Harmonized and Optimized User Satisfaction for Efficient Recommendation

Ethan Reinhart

March 19, 2025

Abstract

In this project, we introduce HOUSER, a method of combining predicted user-item reviews with predicted user-item interactions. This method leverages the concept of a user being satisfied with a purchase by considering how likely a user is to purchase an item combined with how highly the user will rate the item. We leverage the spatial capabilities of GCNs to train link prediction and edge classification models with high accuracy. We optimize the combination of these models and compare them against a baseline GCN model trained over the same data. We address problems within the dataset and potential issues that could arise from our data. Finally, we speculate over model improvement and discuss where this should be taken in the future.

1 Introduction

1.1 General Background

Recommender systems are widely used to predict user preferences and recommend items based on historical interactions. These models typically consider user-item interactions such as purchases, views, and frequency. These are proven beneficial to predict whether a user will purchase an item but fail to predict whether a user will actually be satisfied with the item purchased. This is typically addressed by recommending items above a certain average rating threshold or ignoring graph structure and considering only user preferences of users who have purchased and rated items similarly. These fall short in two ways, the former generalizes too much whereas the latter becomes too specific. These models can be adequate but they fail primarily because generic link prediction optimizes for user purchases as opposed to user satisfaction. One paper mentions that ”relying solely on the precision metric does not assure a high-quality recommendation because items with high accuracy scores do not always guarantee user satisfaction” [9]. This domain is an active area of interest and falls under the MORS (multi-objective recommender systems) where attributes such as accuracy, user satisfaction, diversity, and novelty are optimized in unison [9].

1.2 Specific Problem

To optimize for user satisfaction, we introduce HOUSER. This model combines the output of a pre-trained link prediction and edge classification problem. The former model can accurately predict the probability a user will purchase an item given the neighborhood of users that have purchased similar items. The latter model can accurately predict the probability a user will leave a high rating for the purchased item. The combination of these two results in a new prediction, whether a user will purchase an item and leave a high rating (be satisfied with the purchase).

We argue that generic link prediction or edge classification is not sufficient. Link prediction, even with a rating threshold or a neighborhood of similar purchase-rating pairs, is not fully optimized to determine whether a user will like an item. Edge prediction is not optimized to determine whether a user will purchase an item due to the absence of negative sampling throughout training. Combining the two leverages the capabilities of both to generate a novel representation of user satisfaction while still optimizing user purchase probability. We argue this combined architecture is beneficial to both users and retailers as it is capable of optimizing both profits, whether a user buys an item, and user satisfaction, how a user rates an item.

2 Motivation

2.1 Problem Statement

Some psychological research has been done to determine the effect of recommender systems on user satisfaction. One study found that the two most important metrics to optimize user satisfaction are diversity and accuracy [2]. This evaluation was performed on users who interacted with a recommender system in real-time, measuring their satisfaction with the items recommended. Another study determined that the uses and gratification theory, the motivations for actions that affect user satisfaction, primarily impacts user satisfaction because users with a specific goal become frustrated with too many recommendations while browsing users prefer more recommendations [4].

Computational research has also been done to optimize user satisfaction. PURS addressed the filter bubble problem, where users suffer intellectual isolation that can result from highly personalized searches, by introducing 'unexpected', previously unexplored recommendations to increase user engagement and satisfaction [3]. This proved promising within the domain of content recommendation as user engagement had slightly high retainment, but this has not been extensively tested with item recommendation. Another diversity approach, ISR-MOEA optimizes dataset coverage of the top-K utility patterns [10]. This approach optimizes user satisfaction through the addition of diversity while continuing to optimize top-K utility, such as profit, cost, quantity, etc. This combined approach is well-suited to make both users and retailers happy as user satisfaction as well as retailer profit are both optimized.

2.2 Contributions

Through the combination of link prediction and user satisfaction, we introduce a new model that optimizes post-purchase satisfaction. While many other previous implementations excel at user engagement and satisfaction over provided recommendations, they fail to consider post-purchase / post-consumption user satisfaction. We tackle the issue of post-purchase user satisfaction but acknowledge that little work has been done regarding content recommendation given post-consumer satisfaction. Our implementation provides novelty in creating a recommendation system where users are recommended items that they like and are happy with after purchase. We argue this novel approach has two primary benefits. This approach allows users to be more satisfied with their items after purchase, increasing user satisfaction with the marketplace and user trust in retailers. This approach also allows for higher ratings for retailers as recommending items that users are likely to love after purchase increases the likelihood of benevolent retailer or item ratings. While we recognize this novel approach does not optimize top-K utility, primarily profit, we hypothesize that high user satisfaction, retailer trust, and retailer reviews will lead to more transactions and trust within the marketplace.

3 Data

3.1 Data Collection

The dataset used in the analysis of this model is provided by the McAuley Lab from UCSD over Amazon reviews by item category. We argue that this dataset is sufficient as this was curated for the BLAiR model, another recommender system, that performed well [5]. Due to processing capabilities, we elected to train over the smallest item category subset: Gift Cards. This dataset includes user reviews for items, containing information such as review message, review rating, user ID, item ID, retailer ID, verified purchase, and item metadata. We elected to incorporate only user IDs, item IDs, and ratings in our post-processed dataset.

We also note two specific issues that can affect recommendation and accuracy:

- To keep the dataset size sufficiently large, we include all user-item reviews as purchases, even though not all of these are verified. We recognize that this is not entirely accurate but argue that this is sufficient to draw conclusions.
- The dataset provided contains only user-item reviews. It does not contain all existing user-item purchases. A subset of total purchases is contained in the reviews dataset but due to dataset and computational restraints, we argue that this is again sufficient.
- For the HOUSER model, we determine a threshold 4 or 5 stars is required to

determine whether a user likes an item. We argue that only considering 5-star reviews results in the model not obtaining sufficient information to make accurate predictions. We include 4-star reviews as these are generally positive and are above an average review.

- Due to computational and time constraints, we resorted to subsampling the original dataset. We set manual seeds so that our exact results should be reproducible on any machine.

We analyze the dataset by computing distributions for the number of ratings of each class (Figure 1), the number of reviews each user has placed (Figure 2), and the number of reviews each item has (Figure 3). We observe that there are an average of 1.1483 reviews per user, an average rating of 4.3274 and an average of reviews per item.

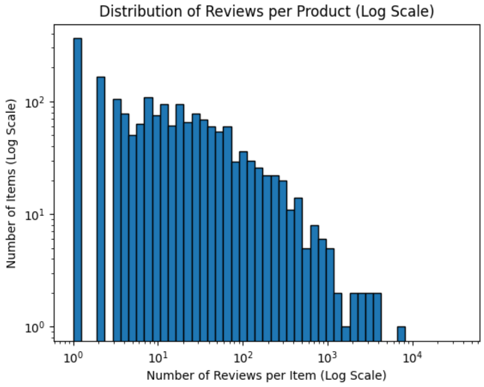

Figure 1: Item Review Distribution (Log Scale)

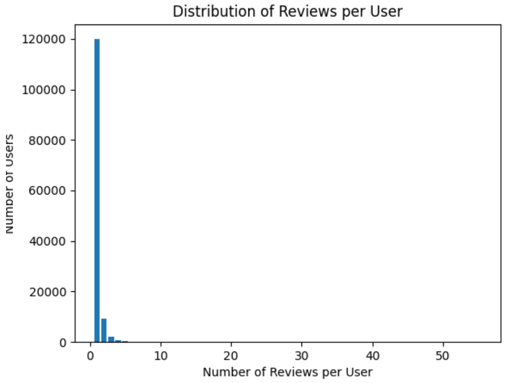

Figure 2: User Review Distribution (Log Scale)

We note that the item and rating distributions are distributed relatively logarithmically and the majority of users/items have left/have a single rating. We note that the majority of ratings are positive which is beneficial for learning the HOUSER model.

For link prediction, edge indexes correspond to user-item ratings in our dataset and labels are binary. For edge classification, edge indexes correspond to user-item ratings in our dataset and labels are normalized ratings over the interval [0,1]. For the HOUSER model and the Evaluation GCN, edge indexes correspond to user-item ratings where the user reviewed the item as 4 or 5 stars and labels are binary.

We ensured that all models received the same testing and training samples. This ensures that no model receives an optimal subset and that the HOUSER model is not trained on testing data.

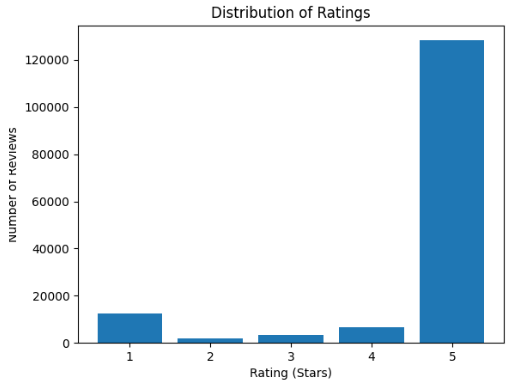

Figure 3: Rating Distribution (Log Scale)

4 Model Types

4.1 Heuristic

The heuristic model in our implementation is based on the commonly used Jaccard heuristic for recommender systems. This has to be adjusted slightly as this heuristic was originally created for non-bitartite graphs. First, this heuristic computes a correlation matrix of size [number of users × number of users] to predict user similarities. This computes similarity by using the Jaccard Heuristic:

$$\frac{U_i \cap U_j}{U_i \cup U_j}$$

where U_i, U_j is the set of items interacted with by user i, user j, respectively, in the training set. For link prediction, an item is recommended by taking the most similar k users and then determining the percentage of users that also purchased that item. For edge classification, as item is recommended by taking the most similar k users and then determining the average review of users that also purchased that item.

4.2 Matrix Factorization (MF) Model

The MF model learns user and item embeddings by factorizing the user-item interaction matrix. This model initializes latent user and item vectors that are learned through training by projecting the dot product of the two vectors to 1D space. The model then backpropagates with a loss function and updates the latent vectors such that user-item pairs have latent vectors that are relatively similar. The model is optimized using AdamW.

4.2.1 Architecture

- User and item embeddings are initialized uniformly.

- The dot product of user and item embeddings is passed through a linear.

- Sigmoid activation is used to introduce non-linearity and ensure output is in range [0,1], the same as normalized rating range.

The link prediction model was trained using binary cross entropy loss and the edge classification model was trained using mean squared error loss.

4.3 Graph Convolutional Network (GCN) Model

The GNN model uses graph convolutional layers to capture higher-order relationships between users and items. First, latent vectors are initialized for all users and items. Three propagation layers are used, allowing information from the 1^{st}, 2^{nd}, and 3^{rd} layers to influence the embeddings of the currently observed node.

With the bipartite graph structure, this allows for the updating of the current user node's latent vector with:

1. Items the user has purchased.
2. Users that share a purchased item with the current user.
3. All items purchased by the user set (2).

This also allows for the updating of the current item node's latent vector with:

1. All users that have purchased the current item
2. All items purchased by users that have purchased the current item
3. All users that have purchased the previous item set

The user-item pairs are then mapped to a 2D vector where each corresponding index in this vector corresponds to the likelihood of an edge at the current user-item pair index. This model also utilizes batch normalization and dropout to improve convergence time as well as decrease the likelihood of overfitting. All models were trained with Adam optimizer.

4.3.1 Architecture

- Three GCN layers.
- Linear layer with ReLU activation.
- Linear layer with Sigmoid activation to project to normalized prediction.

The link prediction model and evaluation GCN were trained using binary cross entropy loss. The edge classification model was trained using mean squared error loss.

4.4 Houser Model

Our HOUSER Model combines the optimized link prediction GCN and edge classification GCN outputs using the harmonic mean or an alpha ratio. These are described as follows

4.4.1 Harmonic Mean

The harmonic mean of two variables, x_1 and x_2 is defined as follows

$$\frac{2 * x_1 * x_2}{x_1 + x_2}$$

The harmonic mean is beneficial in this context as it rewards two variables that are high values and penalizes value discrepancies nonlinearly as well as binding output to the range [0,1]. If x_1 and x_2 are both close to 1, the harmonic mean will be close to 1. If either is low, the overall value will be relatively low and if both are low, it will be even lower. Applying this to our model, we see that in order to have a user buy an item and leave a good rating, we need link prediction and edge classification scores to both be close to 1 to make accurate predictions.

4.4.2 Alpha Combination

Again let x_1 and x_2 be variables. We learn the parameter α according to our pre-defined function

$$(1-\alpha) * x_1 + \alpha * x_2$$

The benefit of this is that α can be learned to optimize the test data, as long as it is not learned over the test data. We use only training data to learn this parameter and guarantee optimization.

5 Metrics

We use metrics commonly used in the analysis of recommender systems. These allow for a multi-faceted analysis and highlight what models excel at versus where they could improve. These are not used during edge classification as they are not relevant. To clarify, when we say relevant items, we mean the number of items the user purchased in the testing set. Additionally, when we say top K, we mean the top K highest-rated items from the model predictions.

5.1 Recall@K

This metric measures the proportion of correctly identified relevant items in the top K recommendations out of the total number of relevant items in the dataset. This is defined as:

$$\frac{\text{Number of relevant items in top K}}{\text{Total number of relevant items}}$$

This metric is beneficial in determining the proportion of your items in the top K predicted items compared to the total number of relevant items. If there are many relevant items, K must be large to attain high accuracy. The range for this metric is 0-1.

5.2 Precision@K

This metric measures the ratio of correctly identified relevant items within the total recommended items inside the list of the top K items. This is defined as:

$$\frac{\text{Number of relevant items in top K}}{\text{K}}$$

This metric is beneficial in determining the proportion of your items in the top K predicted items that are relevant items. If there are few total relevant items, K must be very small to attain high accuracy. The range for this metric is 0-1.

5.3 F1

This metric is a combination of Precision@K and Recall@K using harmonic mean. The harmonic mean nature makes sure if either Precision or Recall has a really high value, then it does not dominate the score. F1 Score has a high value when both precision and recall values are close to 1. This metric is influenced by the choice of K therefore K must be selected to optimize both Recall@K and Precision@K. The range for this metric is 0-1.

5.4 AUC

The AUC (area under curve) metric is defined as the area under the ROC (receiver-operating characteristic curve). The ROC curve is drawn by calculating the true positive rate (TPR) and false positive rate (FPR) at every possible threshold (in practice, at selected intervals), then graphing TPR over FPR. A perfect model, which at some threshold has a TPR of 1.0 and a FPR of 0.0, can be represented by either a point at (0, 1) if all other thresholds are ignored, or by the following:

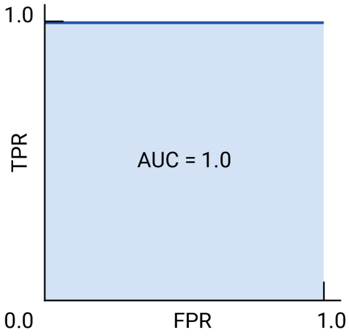

Figure 4: ROC and AUC of a hypothetical perfect model.

5.5 MRR

The MRR (mean reciprocal rank) evaluates how quickly a recommender system can show the first relevant item in the top-K results. To compute the MRR for a user, we look over the top-K items and take the reciprocals rank of the first observed relevant items. The range is 0-1. This is computed as

$$\frac{1}{rank_i}$$

where i is the first observed index of a relevant item in the predictions.

6 Methods

First, the Heuristic Model, Matrix Factorization Model, and GCN models are trained for edge classification and link prediction. Results for link prediction are shown in Figure 5 and results for edge classification are shown in Figure 6.

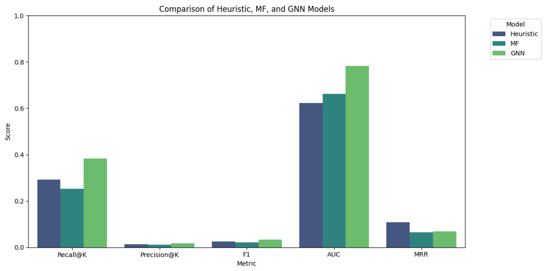

Figure 5: Link Prediction Results

We can safely determine that the GCN models outperform the others and should be used for HOUSER. This is commonly understood as these models are better at considering structural relationships [8] .

Now, we implement an evaluation GCN model to compare against the HOUSER

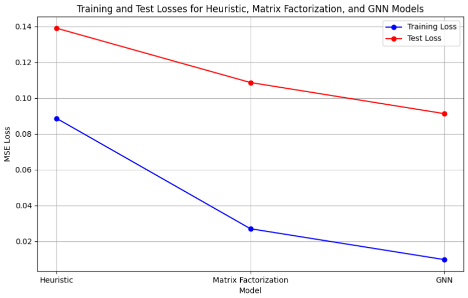

Figure 6: Edge Classification Results

model. The data is reprocessed so that edges correspond to user-item reviews with four or five stars. The evaluation GCN model is trained on this data, which contains the same user-item edges as in the original data, but labels corresponding only to positive reviews.

7 Discussion

7.1 Results

To test, we analyze performance over the testing data for HOUSER and the evaluation GCN. For both model outputs, predictions are normalized to ensure the AUC is a fair approximation for both. For HOUSER, we round our edge prediction scores to binary values to determine if a predicted rating is positive or negative and normalize the output of models before passing to the combination function. Accuracies are compared using the metrics described in (5). The accuracy comparison for Harmonic HOUSER (HH), Alpha Houser (AH), and the evaluation GCN (EGCN) are shown in Figure 7. Exact values shown in Table 1.

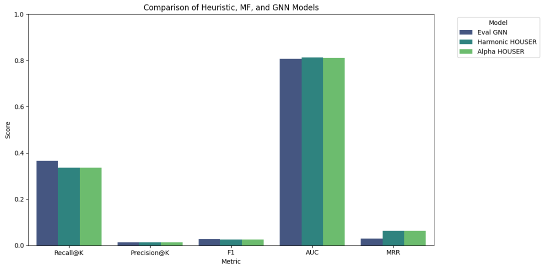

Figure 7: Harmonic HOUSER vs Alpha Houser vs Evaluation GCN

Table 1: Performance Metrics

Model	R@K	P@K	F1	AUC	MRR
AH	0.337	0.013	0.025	0.811	0.064
HH	0.335	0.013	0.025	0.813	0.063
EGCN	0.366	0.014	0.027	0.806	0.029

7.2 Analysis

First, we notice that Precision@K and AUC scores are not statistically different. We observe slight increases in AUC scores and slight decreases in Precision@K for the HOUSER models. However, we observe about an 8% reduction in Recall@K accuracy for the HOUSER models. We reason that this is because some structural representation of the graph is lost when combining results for edge representation and link prediction as the embedded vectors are necessarily distinct for each model. More noticeably, we observe about a 220% increase in Mean Reciprocal Rank (MRR) for both HOUSER models compared with the evaluation GCN model. We reason when relevant items are predicted by HOUSER, they appear, on average, 2.2x higher in the top-K item list. Hence, for example, on average, an item in the top-K item list may appear at index 3 in the HOUSER output whereas it would appear around index 7 in the evaluation GCN output, assuming k ¿ 7. We argue that this metric is very beneficial in recommendations as this would lead to greater retained user engagement as the items they want and will like post-purchase appear to them earlier. We argue this significant increase trumps the slight observed decrease in Recall@K.

We argue that this improvement in MRR is sufficient to determine that this model is more beneficial for user post-purchase satisfaction as items that the user will buy and be happy with post-purchase appear earlier in HOUSER predictions. We argue that the reduction in Recall@K is too insignificant to discredit the improvement in MRR. This is because when relevant items appear in the top-K recommendations they are significantly more beneficial for user satisfaction, even though relevant items are slightly less likely to appear in the top-K recommendations.

We note that the computational efficiency for HOUSER slightly supersedes that of the evaluation GCN as it must run two models and combine outputs. We see that the HOUSER processes data at about 35 iterations per second while the evaluation GCN processes data at around 63 iterations per second, resulting in around a 44% slowdown in the HOUSER model compared with the evaluation GCN model. We recognize this disadvantage and hope to improve this in future work.

8 Future Work

In the future, we hope to optimize for pre-purchase user satisfaction as well by recommending a diverse selection. We also hope to improve the computational efficiency of the model through a fully combined link-prediction and evaluation model capable of outputting predicted user ratings as well as purchase probability. Further, we hope to further optimize HOUSER to improve Recall@K by implementing faster GCN models such as LightGCN [1].

Further, we recognize that industry practices for large-scale commerce utilize item-item collaborative filtering for faster recommendations [6]. While these are computationally efficient, they fail to consider user satisfaction in any capacity. We hope to improve the efficiency of HOUSER to compete with current industry practices while increasing user satisfaction.

We also acknowledge that HOUSER will suffer heavily from the cold start problem as it relies heavily on user-item reviews. While we fail to address this in the paper, we encourage active research in this area or the application of existing solutions [7]. Due to time and computational restraints, we were unable to explore these in the paper.

9 Conclusion

In this project, we implemented and compared three recommender system models: heuristic, matrix factorization, and graph neural network for the tasks of link prediction and edge classification. We combined these metrics to produce HOUSER, a model optimizing user purchase probability and user post-purchase satisfaction. We trained an evaluation GCN to compare against this HOUSER model. We improved MRR metrics but experienced slightly worse computational efficiency and Recall@K. We hope to continue this work in the future, improve Recall@K, and increase the diversity of recommendations. All code is available on GitHub.

References

[1] Xiangnan He et al. "Lightgcn". In: *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (July 2020), pp. 639–648. DOI: 10.1145/3397271.3401063.

[2] XINYUE HE, QI LIU, and SUNHO JUNG. In: *The impact of types of recommendation system on User Satisfaction: A moderated mediation analysis* (Jan. 2024). DOI: 10.20944/preprints202401.1570.v1.

[3] Pan Li et al. "Purs: Personalized unexpected recommender system for improving user satisfaction". In: *Fourteenth ACM Conference on Recommender Systems* (Sept. 2020), pp. 279–288. DOI: 10.1145/3383313.3412238.

[4] Ting-Peng Liang, Hung-Jen Lai, and Yi-Cheng Ku. "Personalized content recommendation and user satisfaction: Theoretical synthesis and empirical findings". In: *Journal of Management Information Systems* 23.3 (Dec. 2006), pp. 45–70. DOI: 10.2753/mis0742-1222230303.

[5] Xinyu Lin et al. "Bridging items and language: A transition paradigm for large language model-based recommendation". In: *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (Aug. 2024), pp. 1816–1826. DOI: 10.1145/3637528.3671884.

[6] G. Linden, B. Smith, and J. York. "Amazon.com recommendations: Item-to-item collaborative filtering". In: *IEEE Internet Computing* 7.1 (Jan. 2003), pp. 76–80. DOI: 10.1109/mic.2003.1167344.

[7] Antiopi Panteli and Basilis Boutsinas. "Addressing the cold-start problem in recommender systems based on frequent patterns". In: *Algorithms* 16.4 (Mar. 2023), p. 182. DOI: 10.3390/a16040182.

[8] Jun Yi et al. In: *Graph neural network-based collaborative filtering for recom-*

mendation systems (2025). DOI: 10.2139/ssrn.5143186.

[9] Fatima Ezzahra Zaizi, Sara Qassimi, and Said Rakrak. “Multi-objective optimization with Recommender Systems: A systematic review”. In: *Information Systems* 117 (July 2023), p. 102233. DOI: 10.1016/j.is.2023.102233.

[10] Lei Zhang et al. “An indexed set representation based multi-objective evolutionary approach for mining diversified top-k high utility patterns”. In: *Engineering Applications of Artificial Intelligence* 77 (Jan. 2019), pp. 9–20. DOI: 10.1016/j.engappai.2018.09.009.