**Setup**

This project aimed to exploit the user artist interactions and side information contained in the lastfm-2k dataset (Cantador, Brusilovsky and Kuflik, 2011) to implement a recommender system. The family of models explored for this task were the matrix factorisation methods of collaborative filtering that involve reconstructing a sparse user-item interaction matrix from two or more reduced dimension, dense, factor matrices. Cortes (2018) demonstrated the benefit of adding extra side information to the matrix factorisation reconstruction terms in the case of predictions where the item or user were not in the training set. This addresses the so-called cold start problem that effects model free collaborative filtering methods (Koren, Bell and Volinsky, 2009). However, it remains unclear if this form of side information encoding is useful to improve the accuracy of the interaction matrix reconstruction term when the predicted user and item are included in the training set. These are known as warm start recommendations (Wang and Wang, 2014).

The initial stage of this project proposed that the artist recommendations in lastfm-2k would benefit from the inclusion of the user social media data and artist tag data contained in the dataset. In this stage the hypothesis was formulated and tested that the matrix factorisation encoding both user-artist direct interactions as well as side information would produce a more accurate warm start recommendation than the matrix factorisation only containing the direct interactions. Given this formulation the null hypothesis was considered the outcome in which there was no difference or worse accuracy from the method incorporating side information that the one without.

In terms of defining accuracy, the primary component for measuring model effectiveness was identified as the magnitude of errors in the values of the user artist interaction matrix. Supplementary to this was the variety of artist that could be recommended. The recmetrics Python package (Longo, 2020) was used to implement these measures. MSE was the metric for reconstruction error and Coverage was used to quantity variety of recommendations. As MSE was the primary metric of concern, it is what was tracked during hyperparameter tuning. The coverage metric required the unfeasible generation of all out of sample recommendations for each tuning step so was left out of this procedure. Both measures were used when reporting on test data. Computational efficiency was measured in elapsed time per procedure and was tracked for all model fitting processes. All model fitting was performed using an 8 core, 16 threads CPU. This led to a prioritisation of highly parallelised model packages.

There were three model variants in the approach, the two in the stated hypothesis and an non-personalised baseline benchmark. For reliability, and under the assumption of asymptotic normality, the MSE between all model variants were tested using an ANOVA and post-hoc Tukey HSD. These tests were implemented using Python’s scipy (Pedregosa et. al 2011) and statsmodels (Seabold & Perktold, 2010) packages respectively. This allowed for a mechanism to reject the null hypothesis as well as assess if performance of models improved over the non-personalised baseline.

As proposed in Stage 1, pseudo ratings were derived from log transforming the user artist listen interactions. This allowed the matrix factorisation methods to take advantage of the log normally distributed data. It’s was on the scale of this transformation that MSE was reported.

**Approach**

The lastfm-2k dataset, as summarised in Stage 1, has three primary tables of interest for modelling. These are the user-artist listen interaction table, the user-friends relation table and the artist-tag table. The user-artist interaction table was prepared for model ingestion in the previous stage by way of log transformation. However, the two side information tables required further pre-processing to be included in the reconstruction term. A matrix representation of the user-friends graph structure was maintained as suggested in Huang et.al (2003). This was achieved by transforming the edge list into an adjacency matrix of dimensions (). Similarly, the artist-tag table was transformed using one-hot-encoding into a matrix of dimensions ()

The candidate models were implemented using the matrix factorisation package, cmfrec (Cortes, 2020). This package offered a suite of multi-threaded matrix factorisation methods. Of particularly interest was the Collective Matrix Factorisation formulation that facilitated reconstruction terms for user and artist side information in addition to the interaction matrix.

For the purpose of this project’s hypothesis three model variants were chosen from cmfrec:

1. collective matrix factorisation
2. traditional matrix factorisation
3. bias only baseline model

The mathematical formulas for each optimisation problem are available in Appendix 1.

The common characteristics between the reconstructions in (a) and (b) were the latent factor matrix terms, the bias vectors for the users and artists, as well as an penalty value for the learnt weights. In addition, (a) incorporates the reconstruction of the side information matrices, sharing part of the factor matrices between all reconstructions. This allowed for the latent factors to generalise to information beyond the user artist interaction matrix. These two methods were chosen to facilitate the measurement of whether the side matrices improve performance over just using the user artist interaction matrix. (c) was used as a baseline. It only contained the bias parameter vectors for artists, this produced a non-personalised rank of the most common artist for all users. If the models weren’t improving over this metric it would imply the information being learnt was not meaningful for recommendations.

To perform fitting, hyperparameter tuning and model evaluation the 92,834 rows of the user artist interaction matrix were split using the Python package Scikit-learn (Pedregosa et al. 2011). The training set was randomly subset 60% of the user artist interaction size and the validation and test set were each randomly subset 20%. The hypothesis focused specifically on warm start performance. To achieve this the validation and test set were filtered to contain only the users and artist that were in the training set. This reduced the size of the validation and test sets by approximately 15%.

Hyperparameter tuning was employed for (a) and (b) using grid search. For both variants penalty weights and the latent factor matrix parameter size were tuned. In addition, the search grid for (b) also evaluated over weights: 0.5 and 1, for the side information specific to this method. In total this resulted in 36 models to fit for model 1 and 140 model fits for model 2. The cmfrec package allows for individual factor matrix hyperparameter tuning when using the collective matrix factorisation method. However, to contain the search space of the grid the author elected to use shared values of these parameters per fitted model. All fitting procedures were performed using alternating least squares optimisation method due to the cmfrec package implementation being multi-threaded (Cortes, 2020). Although grid search was performed over a single threaded for-loop, the highly parallelised fitting procedure was hoped to balance keeping performance efficient.

An addition to the processing pipeline was tested to reduce the dimension of the user and item side matrices prior to matrix factorisation fitting. The two matrices were independently standardised using the training data, and then fitted and transformed using PCA. Proportion of variance explained was validated at 90%, 80% ,70% and 60%. Using default hyperparameter settings it was evident that transforming the sparse binary side matrix into dense, although dimension reduced, continuous matrices had a detrimental effect on accuracy and fitting time efficiency As a result, this pre-processing step wasn’t implemented in the full grid search pipeline and the side matrices remained in sparse binary encoding. As including the side matrices in collective matrix factorisation reconstruction term involves reducing their dimension within the model already this was felt to be justifiable.

Once hyperparameter tuning was complete the parameters with lowest validation set MSE for (a) and (b) were fitted again and evaluated on the unseen test data. Predictions were then generated and final MSE test scores were produced. Following this, recommendation lists were then generated for each test set user allowing the calculation of the supplementary metric of coverage. Finally, the squared test errors of each model were calculated to perform the two significance tests.

**Results**

|  |  |  |  |
| --- | --- | --- | --- |
| Model | Hyperparameters | Test MSE | Coverage |
| Collective Matrix Factorisation (a) | : 10, Latent Params: 60, Side Info Weights: 1 | 0.788 | 0.64 |
| Traditional Matrix Factorisation (b) | : 0.1, Latent Params: 80 | 0.795 | 9.19 |
| Bias Only Baseline (c) | N/A | 2.183 | 0.06 |

Table : Test Data Final Results

The final test data results for the three models are displayed in Table 1. Both Matrix Factorisation methods achieved test MSE less than the baseline. An analysis of variance performed on the MSEs showed there was significant variation among the models (F = 1283, p < 0.05). The post hoc multiple comparison Tukey HSD was then performed with family wise error rate set to 0.05. Appendix 2, Table 2 displays its result. The post hoc analysis showed that (c) had a significantly different MSE compared to both (a) and (b). It also showed that the MSE of (a) was not significantly different from that of (b). This provides evidence that this project’s null hypothesis cannot be rejected. The addition of side information did not improve accuracy in warm start interaction matrix reconstruction for the lastfm-2k dataset.

Although the Test MSE were not significantly different between (a) and (b), the coverage measure indicated that the reconstruction behaviour of the two models were markedly divergent. Variant (a) produced considerably less variety in artists recommendations, only producing 0.64% of the 17,632 potential artists in the lastfm-2k dataset. In comparison, model (b) produced 9.19% of the artists. This result appeared unexpected given the increased generalised information available for the reconstruction of the user artist interaction matrix in (a).

In understanding the difference in coverage between the (a) and (b) attention was drawn to the hyperparameters of each variant’s best performing model. The best performing penalty was observed to be considerably larger for (a). This was further explored by analysing the performance of the two variants over all penalty weights that were searched. The results are displayed in Figure 1. The left pane displays training MSE, validation MSE and fitting time over the  penalty grid for variant (a). The right pane displays the same metrics for (b). For variant (a) it was evident that regularisation played a large role in the out of sample performance. When the penalty was low the collective matrix factorisation method showed evidence of extreme overfitting. With a regularisation weight of 0.01 the training MSE was 0.00 and validation MSE was 9.42. As the penalty increased MSE dropped sharply. The grid optimum was found where training and validation MSE were nearly matched. This occurred with the penalty value of 10 seen in the final model. In contrast, variant (b) did not appear to overfit for small values of penalty weight. The minimum validation MSE occurred with a smaller penalty, and in general there was more out of sample stability over changes in regularisation amount.

Figure 1 also demonstrates that variation in penalty weight was not a determinant of changes in model fitting time. Figure 2 in appendix 2 showed this was not the case for all hyperparameters as time to fit appeared to increase linearly with latent factor parameter size. In general, between model variants a large differential in fitting time was observed. On average (a) took 9 seconds to fit compared to the 0.5 seconds it took to fit (b).

**Conclusions**

This project tested if the addition of side information would improve accuracy in warm start user artist interaction matrix reconstruction for the lastfm-2k dataset. The collective matrix factorisation method was compared to a traditional matrix factorisation method as well as a non-personalised baseline that was fit only on an artist bias vector. Although the test set MSE was lower for the collective matrix factorisation method it was not significantly different from the traditional model. Coverage of artists was notably lower for the collective matrix factorisation method and this was posited to be a result of the model requiring a large regularisation weight to get stable MSE performance. For warm start user artist interaction factorisation this author would advise against the use of collective matrix factorisation as it has been employed here. The model appeared overfitted when evaluated without large amounts of regularisation. It also did not respond favourably to pre-processed PCA of the side information matrices when evaluated on run time efficiency or accuracy. Although the MSE of collective matrix factorisation was competitive the reduction in coverage and increase in fitting time were not appealing properties.

It is possible that the model would respond favourably to more data to train on than what the 60/20/20 train, validation, test split allowed. Advances in the area of probabilistic matrix factorisation using Bayesian methods with priors could potentially assist if data size limitations are a contributing factor. An interesting possibility going forward is adding this form of probabilistic structure to encode the user and artist side information through the use of kernel methods (Zhou et al. 2012)

Figure 1: The Effect of L2 Penalty on Train/Validation and Fitting Time, Latent Factors setting = 60.  
Left Pane: Collective Matrix Factorisation (a), Right Pane: Traditional Matrix Factorisation (b)

**References**

Cortes, D. (2018). Cold-start recommendations in Collective Matrix Factorization. *arXiv preprint arXiv:1809.00366*.

Cortes, D. (2020). [online] *https://github.com/david-cortes/* . Available at: https://github.com/david-cortes/cmfrec [Accessed on 17 May, 2020]

Huang, L. W., Chen, G. S., Liu, Y. C., & Li, D. Y. (2013). Enhancing recommender systems by incorporating social information. *Journal of Zhejiang University SCIENCE C, 14(9), 711-721.*

Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, 42(8), 30-37.

Longo, C. (2020). [online] *https://github.com/statisticianinstilettos* . Available at: https://github.com/statisticianinstilettos/recmetrics [Accessed on 17 May, 2020]

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Vanderplas, J. (2011). Scikit-learn: Machine learning in Python. *the Journal of machine Learning research, 12, 2825-2830.*

Seabold, S., & Perktold, J. (2010, June). Statsmodels: Econometric and statistical modeling with python*. In Proceedings of the 9th Python in Science Conference (Vol. 57, p. 61). Scipy.*

*Simm, J., Arany, A., Zakeri, P., Haber, T., Wegner, J. K., Chupakhin, V., ... & Moreau, Y. (2017, September). Macau: Scalable Bayesian factorization with high-dimensional side information using MCMC. In 2017 IEEE 27th International Workshop on Machine Learning for Signal Processing (MLSP) (pp. 1-6). IEEE.*

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., ... & van der Walt, S. J. (2020). SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature methods, 17(3), 261-272.*

Wang, X., & Wang, Y. (2014). Improving content-based and hybrid music recommendation using deep learning*. In Proceedings of the 22nd ACM international conference on Multimedia (pp. 627-636)*

Zhou, T., Shan, H., Banerjee, A., & Sapiro, G. (2012, April). Kernelized probabilistic matrix factorization: Exploiting graphs and side information. *In Proceedings of the 2012 SIAM international Conference on Data mining (pp. 403-414). Society for Industrial and Applied Mathematics.*

**Appendix 1**

Model formulations:

1. Traditional Matrix Factorisation (Cortes, 2020):
2. The Collective Matrix Factorisation method (Cortes, 2020):

1. Baseline Bias Matrix

Where:

, and are column means for the interaction, user side and item side matrices respectively.  
,,,are the factor matrices to learn, the former two learnt across all least square terms.  
,are the item bias vector and user bias vectors respectively and are the penalty norms to regularise Where appropriate vectors are broadcasted over matrices to perform operations.

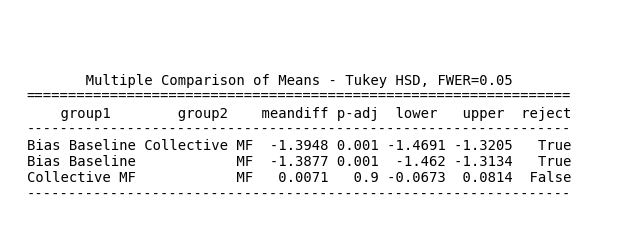
**Appendix 2**

Table 2 Tukey HSD Post Hoc Test

Figure 2: The Effect of Latent Factor Size on Train/Validation MSE and Fitting Time, Latent Factors setting = 60