Recommendation Systems Coursework

Sandeep Thapa NCWN67

I. INTRODUCTION

A. Domain

Music recommendation systems are a class of web applications that assist users to tame the problem of information overload by providing personalized music recommendations[1].

Such a system could be built using 2D algorithms.The 2D recommendation algorithm aims to project all the users 'U' and the items 'I' in same feature space 'F', then it measures the similarities between the two entities U and I in F. Finally, item and user vectors in space F with highest similarities is used for recommendation. Examples of these algorithms are content-based filtering , collaborative filtering, and hybrid filtering.Content-based filtering, uses the descriptions/features of the song and a profile of the user's preferences to generate song recommendations similar to which a given user has liked in the past. The collaborative approach generates song recommendation based on past user-song relationships.The hybrid method, this is where content-based and collaborative-based song recommendation are separately made and then combined to make the final recommendation.

However,3D recommendation algorithms can also be used to model such systems. The context-aware recommendation system(CARS) is one example algorithm. CARS takes into consideration the contextual factors like mood,location and weather etc to recommend relevant songs to the user. Context is any additional information besides users, items and the ratings which may be relevant at the current time to make a recommendation. CARS aims to project users 'U' , items 'I' and context 'C' into feature space 'F'. Then it generate recommendations based on the different similarities measures of these 3 entities in the feature space F. 3D algorithms have shown to generate more relevant music recommendation. This is because it takes in account how users interact with the system within a particular "context".This is important because the preferences for songs within one context may be different from those in another context [2]. For Example, a user 'X' prefers a classical song 'I' when studying in a library. However, user X may also dislike the same item 'I' when outdoors.

B. Related work review

Gediminas Adomavicius et al [1], suggest 3D algorithms such as CARS can be factorized to one 2D recommendation algorithm with a filtering step. They introduce contextual pre-filtering and contextual post-filtering. Pre-filtering is when the original data is filtered to only contain data that is relevant to users' context before it is passed to a 2D recommendation algorithm to generate a recommendation. However, in contextual post filtering data is first passed to a 2D recommendation algorithm to generate recommendations without taking account of the user' context. The generated recommendations are then filtered to match the user's context. Lastly, passing the original data through a 3D recommendation algorithm that directly produce user rating for an item in a particular context is know as contextual modeling.

Contextual modeling in the music domain can be achieved via polynomial regression model figure 1. This is because they can be made to output a user rating for an song for given context by including interaction terms between the users,song and context. However, they have very high time complexity of $O(N^2)$ as they require us to learn a large number of weight w_{ij} variables.

$$\mathbf{(x)} = \mathrm{w}_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} x_i x_j w_{ij}$$

figure 1

Steffen Rendle [3] in 2010 introduced the Factorization machines(FMs) figure 2. The principle idea behind FMs states the following axioms; In a polynomial regression for each variable eg. user, song and context, there exist low-dimensional vectors v_u,v_s and v_c respectively. Secondly, the weight w_{ij} for interaction between 2 variables can be simply obtained by taking the dot product of their respective vectors. Therefore, we can replace the task of learning the weights by learning the vectors. As learning the vectors takes linear time, FMs can model polynomial regression models in $0(n)$ time. The FMs model has the ability to estimate all interactions between features even with extreme sparsity of data.

$$\mathbf{(x)} = \mathrm{w}_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \langle \mathbf{v}_i, \mathbf{v}_j \rangle x_i x_j$$

figure 2

C. Purpose/Aim

In this coursework I aim to implement and evaluate CARS using Factorization Machine. I will be using users' mood as the preferred context.

II. METHODS

A. Data type, source

The nowplayingRS[4] dataset was used in this coursework. The dataset comes with a rich set of song content and user contextual features. The raw form of contextual features are the hashtags that users uploaded in Twitter during the time of listening. The hashtags act as features that give insight of what the listening context and the emotional state of the user is at the time of listening to the track. The processed form

of the contextual features are the sentiment scores that were obtained by passing the hashtags through the Vader sentiment analysis dictionary. The sentiment scores lie in the range $[-1, 1]$, where 1 is the maximum score for positive emotion and vice versa.

The dataset does not contain ratings of the users for different songs under different context. It only contains an array of elements in tabular form, where each element is a listening event (LE) of a user under a unique sentiment score. The rating for a song i_1 by a user u_1 under certain sentiment score 0.2 using the equation 3.

$$\hat{r} = \begin{cases} 1 & if \frac{\sum LE_{i_1 u_1 0.2}}{\sum_{i=1}^{n} LE_{i u_1 0.2}}] \geq 0.5 \\ 0 & \text{otherwise} \end{cases} \quad ((3))$$

The original dataset contained over 9 million LE. Due to the limited RAM size and computational resources available the dataset size was reduced to 2.5 million with a preprocessing step. The preprocessing step included removing a LE if the respective song has been listened to less than 50 times, that are not in UK or US time zones and with no Vader sentiment score.

Additionally, I crawled google images to collect a total of 2,100 images of human facial expressions. There were 7 expressions from the set {Happy, Sad , Fear , Disgust, Neutral, Angry, Surprise}. Each set contained 300 images.

B. Feature extraction and selection methods

Zeno Gantner et al[5] demonstrated that FM is able to produce contextual rating prediction without the usage of song/item features. Therefore, I discarded the provided song/item features in the dataset.

To extract contextual features of the user at any given time, I used a webcam to capture the image of the user to capture. OpenCv built-in face localization library was used to localize the users face.The image was cropped to only contain the user's face. The cropped image was then passed to a deep convolutional neural network to perform classification tasks to predict an emotion expressed by the user. Lastly, Vader sentiment library was utilized to obtained sentiment score for the predicted user emotion. I used transfer learning to train a deep convolutional neural network(CNN) with Resnet-50 architecture to learn facial expressions from the facial expression dataset mentioned above.

C. User profiling and prediction methods

Factorization machine does not require us to project users into feature space that songs are represented by. Hence, the user preference matrix U $\in \mathrm{R}^{N \times F}$, where N and F are users and features of a song respectively, were not required to be modeled.

Factorization machines requires the input to be a row of listening events,where the user and songs needed to be one-hot encoded. Figure 4, illustrates the input. The row vector x^i represents listening event/observation for user in set A, B, C listening to a song in set TI,NH,SW,ST under a unique sentiment score. Last y^i represents an user rating associated with x^i

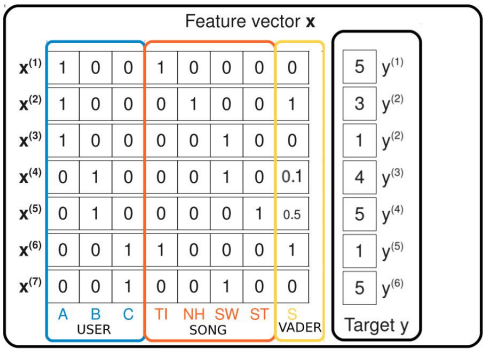

figure 4

D. Evaluation methods

Mean absolute error (MAE)[7] along with precision[6] and recall scores[6] where used to evaluate the performance of the factorization machines to recommend songs to user using mood as context.

III. Implementation

A. Recommendation algorithm

Factorization Machines [3] which performed binary classification task was used as the recommendation algorithm. The factorization machine was optimized using stochastic gradient descent.

B. Output (recommendations/predictions) presentation

The output of the factorization machines are array of probabilities for respective listening events. The probability P is the certainty of that the user will like the song under the given context. If $P \geq 0.5$, the predicted rating for the respective listening event $\hat{r}$ was taken to be 1 else 0.

A python flask server is used to serve a web-page which allows user to interact with the system. The webpage is dynamic and utilizes JSON to provide live video feed,song recommendation and the ability to change user. The live video feed contains bounding boxes around the detected face and displays estimated emotion label. Due to complication with Spotify's(Music service provider) API service, track id was not able to be mapped to song name. Hence, song features available in the dataset such as liveliness,danceability and tempo are presented to the user as a form of song description.

The predicted emotion of the user is converted to sentiment scores and stored. Upon clicking on the buttons to get new recommendation or to change user, a new set of recommendation is presented based on the mean sentiment score.

The system performance deteriorates under bad lighting condition and mild facial expression.

IV. Evaluation results

Upon evaluation on a test set of size $559,080$ listening events, the resulting MAE was 0.0024. There were $557,275$ and 464 true negatives and true positives respectively. While, the false negatives and false positives were $1,334$ and 7 respectively.

The precision of the FMs on the test set was 98%. Meanwhile, the recall of the FMs on the test set was 28%. This indicates that the FMs is only able to identify small fraction of listening events that the user would rate under the given context. However, the high precision indicates the following; if FMs predicts that the user would like the song under the given context then it is very likely that it is true.

For the recommender system, it means that which every song the FMs recommends next, there is a very high chance that the song is relevant to the user under the given context. Furthermore, due to the low recall score only a small fraction of songs gets recommended to the user under the given context. Hence, the variety of songs may be low.

However, as the dataset contains a relatively large number of songs, taking a small fraction yields an adequate number of songs to be recommended to the user every single time. Therefore, the effect of low recall scores are mitigated.

V. Conclusion

A. Limitations

The generated contextual features are not robust. The CNN performs emotion classification tasks with only 60% accuracy. The inadequate performance yields incorrect emotion for the user at any given time. This the generated sentiment scores may not represent users' mood accurately. Thus, this indirectly affects the accuracy of the recommendation using factorization machines.

Too few contextual features considered. FMs can be trained in linear time because it ignores self-interaction terms. For example, it can never capture terms like x_1^2 or $x_1^2 x_2$. Hence, information is lost. This is usually compensated by including many contextual feature terms. However, as this coursework was limited to 1 contextual feature, the generalization ability of FMs is compromised.

The dataset contained skewed vader sentiment scores. Majority of the sentiment score fall in the range 0.6 to 0.8 inclusive. Hence, the factorization machine haven't had opportunity to learn from the training example that fall outside the range. Thus, it's unlikely that optimal weights were found for interaction terms for features of listening events under those context. Therefore, the generalization ability of the factorization machine model decreases outside of the mentioned range above.

The learning rate used for factorization machines was based on experimentation performed by Zeno Gantner et al[4]. I am using FMs on a different dataset than Zeno. Thus, there may exist a different learning rate that is specific to nowplayingRS dataset that could have lead to better convergence when performing SGD.

B. Further developments

In future the algorithm developed by Zeno Gantner[5] which is based on alternating least squares would be a good replacement for SGD to optimize FMs. Their algorithm has lower time complexity than than SGD and requires no learning rate.

References

[1] Gediminas Adomavicius, Bamshad Mobasher, Francesco Ricci, and Alex Tuzhilin, Context-Aware Reccomendation, Aticle in AI Magazine, page 17-18, September 2011

[2] Gediminas Adomavicius, Bamshad Mobasher, Francesco Ricci, and Alex Tuzhilin. Association for the Advancement of Artificial Intelligence

[3] Steffen Rendle,Factorization Machines,Department of Reasoning for Intelligence The Institute of Scientific and Industrial Research,2010

[4] Asmita Poddar,Eva Zangerle,Yi-Hsuan Yang,nowplaying-RS: A New Benchmark Dataset for Building Context-Aware Music Recommender ystems,An open-access article.

[5] Zeno Gantner, Christoph Freudenthaler,Steffen Rendle,Fast Context-aware Recommendations with Factorization Machines,SIGIR'11, July 24–28, 2011, Beijing, China.

[6] Will Koehrsen,Beyond Accuracy: Precision and Recall,Medium March 3,2018

[7] Tumas Racjaitis,Evaluating Recommender Systems: Root Means Squared Error or Mean Absolute Error?,Medium,May 21 2019.