Music Recommendation using Collaborative Filtering and Deep Learning

Anand Neil Arnold, Vairamuthu S.

Abstract: ***The concept of filtering out songs based on the interest of a user is the core principle of today's music streaming (MS) service. Recommendation Systems (RS) are a key component of the MS companies. Different companies use different types of RS. Since the web is now an important medium for almost every kind of business and electronic transaction, it serves up as the driving force for the development of RS technology. There is significant dependency that exists between user and item-based activity which is the basic principle of recommendation. With the rise of digital content distribution, people now have access to music collections on an unprecedented scale. Commercial music libraries easily exceed 15 million songs, which vastly exceeds the listening capability of any single person. With millions of songs to choose from, people sometimes feel overwhelmed.Most common RS are designed using the concept of filtering techniques and deal with the count and similarities between the likenesses of the users. Our approach, in this paper, is to enhance the RS by combining the filtering technique with Deep Learning. It will use the traditional filtering technique and use the album art of the song to recommend new songs. The hybrid RS will scan the album art of the song for unique labels.***

Index Terms: Recommendation system, collaborative filtering, text analysis, object detection, deep learning

I. INTRODUCTION

Gone are the days, when people used to manually search for .mp3/.mp4/.wav music files and save them on their music playing electronic devices. The latest generation of music listeners currently rely on music suggestions provided by music streaming services. Therefore, the music streaming services need a capablerecommendation systems which can suggest music based on the listener’s interests. Most of the streaming services rely on CF algorithms to suggest music. CF is just a single parameter in determining the taste of the listener. To overcome this, top streaming services use a combination of algorithms to form what is called as a hybrid recommender system.

A. Collaborative Filtering

CF or collaborative filtering uses the existing history of the user and recommends music from other user's history which are similar. For recommending, music is classified according the user's history. Traditional classifiers such as the support vector machine and linear regression classify the music by extracting the mel-frequency cepstral coefficients (MFCC) from the audio signal of the music. As the structural complexity of the music is more, the efficiency of traditional classifiers reduces in classifying the music from different genres. CF uses the existing data to recommend the songs which reduces the complexity of the RS.

CF or collaborative filtering uses the existing history of the user and recommends music from other user's history which are similar. For recommending, music is classified according the user's history. Traditional classifiers such as the support vector machine and linear regression classify the music by extracting the mel-frequency cepstral coefficients (MFCC) from the audio signal of the music. As the structural complexity of the music is more, the efficiency of traditional classifiers reduces in classifying the music from different genres. CF uses the existing data to recommend the songs which reduces the complexity of the RS. There are mainly two CF techniques:

1. memory-based
2. model-based.

Memory-based CF techniques can further divided into two approaches:

a. user-item filtering
b. item-item filtering

The **user-item filtering** finds users that are similar based on rating similarities and recommends items to the similar users. Item-item filtering finds users based on similar items that other users liked.

Model-based CF predicts user's rating of unrated items using machine learning algorithms. Machine learning algorithms that are commonly used are SVD, KNN, matrix factorization based algorithm and Deep learning.

B. Yolo (You only look once)

YOLO, one of the fastest object detection algorithm is an extremely good algorithm for real-time object detection. YOLO[17] uses custom darknet-19 having 11 more supplemented network layers on top of the original 19 layer (YOLOv3). YOLOv2 uses an identity mapping and concatenating feature maps for capturing low level features from the previous layer. YOLO got 76.8% mAP on 67FPS & 40 FPS which puts it further that Faster R-CNN using ResNet and SSD[14]. It suggests to have unified network to perform all at once. The input image is divided into a grid (Z x Z) and each grid will predict its corresponding bounding boxes and confidence scores. Each bounding box consists of the coordinates to the center of the box, height and width relative to the whole image and confidence based on IoU (Intersection over Union) between the predicted and the bounding box. YOLOv2 removes all fully connected layers and uses anchor boxes to predict bounding boxes. One pooling layer is removed to increase the resolution of output. And 416×416 images are used for training the detection network now. And 13×13 feature map output is obtained, i.e. 32× downsampled. Without anchor boxes, the intermediate model got 69.5% mAP and recall of 81%. With anchor boxes, 69.2% mAP and recall of 88% are obtained. Though mAP is dropped a little, recall is increased by large margin.

Revised Manuscript Received on May 08, 2018.

Anand Neil Arnold, Pursuing M. Tech in Specialization in Big Data Analytics from VIT University, Vellore 2019 batch. Presently, an intern in Oracle, India

Vairamuthu S., Pursuing M. Tech in Specialization in Big Data Analytics from VIT University, Vellore 2019 batch. Presently, an intern in Oracle, India

II. LITERATURE SURVEY

A. A recommender System Based on Genetic Algorithm for Music Data

The recommender system by Hyun-Tae Kim, Eungyeong Kim, Jong-Hyun Lee and Chang Wook Ahn[1] is a hybrid approach of Collaborative Filtering (CF) and Genetic Algorithm (GA). The proposed system aims to effectively adapt and respond to immediate changes in users' preferences. The experiments conducted in an objective manner exhibit that our system is able to recommend items suitable with the subjective favorite of each individual user. The content-based filtering technique is applied to generate the initial population of GA. The recommender system is divided into three phases:

- Feature extraction
- Feature evaluation
- Interactive phase

Advantages:

The experimental results exhibited that the average scores, which are objectively collected by means of user evaluations, increases by degrees as the generation grows.

Limitations:

It's really hard for people to come up with a good heuristic which actually reflects what we want the algorithm to do. It might not find the most optimal solution to the defined problem in all cases.

B. Building Effective Recommender System Using Machine Learning Based Framework

The proposed paper shows the adaptation of collaborative filtering in Apache Mahout Platforms. Apache Mahout is an apache software foundation project to produce open source implementation of machine learning (ML) techniques like clustering, collaborative filtering and frequent item set mining. It is a scalable ML library.

Advantages:

The problem of scalability was solved by the mahout to a certain extent.

Limitations:

The methodologies are sometimes rejected because of the lack of features availability.

Content-Based Music Recommendation using underlying Music Preference Structure The proposed paper discusses about a content based music recommender system which is based on a set of attributes derived from psychological studies of music preference [7].

Advantages:

Content based music recommender systems can deal with unrated songs.

Disadvantages:

The content based recommender system compromises on quality. Genre and tags can be incorrectly identified.

C. A Music Recommendation System based on Usage History and Automatic Genre Classification

Jongseol Lee, Saim Shin, Dalwon Jang, Sei-Jin Jang and Kyoungro Yoon, in their proposed paper[3] have applied a distance metric learning algorithm in order to reduce the dimensionality of feature vector with a bit of performance degradation. They used 5 and 10-dim feature vector for genre classification without performance degradation. Most music recommendation systems use CF and CBF for finding the common patterns. One of the most common attributes that is used for this filtering is genre.

A paper by Kunal Shah, Akshaykumar Salunke, Saurabh Dongare, Kisandas Antala on different techniques of recommendation systems gave a brief idea on what all techniques can be used in making a good RS.

Another paper by Keita Nakamura and Takako Fujisawa recommended music using lyric network generated by using TF-IDF concept.

D. A Personalized Music Recommendation System Using Convolutional Neural Networks Approach

In the paper by Shun-Hao Chang, Ashu Abdul, Jenhui Chen and Hua-Yuan Liao on "A Personalized Music Recommendation System Using Convolutional Neural Networks Approach", they have used CNN to classify the music and generate a log file and used CF to provide recommendations. There paper suggested that using traditional classifiers such as SVN or KNN can reduce efficiency when applied on complex data.

E. Affective Music Recommendation System Reflecting the Mood of Input Image

"Affective Music Recommendation System Reflecting the Mood of Input Image" by Shoto Sasaki recommended music based on the present mood identified in the image. The RS recommended based on the genre the user listens to and determines its mood (happy, sad, lonely etc.) and predicts music based on it.

III. METHODOLOGY

Our first approach is to create a recommendation system using text analysis. First task is to extract the dataset suitable for prediction. For this, we have taken the FMA and million song dataset from columbia.edu and Github.

Our Recommender System is a hybrid approach between Collaborative filtering and Deep Learning.

The dataset was pre-processed using R and Python. The packages used for in R were readr, dplyr and caTools while pandas and numpy were used in python.

The first stage was to create the collaborative filter using Text Analysis of the pre-processed dataset. The dataset initially was two parts: songs_data.csv and its corresponding metadata.txt which was later merged.

The merged dataset had the following attributes shown in table 1:

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

Table 1

Attributes	Description
Song_id	The unique id assigned to each song.
User_id	The unique id assigned to each user.
Count	The count of each song listened by each user.
Title	The title of the song.
Release	The album with which the song was released.
Artist_name	The singer of the song
Year	The year the song was released.

The dataset was divided into training and test data in the ratio 75:25 ratio. For the start, out of 500000 tuples, we took 1000 random tuples and divided it in training and test data.

A custom recommender was applied which calculated the recommendation score and ranked them based on the score. Based on the recommendation score, the songs were ranked and the songs were recommended to the users.We first calculated the number of times the song was listened to and then arranged based on the count. Once the count is generated, we calculated the recommendation score and ranked them in ascending order.

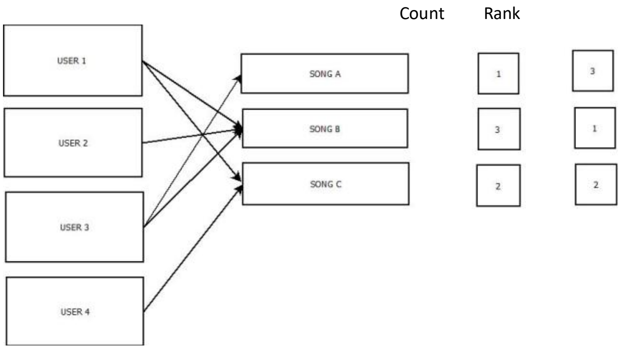


Fig1: System Architecture (Collaborative Filter)

A. Collaborative Filter

The collaborative filtering works based on the behavior of the user. The larger the users, the more efficient the filter becomes.

Algorithm1: (Percentage & Rank calculations)

1. **Collaborative_Filter():**
2. Pre-process the dataset.
3. Divide the dataset into training & Testing (75:25)
4. Count user_id for each unique song to get recommendation score
5. Sort(recommendation_score, ascending=0) <= Rank
6. Print song_title, Rank
7. End;

The next part which is the deep learning part involves recognition of album art using object detection algorithms. In our case, we used the state of the art object detection algorithm You Only Look Once (YOLO) v2. The model is first trained with the custom dataset extracted and compiled from MusicBrainz.

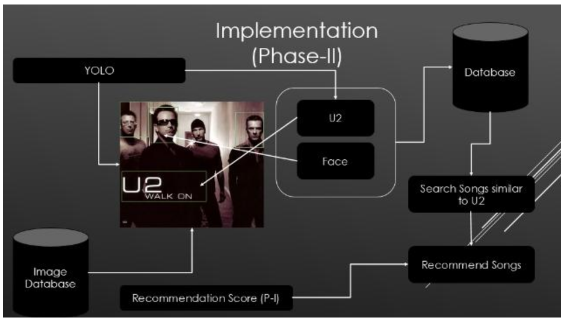


Fig 2: System Design

Using LabelImg tool, we specified the predicted bounding boxes stored in the form of annotations and generated the custom labels. With the custom labels we will train the model using darkflow and pre-trained configurations (cfg) and weights. We ran the model with 50 epochs due to low CPU hardware (i3, 8GB RAM).

B. Album Art Object Detector (YOLO Framework)

The next part of the hybrid recommender uses YOLO framework.

The basic functionality of YOLO is as follows:

1. YOLO first takes an input image and then divides the input image into grids.
2. Image classification and localization are then applied at each grids and the bounding boxes are then predicted.

If an image is divided into a grid of 3x3, then for each grid cell, the label y will have 8 dimensional vector as shown in fig 3[16]:

$$y = \begin{array}{|c|} \hline pc \\ \hline bx \\ \hline by \\ \hline bh \\ \hline bw \\ \hline c1 \\ \hline c2 \\ \hline c3 \\ \hline \end{array}$$

Fig3: Dimensional Vector for each grid

Here,

pc: Probability whether the object exists inside the grid or not.

bx,by,bh,bw: Bounding boxes of each present object (if exists)

c1,c2,c3: Classes that are identified w.r.t the object.

The bounding boxes are calculated in the following manner:

a. The bounding boxes **bx, by** represent the coordinates to the midpoint of the object with respect to this grid.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

b. **bh** is the ratio of the height of the bounding boxes to the height of the corresponding grid cell.
c. **bw** is the ratio of the width of the bounding boxes to the width of the grid cell.

To increase the performance, we have used two concepts: Non Max suppression & Intersection over Union (IoU)

Intersection over Union (IoU)

The **IoU** comes into picture when the accuracy of the predicted bounding box has to be identified. An arbitrary threshold is taken to measure the accuracy of the model. The IoU is calculated by the following equation:

IoU = Area of the intersection/Area of the Union

Algorithm 2: (Intersection over Union)

1. **IoU():**
2. Let arbitrary_threshold = 0.6
3. Iou = area_of_intersection/area_of_the_union
4. If (Iou < arbitrary_threshold)
5. Print "Not accurate enough" & discard
6. Else
7. Print "Above threshold: Accepted"
8. End;

Non Max Suppression

The Non-Max Suppression is all about finding the bounding boxes with maximum probability and the algorithm follows like so:

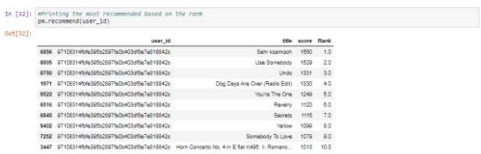
Algorithm 3: (Non Max Suppression)

1. **Non_Max_Suppression():**
2. Let Threshold = 0.6
3. If bounding_boxes < Threshold:
4. Discard them;
5. Else
6. Compare them with the remaining bounding boxes and choose the highest one.
7. End;

After the IoU and Non max suppression have been cleared, we predict the bounding box of the image.

IV. RESULTS

The first part was able to generate the songs using the CF as shown in the figure. On passing the album art to the model, it was able to detect face, band name "Pink Floyd". We tried with new albums as well and it was able to detect the faces of the band but accuracy was not good.



```
In [32]: #Printing the most recommended based on the rank
pm.recommend(user_id)
Out[32]:
```

	user_id	title	score	Rank
6856	[illegible]	Sehr kosmisch	1580	1.0
8806	[illegible]	Use Somebody	1529	2.0
8760	[illegible]	Undo	1331	3.0
1971	[illegible]	Dog Days Are Over (Radio Edit)	1330	4.0
9522	[illegible]	You're The One	1249	5.0
6516	[illegible]	Revelry	1120	6.0
6845	[illegible]	Secrets	1116	7.0
9402	[illegible]	Yellow	1099	8.0
7262	[illegible]	Somebody To Love	1079	9.0
3447	[illegible]	Horn Concerto No. 4 in E flat K495: II. Romanc...	1013	10.0

Fig 4: Collaborative Filter based songs that were recommended.

Fig 5: Object detection on album art (Pink Floyd)

V. CONCLUSION & FUTURE SCOPE

After our experiments, we can conclude by saying that our hybrid RS approach concept will work once the model is trained enough to recognize the labels. Its accuracy is still a major drawback but it can be increased extensively by training the model. We will keep working on training the model to get a better accuracy on the predictions.

VI. APPENDIX

Fig 1: System Architecture for Collaborative Filtering

Fig 2: System Design for the Album Art object Detector

Fig 3: The Dimensional vector generated for each grid

Fig 4: Output of CF recommending top 10 songs

Fig 5: Object detection on album art (Pink Floyd)

VII. ABBREVATIONS

RS: Recommender System

CF: Collaborative Filtering

DL: Deep Learning

IoU: Intersection over Union

CNN: Convolutional Neural Network

KNN: k-nearest Neighbor

REFERENCES

1. Kim, H. T., Kim, E., Lee, J. H., & Ahn, C. W. (2010, April). A recommender system based on genetic algorithm for music data. In 2010 2nd International Conference on Computer Engineering and Technology (Vol. 6, pp. V6-414). IEEE.
2. Chang, S. H., Abdul, A., Chen, J., & Liao, H. Y. (2018, April). A personalized music recommendation system using convolutional neural networks approach. In 2018 IEEE International Conference on Applied System Invention (ICASI)(pp. 47-49). IEEE.

3. Lee, J., Shin, S., Jang, D., Jang, S. J., & Yoon, K. (2015, January). Music recommendation system based on usage history and automatic genre classification. In Consumer Electronics (ICCE), 2015 IEEE International Conference on (pp. 134-135). IEEE.
4. Nakamura, K., Fujisawa, T., & Kyoudou, T. (2017, October). Music recommendation system using lyric network. In Consumer Electronics (GCCE), 2017 IEEE 6th Global Conference on (pp. 1-2). IEEE.
5. Shah, K., Salunke, A., Dongare, S., & Antala, K. (2017, March). Recommender systems: An overview of different approaches to recommendations. In Innovations in Information, Embedded and Communication Systems (ICIIECS), 2017 International Conference on (pp. 1-4). IEEE.
6. Sasaki, S., Hirai, T., Ohya, H., & Morishima, S. (2013, September). Affective music recommendation system reflecting the mood of input image. In Culture and Computing (Culture Computing), 2013 International Conference on (pp. 153-154). IEEE.
7. Soleymani, M., Aljanaki, A., Wiering, F., & Veltkamp, R. C. (2015, June). Content-based music recommendation using underlying music preference structure. In Multimedia and Expo (ICME), 2015 IEEE International Conference on (pp. 1-6). IEEE.
8. Fang, J., Grunberg, D., Luit, S., & Wang, Y. (2017, December). Development of a music recommendation system for motivating exercise. In Orange Technologies (ICOT), 2017 International Conference on (pp. 83-86). IEEE.
9. [9] Xiao, L., Zheng, Y., Tang, W., Yao, G., Ruan, L., & Wang, X. (2013, August). A GPU-accelerated large-scale music similarity retrieval method. In Green Computing and Communications (GreenCom), 2013 IEEE and Internet of Things (iThings/CPSCom), IEEE International Conference on and IEEE Cyber, Physical and Social Computing (pp. 1839-1843). IEEE.
10. Liu, H., Hu, J., & Rauterberg, M. (2009, November). Music playlist recommendation based on user heartbeat and music preference. In Computer Technology and Development, 2009. ICCTD'09. International Conference on (Vol. 1, pp. 545-549). IEEE.
11. Reddy, M. S., & Adilakshmi, T. (2014, January). Music recommendation system based on matrix factorization technique-SVD. In Computer Communication and Informatics (ICCCI), 2014 International Conference on (pp. 1-6). IEEE.
12. Makarand, V., & Sahasrabuddhe, H. V. (2014, January). Novel Approach for Music Search Using Music Contents and Human Perception. In Electronic Systems, Signal Processing and Computing Technologies (ICESC), 2014 International Conference on (pp. 1-6). IEEE.
13. Lin, K., Xu, Z., Liu, J., Wu, Q., & Chen, Y. (2016, August). Personalized music recommendation algorithm based on tag information. In Software Engineering and Service Science (ICSESS), 2016 7th IEEE International Conference on (pp. 229-232). IEEE.
14. https://towardsdatascience.com/review-yolov2-yolo9000-you-only-look-once-object-detection-7883d2b02a65
15. https://towardsdatascience.com/yolov1-you-only-look-once-object-detection-e1f3ffec8a89
16. https://www.analyticsvidhya.com/blog/2018/12/practical-guide-object-detection-yolo-framewor-python/
17. https://pjreddie.com/darknet/yolo/

AUTHORS PROFILE

Anand Neil Arnold, Pursuing M. Tech in Specialization in Big Data Analytics from VIT University, Vellore 2019 batch. Presently, an intern in Oracle, India and lives in Bangalore.