NeoMusic: Find similar artists

Suriya Palanikumar
Computer Software Engineering
San Jose State University
San Jose CA
suriya.palanikumar@sjsu.edu

Sandhya Santhanam
Computer Software Engineering
San Jose State University
San Jose CA
sandhya.santhanam@sjsu.edu

Anuj Sharma
Computer Software Engineering
San Jose State University
San Jose CA
anuj.sharma@sjsu.edu

***Abstract*—Most recommendation systems seen in the industry today use a combination of usage data and item metadata to recommend new and interesting content to its users. While this works well when usage data is available for the items, it falls short in the case of new items when no usage data is available yet. This inadvertently introduces a bias in the recommendation system as it almost never recommends new items to users. This term project explores solutions to the 'cold start' problem in recommender systems. We provide a potential solution by using only metadata to recommend top-k similar artists of an artist to users of Deezer, a music streaming application.**

***Index Terms*—recommender systems, cold-start problem, k-means, hausdorff distance, neo4j, shortest path**

I. Introduction

Recommender systems have played a huge role in the way users consume content online and what kind of content they see. A recommender system is a type of information filtering system that attempts to forecast how a user would rate or prefer an item. For example, on Netflix, which movie to watch, on e-commerce, which product to buy, on Kindle, which book to read, and so on. Over the years, this domain has seen incredible success in this endeavor with the help of neighborhood based collaborative filtering algorithms. The more the user uses a platform, the more successful the recommender system is in attempting to discover the users' true preferences.

Taking a music streaming platform for instance, the more the user interacts with the application, the more data the app gathers to make accurate predictions. Artists that have been on the platform longer will have more usage data than the newer ones. Over time, the more popular artists on the platform tend to attract more attention than the relatively newer ones. If the music recommender system weighs in more on the popular artists than the newer ones, a bias is inadvertently introduced. On the other hand, because the newer artists have little to no usage data the widely popular neighborhood-based algorithms do not match them to users. This is made more challenging because recommending content that is not well-liked by other users is a risky move. The user may not like the recommended content and may end up distrusting the system.

This is called the 'cold start' problem. The phrase is derived from automobiles. The engine has trouble starting when it is cold, but once it reaches its optimal operating temperature, it runs nicely. When it comes to recommendation engines, "cold start" simply indicates that the conditions are not yet ideal for the engine to deliver the greatest benefits. The objective of this project is to propose a solution to solve the cold start problem for a music streaming platform, Deezer. The solution attempts to discover artists who are similar to an artist using only the metadata available for the artists. Going forward all the approaches and techniques used will be discussed with the Deezer use case as an example.

A. Related Work

Deezer, a popular music streaming service provider recently published their research on tackling the cold start problem. They leveraged existing usage data for artists and generated node embeddings for these artists using Variational Graph Encoders. Their approach modeled the artists as nodes of a directed graph. The edges of this artist's graph are attributed with the similarities computed for each of the nodes with others. Using graph encoders, they generated node embeddings to capture latent information about the artists. They also included a 'gravity' component to indicate how popular an artist is. The more popular the artist is, the heavier they are and consequently they attract a lot of other artists. This follows the intuition that many of the listeners will have interacted with the largely popular artists on the platform. Then the solution to discover similar artists is to simply predict the attribute of the link or edge between the nodes. New artists were added as new nodes to the graph and predictions were made. While this approach uses both usage and metadata of the artists, the approach we have taken uses only the artists' metadata to discover similar artists.

II. System Design and Implementation

A. Algorithms Used

The following algorithms are used in the proposed solution to identify similar artists with no usage data available for the artists,

- Use an unsupervised K-Means Clustering technique to identify groups of artists based on the metadata
- Compute distances between artists in each cluster using a distance metric that is asymmetric. i.e. for the distance D between two items x and y, $D(x,y) \neq D(y,x)$
- Use this table of artists of each cluster with their respective distances in Neo4j and compute the Shortest Path between them to select the top-10 artists similar artists for a particular artist

Clustering algorithms have been used previously in the domain of recommender systems to identify groups of similar users. In this project, we are using K-Means to group artists who are similar to each other. This is the first step in our approach because we are only using artist vectors for our project. Once we get clusters of artists, it would suffice to analyze them individually. This not only helps improve the process but also largely reduces the computational overhead. However, we faced an issue with the traditional K-Means algorithm: we noticed that the clusters formed had neighbors in the range between 3 and 2500. Any cluster with less than 10 neighbors would make it impossible for our system to discover top 10 artists. We discovered the K-Means Constrained library which allowed us to set thresholds for minimum and maximum cluster size and used the same for clustering. The following parameters can be specified with the K-means Constrained algorithm along with the value for k:

- size_min - int, optional, default: None. It limits the label assignment so that each cluster is at least size min in size. No constraints will be enforced if None is selected
- size_max - int, optional, default: None. It limits the label assignment to a maximum size of size max for each cluster. No constraints will be enforced if None is selected

The next step in our project workflow is to compute the distances between members of individual clusters using the Hausdorff distance measure. The Hausdorff distance, often known as the Hausdorff metric, is the distance between two subsets of a metric space. It converts a metric space's collection of non-empty compact subsets into its own metric space. It is named after Felix Hausdorff and Dimitrie Pompeiu. Informally, two sets are near in the Hausdorff distance if each of their points is close to one of the other's points. The Hausdorff distance is the longest distance you can be compelled to travel by an opponent who selects a point in one of the two sets from which you must then go to the other set. In other words, it is the longest distance between a point in one set and the nearest point in the other set. This step allows us to take advantage of the fact that the distance between the artists need not be symmetric. The reason why this is important is that some artists can be well-known and popular on the platform and others need not be the same.

The top-k artists have to be recommended from the distance metric calculated using Hausdorff metric. At the end of distance computation, a large interconnected data has been created where each artist might be connected to every other artist. In order to recommend the top k- artists, some graph mining techniques have to be used to reduce the bottlenecks of interconnected data. Hence, data from distance metric has been converted to cypher queries and imported into Neo4j in-order to do graph mining. Ideally, the Shortest Path algorithm is used to compute the distance between two nodes. But in order to recommend the top-k artists Shortest Path algorithm cannot be used as we need to compute the shortest path of one node to all other connected nodes. In order to do so, the Single Source Shortest Path algorithm is used. The Single source shortest algorithm(SSSP) computes the distance between one node to all of its connected nodes.

B. System Design

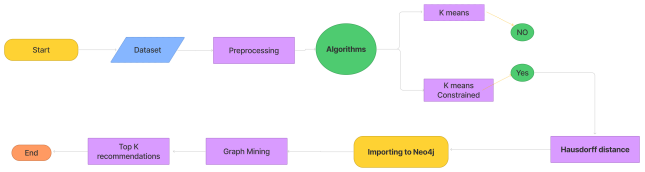

Fig. 1. System Design

In order to find the top-k similar artists, the features dataset of Deezer needs a metric to be computed.The features in the dataset have already been preprocessed and 56 different types of embedding vectors have been provided as Input.Since we do not have the target variable defined, we are using an unsupervised ML approach. The first trial with K-Means clustering failed as it may end up resulting in Cold-Start problem again. Hence, chose K-Means Constrained as a clustering algorithm. From the clusters formed in K-Means Constrained, the hausdorff distance is calculated inorder to find the non-symmetric method. The artists along with their distances have been imported into Neo4j as Cypher queries. The reason for choosing graph mining is that the distances between artists are highly interconnected. In order to determine the shortest distance between each artist, the Single Source Shortest Path algorithm has been used as a graph mining technique. The output of the graph mining algorithm is the top-k similar artists for a selected artist.

C. Visualizations

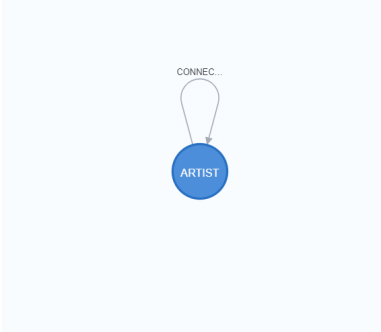

Fig. 2. Schema

In the graph mining methodology, the first step is importing the data to neo4j which creates a general skeleton schema with the logical flow of how data are interconnected. Hence, the nodes represent every artist with the artist name and every artist is connected to another artist with the distance as weights of the edges. The node name in neo4j is :ARTIST and edge name in neo4j Schema is :CONNECTED.

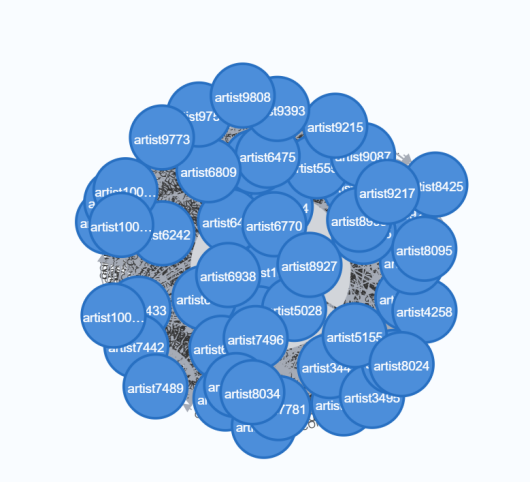

Fig. 3. Densely Connected Neighbors in a Cluster

This cluster visualization in Neo4j shows how densely connected a single cluster with about 200 neighbors are. This is the reason for including graph mining in our approach.

Fig. 4. Connections for a particular artist: 1617

With the Single Source Shortest path graph mining techniques, top-k similar artists are recommended. The image shows the recommended neighbors for Artist Id 1617.

III. Proof of Concept Evaluation

A. Dataset Description

The dataset for the project was taken from Similar Artists Ranking. Deezer, one of the popular music streaming applications made this dataset public along with their research. The csv file 'deezer_features.csv' was used for the project and it consists of 24,270 artists with 56 features. The first column identifies each artist with a unique ID. The remaining vectors represent the metadata of the artists covering three broad categories:

- Music Genre – 32 columns. These attributes indicate the genre of the music that the artist creates
- Artist Country – 20 columns. These columns represent the country the artist is from. The first 19 columns are one-hot encoded vectors of individual countries while the last column is a collective representation of the remaining countries of all artists on Deezer
- Music Mood – 4 columns. These are a representation of the emotion that a particular song evokes, like calm or energetic, or positive or negative

These features were normalized and provided to us. We used the dataset as is for the project.

B. Evaluation Methodology

Since we are following unsupervised learning, the following evaluation metric has been used to evaluate the model's output.

1) Evaluation Method 1 Davies Bouldin Score: Davies Bouldin Score(DBS) is generally used to compute the similarity between similar clusters. i.e., for one cluster, it computes the similarity with its most similar cluster. If the clusters are dispersed heavily then the DBS score would be less. The DBS score for clustering using K-Means constrained is 0.77. DBS score is closer to zero which states that clustering technique with selected k=122 value works well. One important note is that dataset separation is not so clear. For example: Genres in the dataset are not clearly separated. Ie., A 'rock' genre song might also contain some elements of other genres like 'metal'.

2) Evaluation 2 Clustering Visualization: Visualizing the clusters after clustering algorithm is an evaluation technique to determine if the clusters have been separated well. We can see the clusters separated with various colours. If there is a heavy overlap between the clusters, then there could be a possibility that the k selected may not be the optimal value.

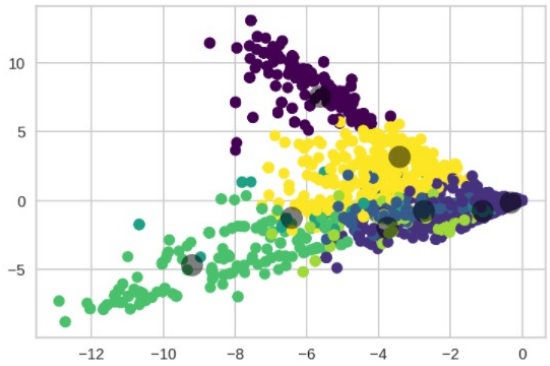

Fig. 5. Cluster Visualization with Centroids

From the image we can see that the clusters are well separated although the data points are dense towards the right edge.

In addition, we also followed an empirical method to compute the value of k to use for the K-Means Constrained algorithm. Empirical technique is a basic evaluation methodology to determine k in K-Means clustering method. It is computed by a formula of square root of N/2. We got a value of 110 and 122 was used in the algorithm as an approximation.

3) Analysis of Results: We used a dataset with artist features alone. With the metadata available we grouped them together into clusters and decided to compute the distances of inter-cluster artists using Hausdorff distance measure. These clusters with artists and their distances were then imported

into Neo4j for dealing with the dense clusters. Single Source Shortest Path algorithm was used to calculate the top 10 artists for a particular artist. We interpreted the results by evaluating the approach from the first step. We evaluated the cluster separation in particular as that plays an important role in separating the similar artists from dissimilar artists. The Davies Bouldin Score helped us observe an interesting situation: The artist vectors themselves are not clearly separated. 32 of the 56 features in the dataset are the embedding vector the music genre of an artist. This can be ambiguous because it is possible for the genres to overlap; an artist can create music that can be categorized as 'Rock' or 'Metal' as both these genres are similar in nature. This is also the reason for dense clusters.

IV. Discussions and Conclusions

A. Things that worked

- For discovering the top 10 similar artists, an unsupervised learning problem, we decided to use clustering. This worked well for the problem as we were able to identify 122 clusters with a maximum of 200 artists in a cluster
- Using an asymmetric distance metric namely Hausdorff distance
- Since we are using distance as a metric for recommending the top k artists, we choose a single source shortest path algorithm

B. Things that did not work

- Our initial idea of using collaborative filtering algorithms to recommend similar artists did not work because we only had artist metadata and not the usage data
- Even though we used the Elbow method and identified ideal cluster size, K means clustering did not work as the number of members were too less for some cluster centers. We then switched to K-Means constrained clustering to overcome this

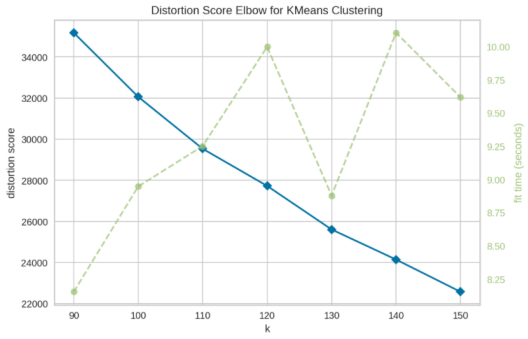

Fig. 6. Distortion Score Elbow for KMeans Clustering

C. Conclusion

Using the metadata of 24,270 artists on Deezer, we proposed a solution to find similar artists of a particular artist. This solution was proposed as a way to solve the cold-start problem for new artists. We adopted an Unsupervised Learning approach to this problem as we had information about the metadata and no usage data. All the features were anonymised and we could not tell one feature from another apart from the broad categories. With more information on the features and usage data, we can convert this into a Supervised Learning problem and apply more of the recommender system algorithms for the task.

V. Project Plan and Task Distribution

A. Who was assigned to what task

Task Name	Assigned Person
Deciding on the project flow	Anuj, Sandhya, Suriya
Clustering Techniques	Sandhya, Anuj
Distance Calculation	Anuj, Suriya
Graph Mining	Suriya, Sandhya
Evaluations	Sandhya, Suriya, Anuj
Report and Presentation	Anuj, Sandhya, Suriya

B. Who ended up doing what

- Sandhya
  - Analysed various methodologies to find the nearest neighbors as K-Means does not satisfy the use case criteria
  - Created the K-Means Constrained Algorithm code logic to determine the nearby neighbors
  - Implemented the clustering visualization evaluation technique
  - Written the "System Implementation" part in the project report
- Anuj
  - Analysed various distance metrics in order to satisfy the non-symmetric property as it has direct impact on cold-start problems
  - Implemented the Hausdorff distance metric for all the clusters generated from K-Means Constrained and computed the distance between each point with all other points in the cluster
  - Implemented the empirical method of the evaluation technique
  - Written the "Introduction" and "Discussions and Conclusions" part of the report
- Suriya
  - Analysed various graph mining algorithms to determine the best fitting algorithm for the use case as simple "Shortest Path" may not be the right graph mining technique for the use case
  - Created and imported the cypher queries into Neo4j. Implemented the Single Source Shortest Path algorithm to get the top-k recommended artists
  - Implemented the Davies Bouldin evaluation technique
  - Written the "Evaluation/ Proof of concept evaluation" part of the report

VI. References

- https://en.wikipedia.org/wiki/Cold_start_(recommender_systems)
- https://arxiv.org/pdf/2108.01053.pdf
- https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html
- https://joshlk.github.io/k-means-constrained/
- https://en.wikipedia.org/wiki/Hausdorff_distance
- https://scikit-learn.org/stable/modules/generated/sklearn.metrics.davies_bouldin_score.html
- https://neo4j.com/blog/graph-algorithms-neo4j-single-source-shortest-path/