Data Clustering

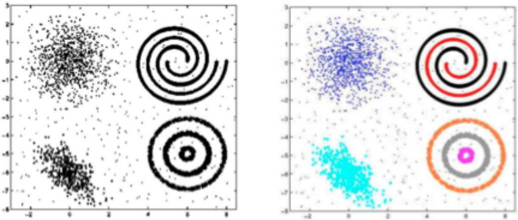
Compiled by: Sujatha Das G. and Cornelia Caragea

Acknowledgments: Rai, Manning

August 20, 2014

What is Data Clustering?

- Data Clustering is an unsupervised learning problem
- Given: N unlabeled examples $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$; the number of partitions K
- Goal: Group the examples into K partitions



(a) Input data (b) Desired clustering

- The only information clustering uses is the similarity between examples
- Clustering groups examples based of their mutual similarities
- A good clustering is one that achieves:
  - High within-cluster similarity
  - Low inter-cluster similarity

Picture courtesy: "Data Clustering: 50 Years Beyond K-Means", A.K. Jain (2008)

Data Clustering: Some Real-World Examples

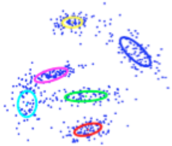
- Clustering images based on their perceptual similarities
- Image segmentation (clustering pixels)

- Clustering webpages based on their content
- Clustering web-search results
- Clustering people in social networks based on user properties/preferences
- .. and many more..

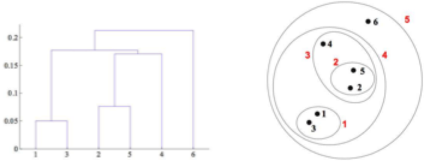
Picture courtesy: http://people.cs.uchicago.edu/~pff/segment/

Types of Clustering

1. **Flat or Partitional clustering** (K-means, Gaussian mixture models, etc.)
   - Partitions are independent of each other



2. **Hierarchical clustering** (e.g., agglomerative clustering, divisive clustering)
   - Partitions can be visualized using a tree structure (a dendrogram)
   - Does not need the number of clusters as input
   - Possible to view partitions at different levels of granularities (i.e., can refine/coarsen clusters) using different K



Flat Clustering: K-means algorithm (Lloyd, 1957)

- **Input:** N examples $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ ($\mathbf{x}_n \in \mathbb{R}^D$); the number of partitions K
- **Initialize:** K cluster centers $\mu_1, \ldots, \mu_K$. Several initialization options:
  - Randomly initialized anywhere in $\mathbb{R}^D$
  - Choose any K examples as the cluster centers
- **Iterate:**
  - Assign each of example $\mathbf{x}_n$ to its closest cluster center

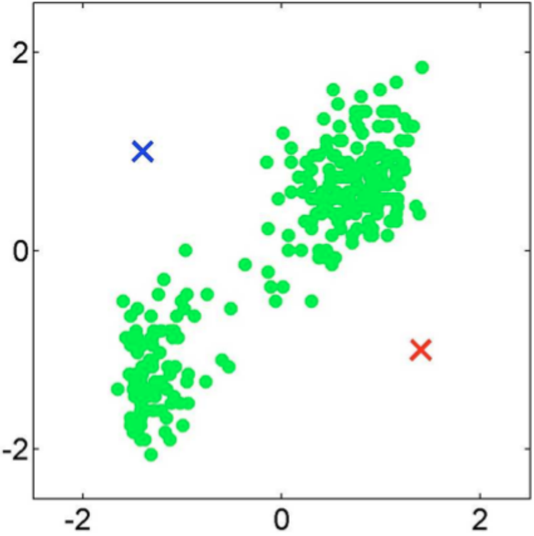
$$\mathcal{C}_k = \{n : \quad k = \arg\min_k ||\mathbf{x}_n - \mu_k||^2\}$$

  ($\mathcal{C}_k$ is the set of examples closest to μ_k)
  - Recompute the new cluster centers μ_k (mean/centroid of the set $\mathcal{C}_k$)

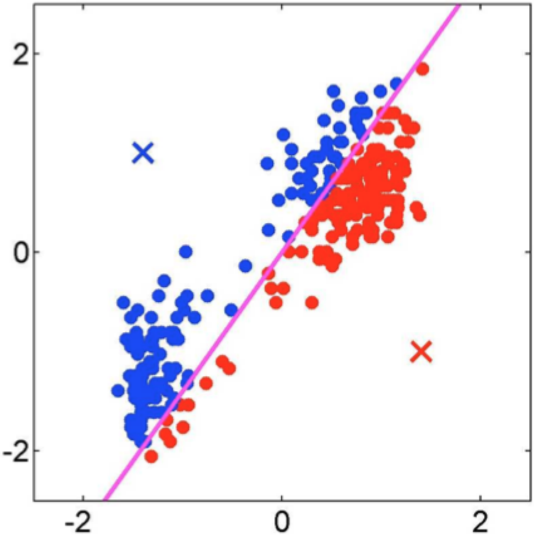
$$\mu_k = \frac{1}{|\mathcal{C}_k|} \sum_{n \in \mathcal{C}_k} \mathbf{x}_n$$

  - Repeat while not converged
- A possible convergence criteria: cluster centers do not change anymore

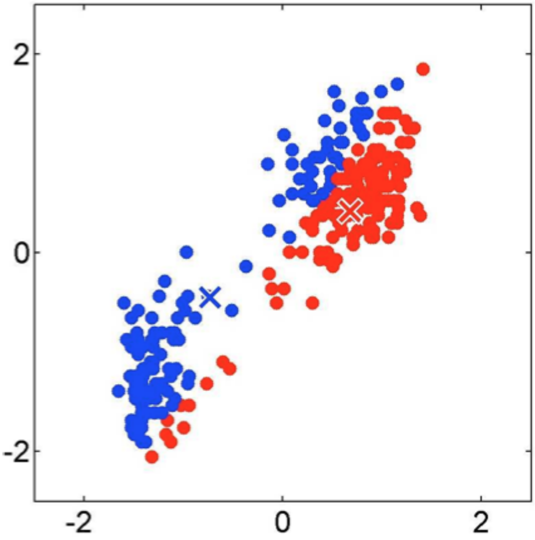
K-means: Initialization (assume $K = 2$)



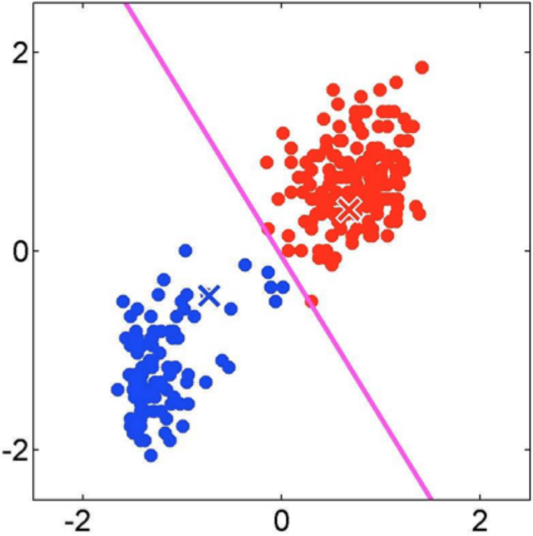
K-means iteration 1: Assigning points



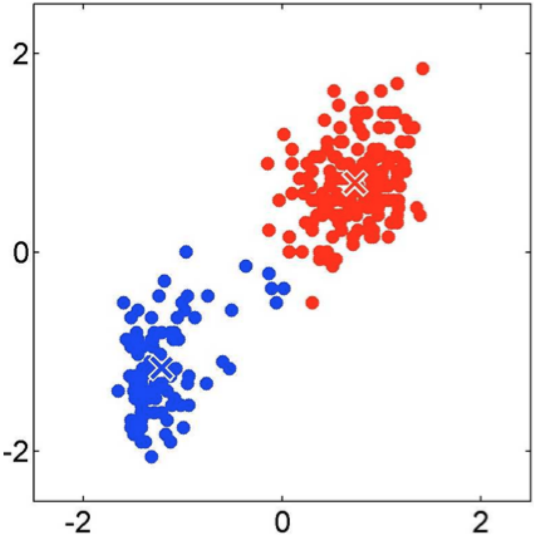
K-means iteration 1: Recomputing the cluster centers



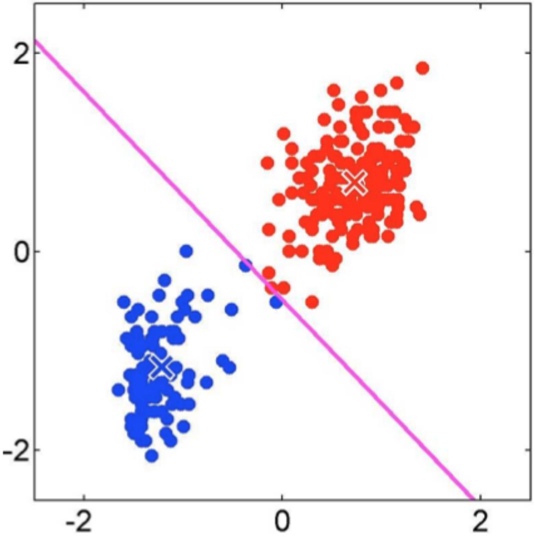
K-means iteration 2: Assigning points



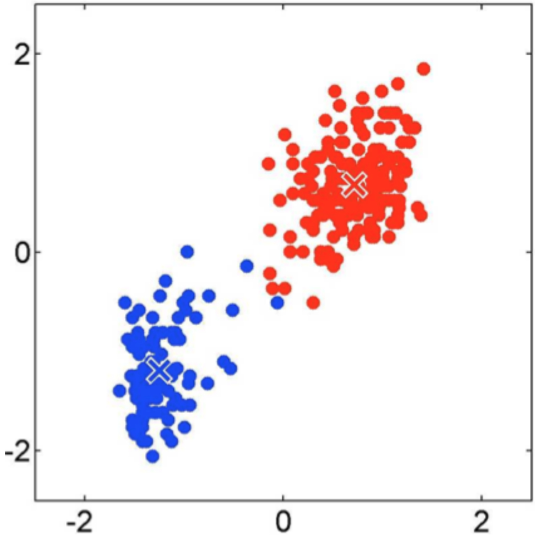
K-means iteration 2: Recomputing the cluster centers



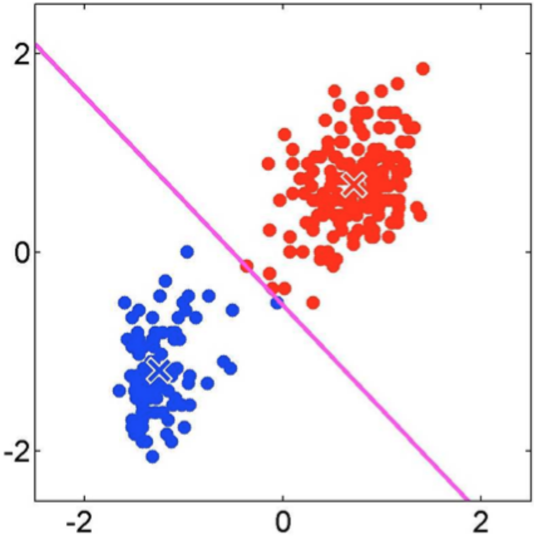
K-means iteration 3: Assigning points



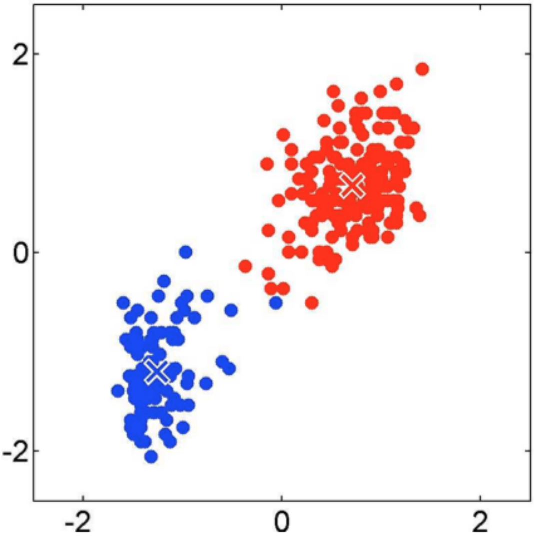
K-means iteration 3: Recomputing the cluster centers



K-means iteration 4: Assigning points



K-means iteration 4: Recomputing the cluster centers



K-means: The Objective Function

The K-means objective function

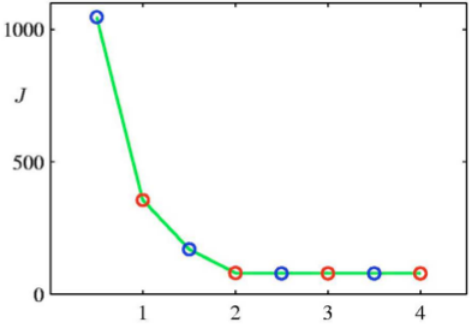
- Let $\mu_1, \ldots, \mu_K$ be the K cluster centroids (means)
- Let $r_{nk} \in \{0, 1\}$ be indicator denoting whether point $\mathbf{x}_n$ belongs to cluster k
- K-means objective minimizes the total distortion (sum of distances of points from their cluster centers)

$$J(\mu, r) = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} ||\mathbf{x}_n - \mu_k||^2$$

- Note: Exact optimization of the K-means objective is NP-hard
- The K-means algorithm is a heuristic that converges to a local optimum

K-means: Choosing the number of clusters K

- One way to select K for the K-means algorithm is to try different values of K, plot the K-means objective versus K, and look at the "elbow-point" in the plot



- For the above plot, $K = 2$ is the elbow point

Picture courtesy: "Pattern Recognition and Machine Learning", Chris Bishop (2006)

K-means: Initialization issues

- K-means is extremely sensitive to cluster center initialization
- Bad initialization can lead to
  - Poor convergence speed
  - Bad overall clustering
- Safeguarding measures:
  - Choose first center as one of the examples, second which is the farthest from the first, third which is the farthest from both, and so on.
  - Try multiple initializations and choose the best result