Tensor Decomposition for Sentiment-Aware Music Recommendations

***Abstract*—Tensor decomposition can be used for the task of recommendations of songs based on a person's current mood. Recommendations are based on the data set #nowplaying-RS dataset.**

I. Introduction

A. Domain

The area of context-aware recommender systems is a branch of recommendation systems concerned with providing more appropriate recommendations for a given context, such as time of day or mood.

B. Related Work

Matrix factorisation has been used to model the interactions of context with user-item ratings. CAMF-C (context aware matrix factorization), CAMF-CI and CAMF-CC [1] are three related methods which extend the standard matrix factorisation technique to include parameters modelling the interaction of contexts and item ratings. CAMF-C is the most general, modelling the global influence of each context on all ratings. CAMF-CI is the most specific, modelling each context-item interaction individually, while CAMF-CC is a middle ground, modelling context-category interactions, where each item is assigned to a group and the context has a uniform influence on the ratings of all items in each group.

Another way of extending matrix factorisation to account for contextual factors is to apply post-filtering of the recommendations from a 2D model to include ratings relevant to the given context. Cui et al. [2] applied contextual post-filtering on top of a two-level SVD (singular value decomposition) model. An initial recommendations list is generated by SVD, as well as a list of items which the user rated highly in the given context. Then, only the recommended items which are similar (in terms of features) to the items in the known list are kept.

Tensor factorisation is yet another technique for modelling context-user-item interactions. This can be thought of as matrix factorisation extended to 3-dimensions, or 3rd order tensors, where the third dimension is context. iTals [3] is an example of this technique which was applied to implicit feedback datasets.

C. Purpose

The purpose of this work is to apply tensor decomposition to the task of context-aware recommendation. Specifically, I will be implementing a recommender system to suggest songs to a user based on their current mood.

II. Methods

A. Data Type and Source

The dataset I will be using is the #nowplaying-RS dataset introduced in [4]. It consists of 361,347 listening events (LEs) from public tweets. Each LE contains a user ID, a track ID, and metadata such as hashtags, timestamp, timezone, language, etc.

To reduce the sparsity of the data and make it more suitable for recommendations, all users who listened to less than 10 tracks, and all tracks listened to by less than 10 users are removed. This is the same process as followed by authors of the original paper on the #nowplaying-RS dataset [4]. Once outliers are removed according to the process described in 2.2, this leaves 2700 ratings across 347 users, 527 songs and 5 contexts, giving a density of 0.3%.

B. Feature Extraction and Selection Methods

The authors used sentiment analysis on the hashtags to obtain a sentiment score. This value is in the range 0-1, where 0 indicates a negative tweet and 1 a positive tweet. This score can then be used to infer the mood of the user at the time they were listening to the song. The range [0,1] is divided into 5 equal intervals and the interval within which each sentiment score lies becomes the value of the context variable.

Since the feedback in the dataset is implicit, an explicit score needs to be generated for each user-item-context combination before an explicit recommendation model can be applied. To do this, the total number of LEs with the same user ID, track ID and mood were counted for the whole dataset. Then, any outliers were removed where the LE count was more than 2 standard deviations from the mean. Because the distribution of LE counts are heavily skewed (most users listen to a song only a few times), a log transform was applied to give a more uniform spread. Finally, the transformed values were brought into the range 1-5 by scaling all values uniformly.

C. User Profiling and Prediction Methods

Tensor rank decomposition is the process of expressing a high-order tensor as a linear combination of rank 1 tensors. Song ratings can be encoded as a 3-dimensional tensor, where the three dimensions are user ID, track ID, and mood.

The alternating least squares algorithm for canonical polyadic decomposition (CP-ALS) is applied to the tensor containing music ratings from the training set. The decomposed tensor is then re-composed into a new tensor containing the predictions of item ratings for all user-item-context combinations.

D.

E. Evaluation Method

The 2700 ratings were split randomly in the ratio 90/10 into train and test, giving 2430 train samples and 270 test samples. CP-ALS was run on the tensor containing only the train samples, and the MAE was measured between the 270 test samples and their corresponding predictions. To measure precision and recall, ratings >= 2.5 were taken to be positive and < 2.5 negative. The experiment was repeated 10 times with random train and test splits each time.

I compare my method against 2 baseline methods, AVG and SVD. In AVG, the predicted rating is just the average rating over all ratings for that item in the training set. SVD is the standard matrix factorisation using singular value decomposition. The context variable has been removed to make the data 2-dimensional, and the rating for a given user-item is the average across all contexts for that user and item.

III. Implementation

A. Recommendation Algorithm

The TensorTools Python package (https://github.com/ahwillia/tensortools) was used to perform the CP-ALS decomposition. The data were loaded from a CSV file and parsed into an array with NumPy. It was then pre-processed as described in 2.1 and 2.2. The resulting set of ratings was then shuffled and the first 90% taken as train samples. A 3D NumPy array was constructed from these training samples where the (i,j,k) entry is the rating of the jth song by the ith user in the kth context (according to the order they appear in the CSV). Then, CP-ALS is run on the tensor which is then reconstructed into the prediction tensor with the same dimensions. For each of the remaining 10% of samples making up the test set, the mean absolute error (MAE) between the ground truth and predicted rating is calculated. The precision, recall and f-measure are also calculated with the threshold of 2.5.

B. Output Presentation

The UI for the system is a simple text-based command-line interface where the user inputs their user ID according the #nowplaying-RS dataset and current mood, where 1 is very sad and 5 is very happy. The UI then presents to the user the Spotify IDs of 10 most recommended songs that they have not already listened to, in decreasing order of predicted rating. The user can then exit or enter another user ID and mood.

IV. Results

The results show that my method outperforms SVD on all metrics except precision. AVG outperforms both methods in terms of MAE but has zero precision and recall, making it practically useless for giving recommendations tailored to a specific user or context. This shows that my method is more suitable for providing recommendations tailored to a given user and context than the baseline approaches

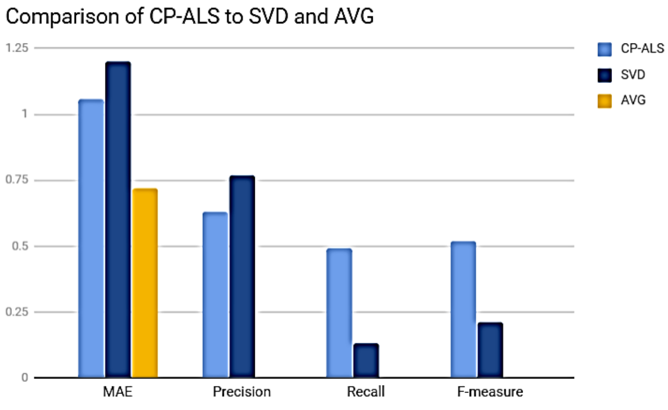

Fig.1 A bar chart to show the comparison of CP-ALS to SVD and AVG. AVG gave 0 for precision, recall and f-measure because all of the predicted ratings were below the 2.5 threshold.

V. Conclusion

A. Limitations

One of main limitations of this method is that it requires a dataset with explicit ratings. Preprocessing of the implicit feedback data allowed my method to perform well in terms of rating items relative to one another, but the heavily skewed distribution of the estimated ratings meant that the MAE was higher than simply taking the average rating across all users and contexts. Furthermore, my method, like all collaborative filtering approaches, suffers from the cold start problem; if a new track is added the model cannot accurately predict who will like it.

B. Further Developments

In order to alleviate the cold-start problem and offer more serendipitous recommendations, the content of tracks, such as tempo and valence, could be used to extend the model to a hybrid of content- and collaborative-based filtering. Furthermore, the model could be developed to work better with implicit feedback. This could be achieved by training the model to rank a list of songs rather than predict explicit ratings.

References

[1] L. Baltrunas, B. Ludwig, F. Ricci, Matrix factorization techniques for context aware recommendation, Proceedings of the fifth ACM conference on Recommender systems, 2011, 301-304.

[2] L. Cui, W. Huang, Q. Yan, F. Yu, Z. Wen, N. Lu, A novel context-aware recommendation algorithm with two-level SVD in social networks, Future Generation Computer Systems, vol. 86, 2018, 1459-1470.

[3] B. Hidasi and D. Tikk. Fast ALS-based tensor factorization for context-aware recommendation from implicit feedback, Machine Learning and Knowledge Discovery in Databases, pages 67–82. 2012.

[4] A. Poddar, E. Zangerle, Y. Yang, #nowplaying-RS: A New Benchmark Dataset for Building Context-Aware Music Recommender Systems, 15th Sound and Music Computing conference, 2018.