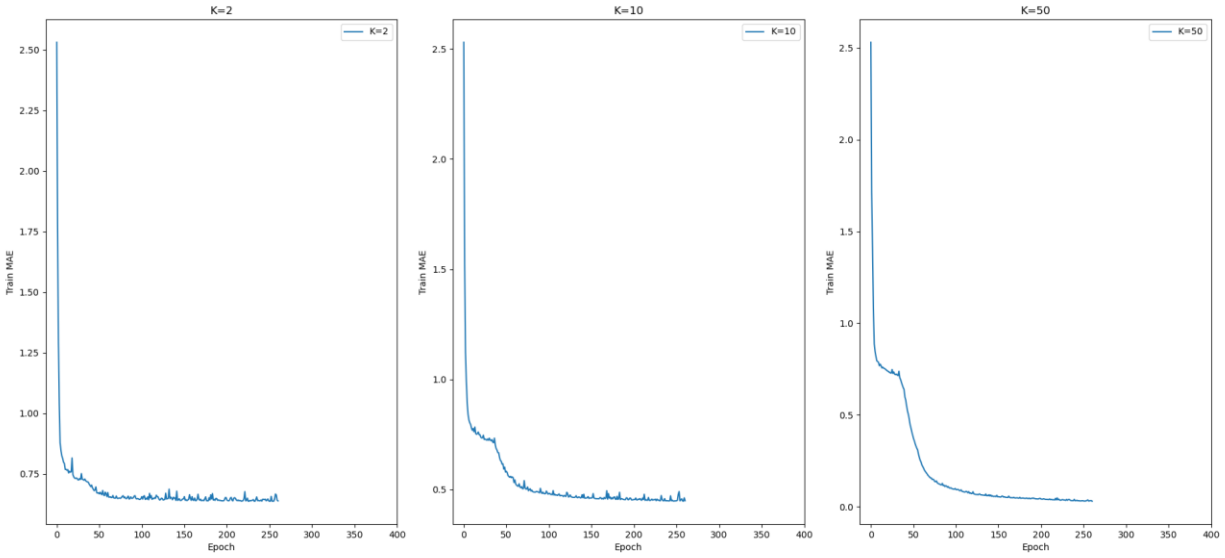
Project B: Recommendation Systems

Alex Lemieux, Matt Carey

05/09/2024

Part 1:

1a: In your caption, reflect on what your trace plots suggest. Do you see underfitting? Overfitting? What happens as K increases?




```
At End: Model for K=2, Train AUC: 0.8332281095487224, Valid AUC: 0.7686093021815764, Train MAE: 0.6384839565737304, Valid MAE: 0.7584105896538 65
At End: Model for K=10, Train AUC: 0.9173055922784397, Valid AUC: 0.71205998782197 67, Train MAE: 0.44828102516386303, Valid MAE: 0.9594721990305406
At End: Model for K=50, Train AUC: 0.9999998875263515, Valid AUC: 0.6524255938997835, Train MAE: 0.030203829368168733, Valid MAE: 1.2146757468931986
```

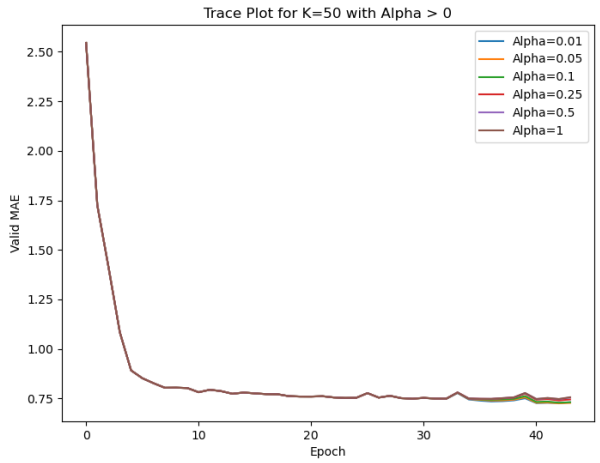
Latent Factors	Train AUC	Valid AUC	Train MAE	Valid AUC
K = 2	0.8332	0.7686	0.6385	0.7584
K = 10	0.9173	0.7121	0.4483	0.9595
K = 50	0.99999	0.6524	0.0302	1.215

Figure 1: Trace plot for K = 2, 10, and 50 from left to right respectively. K describes the number of factors used in model training. Each plot traces the Mean Average Error on the training set as epochs increase. Based on the plots, we see the Mean Average Error dropping continuously for all plots. This makes sense as this is the training error and the model will try its hardest to lower the training error. For some reason the plots could not hold all 1000 epochs of data. As such I copied the scores for the MAE and AUC at the last epoch of each model.

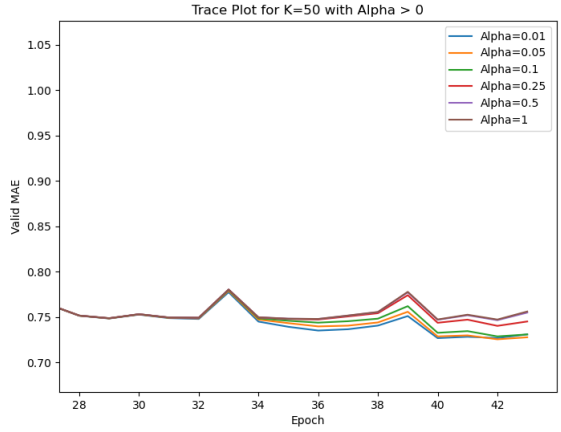
It seems that at high epoch ranges, the K = 10 and K = 50 models overfit the data. This is visible by the discrepancy in performance on training and validation sets. The K=50 demonstrates overfitting the strongest, with the training AUC approaching a perfect score of 1 and the MAE dropping to only 0. This, however, produces the worst performance on the validation set for the AUC and MAE metrics. This is indicative of overfitting and suggests we should train models with few latent factors like K = 2 which produces the best performance on validation sets90090.

1b: Figure and caption: Make trace plot showing mean absolute error vs. epoch completed when α>0. Here, please plot one figure, for K=50. In your caption, please specify (1) which value of α you ultimately selected (you should try several) as well as the batchsize and step size, and (2) whether you ultimately get better heldout error with α>0 than you did in 3a.

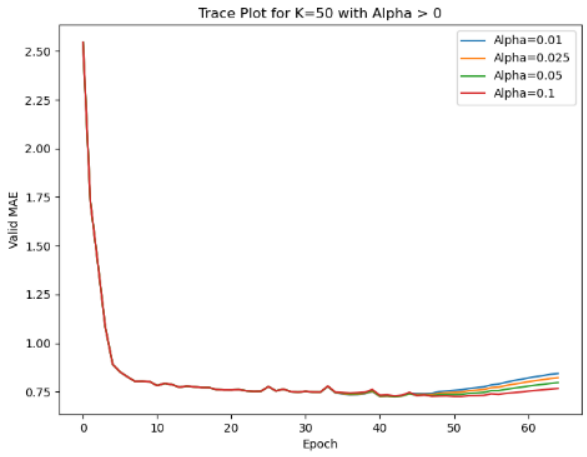
Testing Initial Alpha Range [0.01, 0.05, 0.1, 0.25, 0.5, 1]



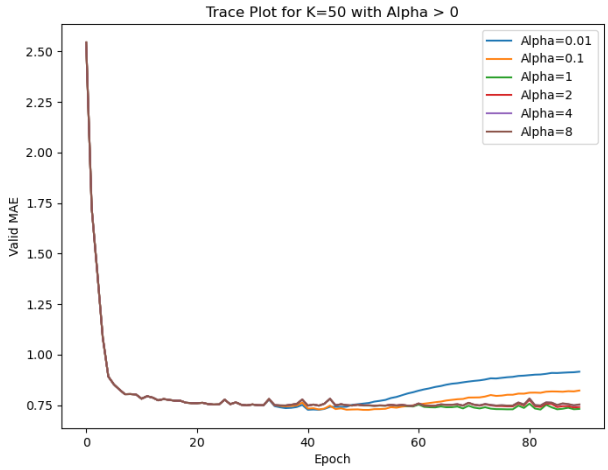
A. *MAE score on validation set for a range of alphas:* [0.01, 0.05, 0.1, 0.25, 0.5, 1].



B. Zoom on plot from A



C. *MAE score on validation set for a more focused range of alphas:* [0.01, 0.025, 0.05, 0.1].



D. *MAE score on validation set for a more general range of alphas:* [0.01, 0.1, 1, 2, 4, 8], but trained at much larger epochs.

Figure 1B:

The above plots were generated using a batch size of 32 and a step size of 0.2. For plots 1C.A-1C.B, n_epochs was set to 50. For plot 1B.C-D, the n_epochs was increased to 100. For plot 1C.D, the n_epochs was increased to 100.The best performing alpha appears to be 0.1.

Although Plot 1B.D shows large alpha values reaching a lower MAE, upon inspection we see that the gradient with respect to U and V go to 0 during training for large alphas. This makes sense as the terms have a higher penalty placed on them but is not ultimately what we desire of the model. It is worth noting that this discrepancy occurs at the higher epochs.

Looking at plots 1B.A-B, we see alpha = 0.1 to boast strong performance, barely losing out to the alpha values of 0.05 and 0.01. In 1B.C, however the model outperforms the others. The other models begin to worsen in performance as can be seen by their increasing MAE. This is common as we use the validation set as overfitting will begin. The minimal reduction in performance at an alpha value of 0.1 indicates that it might be a more general model.

It is worth noting that I did not extend training all the way to 1000 epochs as in part 1A. This was due to the visible drop in performance as epochs increased in 1B.D and training the model at this large epoch is computationally expensive.

1c**:** Figure and caption: For the best model with K=2 factors, consider the learned per-movie vectors for the short list of movies listed in select_movies.csv. Can you make a scatter plot of the 2-dimensional “embedding” vectors of these movies (labeling each point with its movie title), like we saw in lecture? Do you notice any interpretable trends?

Figure 1C: Plotted above is the generated embedding vectors for a shortened list of movies, learned with an alpha value of 0.1, K = 2, and 600 epochs. We can see that sequels such as Star Wars, Jurassic Park, and Scream movies display similar embeddings. It also does a decent job at identifying children's movies (Snow White and the Seven Dwarfs and the Wizard of Oz are nearby although notably not Toy Story and the Lion King), horror films (Scream, Scream 2, Nightmare on Elm Street), and Action Films. This makes sense because these embeddings are learned from user preferences and they would be more likely to watch movies of similar genres or ones targeted towards specific audiences, like children. It is fallible however, as for example, the shining is not paired near other horror movies here. Unique films such as a musical like the Sound of Music, and a children's-horror-holiday blend like Nightmare on Elm Street are not nearby other movies.

Part 2:

2a: Please describe the user and item features you have used.

To construct the feature vectors of the users, we use their age and gender as a boolean. For the item feature vectors, we used the release date. We attempted to use the name in a BoW representation, however this consistently led to inferior results, likely due to overfitting.

2b: Please report which classifier you have used. Please describe how you have done model selection (how you have chosen hyperparameters).

In our final model, we used a RandomForestClassifier. To select hyperparameters, we used the GridSearchCV algorithm to search over a range of hyperparameters such as `n_estimators, max_features, max_depth, min_samples_split, min_samples_leaf, and bootstrap` to find the best ones.

2c: Please try another classifier. Then you have two classifiers and three K values. Please create a 2x3 table showing the AUC value from each combination.

	K = 2	K = 10	K = 50
RandomForest	0.761	0.75858	0.761787
Logistic Regression	0.7594896	0.75921	0.759400