Collaborative Filtering for a Music Recommender System

Edward Bickerton
Department of Computer Science
University of Bristol
rw19842@bristol.ac.uk

***Abstract*—This report was completed as coursework for the module Advanced Financial Technology (COMSM0090). In this report I will give a detailed description of how to build a recommender system (RS) using matrix factorisation techniques and apply this to a digital music RS. A sensitivity analysis is then performed investigating how the number of user-item factors affects the performance of the RS. Finally, I will address the cold start problem, a common issue for collaborative filtering recommender systems.**

***Index Terms*—machine learning, recommender systems, matrix factorisation**

I. Introduction

Consumers have more options than ever before due to the widespread adoption of the internet. This is especially true for digital media such as music, for example the music streaming platform Amazon Music gives users access to 100 million songs [1]. Thus, an effective RS is essential to reduce users' effort in finding items of interest.

The strategy I will use for my music RS is known as collaborative filtering which uses past user-music ratings to predict future ratings (music with high predicted ratings can then be recommended to the user). Collaborative filtering has the benefit of being domain agnostic, meaning a RS developed for digital music can easily be adapted to work on, for example, movies or financial products, whereas content-based filtering such as the Music Genome project [2] requires external information about the items and users.

The dataset used contains digital music ratings from Amazon.com and can be found at [3] (`ratings_Digital_Music.csv` was originally used in [4]). Since the dataset is large, I will use just the first 50,000 rows which contains ratings from 1 to 5 for 1,449 songs from 34,947 unique users. Using explicit feedback such as user ratings results in a very sparse dataset; only 0.10% of user-items have ratings.

II. Matrix Factorisation

Matrix factorisation (a form of latent factor model) characterises both users and items on a set number of factors which are inferred from the ratings pattern [5]. Songs for example could have factors pop vs. classical or calm vs. energetic and for users each factor would measure how much the user enjoys songs that score high on the corresponding factor. Let f be the number of factors (or dimensions) and $q_i,\ p_u \in \mathbb{R}^f$ be vectors of factors for item i and user u respectively. Then the dot product, $q_i^T p_u$ approximates user u's rating of item i, denoted as r_{ui}.

The main challenge thus becomes mapping each user and item to appropriate factor vectors. This is done by minimising the loss function:

$$\sum_{(u,\ i) \in \mathcal{K}} \left(r_{ui} - q_i^T p_u\right)^2 + \lambda \left(||q_i||^2 + ||p_u||^2\right) \quad (1)$$

where $\mathcal{K}$ is the set of $(u,\ i)$ for which r_{ui} is known, the regularisation term $\lambda \left(||q_i||^2 + ||p_u||^2\right)$ is added to prevent overfitting and λ is a constant used to control the amount of regularisation.

The loss function is minimised using stochastic gradient descent (SGD). In short q_i and p_u are modified via:

$$q_i \leftarrow q_i + \alpha \cdot \left(\left(r_{ui} - q_i^T p_u\right) \cdot p_u - \lambda \cdot q_i\right)$$
$$p_u \leftarrow p_u + \alpha \cdot \left(\left(r_{ui} - q_i^T p_u\right) \cdot q_i - \lambda \cdot p_u\right)$$

where α is a learning rate constant. Rather than repeating this process for each user item pair, instead a user matrix $Q = (q_1,\ q_2, \dots,\ q_n) \in \mathbb{R}^{f \times n}$ and item matrix $P = (p_1,\ p_2, \dots,\ p_m) \in \mathbb{R}^{f \times m}$ is learned via SGD, n and m being the number of users and items respectively. We then predict r_{ui} by the element in the u^{th} row and i^{th} column of $Q^T P \in \mathbb{R}^{n \times m}$.

Fig.1 shows us that after many epochs of SGD we get diminishing returns on the performance of the model as the root mean squared error (RMSE) for the test data ceases to decrease significantly. Also, we observe that models with more features (larger f) require more epochs to reach the same levels of performance as those with fewer features.

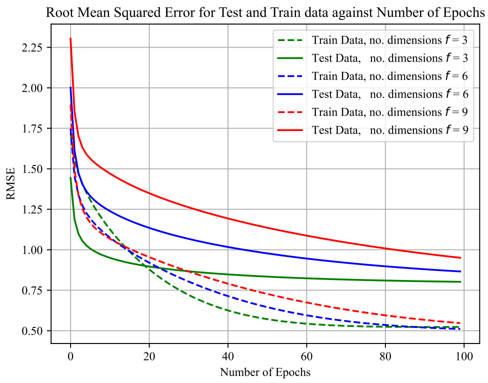

Fig. 1. RMSE for test and train data of three matrix factorisation models with differing numbers of dimensions (features) after each iteration (epoch) of SGD.

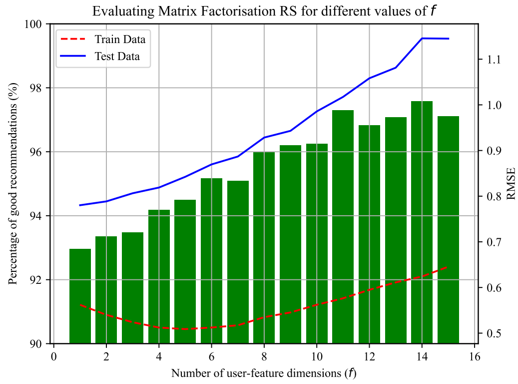

Fig. 2. The bar chart uses the left-hand axis where an item is recommended if it has a predicted rating of 3.5 or greater and the rating is considered good if the actual rating from the test set was 4 or 5. RMSE for test and train data can be seen using the right-hand axis.

III. Sensitivity Analysis

Although fig.2 suggests that the matrix factorisation model more accurately predicts user ratings when the number of features is small (minimising RMSE for the test data), the RS makes better recommendations when using models with more features (see fig.2 bar chart).

IV. Cold Start Problem

The cold start problem refers to the inability of collaborative filtering to make recommendations for both new items and new users; caused by there being little to no rating data. Preference elicitation is one way of dealing with new users and involves either explicitly querying the user upon registration (an efficient method for which was developed in [6]), however this requires effort from the user; alternatively, users' preferences can be determined implicitly by observing behaviours such as Twitter activity [7]. In addressing new items, a common approach is to use a hybrid recommender system which blends collaborative filtering with content-based filtering [8], requiring information such as item taxonomies.

References

[1] Amazon Music Unlimited. Amazon.com. Available at: https://www.amazon.co.uk/music/unlimited/ (Accessed: February 8, 2023).

[2] "Music Genome Project" US Trademark: No. 2731047 United States Patent Office

[3] Amazon digital music ratings dataset. Julian McAuley. Available at: http://jmcauley.ucsd.edu/data/amazon/ (Accessed: February 8, 2023).

[4] McAuley, J. et al. (2015) "Image-based recommendations on styles and substitutes," Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval.

[5] Koren, Y., Bell, R. and Volinsky, C. (2009) "Matrix factorization techniques for Recommender Systems," Computer, 42(8), pp. 30–37.

[6] Zhou, K., Yang, S.-H. and Zha, H. (2011) "Functional matrix factorizations for cold-start recommendation," Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval.

[7] Lin, J. et al. (2013) "Addressing cold-start in App Recommendation," Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval.

[8] Schein, A.I. et al. (2002) "Methods and metrics for cold-start recommendations," Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval.