Web Mining for E-Commerce (CSE 621)

Project 3

Name: Ahmed Sharafeldeen & Youssef Sheta

Date: 3/22/2023

Vectorization

Vectorization is the process of transferring a textual data into vectors of real numbers that can be analyzed and understood by machine learning models (e.g., clustering or classification).

Bag of Words (BoW): The document is represented as a collection of words and their frequencies. This approach disregards the order of words and focuses only on their frequency, which allows for easy and efficient comparison between documents. It involves three operations: Tokenization, Vocabulary creation, and Vector creation

Term Frequency-Inverse Document Frequency (TF-IDF): TF-IDF is a weighting scheme used to evaluate the importance of a term in a document. The scheme takes into account the frequency of a word in a document (term frequency) and the rarity of the word across all documents (inverse document frequency). The goal of this approach is to give higher weights to words that are important in a particular document but rare in the entire corpus.

Word Embeddings: Word embeddings is a technique used to represent words in continuous vector space to capture the semantic meaning of the words. The embeddings are learned through neural networks, which are trained on large text corpora to predict the likelihood of neighboring words given a particular word. The resulting vectors capture the contextual information of words, such as their synonyms, antonyms, and semantic relationships.

Doc2Vec: Doc2Vec is a technique used to represent documents as continuous vectors space to capture the semantic meaning of the documents. The technique extends the idea of word embeddings to entire documents by learning a vector representation for each document in a corpus, in addition to the vector representations of the individual words. This is done through neural networks that predict the words in a document given the context of the entire document.

Latent Semantic Analysis (LSA): LSA is a method used to analyze and represent the meaning of documents based on their co-occurrence with other words in a corpus. LSA works by constructing a matrix of word occurrences in the documents, which is then transformed into a lower-dimensional matrix using singular value decomposition (SVD). The resulting matrix represents the documents in a latent space where semantically similar words and documents are closer together.

N-grams: N-grams refer to consecutive sequences of N words extracted from a text corpus. This method is used to represent text as a sequence of overlapping N-grams, which are then used to model the probability of the next word in the sequence. The approach allows for capturing the local context and co-occurrence of words within a document. This technique is capable of representing more complex connections between words. Examples of N-grams are unigram, bigram, trigram, etc.

Normalization

Normalization is a necessary preprocessing technique that is used to treat your data in a common scale to make sure all of the features are equally treated when performing clustering (i.e., contributed equally to machine learning models). After identifying features used in the project, a normalization approach is adopted.

Min-Max normalization: This method is used to scale the data points of a dataset to a fixed range between 0 and 1. The minimum value of the variable are subtracted from each value, then divide by the range (i.e., max-min).

Z-score normalization: This method rescales the data points to have a mean of 0 and variance 1. The mean is subtracted from each value and then divide it by the standard deviation.

Decimal scaling: This method scales the data by dividing each value by a power of 10. We choose this power so that the largest absolute value of the variable is less than 1.

Unit vector: Using this technique, every observation is scaled to a unit length. Each value is divided by the vector's Euclidean norm to achieve this.

L1/ Manhattan normalization: This approach rescales each data point by dividing it by the sum of its absolute values, so that the resulting vector has L1-norm equal to 1.

Batch Normalization: This technique is used in deep learning to normalize the inputs of each layer of a neural network to improve its performance and training speed. It involves normalizing the activations of each layer by subtracting the mean and dividing by the standard deviation of the batch of inputs.

Similarity or distance metrics

Similarity metrics are used to calculate the distance or similarity between data features that can be used depending on the type of data. After identifying the data which can be numerical, categorical, or text data, we have to choose a similarity metric that is compatible with the type of data, such as Euclidean distance for numerical data. Moreover, the generated matrix from applying this will be used by the clustering algorithms such as K-means, hierarchical clustering, and DBSCAN. In summary, these similarity metrics are important for any clustering project as it help clustering algorithms. Examples of similarity/distance metrics are summarized below:

Euclidean distance: It is the measurement of the straight line between two data points in a multidimensional space. It is commonly used as a distance metric in clustering. For example, the Euclidean distance of (x_1, y_1) and (x_2, y_2) is calculated as follows:

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

Manhattan distance: It is the measurement of the distance between two points as the sum of the absolute differences of their coordinates. It is used mostly in pattern recognition. For example, the Manhattan distance of (x_1, y_1) and (x_2, y_2) is calculated as follows:

$$d = |x_1 - x_2| + |y_1 - y_2|$$

Jaccard similarity: It is used in text mining and recommendation systems as it measures the similarity between two sets of binary data by calculating the intersection over the union of the sets.

$$J = \frac{M_{11}}{M_{01} + M_{10} + M_{11}}$$

Simple Matching Coeficient (SMC): SMC measures similarities between two sets of binary data as follows:

$$J = \frac{M_{11} + M_{00}}{M_{01} + M_{10} + M_{11} + M_{00}}$$

Cosine similarity: It measures the cosine angle between two vectors (i.e., d_1 and d_2). It is mostly used in information retrieval and recommendation systems. It can be computed as follows:

$$\cos(d_1, d_2) = \frac{d_1 \cdot d_2}{\|d_1\| \, \|d_2\|}$$

Mahalanobis distance: It is the measurement of the distance between two points in multidimensional space, but it takes into account the covariance matrix of the data. It is used in outlier detection and classification.

Clustering algorithms

Cluster algorithms aim to partition data points into groups based on similarities in their characteristics. The goal of clustering is to identify patterns and structures within the data which can then be employed to inform decision-making or make prediction. Examples of clustering algorithms are:

K-means: K-means is a partitional clustering algorithm that partitions data points into k distinct clusters based on their similarity to a centroid point. The key concept of this algorithm is to repeatedly update the position of center points and reassign data points to the closest centroid point until convergence is achieved.

DBSCAN: DBSCAN is a density based algorithm that groups data point together that are close to each other ibn density. Moreover, it also identifies noise points that don't belong to any cluster. This algorithm works by identifying three different point types: Core point, Border point, and noise point. Core point is a data point within a data set that has a minimum number of neighboring data points (MinPts)located within a specific radius or distance, known as Eps. Border point is a data point within a data set that has fewer than MinPts within Eps, but it is in the neighborhood of a core point. Noise point is any point that is not a core point or border point.

Hierarchical clustering: Hierarchical clustering builds a hierarchy of nested clusters by iteratively merging or splitting clusters based on a similarity or distance metric. Its Types are agglomerative clustering and divisive clustering. Agglomerative clustering builds the dendrogram from the bottom level (i.e., bottom-up approach). Divisive Clustering starts with all data points in one cluster (i.e., root), then split it into smaller clusters until each data point is in its own cluster (i.e., top-down approach).

BIRCH algorithm: BIRCH is a hierarchical clustering method that uses a tree-based data structure to efficiently cluster large datasets by first building a small, memory-efficient summary of the dataset, known as a "clustering feature" tree, which can be used to partition the data into smaller subclusters. Then, it recursively applies this process to the subclusters until a desired level of clustering is achieved, resulting in high-quality clusters.

Expectation–Maximization (EM) Clustering: This algorithm is a probabilistic clustering algorithm that aims to maximize the likelihood of a set of data points being generated by a mixture of probability distributions. First, this algorithm assign k cluster centers randomly. Then, it iteratively refines the clusters based on two steps: expectation step and maximization step. In expectation step, the algorithm calculates the probability of each data point belonging to each cluster. In maximization step, it re-estimates the parameters of the clusters based on the newly assigned data points.

Mean-Shift Clustering: This clustering algorithm is a non-parametric centroid-based clustering algorithm that aims to identify dense regions of data points in a dataset. The algorithm involves iteratively shifting a kernel function towards the direction of maximum increase in the density of data points until convergence is achieved. The bandwidth parameter of kernel function affects the shape and size of the resulting clusters. As the kernel moves towards the areas of high density, it naturally accumulates data points within its vicinity, forming a cluster. The resulting clusters are centered around the modes of the probability distribution, making it suitable for datasets with arbitrary shapes and sizes.

Spectral Clustering: This algorithm partitions a dataset by analyzing the eigenvectors of a similarity matrix to identify the optimal number of clusters. The algorithm transforms the dataset into a low-dimensional space using a spectral embedding technique, which preserves the pairwise distances between data points. The transformed data is then clustered using a traditional clustering algorithm, such as K-Means, to obtain the final clusters.

Evaluation metrics or methods

There are many methods that can be used to evaluate cluster algorithm and to determine the optimal number of clusters for a given dataset, as shown below:

Entropy: Entropy is a measure of the randomness or uncertainty in a system. The entropy of cluster is the sum of the entropy of each cluster, where the entropy of each cluster is computed based on the proportion of data points in that cluster. A high entropy value indicates that the cluster is poorly defined, with a mixture of different data points, while a low entropy value indicates that the cluster is well-defined, with a homogeneous group of data points. In other word, a good clustering algorithm should minimize the entropy of clusters to ensure that the resulting clusters are distinct and homogeneous.

Purity: Purity measures the extent that a cluster contains only one class of data by computing the sum of max proportion of data point in the cluster.

Rand index: measures the number of pairwise agreements between a clustering K and a set of class labels C, normalized so that the value lies between 0 and 1. Higher values indicate better clustering performance.

Silhouette Score: measures the similarity of data points within clusters and the dissimilarity between clusters, with values ranging from -1 to 1. Higher values indicate better clustering performance.

Dunn's Index: measures the quality of clustering results by comparing the distance between clusters with the size of the clusters themselves. The index calculates the ratio of the minimum distance between clusters to the maximum diameter of any cluster. A high Dunn Index value indicates that the distance between clusters is large compared to the size of the clusters, indicating a good clustering result. Dunn Index may not be suitable for datasets with a large number of clusters or clusters of varying densities, as it tends to favor algorithms that produce a small number of dense clusters. In such cases, other evaluation metrics such as Silhouette Score or Calinski-Harabasz Index should be used instead.

Davies-Bouldin Validity Index: assesses the quality of clustering results by measuring the similarity between clusters and the dissimilarity between clusters. The index calculates the average similarity between each cluster and its most similar cluster, and the results are compared to the dissimilarity between the two clusters. A low Davies-Bouldin Index indicates that the clusters are well-separated and distinct from each other, while a high value indicates that the clusters are not well-separated and may overlap with each other.

Elbow Method: involves plotting the within-cluster sum of squares (WCSS) against the number of clusters and identifying the elbow point, which represents the optimal number of clusters.

Applications of clustering

There are many applications that used cluster algorithms to perform a specific task, such as:
Image segmentation: Clustering can be employed to segment images into distinct regions (i.e., region of interest), based on various characteristics like pixel values, color, texture, or other features.
Medical diagnosis: Clustering can be utilized to form clusters of patients with similar symptoms, medical history, or genetic markers. This can be advantageous in medical diagnosis and treatment.
Anomaly detection: Clustering can be used to detect unusual data points (i.e., outliers) by recognizing the ones that do not belong to any existing cluster or belong to a distinct cluster.
Fraud detection: Clustering can help in identifying fraudulent activities by detecting groups of transactions that deviate from the normal behavior or pattern.
Recommender systems: Clustering can be employed to group users or items based on their preferences or behavior, with the aim of providing personalized recommendations.

Two stories

Clustering can be used to solve challenging problems in real life for its ability to form some patterns and group similar data points based on some metrics, for example, fraud detection and medical diagnosis. The following are two stories that the clustering algorithm can be used to detect the problem that they faced:
Story 1 (fraud detection): A person had recently noticed a sudden surge in fraudulent transactions on his credit card. He was worried that his card had been compromised and wanted to identify the source of the problem. he decided to use clustering to identify patterns in his transaction history. He collected data about all his transactions, including the amount, location, and time of purchase, and used the DBSCAN clustering algorithm to group the transactions based on their similarities. After analyzing the clusters, he noticed that a group of transactions had occurred at unusual times and locations, and involved larger-than-average amounts. Upon further investigation, he discovered that his card had been cloned, and the fraudulent transactions had been made by a group of individuals operating in a different state.

In summary, we can use cluster algorithms to identify any unusual behaviors and prevent any damage caused from these unusual behaviors.

Story 2 (Medical diagnosis): Suppose a physician at a busy hospital. This physician was facing a problem with misdiagnosis. Patients with similar symptoms were being diagnosed with different diseases, leading to confusion and delayed treatment. This physician decided to use clustering to group patients based on their symptoms and medical history. He collected data about patients' symptoms, previous medical conditions, and family history, and used the K-Means clustering algorithm to group them based on their similarities. After analyzing the clusters, he noticed that a group of patients had similar symptoms and previous medical conditions, but were being diagnosed with different diseases. He consulted with other physicians and discovered that these patients had a rare disease that was often misdiagnosed. In summary, using this clustering algorithm, the misdiagnosis rate has been decreased by identifying the problem and take the necessary steps to improve the accuracy of his diagnosis. Moreover, the hospital was able to provide better care to its patients.