Sohil Singh

5/6/20

COMP 162: Data Analytics

Final Project Report

Recommendation systems play an important role in the functioning of popular software such as Youtube, Amazon, and Netflix, providing useful information to people. A prevalent method of determining user preference, filtering options based on previously viewed content allows users to explore matters similar to ones previously viewed. Based on common features between products, the algorithm calculates the closeness through the cosine similarity formula which divides the dot product of two vectors by the product of their respective magnitudes, as depicted in Figure 1. The results range from -1 to 1, with 1 representing exact similarity and -1 complete dissimilarity, meaning a vector would have a cosine similarity of 1 with itself.

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}},$$

Figure 1: Cosine similarity formula [1]

I used a dataset of top songs every year in the decade 2010 to 2019 which contained nine features for each track: bpm, energy, danceability, loudness, liveness, valence, duration, acousticness, speechiness, and popularity [2]. Reading all the data into a pandas dataframe, the algorithm then prompts the user to choose their favorite song out of four random options. The program then computes the cosine similarity, drawn from a scikit-learn function, between the chosen track and all others in the dataset, recommending the song with the max value.

A content based model we covered in class, the Naive Bayes Classifier separates data based on probabilities. This algorithm classifies an object as a programmer defined feature using prior and conditional probabilities to calculate likelihood of an event based on its precursors, mainly used in text files. The cosine similarity method can also be used in text files with an adaptation described in later paragraphs.

Like classification, regression, and clustering, cosine similarity calculates a form of distance, the ones covered in class deal with straight-line distance while the latter relates to angular distance. Straight-line distance equations like Euclidean or Manhattan distance find the direct amount of separation between two points. On the other hand, angular distance refers to the separation of two points from a remote, fixed location. The value received from cosine similarity calculations differs only slightly from Pearson's correlation coefficient found between two variables. Pearson's correlation coefficient needs to be centered around the mean of the vectors, while cosine similarity looks at the dot product of unit vectors [3]. Regression algorithms, best suited for determining relationships between variables, predict the probability of the occurrence of a dependent variable based on an independent variable. Clustering uses the mean value of a centroid location to group data points closest to it. With a completely different goal, recommendation systems calculate a value to find similarities between vectors.

Not covered in this class, collaborative recommendation systems, also used by industry leading companies, offer a different form of recommender based on other user preferences. Instead of looking at the features like classification algorithms, collaborative finds popular content to suggest to the user. An example of this would be the trending section on Twitter. This algorithm requires matrix factorization to align users with their interests [4]. Since there will be a limited number of users running my program a select number of times, I decided not to go with this approach.

A classification based recommender, used by companies listed in above sections along with many others, provides a system for suggesting content with similar features to ones previously viewed by the user. An example of this would be Spotify which makes playlists based on previously listened songs. This algorithm can even be used to suggest similar articles, as the number of words can be counted and put into a vector, however a potential drawback arises since cosine similarity does not take into effect the weight of words. In a text document, participles of little significance such as “a” and “the” often dominante the word count. A classification system using cosine similarity would in essence capture the number of participles, assigning high similarity values to articles with similar amounts. However, this is definitely misleading, so a system term frequency and inverse document frequency is required to place weight on keywords. Term frequency refers to the number of times a word appears in a document, while inverse document frequency measures the appearance of the word throughout all documents [5]. After calculating these values for each of the words, then apply cosine similarity to the documents to find closely related papers.

Content based recommendation approaches change with user preferences, a strength which allows the algorithm to adapt over time. They can suggest similar items that other users have not viewed, so the user views a wide variety of related options, not just popular ones. This system does not share user preferences, keeping their privacy. However, users only receive specialized suggestions, which may lead to redundancy in recommendations known as content overspecialization. These systems also require an extensive amount of metadata that describes content features along with a complete user profile, which requires much effort to produce. This phenomenon is known as limited content analysis [6]. These weaknesses can be offset slightly by combining a collaborative system, to view related items that other users find interesting, an approach prevalent in the industry today.

References:

[1] “Cosine similarity,” Wikipedia, 25-Apr-2020. [Online]. Available: https://en.wikipedia.org/wiki/Cosine_similarity. [Accessed: 06-May-2020].

[2] L. Henrique, “Top Spotify songs from 2010-2019 - BY YEAR,” Kaggle, 26-Dec-2019. [Online]. Available: https://www.kaggle.com/leonardopena/top-spotify-songs-from-20102019-by-year. [Accessed: 06-May-2020].

[3] J. H. J. Herman, “Is there any relationship among cosine similarity, pearson correlation, and z-score?,” Cross Validated, 01-Jun-1966. [Online]. Available: https://stats.stackexchange.com/questions/235673/is-there-any-relationship-among-cosine-similarity-pearson-correlation-and-z-sc. [Accessed: 07-May-2020].

[4] S. Luo, “Intro to Recommender System: Collaborative Filtering,” Medium, 06-Feb-2019. [Online]. Available: https://towardsdatascience.com/intro-to-recommender-system-collaborative-filtering-64a238194a26. [Accessed: 07-May-2020].

[5] N. Sharma, “Recommender Systems with Python - Part I: Content-Based Filtering,” Medium, 05-Feb-2020. [Online]. Available: https://heartbeat.fritz.ai/recommender-systems-with-python-part-i-content-based-filtering-5df4940bd831. [Accessed: 07-May-2020].

[6] F. O. Isinkaye, Y. O. Folajimi, and B. A. Ojokoh, “Recommendation systems: Principles, methods and evaluation,” Egyptian Informatics Journal, 20-Aug-2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1110866515000341#s0050. [Accessed: 07-May-2020].