MULTI-FACETED TRUST BASED RECOMMENDATION SYSTEM

Anonymous authors
Paper under double-blind review

ABSTRACT

Recommendation systems play a decisive role in the choices we make on the internet. They seek to tailor decisions to a user. This makes trust a very important factor in recommendation systems, since it is believed that users are similar to the people that they trust, and hence will make similar choices to those users. Trust and its effects on the choices people make have been widely studied in the context of collaborative recommendation systems. It is understood that trust is not a single-faceted entity but can vary contextually. Recent research in the domain of trust based recommendation systems has shown that taking into account the facets of trust greatly improves the quality of recommendations (Mauro et al., 2019; Fang et al., 2015). We propose a recommendation system that takes multiple facets of trust into account while looking at how suitable a product might be for a particular user. The architecture proposed for this Multi-Faceted Trust Based Recommender (MFTBR) allows for extensibility - new facets of trust can be added without much effort - and dynamicity - trust facets are not weighed arbitrarily. Instead, the weights are optimised for the best result via a neural network. The trust facets considered here are local, global and category-wise trust. MFTBR performs significantly better than basic collaborative filtering - U2UCF (C. Desrosiers, 2011), as well as some established models in the domain of social and trust based recommendation systems - MTR (Mauro et al., 2019) and SocialFD (Yu et al., 2017). Thus, our model provides a better approximation of real-life recommendations, taking into account not only the impact of trust on recommendation, but the context in which trust is established.

1 INTRODUCTION

Recommendation systems are widely used in a variety of domains like product recommenders on e-commerce websites, playlist generators on movie and music streaming websites, and content recommenders for social media and networking websites. In e-commerce websites recommendation systems advertise to the user options for their next purchases that might be to their liking. To accurately predict the users' interests, they often look into what similar users have bought and rated highly. In cases like these, it is often beneficial to look beyond Collaborative Filtering, which looks at similarities between users to recommend products. While Collaborative Filtering achieves good results, it suffers from a number of problems in real world scenarios. One of these is the cold-start problem, i.e., Collaborative Filtering methods can't accurately predict items for new users, or recommend new items to users.

Recent recommendation systems have incorporated information on social trust from social networks, either implicitly inferred or explicitly stated by the users to combat the cold-start user problem, but these methods often treat trust as one-dimensional. It is widely believed in trust network analysis, that trust is multi-faceted and context-dependent, i.e., a user trusts different people in the context of different items/kinds of items.

Consider an example wherein one user places trust in another specifically in the context of gardening tools, but that trust does not extend to TV shows. To take another scenario, it is possible that a user establishes a trust relationship with another at a certain time, but this trust fades over time as the interests and opinions of the two users diverge.

From this we might conclude that a new recommendation algorithm is required – one that takes into account the multiple facets of trust. Here, we attempt to utilise 3 facets of trust, viz, local trust, category-based global trust and global review-feedback based trust, along with the metric of user-user similarity to better recommend products to users. The proposed model is a combination of trust-based techniques and Collaborative Filtering that utilises user-user trust facets that are either explicitly stated or derived/approximated from other metrics.

2 DATASET

The Rich Epinions Dataset (RED) [1] (Meyffret et al., 2014) is an SQL dump that contains data that was extracted from the Epinions website in June 2011. This section explains the details of the website, followed by the structure of the Rich Epinions Dataset.

2.1 EPINIONS WEBSITE

The Epinions website was a consumer reviews platform established in 1999 that allowed users to rate and review items. The items are assigned categories that are arranged in a tree-like structure, where each node is a category that might have multiple sub-categories.
A particular Epinions user could also add other users to their "Web of Trust"[2], which allowed users to curate a list of the people whose opinions they trusted and didn't trust on the website, by adding them to either a "Trust List" or a "Block List". Trusted users' reviews were boosted by the website, while distrusted users were penalised by the algorithm.
Users wrote reviews and accompanied them with ratings between 1 and 5. These reviews in turn could be rated by other users (for example, "Somewhat Helpful", "Helpful", "Very Helpful", etc.). Very short reviews, called "Express Reviews" on the site, could only be assigned "Show" or "Don't Show" tags.
Epinions defined 4 kinds of users[3]:

- **Category leads** were tasked with overseeing a particular category. This included making sure that there was high-quality review coverage of key products in their category and assisting with the selection of Top Reviewers and Advisors for their category.
- **Top reviewers**, write reviews in their category of expertise which are highly rated.
- **Advisors** rated reviews in their category and provided constructive feedback via comments to reviewers on how to improve content quality.
- **Regular users** reviewed items, rated reviews and could trust or block other users.

Users whose reviews had been visited the most number of times were deemed "popular users", and were given ranks.

2.2 RICH EPINIONS DATASET

Figure 2.2 shows the structure of the dataset. Each table in the figure corresponds to a table in the database and each row in the tables corresponds to fields in the respective table in the database. Directed lines denote foreign key references and their corresponding primary keys.

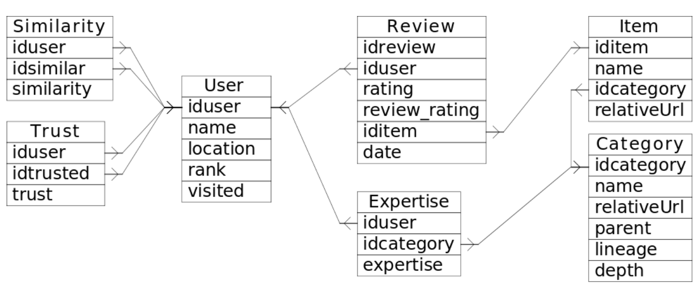

Figure 1: Structure of the Rich Epinions Dataset

1,13,629 users have at least one rating, 47,522 users have at least one trust relation and 21,910 users have at least one review, one trust relation and one computed similarity.
In the User table, 73.4% of the "location" fields are left blank, and ranks are mentioned for the first 1000 users. The Category table has 29 "main" parent categories, while the other 558 categories are

[1] https://projet.liris.cnrs.fr/red/
[2] https://web.archive.org/web/20090420090156/http://www.epinions.com/help/faq/?show=faq_wot
[3] https://web.archive.org/web/20090430060527/http://www.epinions.com/help/faq/show_ faq_recognition

sub-categories of the main categories. The Expertise table provides category-wise details of the Category Leads, Top Reviewers and Advisors. The Trust table has only positive trust statements, so there's no data on whom users distrust.

Table 1: RED statistics

Table	**Number of Entries**
User	131228
Category	587
Expertise	556
Reviews	1127673
Similarity	3689606
Trust	317755

3 RELATED WORK

3.1 EXPLICIT TRUST IN RECOMMENDATION SYSTEMS

Massa & Avesani (2007) and Massa & Bhattacharjee (2004) stated that Collaborative Filtering (CF) techniques might be lacking in cases where the total number of items is very large while the number of items rated by each user is small. In such cases, similarity metrics between two users (like the Pearson's Coefficient) can usually be computed for a very small portion of the user base, and is based on a small number of overlapping items - producing a noisy and unreliable recommendation metric. In order to overcome this, they suggested using explicit trust in conjunction with trust propagation whenever available, in order to better represent the users in the user base. Jamali & Ester (2010), who developed the SocialMF algorithm, leveraged the added insight that explicit trust gave them to improve matrix factorisation techniques. They made the features of every user dependent on the feature vectors of their direct neighbors in the social network. The latent features of users indirectly connected in the social network are thus dependent on each other, which results in the propagation of explicit trust. Yu et al. (2017) developed SocialFD, which also attempts to optimise the social matrix factorisation problem, by classifying items into two buckets: "like" and "dislike". During the training stage, constraints are imposed to guarantee that users are spatially closer to their friends and their "liked" items and further from their "disliked" items. If user u likes item i and has a friend user k, SocialFD not only pulls u and i closer but also pulls k and i, propagating trust.

3.2 IMPLICIT TRUST IN RECOMMENDATION SYSTEMS

Guo et al. (2014) explained that explicit trust could be noisy in terms of user preference in cases where trusted friends have different tastes in certain areas. They state that most datasets with explicit trust information are usually very sparse - the number of trust statements are much smaller than the number of ratings. Although trust propagation can alleviate this issue to some extent, this might create more noise in the dataset. They stress the importance of implicit trust in recommendation systems, which is usually inferred from user behaviour as well as context. Oard et al. (1998) explored various sources of implicit feedback as well as two methods of utilising this feedback. The first method is an approach that converts a user's explicit feedback into ratings, and then learns from the user's past behaviour to predict future ratings for products. The second seeks to capture the user's preferences without converting their feedback into ratings. Fazeli et al. (2014) looked at both implicit and explicit trust in the context of matrix factorisation recommendation techniques and sought to explore the question of if, in social matrix factorisation, implicit trust can be a replacement for explicit trust when explicit trust relations are not available. They found that the algorithm SocialMF (Jamali & Ester, 2010) when trained using implicit trust information performed comparably to when it was trained with explicit trust. Kim & Kim (2012) derived implicit trust relationships between users from users' tagging information. They utilised Kullback-Leibler divergence (Kullback & Leibler, 1951) to extract implicit trust relationships.

3.3 MULTI-FACETED TRUST IN RECOMMENDATION SYSTEMS

Fang et al. (2015) proposed a framework that took interpersonal and impersonal aspects of trust into account. Four interpersonal facets (benevolence, competence, integrity and predictability) were computationally modeled based on users' past ratings, and considered along with impersonal aspects (for example, degree centrality). Two logistic regression models were developed and trained by accommodating these factors and then applied to predict continuous values of users' trust and distrust. The trust and distrust values were applied to 3 trust-aware algorithms - TidalTrust (Golbeck, 2006), Merge (Guo et al., 2012) and SocialMF (Jamali & Ester, 2010). MTR, developed by Mauro et al. (2019) fused 3 facets of trust in order to create a trust metric (t_{uvi}). They looked at whether user u could trust user v in the context of an item i. The facets considered were: social relation between u and v - derived from Jaccard's Similarity and explicit trust statements, feedback on v's profile, and the global feedback on v's review of item i. t_{uvi} is then combined with the similarity between u and v, determined by Pearson's Similarity and used to define v's influence on u in the context of item i ($infl_{uvi}$) in the system. This influence is used to predict the rating that u would give i, which is then used to determine if u should be recommended i.

4 MFTBR

4.1 OVERVIEW

Leveraging the information present in the publicly available Rich Epinions Dataset (Meyffret et al., 2014), we propose a model that utilises the information available as distinct trust facets, which are both local and global in nature. This results in a degree of personalisation to each user, while also considering global and category-wise factors that influence larger parts of the user base collectively.

4.2 ARCHITECTURE

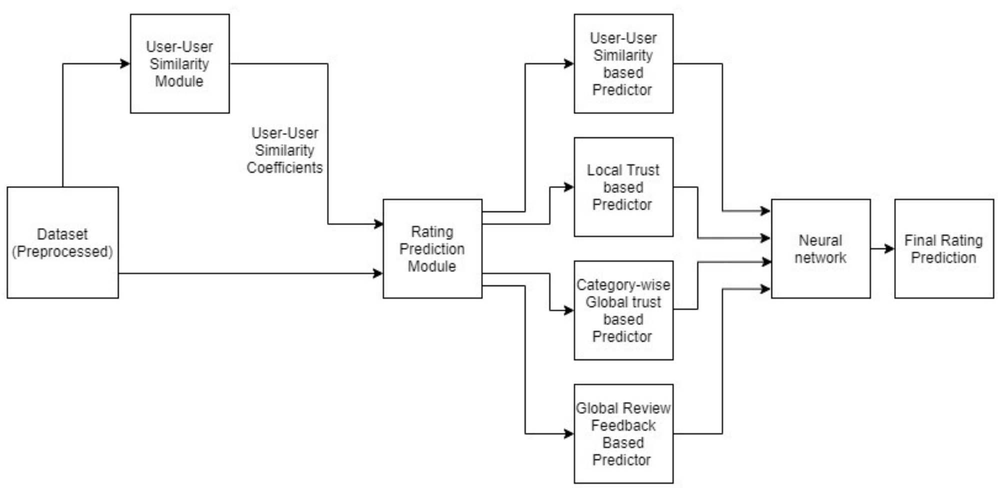

Figure 2: Modules of MFTBR

4.3 MODULES

The modules represented in the architecture diagram are further explained. In each of the predictor modules, we delve deeper into the trust facets considered.

4.3.1 USER-USER SIMILARITY MODULE

This module is responsible for calculating the Pearson's Similarity between each pair of users. Pearson's correlation coefficient is used to quantify similarity between each pair of users, and is defined as:

$$PC(u,v) = \frac{\sum_{i \in I_{uv}} (r_{ui} - \overline{r}_u)(r_{vi} - \overline{r}_v)}{\sqrt{\sum_{i \in I_{uv}} (r_{ui} - \overline{r}_u)^2 \sum_{i \in I_{uv}} (r_{vi} - \overline{r}_v)^2}} \tag{1}$$

Here, I_{uv} is the set of items that are rated by users u and v. $R \in \mathbb{R}^{UXI}$ the user-item rating matrix, where each value r_{ui} is the rating given by user u to item i. The sign of the correlation coefficient indicates if the users are directly or inversely related, while its magnitude which $\in$ [0, 1] represents the strength of the correlation between the users. It can be noted that this formula is resistant to user bias: different users have differing levels of generosity when rating items. By subtracting the average user rating from the rating for each item, it is ensured that the user's bias does not play a role in determining the similarity between any two users.

4.3.2 User-User Similarity Based Predictor

All similarity values previously calculated by the User-User Similarity Module are stored, and used here to predict the rating for a user-product combination. As in User-to-User Collaborative Filtering, we assume that if user u and user v rated products similarly, then the rating that u might give to a new item i would be similar to the rating that v gave i(C. Desrosiers, 2011). Following from 1, U2UCF estimates u's rating of i ($\hat{r_{ui}}$) as:

$$\widehat{r_{ui}} = \overline{r}_u + \frac{\sum\limits_{v \in N_i(u)} PC_{uv}(r_{vi} - \overline{r}_v)}{\sum\limits_{v \in N_i(u)} |PC_{uv}|} \tag{2}$$

Here, N_i refers to the set of neighbours of u who have rated item i. PC_{uv} refers to the similarity between u and v.

4.3.3 Local Trust Based Predictor

The Local trust facet is extracted from the explicit user-user trust statements available in the dataset. This local trust is asymmetric. "User A trusts User B" does not imply "User B trusts User A". For predicting the rating of user u for item i, we use the following formula:

$$\widehat{r_{ui}} = \overline{r}_u + \frac{\sum\limits_{v \in T(u)} \sigma(v,i)(r_{vi} - \overline{r}_v)}{\sum\limits_{v \in T(u)} \sigma(v,i)} \tag{3}$$

Here, $T(u)$ refers to the set users that u has explicitly trusted. $\sigma(v, i)$ is 1 if user v has rated item i, and 0 if v has not rated i.

4.3.4 Category-wise Global Trust Based Predictor

Global trust is the trust that a large number of users, or communities as a whole have on a user. Global trust is available in the dataset in the form of ranks of users, as well as expertise levels held by them. Here, the influence exerted by an expert is considered to be limited to their category of expertise. MFTBR only considers the expertise of a user in this trust facet, since ranks are only present for the first 1000 users (0.883% of the user base) in the Epinions dataset. The "Expertise" table has details of the list of users who are experts in each category. Experts are further divided into 3 levels as detailed in Section 2.1. Each category may be associated with a parent category. The experts from the parent category, if applicable, also affect the predictions.

$$\widehat{r_{ui}} = \overline{r}_u + \frac{w_1 * \sum\limits_{v \in E_1} \sigma(v,i)\alpha(v,C_1)(r_{vi} - \overline{r}_v)}{\sum\limits_{v \in E_1} \sigma(v,C_1)\alpha(v,i)} + \frac{w_2 * \sum\limits_{x \in E_2} \sigma(x,i)\alpha(x,C_2)(r_{xi} - \overline{r}_x)}{\sum\limits_{x \in E_2} \sigma(x,C_2)\alpha(x,i)} \tag{4}$$

Here, C_1 is the category of item i, C_2 is C_1's parent category and E_1 and E_2 are the sets of experts in categories C_1 and C_2 respectively. $\alpha(v, C_n)$ returns the weight awarded to user v's rating based on the level of expert that they are in category C_n. w_1 and w_2 are the weights assigned to the current category and the parent category respectively. Various combinations of w_1 and w_2 have been considered, and are detailed in Table 6.

4.3.5 Global Review Feedback based Predictor

The Global Review Feedback based Predictor Class looks at the tags associated with each review, which in the Epinions dataset consist of "Very Helpful", "Helpful", "Somewhat Helpful", "Not

Rated" and "Show". Of these ratings, "Very Helpful", "Helpful" and "Somewhat Helpful" tags are useful to this facet, and are assigned weights. For predicting the rating, we use

$$\widehat{r_{ui}} = \overline{r}_u + \frac{\sum\limits_{v \in R} \beta(v,i)(r_{vi} - \overline{r}_v)}{\sum\limits_{v \in R} \beta(v,i)} \tag{5}$$

Here, R is the subset of the review matrix with ratings for item i that have been rated by users and $\beta(v,i)$ returns a weight for each rating given by user v to item i based on the feedback that the review has received.

4.3.6 NEURAL NETWORK

The neural network is responsible for combining predictions of Trust facets and making the final prediction. This brings a dynamic aspect to the recommendation system, by adjusting the weights as required to make optimal rating predictions. The architectural details of the network in terms of the number of layers used and number of nodes at each layer are represented diagrammatically in Figure 4.3.6. The optimiser and activation function are the common through all the layers and mentioned in Table 2 along with other design choices. The neural network was trained according to the specifications mentioned in Table 3.

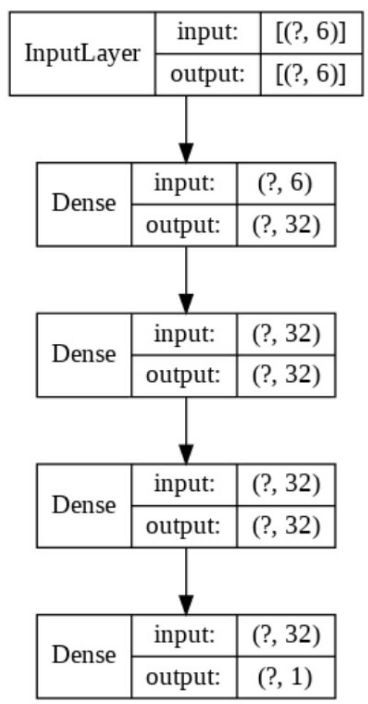

Figure 3: Neural Network Architecture

Table 2: Neural Network Details

Detail	Value
Optimiser	Adam
Loss Metric	Mean Squared Error
Error Metric	Mean Absolute Error
Activation Function	Rectified Linear Unit

Table 3: Learning Rate and corresponding number of epochs

Learning rate	Number of Epochs
0.001	200
0.0005	100
0.00025	20
0.00005	10
0.00001	10

5 TEST METHODOLOGY

To evaluate the MFTBR model, its results were compared against a number of established baselines, namely, User-User Collaborative Filtering (C. Desrosiers, 2011), SocialMF[4] (Jamali & Ester, 2010), SocialFD[5] (Yu et al., 2017) and MTR (Mauro et al., 2019).
To evaluate each of these algorithms, the dataset was divided into 5 folds and tested with each of the folds as the test fold, as is the case with k-fold cross validation. Metrics commonly used to assess recommendation systems were chosen to evaluate the performance of MFTBR against the baselines:

1. Mean Absolute Error (MAE)
2. Root Mean Square Error (RMSE)
3. Rating Coverage (RC)

The average results over all the folds are presented in Table 4.

6 RESULTS

The performance of the baselines and MFTBR were evaluated based on the test metrics described. Table 4 summarises the results. The rating coverage of the model was improved by considering average user rating and average product rating values - in addition to the facet predictions - as input to the neural network. This resulted in complete rating coverage, enabling the model to recommend products to infrequent users as well as new users, eliminating the cold start problem. The rating coverage values after each iteration of the model, are documented in Table 5.

Table 4: Comparison of the performance of MFTBR against baselines

Baseline	MAE	RMSE	RC
U2UCF	0.90856	1.20078	0.9444
SocialMF	0.89809	1.20276	1
SocialFD	0.86120	1.12772	1
MTR	0.86336	1.14640	0.9442
MFTBR	0.83230	1.07145	1

Table 5: Rating Coverage with each iteration of MFTBR

Iteration of MFTBR	RC
Iter 1: Four Factors	0.5799
Iter 2: With Average User and Average Product Ratings	0.8020
Iter 3: Post addition of Neural Network	1

[4]https://github.com/Coder-Yu/RecQ/blob/master/algorithm/rating/SocialMF.py
[5]https://github.com/Coder-Yu/RecQ/blob/master/algorithm/rating/SocialFD.py

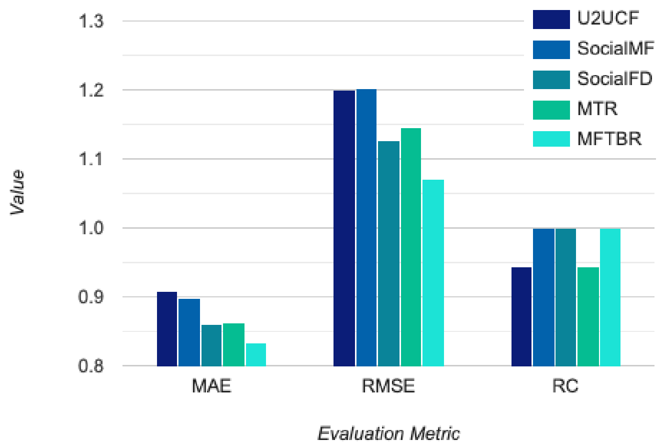

Figure 4: Comparison of the performance of MFTBR against baselines

7 Conclusion

In this work, we present a recommendation system that takes into account the trust established between users while incorporating the multi-faceted nature of this trust into the recommendations. Including both explicit and implicit sources of trust via local and global facets enables us to make unbiased predictions that better resemble the real world scenario. MFTBR generates viable predictions even when data is sparse, by relying on multiple facets some of which are independent of the user in question, and making appropriate imputations where necessary. This eliminates the cold start problem for both users as well as items.

MFTBR provides a modular architecture for trust-based recommendation systems that allows the integration or removal of any number of trust facets. To arrive at the final set of facets to be included in the model, it was necessary to experiment with different facet combinations. This process was made significantly easier by the independent processing of the trust facets. Since the trust facet weights are not decided through an arbitrary heuristic, but are computed through the neural network for optimal results, a measure of dynamicity is introduced which is lacking in previous models. Moreover, trust is a measure that cannot be quantified by a static set of facets because it relies on external factors that are both source dependent and constantly evolving. Given this, the extensible and dynamic nature of MFTBR ensure that it remains relevant and accurate in various scenarios.

It is due to the above outlined features that MFTBR was able to outperform the baselines across all evaluation metrics as detailed in Table 4.

Future work could include extending this approach into a Top-K recommendation system which would provide more metrics to measure the results against. In case more facets are added, it would be beneficial to ensure that they are not correlated with each other. Feature engineering methods could be applied to remove redundant facets. Also, this method is yet to be evaluated on other datasets, which might provide some insight into how well it would work in different environments - like those with different subsets of reviews.

References

G. Karypis C. Desrosiers. A comprehensive survey of neighborhoodbased recommendation methods. In P.B. Kantor L. Rokach, B. Shapira (ed.), *Recommender Systems Handbook*, pp. 107–144. Springer, 2011.

Hui Fang, Guibing Guo, and Jie Zhang. Multi-faceted trust and distrust prediction for recommender systems. *Decision Support Systems*, 71:37–47, 2015.

Soude Fazeli, Babak Loni, Alejandro Bellogin, Hendrik Drachsler, and Peter Sloep. Implicit vs. explicit trust in social matrix factorization. In *Proceedings of the 8th ACM Conference on Recommender systems*, pp. 317–320, 2014.

Jennifer Golbeck. Generating predictive movie recommendations from trust in social networks. In *International Conference on Trust Management*, pp. 93–104. Springer, 2006.

Guibing Guo, Jie Zhang, and Daniel Thalmann. A simple but effective method to incorporate trusted neighbors in recommender systems. In *International conference on user modeling, adaptation, and personalization*, pp. 114–125. Springer, 2012.

Guibing Guo, Jie Zhang, Daniel Thalmann, Anirban Basu, and Neil Yorke-Smith. From ratings to trust: An empirical study of implicit trust in recommender systems. In *Proceedings of the 29th Annual ACM Symposium on Applied Computing*, SAC '14, pp. 248–253, New York, NY, USA, 2014. Association for Computing Machinery. ISBN 9781450324694. doi: 10.1145/2554850.2554878. URL https://doi.org/10.1145/2554850.2554878.

Mohsen Jamali and Martin Ester. A matrix factorization technique with trust propagation for recommendation in social networks. In *Proceedings of the Fourth ACM Conference on Recommender Systems*, RecSys '10, pp. 135–142, New York, NY, USA, 2010. Association for Computing Machinery. ISBN 9781605589060. doi: 10.1145/1864708.1864736. URL https://doi.org/10.1145/1864708.1864736.

Hyunwoo Kim and H. Kim. Improving recommendation based on implicit trust relationships from tags. 2012.

S. Kullback and R. A. Leibler. On information and sufficiency. *Ann. Math. Statist.*, 22(1):79–86, 03 1951. doi: 10.1214/aoms/1177729694. URL https://doi.org/10.1214/aoms/1177729694.

Paolo Massa and Paolo Avesani. Trust-aware recommender systems. In *Proceedings of the 2007 ACM Conference on Recommender Systems*, RecSys '07, pp. 17–24, New York, NY, USA, 2007. Association for Computing Machinery. ISBN 9781595937308. doi: 10.1145/1297231.1297235. URL https://doi.org/10.1145/1297231.1297235.

Paolo Massa and Bobby Bhattacharjee. Using trust in recommender systems: an experimental analysis. In *International conference on trust management*, pp. 221–235. Springer, 2004.

Noemi Mauro, Liliana Ardissono, and Zhongli Filippo Hu. Multi-faceted trust-based collaborative filtering. *Proceedings of the 27th ACM Conference on User Modeling, Adaptation and Personalization*, Jun 2019. doi: 10.1145/3320435.3320441. URL http://dx.doi.org/10.1145/3320435.3320441.

Simon Meyffret, Emmanuel Guillot, Lionel Médini, and Frederique Laforest. Red: a rich epinions dataset for recommender systems. 06 2014.

Douglas W Oard, Jinmook Kim, et al. Implicit feedback for recommender systems. In *Proceedings of the AAAI workshop on recommender systems*, volume 83. WoUongong, 1998.

Junliang Yu, Min Gao, Wenge Rong, Yuqi Song, and Qingyu Xiong. A social recommender based on factorization and distance metric learning. *IEEE Access*, PP, 10 2017. doi: 10.1109/ACCESS.2017.2762459.

A Appendix

The Epinions dataset provides information regarding the category of each product and the parent category for each category, if applicable. While experimenting with the Category-based Global Trust Predictor (Section 4.3.4), weights were assigned for the ratings given by experts of the immediate and parent categories.
Since it is possible to logically deduce that the immediate category rating should be weighted more than the parent category rating, the ratios chosen for experimentation were 100:0, 75:25, 70:30 and 50:50 (Immediate Category: Parent Category). Modifying the weights did not have a huge impact on the error of the model and these results are seen in Table 6.

Table 6: Impact of Category Weight Combinations on MFTBR

Immediate Category: Parent Category	**MAE**	**RMSE**
100:0	0.83324	1.07169
75:25	0.83279	1.07230
70:30	0.83426	1.07270
50:50	0.83091	1.07133