Unsupervised learning

UNSUPERVISED LEARNING

Unsupervised learning models are used when we only have the input variables (X) and no corresponding output variables. They use unlabeled training data to model the underlying structure of the data.

CLUSTERING

What is clustering?

Clustering is used to group samples such that objects within the same cluster are more similar to each other than to the objects from another cluster.

What are the goals for clustering?

-We want observations in different clusters to be dissimilar to each other

-Homogeneity: We want observations in the same cluster to be similar to each other.

-Find natural groupings.

We want to minimize the variance inside a cluster.

What are the inputs and outputs?

The inputs will be a set of numerical inputs normally scaled, and without outliers.

The outputs will be a set of labels, one for each observation, and also a set of centroids, one for each cluster.

To achieve good clustering, you need to:

-Choose the right distance metric.

-Have good intuition behind your data.

K-MEANS

K-means clustering is an unsupervised clustering algorithm that takes a bunch of unlabeled points and tries to group them into “k” number of clusters, where each point in the cluster is similar to each other.

Cluster is a collection of similar objects that are dissimilar to the others.

The “k” in k-means denotes the number of clusters you want to have in the end. If k = 5, you will have 5 clusters on the data set.

The cluster means are usually randomized at the start (often by choosing random observations from the data) and then updated as more records are observed.

At each iterations, a new observation is assigned to a cluster based on which cluster mean it is nearest and then the means are recalculated, or updated , with the new observation information included.

What are common use cases for k-means clustering?

Customer segmentation is probably the most common use case for k-means clustering.

It is also used for finding groups in types of complaints, types of consumer behaviors, summarization of data, finding anomalies, for example fraud detection, and the list goes on.

Anomalies can be considered small clusters with points very far away from any centroid.

How does k-means work?

Step 1: Determine K value by Elbow method and specify the number of clusters K

Step 2: Randomly assign each data point to a cluster

Step 3: Determine the cluster centroid coordinates

What is a centroid?

Simple, it is the center point of a cluster. For example, if we want to find 3 clusters, then we would have 3 centroids, or centers, one for each cluster.

Step 4: Determine the distances of each data point to the centroids and re-assign each point to the closest cluster centroid based upon minimum distance. The smaller the distance, more similarity. The bigger the distance, less similarity.

Step 5: Calculate cluster centroids again

Step 6: Repeat steps 4 and 5 until we reach global optima where no improvements are possible and no switching of data points from one cluster to other.

Common distance metrics:

-Euclidean distance: The distance can be defined as a straight line between two points (most common distance metric).

-Manhattan distance: The distance between two points is the sum of the (absolute) differences of their coordinates.

-Cosine distance

Example Code:

```
# Import kmeans and vq functions
from scipy.cluster.vq import kmeans, vq

# Compute cluster centers
centroids,_ = kmeans(df, 2)

# Assign cluster labels
df['cluster_labels'], _ = vq(df, centroids)

# Plot the points with seaborn
sns.scatterplot(x='x', y='y', hue='cluster_labels', data=df)
plt.show()
```

Dimensionality Reduction

What is dimensionality reduction?

Dimensionality Reduction is used to reduce the number of variables of a data set while ensuring that important information is still conveyed. Dimensionality Reduction can be done using feature extraction methods and Feature Selection methods. Feature Selection selects a subset of the original variables. Feature Extraction performs data transformation from a high-dimensional space to a low-dimensional space. Example: PCA algorithm is a Feature Extraction approach.

Why would we want to use dimensionality reduction techniques to transform our data before training?

Dimensionality reduction allows us to:

- Remove collinearity from the feature space
- Speed up training by reducing the number of features
- Reduce memory usage by reducing the number of features
- Identify underlying, latent features that impact multiple features in the original space

Why would we want to avoid dimensionality reduction techniques to transform our data before training?

Dimensionality reduction can:

- Add extra unnecessary computation

- Make the model difficult to interpret if the latent features are not easy to understand
- Add complexity to the model pipeline
- Reduce the predictive power of the model if too much signal is lost

Popular dimensionality reduction algorithms

1. Principal component analysis (PCA) - uses an eigen decomposition to transform the original feature data into linearly independent eigenvectors. The most important vectors (with highest eigenvalues) are then selected to represent the features in the transformed space.
2. Non-negative matrix factorization (NMF) - can be used to reduce dimensionality for certain problem types while preserving more information than PCA.
3. Embedding techniques - For example, finding local neighbors as done in Local Linear Embedding, can be used to reduce dimensionality
4. Clustering or centroid techniques - each value can be described as a member of a cluster, a linear combination of clusters, or a linear combination of cluster centroids

By far the most popular is PCA and similar eigen-decomposition based variations

Most dimensionality reduction techniques have inverse transformations, but signal is often lost when reducing dimensions, so the inverse transformation is usually only an apporximation of the original data.

PCA

Principal component analysis (PCA) is an unsupervised technique used to preprocess and reduce the dimensionality of high-dimensional datasets while preserving the original structure and relationships inherent to the original dataset so that machine learning models can still learn from them and be used to make accurate predictions.

How do we select the number of principal components needed for PCA?

Selecting the number of latent features to retain is tipically done by inspecting the eigenvalue of each eigenvector. As eigenvalues decrease, the impact of the latent feature on the target variable also decreases.

This means that principal components with small eigenvalues have a small impact on the model and can be removed.

There are various rules of thumb, but one general rule is to include the most significant principal components that account for at least 95% of the variation in the features.

Source:

https://becominghuman.ai/comprehending-k-means-and-knn-algorithms-c791be90883d

https://www.dataquest.io/blog/top-10-machine-learning-algorithms-for-beginners/#:~:text=The%20first%205%20algorithms%20that,are%20examples%20of%20supervised%20learning.