CSE 158 Report

Assignment 2

Yishai Silver and Ethan Spraggon

December 2, 2021

1 Dataset

Our project uses one dataset based on social recommendation data. The dataset is from the LibraryThing website where users write comments and rate different books. That being the case, not all users provide ratings in their reviews. In light of our predictive task (rating prediction) such users have been removed from the dataset.

Additionally, the dataset comes with a trust network which denotes relationships between users. Generally speaking, these relationships denote friendships and shared interests. Throughout this assignment, we hope to leverage this data in particular to improve our rating predictions for each user.

One last key piece of information that we wanted to focus on were the time-stamps associated with each review. Such time-stamps give our data a sequential quality that we hope to use.

There are a few notable statistics within the dataset. The average scores for books across all reviews is about 3.8. There are 61342 items and 24878 users from the data we selected. Most every gave a rating higher than 1 since it was implied that comments without a star rating were 0 or the user had no preference toward the book at all.

Our attempts to use this dataset in its entirety were unsuccessful. We were working on a machine that had 1 GPU, 8 CPU's, and 16G RAM, but this machine was not powerful enough to contain all of the parameters that we needed at the same time; it would often crash when we were attempting to create features for our models. As such, we used only 100,000 samples from the dataset.

2 Predictive Task

We chose to predict the star-rating a user would give a certain book because it is an interesting problem in and of itself and the predictions could likely be used to build an interaction prediction pipeline. Likewise, as students enrolled in 158 and not 258, we thought it would be interesting to give ourselves a more interesting challenge. In terms of evaluation, we think mean squared error would be appropriate.

2.1 Data Considered:

2.1.1 Sequential Activity

Each interaction has a timestamp associated with it. As such, we can try to leverage the user's past interaction and rating history to improve the predictions we make. As an example, if we are trying to predict a user's rating on the last "Harry Potter" movie, it would probably be helpful to know their ratings for all of the other "Harry Potter" movies.

2.1.2 Friends

We wanted to leverage the social data provided in edges.txt. People tend to associate with one another on the basis of similar beliefs and interests, so we believe that the opinions of a user's friends might be indicative of the user's own opinions. The users provided in the dataset have an average of 4.84 friends, so we chose to round up and consider the 5 most indicative friends, as revealed by finding their cosine similarity. That being the case, we chose to have a friend's opinion feature of length 15. For each friend, there are 4 values: one for whether or not the friend is a real user or just empty values (this value takes on either a 1 or a 0; 0 if there is no friend), one denoting the friend's similarity to the user in question (0 if there is no friend), one denoting the friend's Jaccard similarity to the book in question (comparing the set of users belonging to the friend's friends and the set of users who have interacted with the book), and one to denote the rating that the user gave the book in question (if they gave a rating; 0 if there is no friend). Which friends are included is dependent first upon which friends have interacted with the book in question and then upon the magnitude of their similarities.

Similar to one of the features above, we also thought it'd be interesting to use the Jaccard similarity between the set of friends belonging to the user and the set of users belonging to the book.

2.1.3 Review Text

There's a lot of information provided by the text in the reviews, and we thought it would be appropriate to include this information as part of the features that we provide to our network. That being the case, we acknowledge that the text the user provides in his own review will not be available at runtime. As such, we do not believe that the user's own review should be used. Instead, we intend on using a bag of words vector that includes the frequencies of words in other reviews.

Another issue of using text mining in the context of this dataset is that there is no guarantee information about the work will be transmitted in the vector. In other words, consumers know who are leaving a review may not be inclined to specify the product for which they are leaving a review; they might only say something like "it's very good." As such, there is only so much information that we can use to indicate what kind of book it is.

2.2 Model Evaluation

2.2.1 Loss Function

We found that Mean Squared Error (MSE) is commonly used as a loss function for rating prediction tasks. We briefly tried training our models to predict via one-hot vectors with the Categorical Cross Entropy function as our loss function, but we found that it was less effective than MSE.

2.2.2 Baselines

Using the average rating across all reviews for our prediction resulted in an MSE of 0.9910.

Using an item's average rating (or the global average if it is a new item) for the prediction resulted in an MSE of 1.108.

Using the user's average rating (or the global average if it is a new item) for the prediction resulted in an MSE of 0.987.

3 Model

3.1 Linear Regression

We started experimenting with a simple Linear Regression model that encompasses a few different experiments, each with a different feature vector. The first experiment included all three feature: Unigrams on the comment on the book, the top five friends for the user, and a Jaccard score for friend similarity. The second experiment is without the first feature. The third experiment is without the middle feature, and the forth experiment is without the last feature. We chose this model first due to it's simplicity. We optimized it slightly with a Ridge model, adjusting the regulariser but found no significant improvement on our results. We ran this linear regression on each experiment, predicting both on the test data and the training data. Again, there was not much difference between predicting on the test versus the training data. In fact, all of our different experiments did not seem to change the result by very much. This could be because our features are too complicated.

We conducted several ablation experiments with five epochs (these scores are for

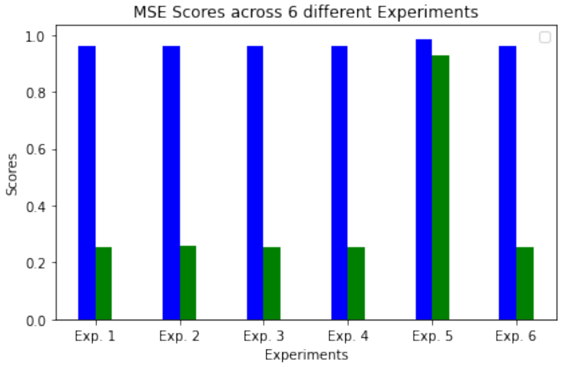


Figure 1: A Measure of MSE scores across different experiments.

]

the validation dataset).

- Including all of the features: MSE of 0.962
- Excluding bag-of-words: MSE of 0.962
- Excluding Friends Feature: MSE of 0.962
- Excluding User-Item Jaccard Similarity: MSE of 0.962
- Excluding User-Average: MSE of 0.986
- Excluding Item-Average: MSE of 0.962

The results of our experimentation here did not reveal much as to what features could be used to improve predictions in the context of linear regression (except for including the a user's average rating, which seemed to improve the predictions).

3.2 Markov Chain

Many of the papers we read praised Factorized Personalized Markov Chains (FPMC) as a valuable means of predicting user-item interactions [1]. We thought we might build upon this work and adapt it to predict user-item ratings. To test this hypothesis, we trained both a non-personalized and Markov Chain as well as a personalized Markov Chain and then added the predictions of these functions to the global average. We expected that this would perform better than than the simply predicting the global average itself, but this was not the case: The global average had an MSE of 0.991, whereas modified Markov Chain functions had 1.012 and 1.026, respectively. In other words, we found that our hypothesis was wrong and that we could not reasonably use Markov Chains to guide or otherwise improve our predictions.

3.3 Fully Connected

3.3.1 Justification

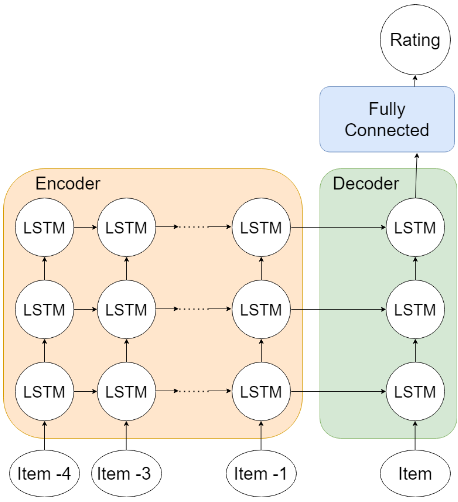
One might recall that we made it possible for social features to include empty values and that these values were denoted as being such by the first dimension in these features (which is either a 1 or a 0). That being the case, linear regression only discovers how to weigh dimensions independently of one another. As such, linear regression models can't learn to evaluate some dimensions in lieu of others. Multi-layer networks, however, are capable of doing so. Therefore, we explored the use of fully connected neural network to predict a user's rating for a particular item. Another reason for doing so were the lackluster results of our Markov Chain experiment (and the relatively good results of linear regression).

Our network consists of 8 layers, with each containing 4096, 8192, 4096, 2048, 1024, 512, 256, and 1 neurons, respectively. Likewise, the ReLU activation function is applied to the outputs of the first 7 layers in our network.

3.3.2 Results

We found that this model performed reasonably well relative to all of the other models we evaluated. For our feature vectors, we included five components: a bag-of-words vector, the friends feature described above, the Jaccard similarity between the user's of friends and the users who have interacted with the item, the user's average rating (or the global average if they're a new user), and the item's average rating (or the global average if it is a new item).

We conducted several ablation experiments with five epochs and the results seemed to be more revealing than our experiments in the context of linear regression.



]

Figure 2: Structure of LSTM Seq2Seq

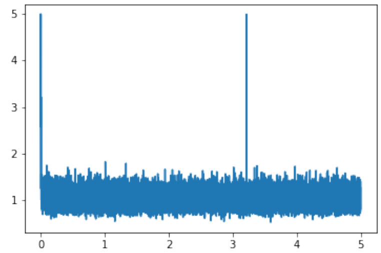
- Including all of the features: MSE of 0.875
- Excluding bag-of-words: MSE of 0.820
- Excluding Friends Feature: MSE of 1.336
- Excluding User-Item Jaccard Similarity: MSE of 0.915
- Excluding User-Average: MSE of 1.074
- Excluding Item-Average: MSE of 0.858

These results suggest that the friends features that we created were actually contributing to our model's performance. Similarly, the decrease in performance from the exclusion of user-item Jaccard similarity shows quite plainly that the social data can be used to improve model performance. On the other hand, our experiments suggest that the high dimensionality of our bag-of-words vector was detrimental to our model and that, despite their performance as baselines, using item averages to guide performance may not be beneficial.

3.4 LSTM Seq2Seq

3.4.1 Justification

We believe that activity could be viewed simply as a sequence of individual events. As such, we thought that we should explore models which are commonly used for predicting sequences. One such model of particular interest is a seq2seq network. We believed that it would perform similarly to that of a fully connected neural network (thus allowing us to learn the importance of the various features we provide it) while also going so far as the consideration of features for the items in the user's past.



]

Figure 3: LSTM Seq2Seq: MSE Over Training

Another reason we thought this would be an interesting model for us to explore was because we believed it would be capable of using temporal features without the need to store massive amounts of data via sparse matrices (as is the case with Markov Chains).

For features representing past items we use all feature components used in the features used for a regular fully connected model, but we also append the star-rating provided for those models. We use a total of five past items to represent user history. If the user does not have 5 items in their past, we add 0 vectors to fill in. To distinguish between 0 vectors and real items, we add another dimension that contains 1 for real items and 0 for 0 vectors.

As for the model itself, both the encoder and the decoder modules have 3 LSTM layers that have 512, 1024, and 2048 neurons, respectively. The fully connected module that parses the output of the decoder module consists of 5 layers that have 2048, 1024, 512, 256, and 1 neuron respecetively. The ReLU activation function is applied to the first 4 of these layers. A diagram of this model can be found in Figure 2.

3.4.2 Results

We found that using the features we discovered to offer the best predictions in the context of a fully connected network offered even better results in the context of our LSTM Seq2Seq network: over the course of 5 epochs, our best MSE for the validation dataset was 0.791, our best score yet. We believe that this improvement might be the result of both casual trajectories in how a user rates movies (shifting mentalities that are independent of the movies they are watching) as well as the possible convergence of friends watching movies (increases in similarities might be associated to an immediate increase in the popularity of a movie). Both of these could be found relatively easily in a Seq2Seq network but would be much harder to see in a fully connected network (especially because we didn't think it would be appropriate to consider a user's past items as full features; this requires additional experimentation).

A plot of MSE scores over the course of training can be found in Figure 3.

4 Literature

The dataset we used came with link to a paper by McAuley et. al. in which they focused on using personalized feature projection to improve one class recommendation [2]. In other words, they used matrices instead of vectors to represent user preferences. They found that this works better because it better represents more complicated relationships between a users biases and the items they interact with; using vectors to reach a singular value oversimplifies a user's opinion, whereas using matrices allows for higher dimensional, more complicated representations of relationships between users and the items they interact with.

The dataset was linked to an additional paper by McAuley et. al. that proposes a new model to leverage both sequential and social behavior [3]. That being the case, this paper does not use our dataset, rather it focuses on datasets named "Epinions," "Ciao," "Foursquare," and "Flixster." The "Epinions" dataset contains the same data contained by LibraryThing, but for a wider variety of items (whereas LibraryThing data is concerned only with books). Likewise, the "Ciao" dataset contains the same data (and is also oriented towards general products), but also includes timestamp information for when a social link is established and is a smaller dataset in general. Flixster and Foursquare are similar datasets, but are oriented towards movies and venues, respectively. Despite these datasets being very similar to our own, the paper's authors focused on predicting whether or not a user would interact with a certain item, and not necessarily what score a user would give an item. That being the case, this paper inspired our exploration of the LSTM seq2seq model.

Another paper that we found particularly interesting was also by McAuley et. al. [5]. It denoted that we were correct in using MSE as our loss function as well pointed us to some state-of-the-art methods for predicting user-item ratings (though such methods do not consider social data). In particular, it covers Latent Dirichlet Allocation (LDA), which gives a vector to each document and uses that vector to encode the fraction of words associated with a certain topic. Likewise, each topic has a vector encoding the extent to which each word is associated with that topic and there is a vector to model the distribution of these topics themselves. Though not directly related, LDA performs well for sentiment analysis and thus can be leveraged to predict ratings. The paper, however, proposes a new method called "Hidden Factors as Topics" that links rating parameters to review parameters. Intuitively, this is similar to what we attempted, but in practice this focuses on ratings and reviews (text) separately.

We originally intended on evaluating our model on the various datasets described above, but due to the constraints of the computer we were developing on we ended up focusing on improving our predictions for the LibraryThing dataset.

5 Results

In addition to the model-specific results above, we generally found that we could leverage social data to improve the predictions of our model. Likewise, we also found that we could improve our results by leveraging sequential and historical data.

Our codebase can be found on github [5].

References

[1] Rendle, S., Freudenthaler, C, and Schmidt-Thieme, L.,"Factorizing Personalized Markov Chains for Next-Basket Recommendation", 2010

[2] Zhao, T., McAuley, J., King, I., "Improving Latent Factor Models via Personalized Feature Projection for One Class Recommendation"

[3] Cai, C., He, R., McAuley, J., "SPMC: Socially-Aware Personalized Markov Chains for Sparse Sequential Recommendation", University of California, San Diego

[4] McAuley, J., Leskovac, J., "Hidden Factors and Hidden Topics: Understanding Rating Dimensions with Review Text"

[5] https://github.com/yishaiSilver/Social-Consumer-Recommender-Systems