

07-380 Notes

Clustering

Carnegie Mellon University
School of Computer Science

Contents

1 Unsupervised Learning: Clustering	1
1.1 Examples	2
1.2 K-means clustering	3
1.2.1 Algorithm	3
1.2.2 Optimization Formulation	6
1.2.3 Alternating Minimization	6

1 Unsupervised Learning: Clustering

Supervised learning is the process of learning a function that maps an input to an output based on example input-output pairs. In contrast, **unsupervised learning** is the process of learning a function that maps an input to an output based on input data without example output pairs.

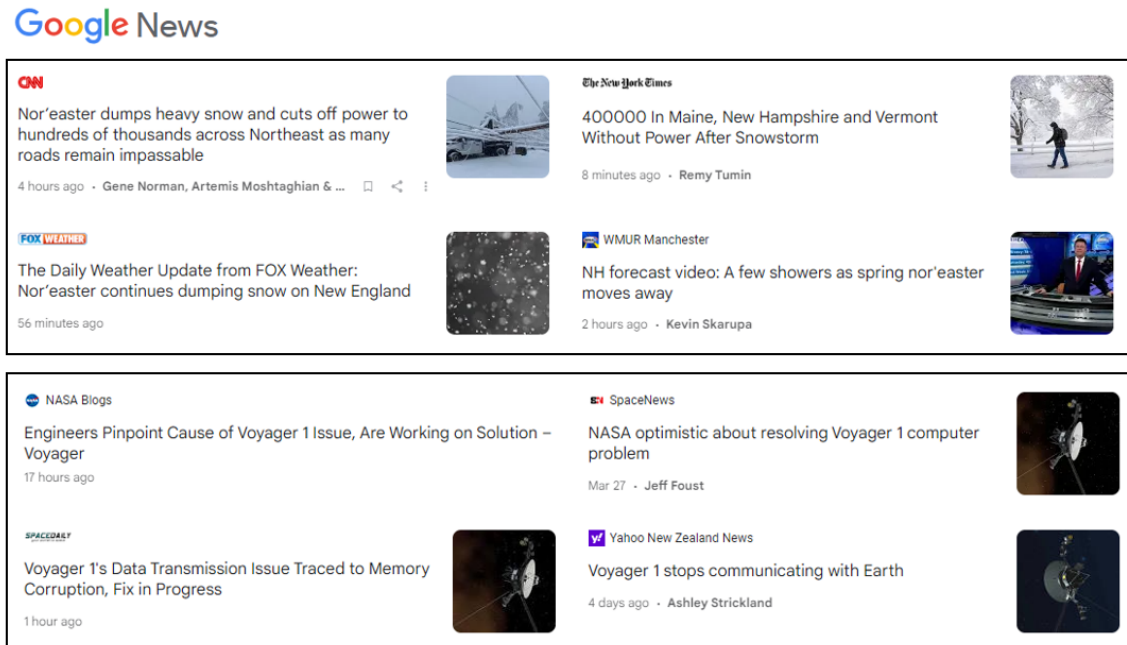
Clustering is a type of unsupervised learning that groups similar data points together. It is essentially classification when we don't have labeled data.

Even though we don't have labeled classes for our data points, clustering can be incredibly valuable. For example, clustering can be used to segment customers based on their purchasing behavior, to group similar documents together, or to identify patterns in data.

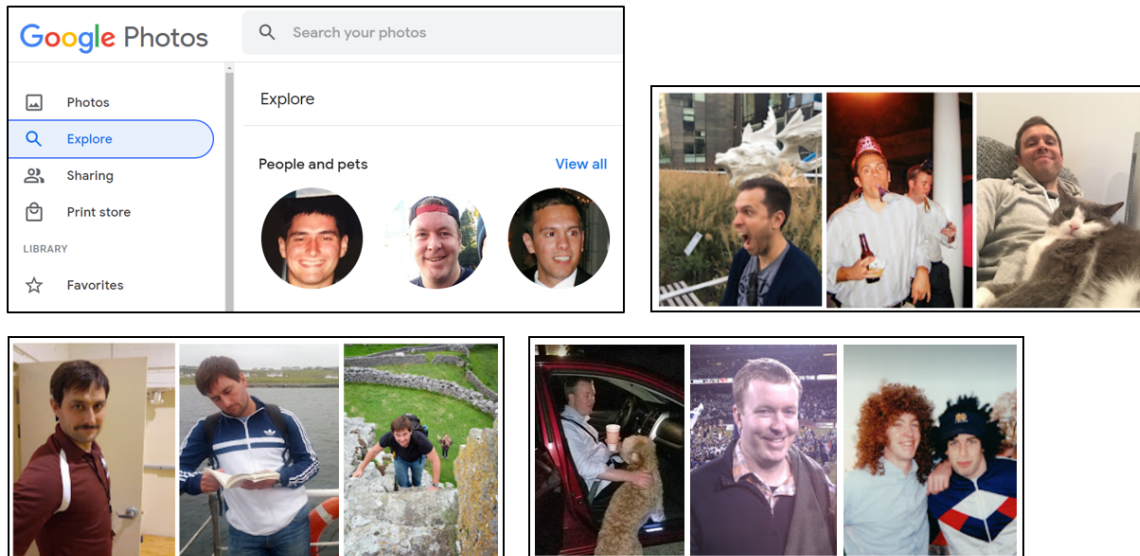
Clustering can also be used to analyze the underlying structure of the data or as a preprocessing step for a later task. For example, Google Photos can cluster photos without any labels, and then prompt the user to label the cluster with the name of the person that appears in the cluster photos.

1.1 Examples

Example: Google News Google News uses clustering to group similar news articles together. When a new article is published, Google News uses clustering to determine which cluster the article belongs to. This allows Google News to group similar articles together and present them to the user.



Example: Google Photos Google Photos uses clustering to group similar photos together. Despite going out of your way not to identify your college roommates, Google Photos can still group photos containing them together.



1.2 K-means clustering

K-means clustering is a popular clustering algorithm that groups data points into K clusters.

Given a dataset of N unlabeled M -dimensional data points, $\{\mathbf{x}^{(i)}\}_{i=1}^N$ with $\mathbf{x}^{(i)} \in \mathbb{R}^M$, the goal is to assign each point to a cluster such that we minimize the sum of the squared distances between each point and the center of its assigned cluster. The center of each cluster is the mean of all data points assigned to that cluster.

1.2.1 Algorithm

The K-means clustering algorithm is as follows:

Plain English version:

1. Randomly initialize K cluster centers
2. Assign each point in the training data to the cluster with the closest center
3. Update the cluster centers by computing the mean of all data points assigned to each cluster
4. Repeat steps 2 and 3 until the cluster centers no longer change

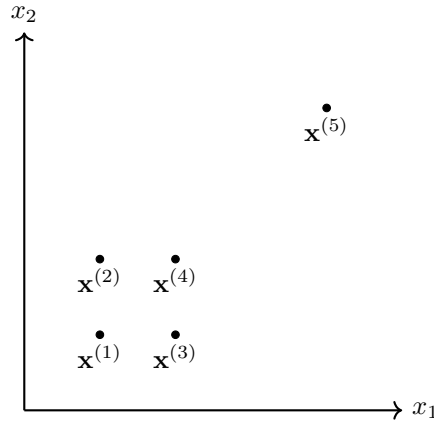
Repeated with a few more details:

1. Randomly initialize K M -dimensional points to represent the initial cluster means, $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \dots, \boldsymbol{\mu}_K$.
2. Assign each data point a cluster index, $z^{(i)} \in \{1, 2, \dots, K\}$, by finding the cluster centroid that is closest to the data point:

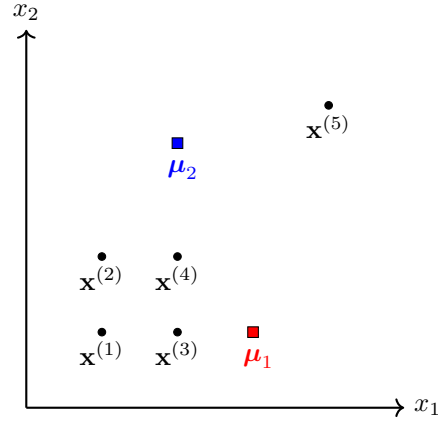
$$z^{(i)} = \underset{k}{\operatorname{argmin}} \|\mathbf{x}^{(i)} - \boldsymbol{\mu}_k\|_2^2$$

3. Update the cluster means by computing the mean of all data points assigned to each cluster, $\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{i=1}^N \mathbf{x}^{(i)} \mathbb{I}\{z^{(i)} = k\}$, where N_k is the number of data points assigned to cluster k .
4. Repeat steps 2 and 3 until the cluster means no longer change (i.e., cluster centers converge)

Example: Let's run the algorithm with $K = 2$ on the following five data points in $\mathbb{R}^2$:

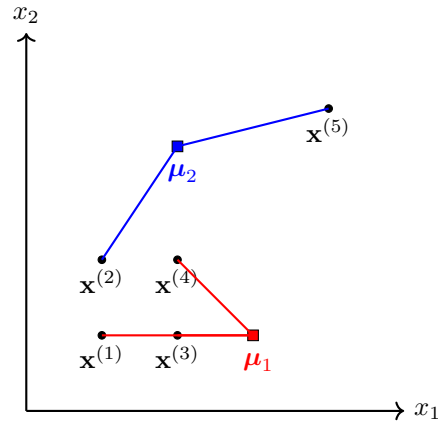


Step 1: Random Initialization We randomly initialize the cluster centers. For example, we might initialize the cluster centers as follows:



Where the two cluster centers are $\mu_1 = [3, 1]^\top$ and $\mu_2 = [2, 3.5]^\top$.

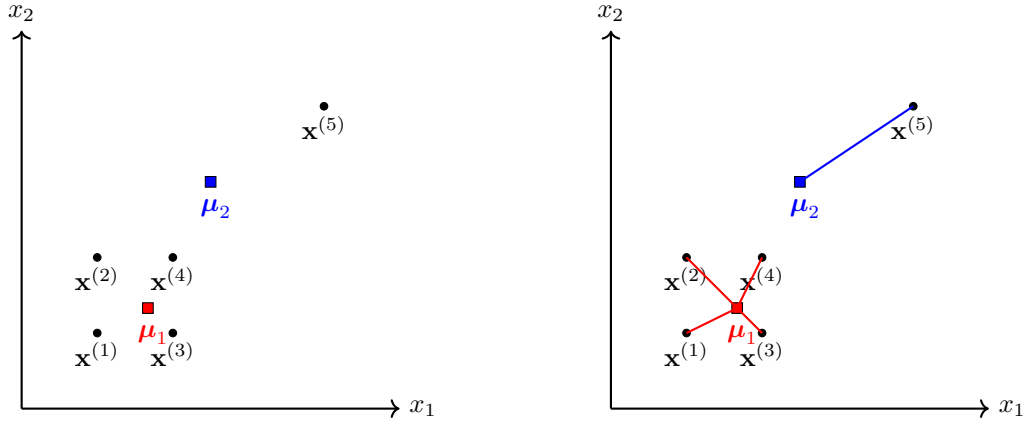
Step 2: Assign Data Points to Clusters We assign each data point to the cluster with the closest center. For example, we would assign the data points as follows:



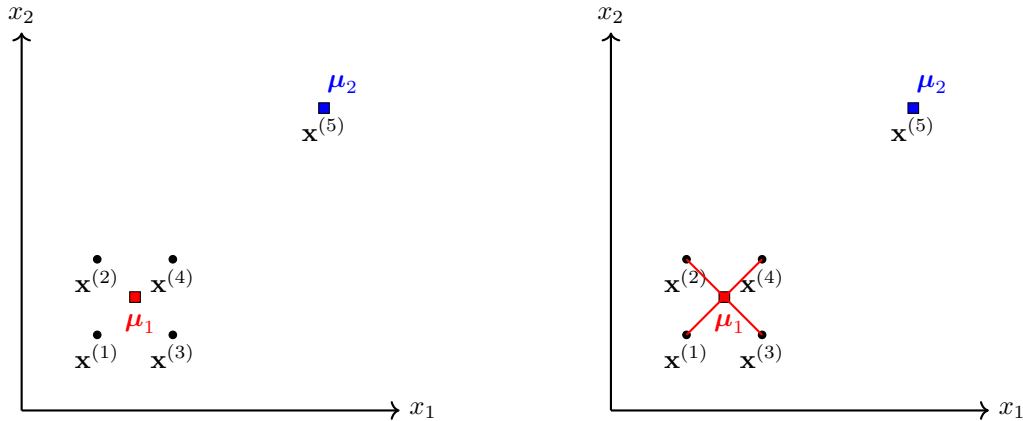
Where the data points are assigned to clusters as follows: $\{z^{(1)} = 1, z^{(2)} = 2, z^{(3)} = 1, z^{(4)} = 1, z^{(5)} = 2\}$.

Step 3: Update Cluster Centers We update the cluster centers by computing the mean of all data points assigned to each cluster. Cluster 1 contains $\mathbf{x}^{(1)}$, $\mathbf{x}^{(3)}$, and $\mathbf{x}^{(4)}$, so $\boldsymbol{\mu}_1 = [5/3, 4/3]^\top \approx [1.67, 1.33]^\top$, and cluster 2 contains $\mathbf{x}^{(2)}$ and $\mathbf{x}^{(5)}$, so $\boldsymbol{\mu}_2 = [2.5, 3]^\top$ (below, left).

Repeat Step 2: With the updated centers, $\mathbf{x}^{(2)}$ is now closer to $\boldsymbol{\mu}_1$ than to $\boldsymbol{\mu}_2$, so the assignments become $\{z^{(1)} = 1, z^{(2)} = 1, z^{(3)} = 1, z^{(4)} = 1, z^{(5)} = 2\}$ (below, right).



Repeat Step 3: Cluster 1 now contains $\mathbf{x}^{(1)}$ through $\mathbf{x}^{(4)}$, so $\boldsymbol{\mu}_1 = [1.5, 1.5]^\top$, and cluster 2 contains only $\mathbf{x}^{(5)}$, so $\boldsymbol{\mu}_2 = [4, 4]^\top$ (below, left). **Repeat Step 2:** With these centers, every point stays in its current cluster (below, right).



Converged: The cluster assignments didn't change, so repeating step 3 would leave the cluster means unchanged as well. The algorithm has converged and we stop.

1.2.2 Optimization Formulation

Our goal is to minimize the sum of the squared distances between each point and the center of its assigned cluster. We can formulate this as an optimization problem:

- Input: $\{\mathbf{x}^{(i)}\}_{i=1}^N$ with $\mathbf{x}^{(i)} \in \mathbb{R}^M$
- Output: $z^{(i)} \in \{1, 2, \dots, K\}$, the cluster assignment for each data point i
- Output: $\boldsymbol{\mu}_k \in \mathbb{R}^M$, the center of cluster k

$$\min_{\{z^{(i)}\}, \{\boldsymbol{\mu}_k\}} \sum_{i=1}^N \|\mathbf{x}^{(i)} - \boldsymbol{\mu}_{z^{(i)}}\|_2^2$$

1.2.3 Alternating Minimization

The objective is actually quite difficult to optimize directly. Instead, we can use an **alternating minimization** approach. We alternate between two steps: (1) fixing the cluster assignments and updating the cluster centers, and (2) fixing the cluster centers and updating the cluster assignments.

1. **Fix cluster assignments, update cluster centers:** Given the cluster assignments, we can update the cluster centers by computing the mean of all data points assigned to each cluster.

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{i=1}^N \mathbf{x}^{(i)} \mathbb{I}\{z^{(i)} = k\}$$

2. **Fix cluster centers, update cluster assignments:** Given the cluster centers, we can update the cluster assignments by assigning each data point to the cluster with the closest center.

$$z^{(i)} = \underset{k}{\operatorname{argmin}} \|\mathbf{x}^{(i)} - \boldsymbol{\mu}_k\|_2^2$$