**CONSIDER renaming model 1 model 2 etc**

**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 dense, reduced dimension, factor matrices. Cortes (2018) demonstrated the benefit of adding extra side information to the matrix factorisation reconstruction terms for predictions where the item or user were not in the training set. This addresses the so-called cold start problem that effects model free matrix factorisation (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 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 user social media data and artist tag data. In this stage the hypothesis was formulated that the matrix factorisation encoding both the user-artist direct interactions as well as the 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, two components were identified for measuring model effectiveness. The first was from the magnitude of errors in the values of the user artist interaction matrix and the second was from the categorical classification errors in the generated recommendation. The recmetrics python package (Longo, 2020) was used to implement accuracy measures in each of these cases. RMSE was the metric for reconstruction error and Recommender Precision and Recall for the predicted recommendations. However, it was computationally expensive, measured in processing time, to generate recommendations for every iteration of hyperparameter tuning. As a result, only the primary metric of RMSE was the tracked during hyperparameter tuning. Both metrics were used when reporting on test data. As referenced earlier computational efficiency was measured using processing time per procedure and was tracked for all model fitting processes.

There were three model variants in the approach, the two in the stated hypothesis and a nonspecific baseline benchmark. For reliability, and under the assumption of asymptotic normality, the MSE between all model variants were tested using an F test and post-hoc Tukey HSD. This allowed for a mechanism to reject the null hypothesis as well as assess if performance of models improved over a nonspecific 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 RMSE 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 matrix factorisation methods, of particularly interest was the Collective Matrix Factorisation formulation that facilitated the ingestion of user and artist side information into the reconstruction term of the user artist interaction matrix.

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

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

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

The common characteristics between the reconstructions in model 1 and model 2 were the latent factor matrix terms, bias vectors for the users and artists, as well as the penalty for the factor matrices and bias terms. In addition, the model 2 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. Model 3 was used as baseline as it only contains bias parameters vectors for the artist, this essentially produces 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 both randomly subset 20% each. As the hypothesis focused specifically on warm starts the validation and test set were then filtered to contain users and artist that were in the training set, reducing their size by approximately 15%.

Hyperparameter tuning was employed for models 1 and 2 using grid search over a list of penalty weights and the latent factor matrix parameter size. For model 2 the search grid was also evaluated over a binary set of weights values, either 0.5 or 1, given to 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 hyperparameter tuning over each individual factor matrix size and penalty using the collective matrix factorisation method. However, for this project shared hyper parameter weights were used per model to prevent the rapidly increasing the grid size. All fitting procedures were performed using alternating least squares optimisation method as is noted to be the fastest optimiser in the cmfrec package (Cortes, 2020).

An additional pipeline was tested to reduce the dimension of the user and item side matrices prior matrix factorisation fitting. The two matrices were independently standardised using the training data, and then fitted and transformed using PCA and evaluated on validation data. 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, continues matrices had a significantly detrimental effect on fitting time. For example, there was 50 seconds fitting time for 60% variance explained compared to 9 seconds fit for matrices prior to dimension reduction. Similarly, performance appeared to be detrimentally affected. 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.

Once hyperparameter tuning was complete the results with lowest validation set RMSE per model were fitted again and evaluated on the unseen test data. Predictions were then generated and final RMSE test score were produced. Following this recommendation lists were then generated for each test set user allowing the calculation of precision and recall on the artist categorical recommendations. This also facilitated the generation of the auxiliary metric of coverage to get an understanding of what proportion of the total artists set each model is recommending. Finally, the squared test errors of each model were calculated to perform significance testing.

**Results**

The grid search results for model 2 indicated a clear ranking in the RMSE benefit of the tested hyperparameters. Of the 140 fitted models, the top 28 were all the combinations with the penalty set to 10. Next up was the weights associated the side information matrices. The top 7 fitted models all contained equal weighting between the user artist matrix and the side information. Unsurprisingly due to the very high regularisation in the top scoring model, parameter size of the latent matrix did not seem relevant. With the near highest and lowest parameters 60, 70 and 20,30 respectively sitting very near each other at the top.

In contrast the top 3 RMSE results for model 1 appeared to be with the larger parameter latent factor matrices, i.e. 80,70,60 respectively. penalty was lower for these fitted models that the top parameters in model 2 at the value of 0.1.

The auxiliary measure of coverage showed a striking difference between the two matrix factorisation models despite their very similar RMSE. Model 1 recommended 9.19% of the 17632 artists in the dataset, Model 2 recommended 0.64% of the artists, while the baseline model 3 recommended 0.06%. It was suspected that the difference in coverage is likely driven by the large regularisation in the lowest RMSE model 2 model

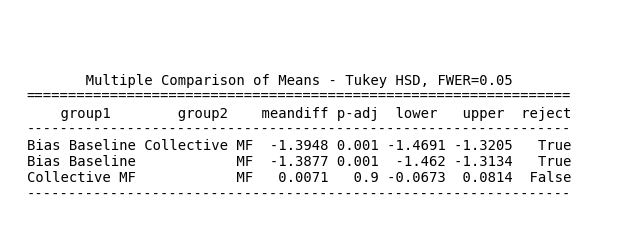
****

Figure 1Tukey HSD Post Hoc Test

An analysis of variance was performed on these results. There was significant variation among the model MSEs (F = 1283, p < 0.05). A post hoc multiple comparison Tukey HSD was performed. Figure 1 displays its results. The post hoc analysis showed that the baseline bias model was significantly different from both the collective matrix factorisation method as well as the traditional matrix factorisation method. The result also shows that although the collective matrix factorisation method reported a lower MSE, it was not significantly different at a family wise error rate of 0.05. As a result, the null hypothesis that there is no difference between models that incorporate side information and those that don’t for reconstructing the user artist interaction matrix for the last lastfm-2k cannot be reject.

**Conclusions**

**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.*

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).*

**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 and vectors are broadcasted over matrices to perform subtraction operations.