MACHINE LEARNING WITH PYTHON

K-MEANS CLUSTERING

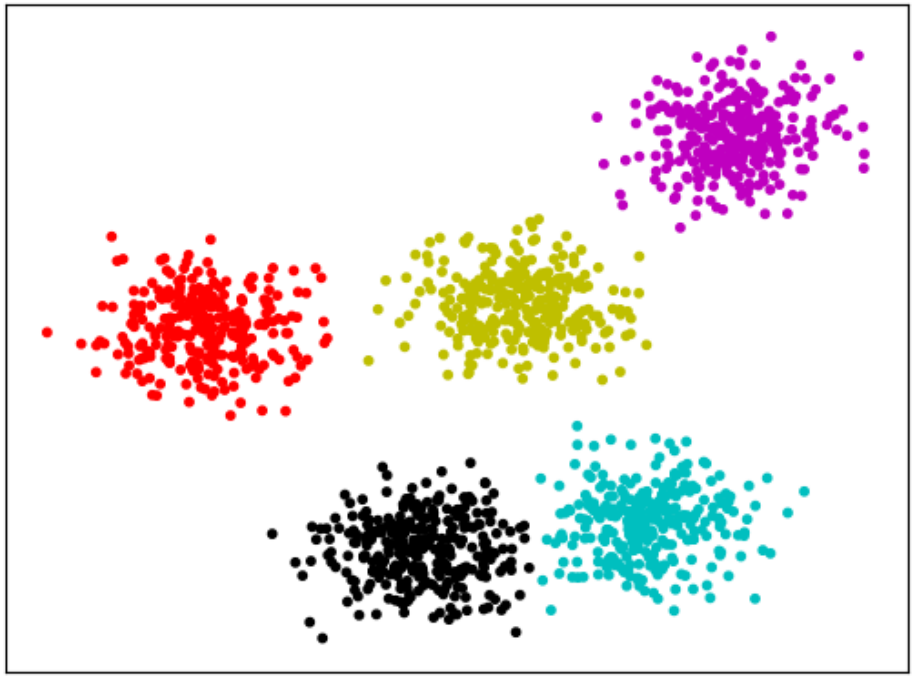
Themistoklis Diamantopoulos

Clustering

- Split data into clusters according to features
- K-Means
  - Given the number of clusters k
  - Split the data into clusters
  - Each cluster has a centroid
  - Minimize sum-of-squared-errors:

$$J = \sum_{i=1}^{k} \sum_{x \in S_i} \left\| x - \mu_i \right\|^2$$

where x are datapoints and $i=1\ldots k$ refer to clusters S_i with centroids μ_i

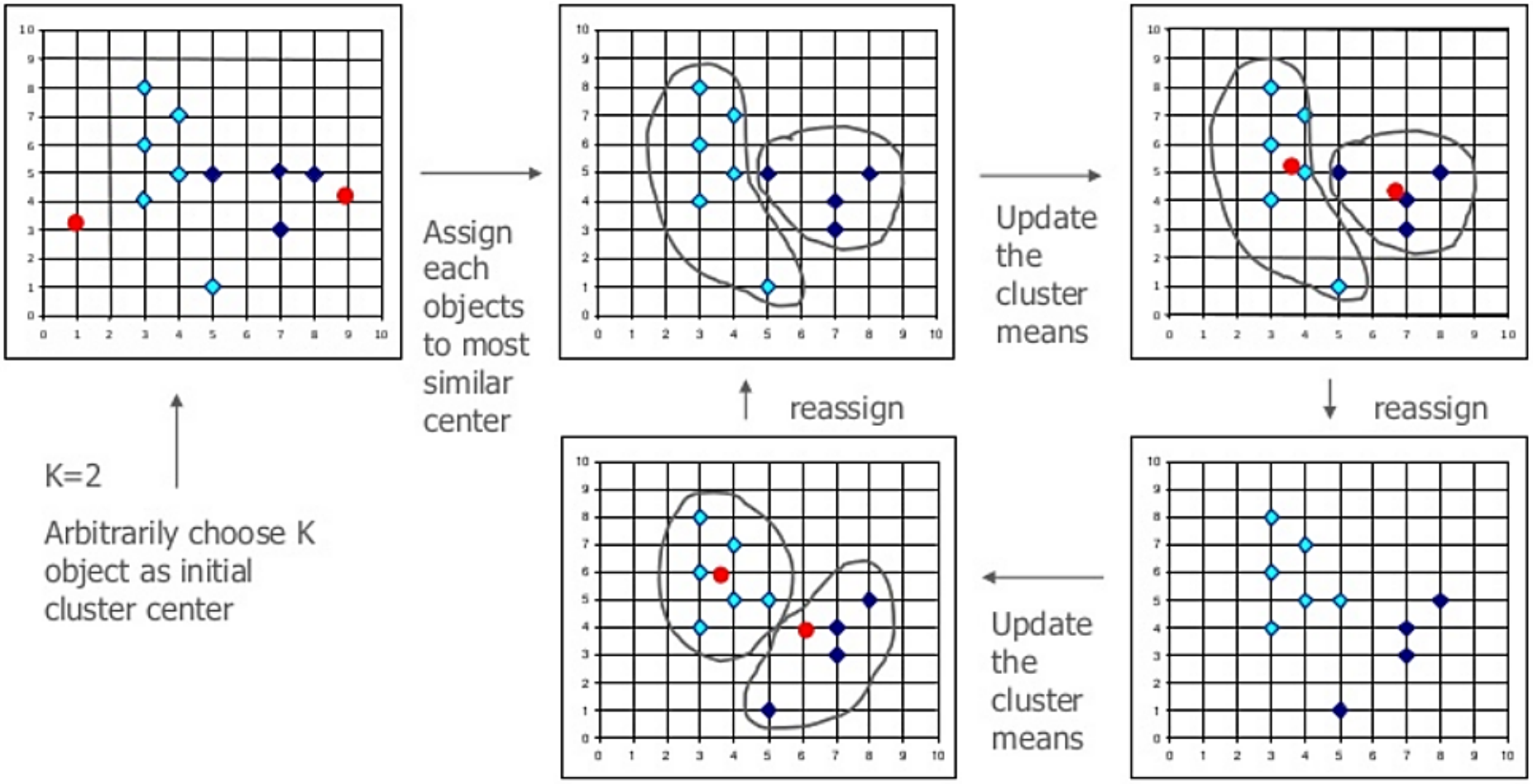


Euclidean distance:

$$\sqrt{\sum_{i=1}^{k} (x_i - y_i)^2}$$

Source: https://www.imperva.com/blog/2017/07/clustering-and-dimensionality-reduction-understanding-the-magic-behind-machine-learning/

K-Means Clustering



Source: http://bdewilde.github.io/blog/blogger/2012/10/26/classification-of-hand-written-digits-3/

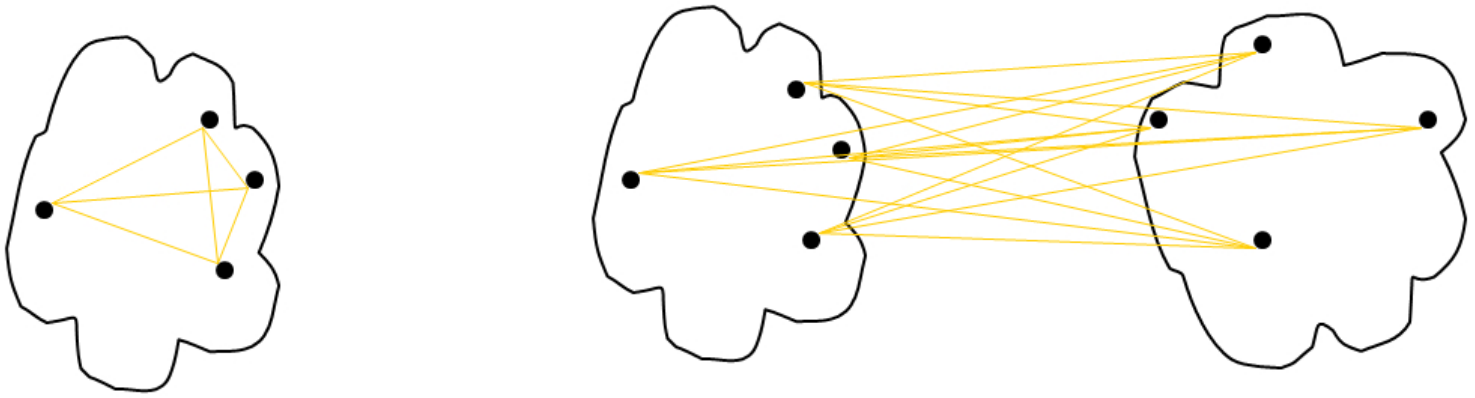
Evaluation

- Cohesion (Within Cluster Sum of Squares)

$$SSE = WSS = \sum_i \sum_{x \in C_i} (x - m_i)^2$$

- Separation (Between Cluster Sum of Squares)

$$BSS = \sum_i |C_i| (m - m_i)^2$$



cohesion separation

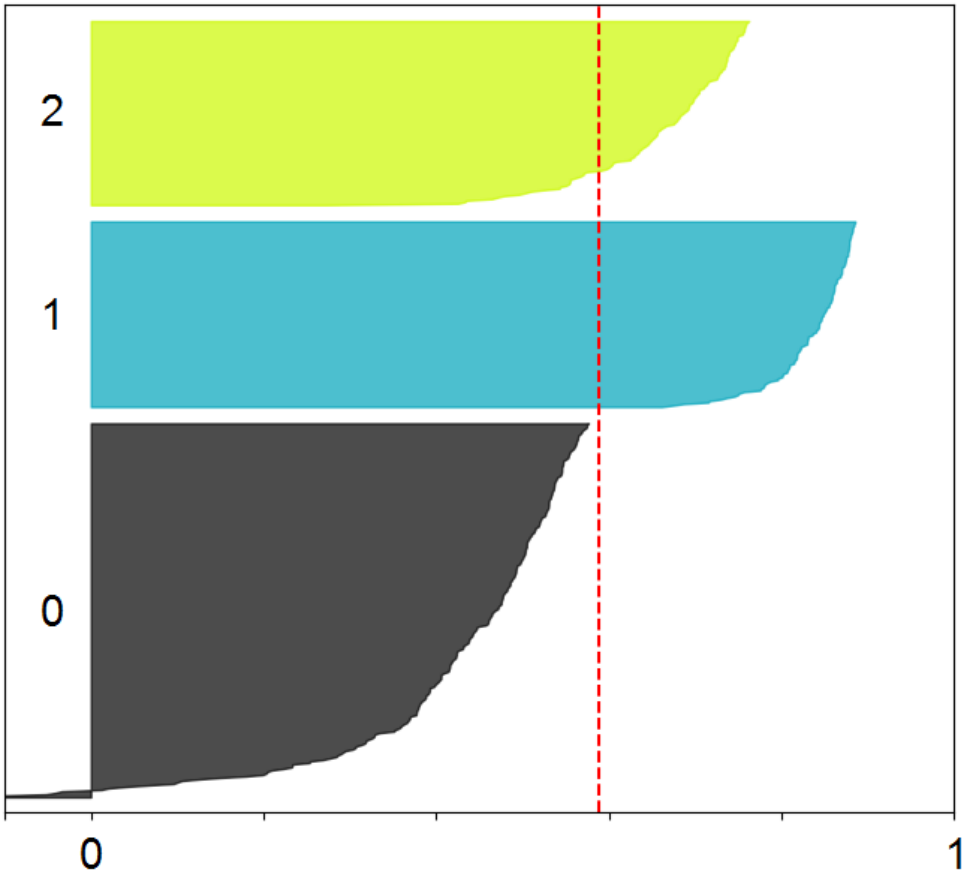
Source: https://codeahoy.com/2017/02/19/cluster-analysis-using-k-means-explained/

Silhouette

- Silhouette value of *x*:

  $s = (b - a) / \max(a, b)$

  where
  - *a* = average distance of x to the points in its cluster
  - *b* = min(average distance of x to points in another cluster)
- Value:
  - Positive/close to 1: possibly assigned to proper cluster
  - Negative: possibly assigned to wrong cluster
  - Close to 0: on border of 2 clusters



Source: http://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_silhouette_analysis.html