How Linear Algebra is Related to the K-Means Clustering Algorithm – An Unsupervised Machine Learning Algorithm

Introduction

At some point in our lives, we all have studied linear algebra or at least something related to it. However, when it comes to Data Science and Machine Learning, not all of us have some kind of exposure to it. Linear algebra comprises of algorithms and is a much more powerful tool than we think. It is the fundamental mathematical tool in data science and machine learning. Linear algebra is the prime mover / tool / framework that is used to represent and manipulate data, in a structured, efficient and mathematical form, which can then be analyzed, calculations performer upon, and interpreted.

Linear algebra, which is used to represent and manipulate data as vectors and matrices, is essential to several areas of data science, including physics, engineering, computer science, data science, machine learning, data mining, signal processing, and image processing. Several algorithms in data science rely on linear algebra concepts such as eigenvalues, eigenvectors, matrix factorization, and linear transformations.

Linear algebra is used in a wide range of applications from data representation (where the data is represented in a matrix form and can then be manipulated using matrix operations), to dimensionality reduction (where techniques such as principal component analysis and singular value decomposition are utilized to reduce the dimensionality of large datasets), to machine learning (where machine learning algorithms, such as linear regression, logistic regression, and support vector machines use linear algebra to represent the relationships between input variables and output variables), and, to optimization (where optimization problems are solved by minimizing the error in a machine learning model).

As mentioned above, one of the key utilizations of linear algebra is in the field of machine learning, where algorithms and statistical models that enable computer systems to learn patterns from data, and make predictions or decisions without being explicitly programmed, are developed. Machine learning models are explicitly designed to mechanically recognize and analyze patterns and relationships in large amounts of data in an intelligible manner, and to then use these patterns to make predictions or decisions about new data. Overall, machine learning has a wide range of applications, ranging from image and speech recognition to natural language processing, to predictive modeling in finance and healthcare.

One of the key tasks in machine learning is clustering, which involves grouping similar data points together based on some similarity metric. Clustering is used in a variety of applications, such as customer segmentation, image processing, and anomaly detection. One popular clustering algorithm is the k-means algorithm, which involves finding k centroids that minimize the sum of squared distances between each data point and its nearest centroid.

In this paper, we will discuss the mathematical concepts behind the k-means algorithm and its implementation using linear algebra.

Within machine learning there are several types of learning paradigms, including supervised learning, unsupervised learning, and reinforcement learning, the k-means clustering algorithm is a popular form of unsupervised learning technique and recommendation algorithm that requires no labeled response for the given input data. It is used for clustering, which involves grouping data points into clusters based on their similarity, in other words, K-means clustering is utilized for clustering data points into k clusters based on their similarity. It is widely used in recommendation systems to group similar users or items based on their preferences or attributes.

Overall, Machine Learning, particularly unsupervised learning and recommendation algorithms like k-means clustering, provide powerful tools for data analysis, pattern discovery, and decision-making in various domains.

The k-means clustering algorithm can be applied to various domains, such as customer segmentation, image recognition, document clustering, and anomaly detection. In the context of recommendation systems, k-means clustering can be used to group users or items based on their preferences or characteristics, enabling personalized recommendations for users based on the preferences of similar users or the characteristics of similar items.

In the real-world, k-means clustering in recommendation systems utilization is in Netflix's movie recommendation algorithm. The algorithm uses k-means clustering to group similar users based on their movie preferences, and then recommends movies that were highly rated by users in the same cluster. Another example is Amazon's product recommendation algorithm, which uses k-means clustering to group similar products based on their attributes, such as price and category, and then recommends related products to customers based on their purchase history.

(Unsupervised learning refers to the process of extracting patterns or structures from data without any prior knowledge or labeled examples. It aims to discover hidden patterns, group similar data points, or reduce the dimensionality of the datasets.)

Main Body

The main objective of the k-means algorithm is related a lot to linear algebra. This algorithm states that “it aims to minimize the sum of squared distances between data points and their respective cluster centroids”.

1. Because we’re trying to group similar data points into partitions or clusters, we term it as ‘k’ clusters and the first step is to pick several random k clusters. Alternatively, we can also choose the number of clusters we want depending on the dataset.
2. Next, we pick k number of data points to be the centers of these clusters or to be the ‘cluster centroids’. This can be done randomly by the algorithm as well.
3. We then assign each data point with the closest cluster center and using the Euclidian distance formula, we find the distance between the points and the cluster centers. The Euclidian distance formula is given as follows –

$$d((x,y),(a,b)) = \sqrt{(x-a)^2 + (y-b)^2}$$

Where (x,y) and (a,b) are dimensional vectors representing the two points from which we can calculate the distance.

4. Following this, we must re-compute the centroid of each cluster which as it happens, is also represented as a vector (the re-computing of the centroids is done based on the mean of all the data points assigned to that particular cluster).

To do this, we must compute the mean of the data points associated with each cluster. And this is done with a series of matrix operations such as addition and division.

$$z_j = (1/|G_j|) \sum_{i \epsilon G_j} x_i$$

This essentially means that 1 is divided by the absolute value |Gj| which is the number of data points with respect to cluster j and is then multiplied by the sum of the number of data points with respect to cluster j. This centroid is computed using matrix division as Gj and |Gj| can be represented as matrices.

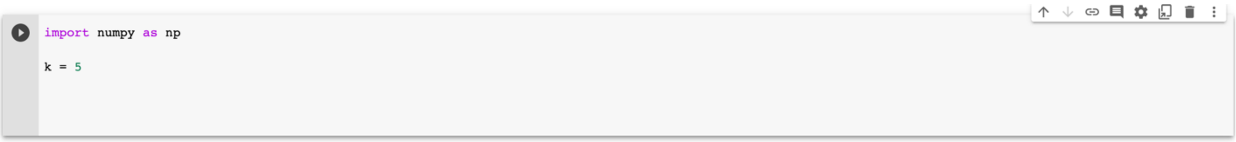
Finally, we must keep doing steps 3 and 4 repeatedly until the data points fall in the same cluster or if the centroids no longer change or any other criteria which would allow it to stop. We see that these steps are heavily dependent on linear algebra, mainly, vector and matrix operations. Finding the Euclidian distance between two points involves vector subtraction and computing the mean of a set of vectors is also a linear algebra operation.

We can easily implement this k-means algorithm in python using a simple package known as sklearn.cluster.KMeans. However, since I want to emphasis the linear algebra aspect of the k-means algorithm, I won’t simply use the KMeans package.

Note that the following code is only meant for informative purposes and just gives you a broad understanding as to how the k-means algorithm works in python.

We will use the Numpy package which is very useful in the field of linear algebra as it provides several linear algebra functions which can be used to implement the k-means algorithm.

Step 1

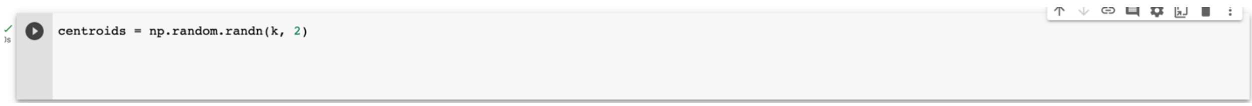


```
import numpy as np

k = 5
```

As we spoke about in our writeup, we choose the number of clusters. In my instance here, I have chosen k as 5, i.e., there are 5 clusters. You can choose how many ever clusters as you want.

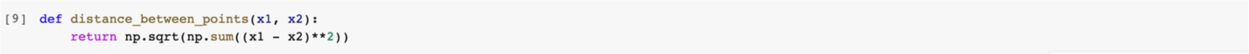
Step 2



```
centroids = np.random.randn(k, 2)
```

In this step, we are just picking the number of cluster centroids and I did this with the random.randn function which generates ‘k’ random centroids and the 2 besides that means that each randomly generated centroid is a 2 Dimensional (2-D) array representing a particular point. It is also a known fact that the k-means algorithm generally operates in a 2-D space.

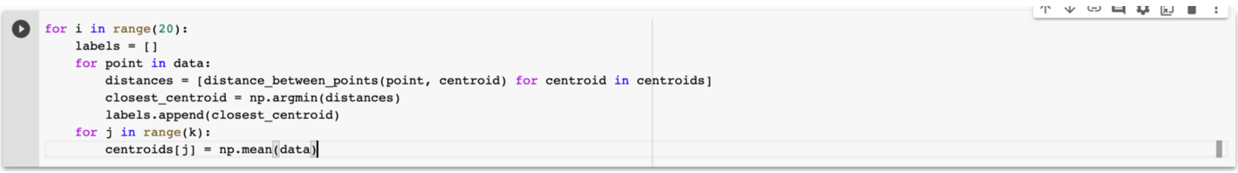
Step 3



```
[9] def distance_between_points(x1, x2):
        return np.sqrt(np.sum((x1 - x2)**2))
```

Here, we apply the Euclidean distance formula and find the distance between two points assuming we have some data. This code snippet gives a general idea as to how to implement the Euclidian distance formula in python.

Step 4



```
for i in range(20):
    labels = []
    for point in data:
        distances = [distance_between_points(point, centroid) for centroid in centroids]
        closest_centroid = np.argmin(distances)
        labels.append(closest_centroid)
    for j in range(k):
        centroids[j] = np.mean(data)
```

This code snippet essentially uses a for loop and loops it 20 times as shown in the bracket. (Because we specified earlier that we had to repeat those two steps over and over again.)

The main objective of the first part of the above code is to use the distance_between_points function and find the distances between the data points and each centroid (that is what ‘distances’ is for). Then, it selects the centroid with the minimum distance (thus the ‘closest_centroid’ aspect in the code snippet). ‘labels’ is just a list which stores the cluster assignments for each point.

The second for loop is just like step 4 in what I mentioned above in the writeup.

Once again, the complete code is a bit more complex and beyond my capabilities at the moment, but I hope that this gives you a better understanding of the k-means algorithm and how linear algebra is so important within it.

Applications of the k-means algorithm are –

1. Image segmentation: In image processing, the k-means algorithm is used to segment an image into regions based on color similarity. Similar pixel values are grouped and a lot of linear applications are used to calculate distances, centroids, etc.
2. Anomaly detection: The k-means algorithm can be used to detect anomalies in data by identifying data points that do not belong to any cluster. Once again, linear algebra operations are used for determining the distances between data points and the centroids.
3. Customer segmentation: In marketing, the k-means algorithm is used to segment customers based on their buying behavior, trends, past purchases, etc.

Conclusion

Thus, from all the information above, we can understand that the k-means algorithm relies heavily on a lot of linear algebra aspects such as vectors, matrices, and matrix operations (as we saw in the formulas). The k-means algorithm also has numerous benefits when it comes to real world problems. Netflix – the massive streaming platform which almost all of us use, implements the k-means algorithm mainly to give movie/series recommendations to users based on their browsing history, most watched genre, etc. This is similar to what I spoke about regarding customer segmentation. I feel like it is extremely useful for companies but the only critical point I would have - is that sensitive information of users may be compromised and obtained without their knowledge, thus leading to a violation of data ethics. However, all in all, linear algebra plays a crucial role in the world of Data Science and Machine Learning as we have seen with the k-means clustering algorithm. It is a kind of a foundation on which many machine learning models are built on and thus should be investigated thoroughly to maximize its potential uses and benefits.

References

https://stackoverflow.com/questions/15604647/k-means-clustering-algorithm

https://medium.com/mlearning-ai/introduction-to-applied-linear-algebra-k-means-clustering-c6885cad0f7f

https://www.simplilearn.com/tutorials/machine-learning-tutorial/k-means-clustering-algorithm

https://journals.sagepub.com/doi/abs/10.1243/095440605X8298?journalCode=picb

https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html

https://builtin.com/data-science/cluster-analysis