Unsupervised Learning

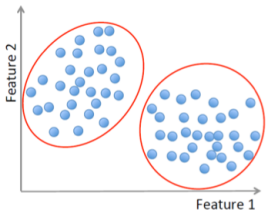
Training data: "examples" x.

$$x_1, \ldots, x_n, \quad x_i \in X \subset \mathbb{R}^d$$

- **Clustering/segmentation:**

$$f\colon \mathbb{R}^d \to \{C_1, \ldots, C_k\} \text{ (set of clusters).}$$

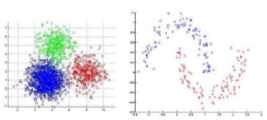
Example: Find clusters in the population, fruits, species.



Methods: K-means, Gaussian mixtures, hierarchical clustering, spectral clustering, etc.

Notion of Similarity

- Choice of **similarity** measure very important for clustering
- Similarity is inversely related to **distance**
- Different ways to measure distances:
  - Euclidean distance: $d(\mathbf{x}, \tilde{\mathbf{x}}) = \|\mathbf{x} - \tilde{\mathbf{x}}\|_2$
  - Manhattan distance: $d(\mathbf{x}, \tilde{\mathbf{x}}) = \sum_{i=1}^{d} |\mathbf{x}_i - \tilde{\mathbf{x}}_i|$
  - Kernelized distance: $d(\mathbf{x}, \tilde{\mathbf{x}}) = \|\phi(\mathbf{x}) - \phi(\tilde{\mathbf{x}})\|$



Clustering : K-Means

Algorithm K-Means:

Initialize randomly $\mu_1, \ldots, \mu_k$

Repeat

Assign each point xi to the cluster with the closest μ_j

Calculate the new mean for each cluster as follows:

$$\mu_j = \frac{1}{|C_j|} \sum_{x_i \in C_j} x_i$$

Until convergence*.

*Convergence: Means no change in the clusters OR maximum number of iterations reached.

- **Goal:** Assign each example $(x_1, \ldots, x_n)$ to one of the k clusters $\{C_1, \ldots, C_k\}$.
- μ_j is the mean of all examples in the j^{th} cluster.
- **Minimize:**

$$J = \sum_{j=1}^{k} \sum_{x_i \in C_j} ||x_i - \mu_j||^2$$

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

K-Means: pros & cons

+ Easy to implement

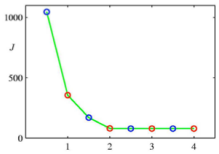
  BUT...

- Need to know K
- Suffer from the curse of dimensionality
- No theoretical foundation

Question 1

How to set k to optimally cluster the data?

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



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

G-means algorithm

1. Initialize k to be a small number
2. Run k-means with those cluster centers, and store the resulting centers as C
3. Assign each point to its nearest cluster
4. Determine if the points in each cluster fit a Gaussian distribution (Anderson-Darling test).
5. For each cluster, if the points seem to be normally distributed, keep the cluster center. Otherwise, replace it with two cluster centers.
6. Repeat this algorithm from step 2. until no more cluster centers are created.

Question 2

How to evaluate your model?

- Not trivial (as compared to counting the number of errors in classification).
- Internal evaluation: using same data. high intra-cluster similarity (documents within a cluster are similar) and low inter-cluster similarity. E.g., Davies-Bouldin index that takes into account both the distance inside the clusters and the distance between clusters. The lower the value of the index, the wider is the separation between different clusters, and the more tightly the points within each cluster are located together.
- External evaluation: use of ground truth of external data. E.g., mutual information, entropy, adjusted rand index, etc.

Question 3

How to cluster non circular shapes?

There are other methods: spectral clustering, kernelized K-means, DBSCAN, BIRCH, etc. that handle other shapes.

Question 4

How to initialize cluster centers?

- 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 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

Question 5

Other limitations

- Makes **hard assignments** of points to clusters
  - A point either completely belongs to a cluster or not belongs at all
  - No notion of a **soft assignment** (i.e., probability of being assigned to each cluster: say $K = 3$ and for some point x_i,
  - $p_1 = 0.7$; $p_2 = 0.2$; $p_3 = 0.1$)
  - **Gaussian mixture models** allows soft-assignments
- Sensitive to **outlier examples**
  - **K-median** algorithm is a more robust alternative for data with outliers
  - Reason: Median is more robust than mean in presence of outliers