CHEME 5820: Machine Learning for Engineers	Spring 2025

Lecture 1c: Unsupervised learning and clustering

Lecturer: Jeffrey Varner

Disclaimer: *These notes have not been subjected to the usual scrutiny reserved for formal publications.*

Lecture Summary

- **Unsupervised learning** is a type of machine learning that involves training algorithms on unlabeled data. The goal of unsupervised learning is to identify patterns and structures in data without explicit guidance. Unsupervised learning is particularly useful when dealing with large volumes of unstructured data or when the desired outcomes are unknown.
- **Clustering** is a common unsupervised learning technique that involves dividing a dataset into distinct groups, or clusters, based on the similarity of data points. Clustering algorithms aim to group data points that are more similar to each other than to those in other clusters.
- **K-means clustering** is a popular and straightforward clustering algorithm that partitions a dataset into k clusters. The algorithm iteratively assigns data points to the nearest cluster center and updates the cluster centers based on the mean of the assigned points.

---

1 Introduction

This lecture introduces the first of several unsupervised learning approaches we will explore: K-means clustering. Unsupervised learning is a branch of machine learning focused on the discovery of hidden patterns and insights from unlabeled data without explicit human guidance. Unlike supervised learning, which relies on labeled datasets, unsupervised learning models are given raw, unstructured information and tasked with identifying inherent structures, relationships, and similarities within the data. This approach is particularly valuable for exploratory data analysis, clustering similar data points, and uncovering previously unknown trends in datasets. The K-means clustering algorithm can be used to identify hidden patterns and structures in data without explicit guidance, or labels. Thus, it is particularly useful when dealing with large volumes of unstructured data or when the desired outcomes are unknown.

2 K-means clustering

The K-means algorithm, originally developed by Lloyd in the 1950s but not published until much later in 1982 (1), is an example of `unsupervised learning`. Unsupervised learning focuses on discovering patterns and structures within data without the guidance of labeled examples or explicit feedback. Thus, unlike supervised learning (which we will explore in future lectures), where algorithms are trained on labeled datasets, unsupervised learning algorithms operate with raw, unlabeled data to identify inherent groupings, anomalies, or relationships. This approach is particularly valuable when dealing with large volumes of unstructured data or when the desired outcomes may be unknown. Typical applications of unsupervised learning include clustering (which we are discussing today), dimensionality reduction, and anomaly detection.

K-means is a popular unsupervised machine learning algorithm used for clustering data points into K distinct groups based on their similarity. In this approach, the algorithm partitions the dataset into K

(specified by you) clusters, with each cluster represented by a centroid (the mean of the data points in the cluster). Then the algorithm iteratively assigns data points to the nearest cluster centroid and updates the centroids based on the mean of the assigned points.

2.1 Problem formulation

Suppose we have a dataset $\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n \in \mathbb{R}^m\}$, where each data point $\mathbf{x}_i$ is a m-dimensional vector. The goal of the k-means algorithm is to partition the dataset into K clusters, $\mathcal{C} = \{c_1, c_2, \ldots, c_K\}$, where each cluster c_j is represented by a centroid $\mu_j \in \mathbb{R}^m$. To enforce that $\mathcal{C}$ is a valid partition of $\mathcal{D}$, we require that each data point $\mathbf{x}_i$ is assigned to exactly one cluster c_j, i.e., $c_i \cap c_j = \emptyset$ for $i \neq j$ (no shared members). In addition, the union of all clusters covers the entire dataset, i.e., $\cup_{j=1}^{K} c_j = \mathcal{D}$.

Puesdo code for the k-means algorithm is shown in Algorithm 1.

Algorithm 1 Unsupervised naive k-means clustering (Lloyd's algorithm)

Input: $\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n \in \mathbb{R}^m\}$, number of clusters K and initial centroids $\{\mu_1, \mu_2, \ldots, \mu_K \in \mathbb{R}^m\}$
Output: Cluster assignments $\mathcal{C} = \{c_1, c_2, \ldots, c_K\}$ and updated cluster centroids $\{\mu_1, \mu_2, \ldots, \mu_K\}$
flag $\leftarrow$ `false`   ▷ flag to indicate convergence: `true` for convergence and `false` otherwise
while flag is `false` **do**
    for $\mathbf{x} \in \mathcal{D}$ **do**   ▷ Iterate over all data points in $\mathcal{D}$
        $c_i \leftarrow \arg\min_j \|\mathbf{x} - \mu_j\|^2$ ▷ Assign data point $\mathbf{x}$ to the closest cluster centroid (Euclidean distance)
    end for

    $\hat{\mu} \leftarrow \mu$ ▷ Store the current best cluster centroids
    for $j = 1$ to K **do**   ▷ Iterate over all clusters
        $\mu_j \leftarrow \dfrac{1}{|c_j|} \cdot \sum\limits_{\mathbf{x} \in c_j} \mathbf{x}$ ▷ Update cluster centroid μ_j where $|c_j|$ is the number of data points in cluster c_j
    end for

    if $\|\mu - \hat{\mu}\| < \epsilon$ **then**   ▷ Check for convergence: based on the change in cluster centroids
        flag $\leftarrow$ `true`   ▷ Set flag to `true` to terminate the algorithm
    end if
end while

---

3 Estimating the number of clusters

The K-means algorithm is simple and intuitive, but it has some limitations. One of the main drawbacks of k-means is that it requires the number of clusters K to be specified in advance, which can be challenging when the number of clusters is unknown. There are several methods to estimate the number of clusters, including the elbow method, the silhouette method, or performance metrics such as the Davies-Bouldin index, the Dunn index or the Calinski-Harabasz index.

3.1 Silhouette method

The silhouette method is a technique to evaluate the consistency of data within clusters. The silhouette score ranges from -1 to 1, where a high score indicates that the data point is well matched to its own cluster and poorly matched to neighboring clusters. If most objects have a high silhouette score, then the clustering configuration, i.e., the number of clusters k is appropriate. However, if many points have a low or

negative value, then the clustering configuration may have too many or too few clusters. Assume that we have clustered the data using k-means into K clusters $\mathcal{C} = \{c_1, c_2, \ldots, c_K\}$. Then, for a data point $\mathbf{x}_i \in c_i$, let $a(i)$ denote the average distance between $\mathbf{x}_i$ and all other points in the same cluster c_i:

$$a(i) = \frac{1}{|c_i| - 1} \sum_{j \in c_i, j \neq i} d(i, j) \tag{1}$$

where $|c_i|$ denotes the number of data points in cluster c_i, and $d(i, j)$ denotes the distance between data points $\mathbf{x}_i \in c_i$ and $\mathbf{x}_j \in c_i$. Next, we define $b(i)$, the mean dissimilarity of $\mathbf{x}_i$ to all other points not in the same cluster c_i:

$$b(i) = \min_{j \neq i} \frac{1}{|c_j|} \sum_{j \in c_j} d(i, j) \tag{2}$$

Putting it all together, the silhouette score $s(i)$ for data point $\mathbf{x}_i$ is defined as:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \quad \text{if} |c_i| > 1 \tag{3}$$

If $|c_i| = 1$, then $s(i) = 0$. This gives a silhouette score for each data point as:

$$s(i) = \begin{cases} 1 - a(i)/b(i) & \text{if } a(i) < b(i) \\ 0 & \text{if } a(i) = b(i) \\ b(i)/a(i) - 1 & \text{if } a(i) > b(i) \end{cases} \tag{4}$$

From this definition we see that $-1 \leq s(i) \leq 1$. The mean $s(i)$ over all data points in a cluster is a measure of how tightly grouped all the points in the cluster are. Thus the mean of $s(i)$ over all data in the entire dataset is a measure of how appropriately the data have been clustered. If there are too many or too few clusters, as may occur when a poor choice of K is made, then some of the clusters will typically display much narrower silhouettes than the rest. Thus silhouette plots and means may be used to visualize the natural number of clusters within a dataset.

3.2 Calinski-Harabasz index

The Calinski–Harabasz index (CHI), also known as the Variance Ratio Criterion, is a widely used metric for assessing the quality of clustering algorithms (2). To compute the CHI, we first need to perform clustering for different values of k. Then, we compute the CH index for each clustering result. Finally, the value of k that yields the maximum CH index is chosen as the optimal number of clusters. The CH index is defined as:

$$\text{CHI} = \left(\frac{n - k}{k - 1} \right) \cdot \left(\frac{\sum_{i=1}^{k} n_i \|\mathbf{c}_i - \mathbf{c}\|^2}{\sum_{k=1}^{k} \sum_{\mathbf{x} \in C_i} \|\mathbf{x} - \mathbf{c}_i\|^2} \right) \tag{5}$$

where n denotes the number of feature vectors $\mathbf{x}_i$ that we are clustering, k denotes the number of clusters, n_i denotes the number of points in cluster i, $\mathbf{c}_i$ denotes the centroid for cluster i and $\mathbf{c}$ denotes the overall centroid of the data. The numerator measures how well the clusters are separated from each other (the higher, the better), while the denominator measures the compactness or cohesiveness of the clusters (the smaller, the better).

4   Summary and Conclusion

In this lecture, we introduced the concept of unsupervised learning and discussed two common unsupervised learning algorithms: k-means clustering and self-organizing maps. Unsupervised learning is a type of machine learning that involves training algorithms on *unlabeled data* to identify patterns and structures within data without explicit guidance. Clustering is a common unsupervised learning technique that involves dividing a dataset into distinct groups, or clusters, based on the similarity of data points. We explored k-means clustering, which partitions a dataset into k clusters and identify patterns that may not be immediately apparent.

References

1. Lloyd S. Least squares quantization in PCM. IEEE Transactions on Information Theory. 1982;28(2):129–137. doi:10.1109/TIT.1982.1056489.

2. T Caliński and J Harabasz. A dendrite method for cluster analysis. Communications in Statistics. 1974;3(1):1–27. doi:10.1080/03610927408827101.