Music Recommendation System Using Deep Learning

Tracey Kong, Kyra Abbu, Donald Chen

I. Description[1]

Our project is a music playlist recommender. It takes a playlist as an input and will create a set amount of playlists based on the genres of the given playlist. We will use two variations of two algorithms, Bag of Words and Sequence to Sequence Learning, to classify the input playlists. Approximate Nearest Neighbors and Spotify's ANNOY library will be used on the output to find similar playlists.

The Bag of Words (BoW) algorithm creates a feature vector where each feature is a unique genre of the songs from an input playlist. The value of each feature is based on its frequency in the input playlist, i.e., [rock: 5, indie-rock: 3, blues: 2, soft-rock: 7]. The vector can be used to find the closest playlists based on the genres. The two variations of BoW we will be using include the base version and a weighted version.

Our Sequence-to-Sequence (seq2seq) models use a Recurrent Neural Network (RNN) to read an element of the input sequence and use the result as an influence to the result of the next element in the input sequence. The two variations of the seq2seq algorithms we will be using are the base (unidirectional) Sequence to Sequence and BiDirectional Sequence to Sequence.

We are also exploring the possibility of using Top-N-Targets-Balanced Recommendation Based on Attentional Sequence-to-Sequence Learning. This is an algorithm with proven results of taking into account the changing interests and needs of the users as well as possible mistakes in choosing a song which can decrease the accuracy.

II. Contribution

Each person will contribute to generating figures for the performance measures and the data filtering process.

Tracey will implement and train the base and bidirectional seq2seq models to correctly label playlists as well as building a simple web app for the application.

Donald will work on the base and weighted BoW algorithms to label playlists as well as building a simple web app for the application.

Kyra's main contribution will be implementing a K-Nearest-Neighbors algorithm for returning playlists as well as using the Approximate Nearest Neighbors Algorithm using Spotify ANNOY library. She will also perform the evaluations using the formula stated on the Evaluation section, specifically using KKBOX dataset against our dataset to evaluate based on four measures: novelty, diversity, freshness, and popularity.

III. Related Topics

1. Classifying the input playlist using a **base Sequence 2 Sequence model**. (Tracey)
2. Classifying the input playlist using a **BiDirectional Sequence 2 Sequence model**. (Tracey)
3. Classifying the input playlist using a **base Bag of Words algorithm**. (Donald)

[1] Our project is based on Papreja 2019's study: Papreja, P., Venkateswara, H., & Panchanathan, S. (2019). Representation, Exploration and Recommendation of Playlists. *Machine Learning and Knowledge Discovery in Databases Communications in Computer and Information Science,* 543-550. doi:10.1007/978-3-030-43887-6_50

4. Classifying the input playlist using a **weighted Bag of Words algorithm.** (Donald)
5. Returning playlists that are most similar to the input playlist using **K-Nearest-Neighbors algorithm**. This is also used for the evaluation process. (Kyra)
6. Generating a **Precision-Recall Curve** and other performance measures to evaluate the performance of the models. (All)
7. **Data Filtering** - remove outlier playlists from the dataset. (All)

IV. Dataset

Spotify Dataset[2] contains a million user created playlists. The dataset was sampled randomly but is filtered for playlist quality and offensive content. This makes the data skewed.

Every Noise Dataset[3] contains 5000 playlists with specific genres for each playlist.

V. Timeline

Date	Goals
10/26 ~ 10/30	[x] Do research on seq2seq and BoW
11/2 ~ 11/6	[] Clean up the dataset by removing outlier playlists [] Correctly label playlists with the appropriate genre (for evaluation) [] Implement and train the seq2seq and BoW models over the dataset
11/9 ~ 11/13	[] Continue to implement and train the seq2seq and BoW models [] Build the recommendation system by populating a KD-tree
11/16 ~ 11/20	[] Implement the KNN algorithm to retrieve results from the recommendation system [] Build a simple web app to visualize the recommendation system
11/23 ~ 11/27	Thanksgiving week! Happy Thanksgiving!
11/30 ~ 12/4	[] Generate precision-recall curves and create other performance measures for the models [] Type up the documentation
12/7 ~ 12/11	[] Prepare presentation slides [] Prepare deliverables
12/15	Final submission and presentation of the project

VI. Deliverables

2 https://www.aicrowd.com/challenges/spotify-million-playlist-dataset-challenge
3 http://everynoise.com/

Report A report will be provided, consisting of the Summary of the Project, Project Progress, Target vs. Actual Accomplishments, Analysis, Contributions, Limitations, Conclusion

Jupyter Notebook/Python Files We will create a Jupyter Notebook, which will consist of the Python implementations. However, as needed, .py files will be provided.

GitHub Repository The GitHub Repository will contain .py, README, JSON files. This will also contain the web application.

Application Currently, we are exploring different websites to host our web application. However, one of the deliverables will be the web application. The source code will also be in the GitHub repository. We are exploring Heroku and GitHub to host our web application.

YouTube Video Demo (As Needed) We will create a YouTube Video Demo, which will be a tutorial on how to use our web application or how to recreate it using the source code on GitHub.

VII. Evaluation

[4]Recommendation systems are difficult to evaluate without user-labeled data. User-labeled data can be in the form of gathering the number of skips from the user among the songs in the playlist recommended. One evaluation method we will explore are offline evaluation methods, which will not contain user-labeled data from an actual user. Instead, we will use data that mimics user evaluations. If this is not a scalable solution, we will use the Approximate Nearest Neighbors Algorithm using Spotify ANNOY Library to analyze a tree structure that contains playlist embeddings. This library will randomly select a query playlist. It will be evaluated in terms of genre and length of the songs spanning nine genre labels and length of 30-250 with 20 bins.

Chou et al.[5] provides five measurements to evaluate a music recommendation system. The first measure is based on novelty. We will evaluate our model through using this formula in calculating the novelty. We will evaluate this in the artist level. Given the set of artists that the user has listened to and the set of artists the top-N songs recommend to the user, novelty will be calculated as follows:

$$novelty_u = \frac{|RA_u \setminus UA_u|}{|RA_u|}. \qquad (1)$$

We will also evaluate our model by calculating diversity, which is how diverse the top-N songs are in terms of the genre of the songs. We will use the Spotify dataset or the KKBOX dataset referred to in the paper and compute the genre histogram. The diversity is as follows:

---

[4] https://towardsdatascience.com/building-music-playlists-recommendation-system-564a3e63ef64

[5] Szu-Yu Chou, Y. Yang and Yu-Ching Lin, "Evaluating music recommendation in a real-world setting: On data splitting and evaluation metrics," *IEEE Xplore*, doi: 10.1109/ICME.2015.7177456.

$$diversity = -\frac{1}{U}\sum_{u=1}^{U}\sum_{g=1}^{G} h_u(g)\log(h_u(g))\,, \qquad (2)$$

We will also evaluate our model based on freshness, which is taking the average of the release date of the recommended songs by referring to the information from the Spotify dataset or the KKBOX dataset. The formula below will reflect if the top-N songs recommended are composed of new or old songs.

$$freshness = \frac{1}{U}\frac{1}{N}\sum_{u=1}^{U}\sum_{i=1}^{N} RD_{ui}\,, \qquad (3)$$

We will also evaluate based on popularity by taking the average of the top-N songs using the formula below and comparing it to the Spotify or KKBOX dataset.

$$popularity = \frac{1}{U}\frac{1}{N}\sum_{u=1}^{U}\sum_{i=1}^{N} PS_{ui}\,. \qquad (4)$$

These measures will be used as a means to evaluate our music recommendation system.