Sentiment-Enhanced Neural Collaborative Filtering

Leroy Souz
Rutgers University, New Brunswick
lms548@rutgers.edu

ABSTRACT

Neural Networks have been widely used in many applications including recommendation systems and have proven to provide a significant improvement in performance compared to traditional methods. Recent developments in Natural Language Processing has also facilitated better capturing of the semantics of underlying text.

This paper presents a novel architecture that integrates deep learning techniques with sentiment opinion mining to improve the quality of inferred ratings. Deep learning techniques have proven to be effective in capturing complex user-item interactions. However, the sentiment expressed in user reviews provides valuable information ratings cannot provide and can be leveraged to further improve the rating inference. By incorporating the sentiment scores, our proposed architecture aims to capture the emotional aspects of user feedback and combine them with the collaborative filtering capabilities of Neural Networks.

We conduct experiments on the Amazon CDs and Vinyls dataset to evaluate the effectiveness of our proposed architecture. The results demonstrate that the integration of sentiment analysis with deep learning leads to significant improvements in rating accuracy compared to existing deep learning models.

KEYWORDS

Rating Inference, Sentiment Opinion, Recommendation Systems, Deep Learning

1 INTRODUCTION

Recommendation systems leverage historical user interaction data to predict user preferences and recommend items accordingly. Among the multitude of approaches for building recommender systems, Collaborative Filtering (CF) has emerged as a foundational technique, excelling at capturing patterns of user-item interactions. One of the most popular CF methods is Matrix Factorization [7] which decomposes the user-item interaction matrix into two lower-dimensional matrices that capture user and item latent factors. Unfortunaely, such models are limited to being able to capture only the linear relationships between users and items. To overcome this issue there has been a lot of research done in levraging Neural Networks for recommendation systems [3, 4]. These models aim to capture more complex user-item interactions by learning non-linear relationships between users and items with the help of multiple connected layers.

However, ratings by itself are not enough to capture the user opinion towards the item due to their finite set of opinion choices (making the user choose from a short range of numbers). They fail to capture user sentiment towards the item. There are possiblities that a user might provide a high rating for a product because he was impressed with the quality but express negative sentiments towards other certain aspects of the product in the review. This paper aims to address this limitation by providing an architecture that integrates sentiment analysis with deep learning techniques to improve the quality of inferred ratings. To achieve this, we use a Neual Collaborative Filtering model [4] as the base model and incorporate sentiment scores obtained from user reviews using VADER [1] as an additional input to the model. Due to the lack available of user reviews during inference, we train another NCF model that learns to predict the sentiment score for user-item pairs.

We conduct experiments on the Amazon CDs and Vinyls dataset to evaluate the effectiveness of our proposed architecture. The results demonstrate that the integration of sentiment analysis with deep learning leads to significant improvements in rating accuracy compared to a standalone Neural Collaborative Filtering Model with an increase in 13% accuracy.

2 RELATED WORK

Using sentiment analysis for understanding user opinion has been researched in the past. [2, 5, 6, 8, 9]. A description of some of the work is provided below.

Leung et al. [8] explored how the strength of words can be determined to be used in a CF based model. Techniques such as POS tagging and negative tagging were used and strength of words was calculated based on their importance. The word strength was then used to build a opinion word corpus and train a CF model that did rating inference based on the sentiment of the words.

The paper by Ganu et al. [2] explored how textual content in user reviews can improve rating predictions in a recommendation system. They classified review sentences into categories (e.g., food, service, ambience) and sentiments (positive, negative) using machine learning techniques like Support Vector Machines. This classification was used to calculate text-based ratings that incorporated both sentiment and topic information. The study demonstrated that using structured textual data outperforms relying solely on star ratings, particularly in personalized restaurant recommendations.

Hu and Liu [5] proposed a method to extract product features from customer reviews and analyze the sentiment associated with each feature. Techniques like POS tagging and association rule mining were used to identify explicit features, while opinion words nearby these features determined sentiment. The system summarized reviews by aggregating opinions on features, offering structured insights for potential customers.

Conference'17, Washington, DC, USA

DOI: 10.1145/nnnnnnn.nnnnnnn

3 PROBLEM FORMALIZATION

We aim to predict user ratings $\hat{r}$ for items, using both collaborative filtering (CF) techniques and sentiment analysis scores s derived from textual opinion mining (e.g., VADER). The proposed model combines Matrix Factorization (MF) and Neural Networks (NNs) to capture user-item latent interactions as well as sentiment-based contextual features.

The primary objective is to learn a function $f(u, i, s)$ that maps user u, item i, and the sentiment score s to a predicted probability distribution over possible ratings. Specifically, we aim to predict $\hat{\mathbf{r}}_{u,i} = f(u, i, s) \in \Delta^{K-1}$, where Δ^{K-1} is the K-dimensional probability simplex corresponding to K rating categories (e.g., $K = 5$ for a 5-star rating system). The final predicted rating $\hat{r}_{u,i}$ can then be derived as the category with the highest predicted probability.

The model is trained to minimize the categorical cross-entropy loss, defined as:

$$\mathcal{L} = -\frac{1}{|R|} \sum_{(u,i)\in R} \sum_{k=1}^{K} \mathbb{I}(r_{u,i} = k) \log(\hat{\mathbf{r}}_{u,i}[k]),$$

where: - $r_{u,i}$ is the true rating, and $\hat{\mathbf{r}}_{u,i}[k]$ is the predicted probability for the k-th rating category. - $\mathbb{I}(\cdot)$ is an indicator function that is 1 when the condition is true and 0 otherwise.

The model combines MF embeddings, which capture linear user-item latent interactions, with NN embeddings, which capture non-linear patterns and integrate sentiment information. The inclusion of sentiment scores s ensures that textual opinions are factored into the prediction process, allowing for more accurate and context-aware recommendations.

4 THE PROPOSED MODEL

A visualization of the model can be seen in Figure 1.

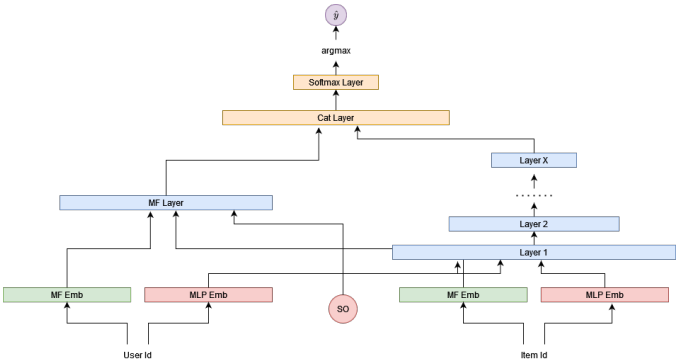

Figure 1: Model Architecture

Components of the model include:

4.1 MLP Layers

The input of a model is a vector:

$$\mathbf{z}_{u,i} = [\mathbf{p}_u^T, \mathbf{q}_i^T, s_{u,i}],$$

where:

- $\mathbf{p}_u \in \mathbb{R}^{d_{nn}}$ is latent embeddding vector for user 'u'
- $\mathbf{q}_i \in \mathbb{R}^{d_{nn}}$ is latent embeddding vector for item 'i'.
- $s_{u,i}$ (scalar) is the sentiment score for user 'u' for the item 'i'.

We concatenated these embeddings to get $\mathbf{z}_{u,i}$ and pass it through several nonlinear layers in the NN for feature transformation.

4.2 Matrix Factorization

The MF component learns another pair of user-item embeddings to approximate the rating:

$$\text{mf\_x}_{u,i} = (\mathbf{P}_u \odot \mathbf{Q}_i) \cdot s_{u,i},$$

where:

- $\mathbf{P}_u \in \mathbb{R}^{d_{mf}}$ is the latent vector for user 'u'.
- $\mathbf{Q}_i \in \mathbb{R}^{d_{mf}}$ is the latent vector for item 'i'.
- $s_{u,i} \in \mathbb{R}$ is the sentiment score for the user-item pair.
- $\odot$ denotes element-wise multiplication.

The user and item embedding vectors are multipled together and scaled by $s_{u,i}$ to obtain MF layer.

4.3 Combining MLP and MF

These two layers are then concatenated vertically and passed through a final MLP layer of size C to be sent through a softmax layer to get the final rating prediction $\hat{r}_{uv}$ as:

$$\hat{r}_{uv} = \phi(\mathbf{W} \cdot \text{concat}(\text{mf\_x}_{u,i}, \mathbf{z}_{u,i})),$$

where:

- $\mathbf{W} \in \mathbb{R}^{C\times(d_{nn}+d_{mf})}$ is the weight matrix.
- $\phi(\cdot)$ is the softmax function.

4.4 Obtaining the Sentiment Score

When doing rating inference for items, we first need obtain the sentiment score for the user-item pair. This was readily avaiable in the training set through conversion of reviews to SO using VADER. But during inference, we do not have access to reviews for all items. Thus we train a seprate NCF model that learns to predict the sentiment score for a user-item pair. This model is trained using the same architecture as the rating inference model but with the sentiment score as the target variable.

The objective function being minimized is:

$$\mathcal{L}_{\text{sent}} = \frac{1}{|R|} \sum_{(u,i)\in R} (s_{u,i} - \hat{s}_{u,i})^2,$$

where:

- $s_{u,i}$ is the true sentiment score.
- $\hat{s}_{u,i}$ is the predicted sentiment score.
- R is the set of user-item pairs.

The model driagram can be seen in Figure 2.

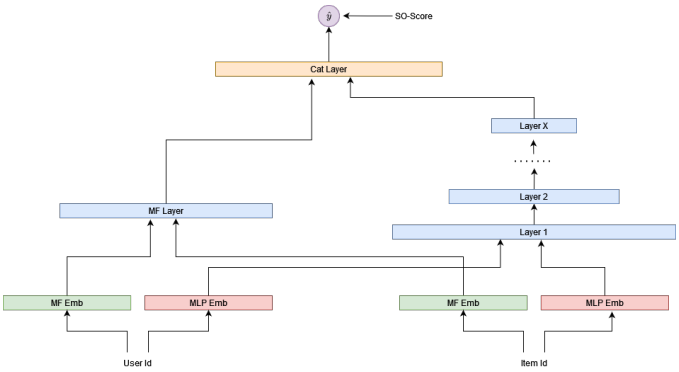

Figure 2: Model Architecture

5 EXPERIMENTS

We experiment with the 5-core Amazon CDs and Vinyls dataset [10] to test for two things:

(1) We check if the proposed model outperforms the baseline NCF model in terms of rating prediction accuracy.
(2) We also check if it is better at top-k recommendation tasks.

5.1 Dataset

The dataset consists of user-item interactions and reviews for CDs and Vinyls on Amazon. It contains 1,443,475 samples with features including 'reviewer id', 'item id', 'name', 'verified', 'review', 'rating', 'time', 'summary', 'unix time', 'style', 'vote', 'image'. We only use the 'reviewer id', 'item id', 'rating', and 'review' columns for our experiments.

We preprocess the dataset by dropping any N/A values. There are around 300 NANs in the dataset for the features we are focusing on so dropping them doesn't lead to a huge loss of information. We then covert ids to integers and also remove users with less than 5 reviews to maintian the 5-core nature of the dataset. We also convert reviews to a sentiment score using VADER. We then split the dataset into a 80-20 train-test split by splitting each users reviews into 80% training and 20% testing. This way we have a fair representation of users in both the training and testing set.

There is a huge imbalace in the classes with 5-star ratings being the most common and 33x more than 1 star ratings. We thus make sure to use a weighted loss function during training to account for this imbalance. The weights for a class is calculated as:

$$w_c = \frac{n}{|C| \cdot n_c}$$

where:

- n is the total number of samples.
- n_c is the number of samples in class c.
- $|C|$ is the number of classes.
- w_c is the weight for class c.

5.2 Rating Inference

We design both models with a similar architecture, ensuring they have the same number of layers and hidden units. Both models are trained for 10 epochs using a batch size of 128. Not much fine tuning is done except preventing overfitting.

To compare the rating inference capabilities of both models, we focus solely on the second half of the model during testing. Instead of predicting sentiment scores, we utilize the precomputed sentiment scores available in the testing set. This approach allows us to directly evaluate whether the inclusion of sentiment scores enhances the model's performance. We compare these predictions to a standalone NCF model on metrics such as RMSE, MAE, and accuracy and precision. The results are shown in Table 1.

As can be seen, the NCF model with sentiment opinion outperforms the standalone NCF model in all metrics. The RMSE and MAE are significantly lower, and the accuracy and precision are higher. This demonstrates that the inclusion of sentiment scores improves the model's ability to infer ratings, leading to more accurate and precise predictions.

Model	RMSE	MAE	Accuracy	Precision
NCF	1.74	1.14	0.44	0.63
NCF + SO	1.20	0.65	0.70	0.56

Table 1: Rating Inference Results

5.3 Top-k Recommendation

We now evaluate the models on a top-k recommendation task. We use the same testing set as before and generate the top 10 recommendations for each user. We do this by passing user-item pairs for a sample of 1000 users through the entire model. The first part predicts, the sentiment score and the second part predicts the rating using this NCF + SO. We then rank the items based on the predicted ratings and calculate the precision@k, recall@k, f1 score, and NDCG for each user and average them over all users to get the metrics. These can be seen in Table 2.

Model	Precision@k	Recall@k	F1 Score	NDCG
NCF	0.0022	0.00090	0.00121	0.00192
NCF + SO	0.0006	0.00015	0.00024	0.00047

Table 2: Comparison of Average Metrics for NeuMF Models

Both models perform terribly on the top-k recommendation task but the NCF + SO model performs comparatively worse than the standalone NCF model. We can attribute this to multiple reasons. The poor score on both models can be attributed to the imbalanced nature of the dataset. Data analysis reveals that many users provide similar ratings for all the items they have reviewed. This causes the model to predict uniform ratings across items, leading to less diverse recommendations. The model is tested on how many of the top 10 recommendations overlap with the test set. Due to most of the ratings being similar, the model is penalized for providing valid recommendations that are not in the test set.

The poor score of the NCF + SO model can be attributed to the need of seprate model to predict the sentiment score. The model is not perfect at predicting sentiment scores for (user,item) pairs and this leads to the rating inference model being fed inacurate sentiment scores. This leads to the model not being able to accuratey predict the ratings.

6 CONCLUSIONS AND FUTURE WORK

The integration of sentiment analysis into recommendation systems has shown promising results in improving the accuracy of inferred ratings. By combining a Neural Collaborative Filtering model with sentiment scores derived from user reviews using tools like VADER, the paper demonstrates a significant enhancement in prediction accuracy, as evidenced by a 13% improvement on the Amazon CDs and Vinyls dataset. This suggests that incorporating emotional aspects of user feedback can address the limitations of traditional models that rely solely on numerical ratings.

However, there are still many areas for improvement. Future work could focus on obtaining a more accurate sentiment score for user reviews using state-of-the-art models like BERT. Work could also be done on creating a better model to learn the sentiment score for user-item pairs. Additionally, the top-k recommendation task could be improved by addressing the issue of imbalanced ratings by using techniques like oversampling. The two models, SO predictor and rating predictor, were trained seprately and could be improved by training them together.

ACKNOWLEDGEMENT

Special thanks to all the authors of the papers referenced in this work, Dr Zhang for his guidance and support, and ilabs for providing the computational resources.

REFERENCES

[1] Eric Gilbert C.J. Hutto. 2016. VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text. *AAAI* (2016).

[2] Gayatree Ganu, Noemie Elhadad, and Amélie Marian. 2009. Beyond the Stars: Improving Rating Predictions using Review Text Content.. In *WebDB*, Vol. 9. Citeseer, 1–6.

[3] Xiangnan He, Xiaoyu Du, Xiang Wang, Feng Tian, Jinhui Tang, and Tat-Seng Chua. 2018. Outer Product-based Neural Collaborative Filtering. *IJCAI* (2018).

[4] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. In *WWW*. 173–182.

[5] Minqing Hu and B. Liu. 2004. Mining Opinion Features in Customer Reviews. In *AAAI Conference on Artificial Intelligence*. https://api.semanticscholar.org/CorpusID:5724860

[6] Niklas Jakob, Stefan Hagen Weber, Mark Christoph Müller, and Iryna Gurevych. 2009. Beyond the stars: exploiting free-text user reviews to improve the accuracy of movie recommendations. In *Proceedings of the 1st international CIKM workshop on Topic-sentiment analysis for mass opinion*. ACM, 57–64.

[7] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 8 (2009), 30–37.

[8] Cane WK Leung, Stephen CF Chan, and Fu-lai Chung. 2006. Integrating collaborative filtering and sentiment analysis: A rating inference approach. In *Proceedings of the ECAI 2006 workshop on recommender systems*. 62–66.

[9] Tetsuya Nasukawa and Jeonghee Yi. 2003. Sentiment analysis: capturing favorability using natural language processing. In *Proceedings of the 2nd International Conference on Knowledge Capture (K-CAP '03)*. Association for Computing Machinery, New York, NY, USA, 70–77. DOI: http://dx.doi.org/10.1145/945645.945658

[10] Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying Recommendations using Distantly-Labeled Reviews and Fine-Grained Aspects. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (Eds.). Association for Computational Linguistics, Hong Kong, China, 188–197. DOI: http://dx.doi.org/10.18653/v1/D19-1018