

Recommender Systems

ffgt86

I. INTRODUCTION

A. Domain

A recommender system (RS) seeks to predict the 'rating' that a user would give to an item. There are many flavours of RS: content-based, collaborative, knowledge-based, context-aware, *e.t.c.* This coursework is concerned with context-aware recommender systems (CARS).

A context is "any information useful to characterize the situation of an entity (e.g. a user or an item) that can affect the way users interact with systems" [1]. User preferences, and therefore ratings, may differ from one context to another. Context types include physical, social, and modal (i.e. pertaining to mood).

B. Purpose / aim

The purpose of this coursework is to develop a CARS. The focus is on the implementation and evaluation, rather than performance, of the system.

II. METHODS

A. Data type, source

The dataset used is MusicMicro¹ [3]. MusicMicro is relatively simple. It consists of users, artists, cities, countries, tracks, and events, which join them all together. A user u listens to a track i by artist a in a country c_o and a city c_i . Also included is temporal metadata: the day of the week, and month of the year, that the listening event took place. Importantly, songs are not assigned *ratings*, per se. The 'rating' is binary: whether a song was listened to or not. A prediction, therefore, lies in $[0, 1]$.

No user metadata is available. As the MusicMicro dataset is rather limited, the only available context is *physical*. The physical context comprises spatiotemporal data such as time, location, activity, and environmental conditions. In the MusicMicro dataset, this corresponds to the country and city in which the listening event took place and the day of the week, and month of the year that it occurred.

B. Feature extraction and selection methods

Feature selection was based on the author's finding in [3], in which the city the song was listened to, and the day of the week it was listened to, proved to be poor predictors. Instead, broader categories are used: whether or not the listening event occurred on a weekend, the country the event took place in, and the season during which it occurred.

C. User profiling and prediction methods

Only the top 100 most active users - those with the most recommendations - were used in order to constrain the time taken to calculate recommendations. This has the convenient side-effect of negating the 'cold start' problem. Prediction uses singular value decomposition (SVD), with the number of latent factors $k = 25$. SVD seeks to decompose a ratings matrix R into two singular value matrices, V and Q , such that $R \approx V \times Q^T \Sigma = \hat{R}$.

Users cannot be profiled save by their listening events.

D. Evaluation methods

The performance of the CARS will be evaluated on ratings prediction accuracy using mean absolute error (MAE)

$$MAE = \frac{1}{|\tau|} \sum_{(u,i) \in \tau} |\hat{r}_{ui} - r_{ui}|$$

and mean absolute error with context (MAE_B):

$$MAE_B = \frac{1}{|\tau|} \sum_{(u,i) \in \tau} |\hat{r}_{ui} - r_{ui} - \sum_{j=1}^k B_{jc_j}|$$

It will be evaluated on usage prediction accuracy using precision:

$$precision = \frac{TP}{TP+FP}$$

and recall:

$$recall = \frac{TP}{TP+FN}$$

where TP is the number of true positives, FP is the number of false positives, and TN is the number of true negatives.

III. IMPLEMENTATION

A. Recommendation algorithm

The recommendation algorithm used is Context-Aware Matrix Factorisation (CAMF). There are three varieties of CAMF:

- CAMF-C: a parameter for each contextual condition
- CAMF-CC: a parameter for each contextual condition and item category
- CAMF-CI: a parameter for each contextual condition and item

CAMF-CC was found to offer superior results, but there is no clear categorisations in the MusicMicro dataset. CAMF-CI was found to take a prohibitively long time to train, so CAMF-C was used instead.

B. Output (recommendations / predictions) presentation

Recommendations are displayed on a simple website originally developed for the Software, Systems and Applications III Web Technology submodule. This application uses Python, Flask, and Bootstrap in a much more user-friendly format than a simple CLI.

¹<http://www.cp.jku.at/datasets/musicmicro/index.html>

The physical context is easy to implicitly retrieve. In this instance, the user's IP address is used to calculate location, and temporal information can be similarly easily obtained. Drop-downs are provided to allow user selection of contextual conditions for testing.

IV. EVALUATION RESULTS

MAE was calculated as 0.223. MAE_B was calculated as 0.213, a reduction of over 4.66%.

There were 17280 true positives, 76 false positives, 1979612 true negatives, and 8332 false negatives, leading to an accuracy of 0.996 and a recall of 0.675.

V. CONCLUSION

A. Limitations

Many of the limitations in this coursework are due to the simplicity of the dataset. This was a deliberate choice. The code written is robust, extensible, and could easily be adapted to another dataset. This coursework is intended to demonstrate how CAMF-C, rather than the nuances of the data itself. That said, it is worth highlighting the following:

- SVD is only performed once. With a limited dataset, and a constrained number of users (100), it is possible to perform SVD live, after every user rating, to update recommendations appropriately. However, without the functionality for users to rate songs, doing so is pointless.
- Contextual information is limited. A consequence of the dataset. In [2], the music dataset is evaluated using 8 contextual factors and 27 contextual conditions. In this coursework, the inclusion of countries increased the number of contextual conditions to approximately 24 per factor, with only 3 contextual factors: `weekend`, `season`, and `country`. Even so, results show a marked decrease in MAE.
- Countries are not one-hot encoded. There is no inherent order in the numerical ID of a country, and so no underlying pattern for a weight to be optimised for. This means that the impact of the country contextual condition must be viewed as dubious at best. It is possible that encoding season cyclically may also have yielded better results.
- It was not possible to categorise tracks. This permitted only the use of CAMF-C, the most basic type. CAMF-CC (with weights per item *category* and contextual condition) is believed to be the most accurate.
- The number of latent dimensions k in SVD was chosen qualitatively. A superior implementation would use a training and test set to optimise k without overfitting. That said, $k = 25$ is a reasonable estimate.

B. Further developments

This coursework could be extended by comparing the three varieties of CAMF. The UI could also be expanded, with functionality allowing a user to rate tracks. The use of a more

complex dataset - such as `#nowplaying-RS`² - would allow the exploration of more complex contexts such as mood.

REFERENCES

- [1] Gregory D. Abowd et al. "Towards a Better Understanding of Context and Context-Awareness". In: *Handheld and Ubiquitous Computing* (1999), pp. 304–307. DOI: 10.1007/3-540-48157-5_29.
- [2] Linas Baltrunas, Bernd Ludwig, and Francesco Ricci. "Matrix factorization techniques for context aware recommendation". In: *Proceedings of the fifth ACM conference on Recommender systems - RecSys '11* (2011). DOI: 10.1145/2043932.2043988.
- [3] Markus Schedl. "Leveraging Microblogs for Spatiotemporal Music Information Retrieval". In: *Lecture Notes in Computer Science* (2013), pp. 796–799. DOI: 10.1007/978-3-642-36973-5_87.

²<https://zenodo.org/record/3248543#.XlELYG52vxB>