CHEME 5820: Machine Learning for Engineers **Spring 2025**

Lecture 1c: Unsupervised learning and K-means clustering

Lecturer: Jeffrey Varner

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

Topics

- **Unsupervised learning** is a type of machine learning that involves training algorithms on unlabeled data. Unsupervised learning aims 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 typical 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 discovering 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 identify hidden patterns and structures in data without explicit guidance or labels. Thus, it is beneficial when dealing with large volumes of unstructured data or when the desired outcomes are unknown. In Chemical and Biomolecular Engineering, K-means clustering can be used to group similar molecules based on their chemical properties (1), identify patterns in gene expression data (2, 3), or segment customers based on purchasing behavior (4). Thus, K-means clustering has many applications in chemical engineering, bioengineering, and related fields.

2 K-means clustering

The K-means algorithm, originally developed by Lloyd in the 1950s but not published until much later in 1982 (5), 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 for clustering data points into K distinct groups based on similarity. In this approach, the algorithm partitions the dataset into K (specified by you) clusters, each representing 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 $\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n \in \mathbb{R}^m\}$, 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, $\mathcal{C} = \{c_1, c_2, \dots, c_K\}$, 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., $\cup_{j=1}^{K} c_j = \mathcal{D}$. The problem of K-means clustering can be formulated as an optimization problem, where the objective is to minimize the sum of squared distances between each data point and its assigned cluster centroid:

$$\min_{\mathcal{C}} \sum_{j=1}^{K} \sum_{\mathbf{x} \in c_j} ||\mathbf{x} - \mu_j||_2^2 \tag{1}$$

where $||\star||_2^2$ denotes the squared Euclidean distance between two vectors (in this case, a feature vector and the cluster centroid). The K-means algorithm aims to find the optimal cluster assignments $\mathcal{C}$ and cluster centroids μ that minimize this objective function. Puesdo code for K-means (Lloyd's algorithm) is shown in Algorithm 1.

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

```
Input: $\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n \in \mathbb{R}^m\}$, number of clusters $K$ and initial centroids $\{\mu_1, \mu_2, \dots, \mu_K \in \mathbb{R}^m\}$
Output: Cluster assignments $\mathcal{C} = \{c_1, c_2, \dots, c_K\}$ and updated cluster centroids $\{\mu_1, \mu_2, \dots, \mu_K\}$
flag ← 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
        $\mu_j \leftarrow \frac{1}{|c_j|} \cdot \sum_{\mathbf{x} \in c_j} \mathbf{x}$▷ Update cluster centroid $\mu_j$ 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 ← true    ▷ Set flag to true to terminate the algorithm
    end if
end while
```

The time cost of Lloyd's algorithm is $\mathcal{O}(nmk)$, where n is the number of feature vectors, m is the number of dimensions of each feature vector, and k is the number of clusters.

3 Issues with K-means clustering

The K-means algorithm is simple and intuitive but has some limitations (6). 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 this choice is not apparent. Furthermore, the algorithm is sensitive to the initial choice of cluster centroids, which can lead to suboptimal solutions. Verification of the computed solution's global optimality is generally challenging, and the subject of ongoing research, see (7–9). Finally, K-means is unsuitable for clustering datasets with non-convex shapes or varying cluster sizes, as it assumes that clusters are spherical and have similar sizes. See Example 1 for an illustration of K-means failure.

Example 1 (K-means failure)**.** *Consider two circles, each of radius 1, and their centers are a distance d apart. As long as $d > 2.08$, K-means yields the correct answer. But if $d \leq 2.08$, K-means fails. This is the failure of K-means and not of Lloyd's algorithm.*

Despite these limitations, K-means is a widely used clustering algorithm that can be effective in many scenarios.

3.1 Estimating the number of clusters

Of K-means' shortcomings, the need to specify the number of clusters K in advance can be addressed with several heuristic methods. 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.

Silhouette method

The silhouette method is a technique to evaluate the consistency of data within clusters. The silhouette score ranges from -1 to 1. A high score indicates that the data point is well-matched to its 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, 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:

$$a(i) = \frac{1}{|c_i| - 1} \sum_{j \in c_i, j \neq i} d(i, j) \tag{2}$$

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:

$$b(i) = \min_{j \neq i} \frac{1}{|c_j|} \sum_{j \in c_j} d(i, j) \tag{3}$$

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

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \quad \text{if} |c_i| > 1 \tag{4}$$

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

$$s(i) = \begin{cases} 1 - a(i)/b(i) & \text{if } a(i) < b(i) \\ 0 & \text{if } a(i) = b(i) \\ b(i)/a(i) - 1 & \text{if } a(i) > b(i) \end{cases} \tag{5}$$

This definition shows $-1 \leq s(i) \leq 1$. The mean $s(i)$ of all data points in a cluster measures how tightly grouped the points in the cluster are. Thus, the mean of $s(i)$ over all data in the entire dataset measures 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, some clusters will typically display much narrower silhouettes than the rest. Thus, silhouette plots and means may be used to visualize a dataset's natural number of clusters.

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 (10). 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:

$$\text{CHI} = \left(\frac{n-k}{k-1}\right) \cdot \left(\frac{\sum_{i=1}^{k} n_i \|\mathbf{c}_i - \mathbf{c}\|^2}{\sum_{k=1}^{k} \sum_{\mathbf{x} \in C_i} \|\mathbf{x} - \mathbf{c}_i\|^2}\right) \tag{6}$$

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). In contrast, 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 a common unsupervised learning algorithm: K-means clustering. Unsupervised learning aims to identify patterns and structures in data without explicit guidance or labeled examples. The K-means algorithm is a popular clustering technique that partitions a dataset into K clusters based on the similarity of data points within each cluster. We introduce Lloyd's algorithm for K-means clustering, which iteratively assigns data points to the nearest cluster centroid and updates the centroids based on the mean of the assigned points. While K-means clustering is a simple and intuitive algorithm, it has some limitations, such as specifying the number of clusters in advance. Toward this challenge, we discussed the difficulties of estimating the number of clusters in K-means clustering and introduced two methods for determining the number of clusters: the silhouette method and the Calinski-Harabasz index.

References

1. Hadipour H, Liu C, Davis R, Cardona ST, Hu P. Deep clustering of small molecules at large-scale via variational autoencoder embedding and K-means. BMC Bioinformatics. 2022;23(4):132. doi:10.1186/s12859-022-04667-1.

2. Hruschka ER, de Castro LN, Campello RJGB. In: Abraham A, Grosan C, Ishibuchi H, editors. Clustering Gene-Expression Data: A Hybrid Approach that Iterates Between k-Means and Evolutionary Search. Berlin, Heidelberg: Springer Berlin Heidelberg; 2007. p. 313–335. Available from: `https://doi.org/10.1007/978-3-540-73297-6_12`.

3. Wu FX. Genetic weighted k-means algorithm for clustering large-scale gene expression data. BMC Bioinformatics. 2008;9(6):S12. doi:10.1186/1471-2105-9-S6-S12.

4. Shaikh N, Shahu H, Patel R, Patel D. Customer Segmentation Using K-means Clustering. In: Somani AK, Mundra A, Gupta RK, Bhattacharya S, Mazumdar AP, editors. Smart Systems: Innovations in Computing. Singapore: Springer Nature Singapore; 2024. p. 135–147.

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

6. Raykov YP, Boukouvalas A, Baig F, Little MA. What to Do When K-Means Clustering Fails: A Simple yet Principled Alternative Algorithm. PLoS One. 2016;11(9):e0162259. doi:10.1371/journal.pone.0162259.

7. Peng J, Wei Y. Approximating K-means-type Clustering via Semidefinite Programming. SIAM Journal on Optimization. 2007;18(1):186–205. doi:10.1137/050641983.

8. Iguchi T, Mixon DG, Peterson J, Villar S. Probably certifiably correct k-means clustering; 2016. Available from: `https://arxiv.org/abs/1509.07983`.

9. Clum C, Mixon DG, Villar S, Xie K. Sketch-and-solve approaches to k-means clustering by semidefinite programming; 2022. Available from: `https://arxiv.org/abs/2211.15744`.

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