} 1 & \text{if k=argmin}_{\text{j}} \| \text{x}^{\text{i}}\text{-}\mu_{\text{j}} \|^2 \\ 0 & \text{otherwise.} \end{cases}$$

Simply assign the point xi to the closest cluster judged by its sum of squared distance from clusters centroid

$$\frac{\partial J}{\partial \mu_k} = 2\sum_{i=1}^{m} w_{ik}(x^i - \mu_k) = 0$$

$$\mu_k = \frac{\sum_{i=1}^{m} w_{ik} x^i}{\sum_{i=1}^{m} w_{ik}}$$

Compute the centroid of each cluster

$$\frac{1}{m_k}\sum_{i=1}^{m_k} \| x^i - \mu_{c^k} \|^2$$

Ensure that the clustr layout is not changin

Careful

- Standardize the data
- Different initializations may lead to different clusters

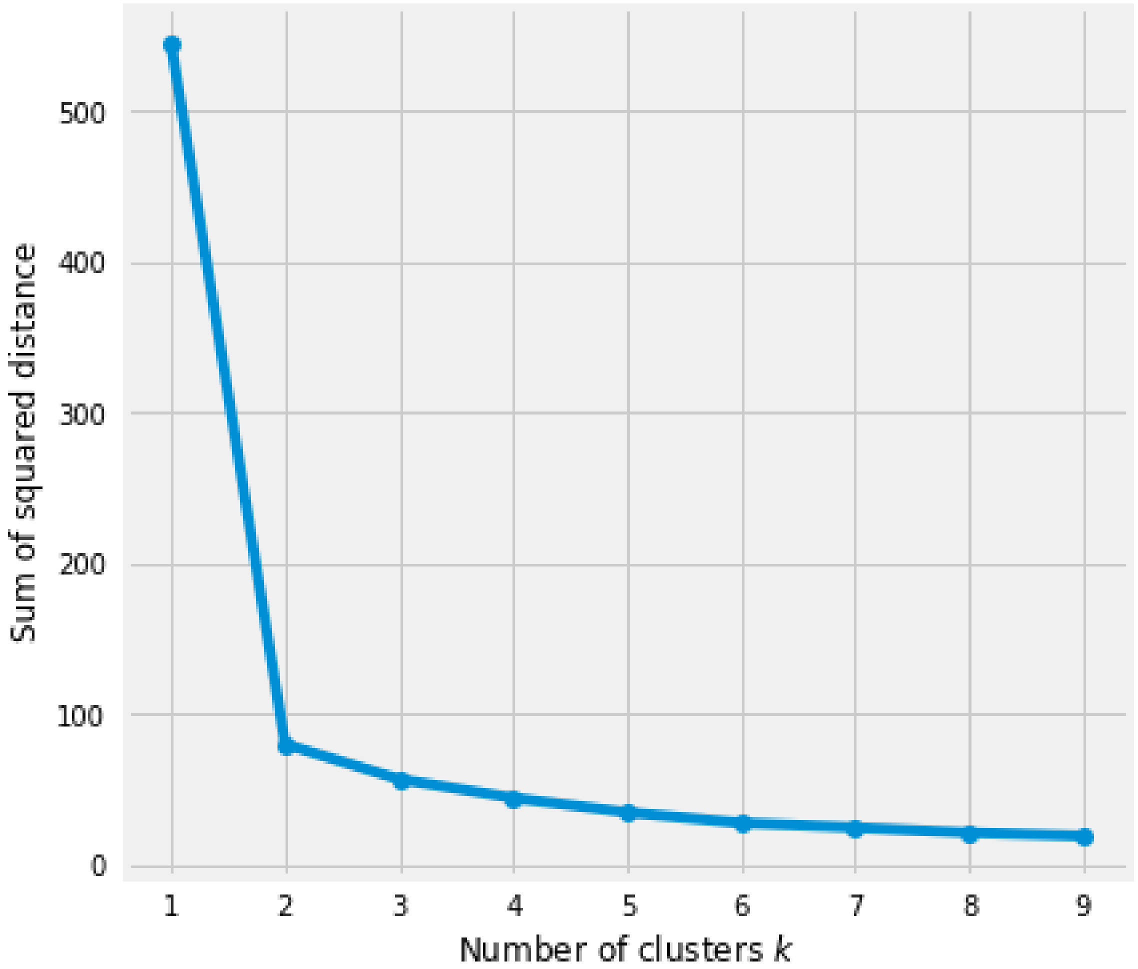
Applications

- market segmentation
- document clustering
- image segmentation
- image compression

Evaluation methods

- Clustering analysis doesn’t have a solid evaluation metric that we can use to evaluate the outcome of different clustering algorithms.
- Since k-means requires *k* as an input and doesn’t learn it from data, there is no right answer in terms of the number of clusters that we should have in any problem.
  - Domain knowledge and intuition may help but usually that is not the case.
- In the cluster-predict methodology, we can evaluate how well the models are performing based on different *K* clusters since clusters are used in the downstream modeling.

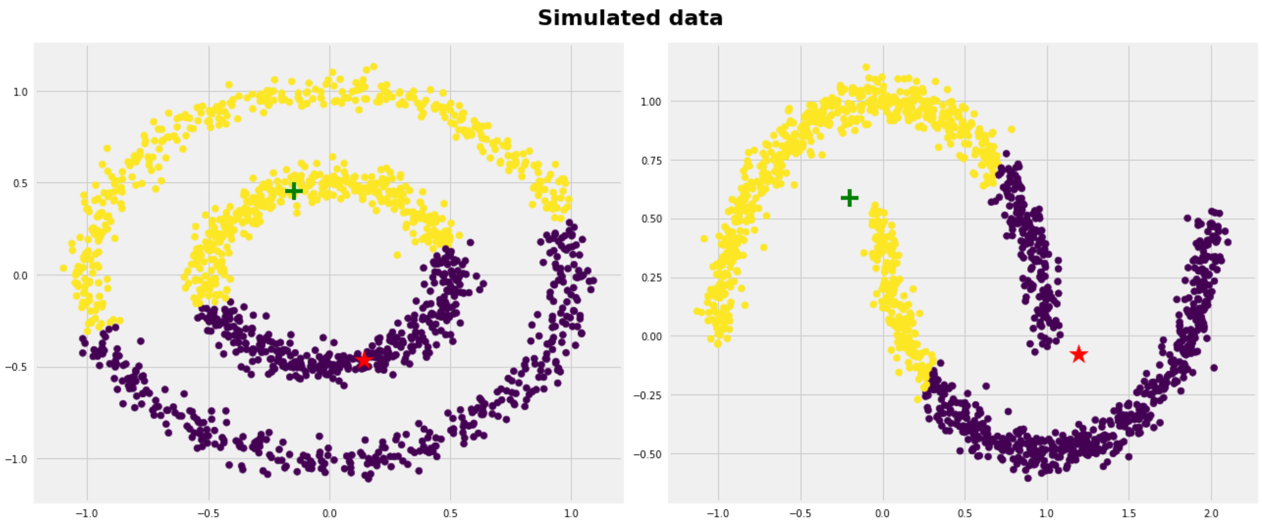
Elbow method



Elbow method gives us an idea on what a good *k* number of clusters would be based on the sum of squared distance (SSE) between data points and their assigned clusters' centroids. We pick *k* at the spot where SSE starts to flatten out and forming an elbow.

Drawbacks

- Kmeans algorithm is good in capturing structure of the data if clusters have a spherical-like shape.
- K-mean does a poor job with complex clusters
  - Potential solution is to use cluster



Takeaway

- Scale / standardize the data when applying k-means algorithm.
- K-means gives more weight to the bigger clusters
- K-means assumes spherical shapes of clusters and doesn’t work well when clusters are in different shapes such as elliptical clusters.
- If there is overlapping between clusters, kmeans doesn’t have an intrinsic measure for uncertainty
- Kmeans may still cluster the data even if it can’t be clustered