

UniWalk: Explainable and Accurate Recommendation for Rating and Network Data

Haekyu Park* Hyunsik Jeon* Junghwan Kim* Beunguk Ahn* U Kang*

Abstract

How can we leverage social network data and observed ratings to correctly recommend proper items and provide a persuasive explanation for the recommendations? Many online services provide social networks among users, and it is crucial to utilize social information since recommendation by a friend is more likely to grab attention than the one from a random user. Also, explaining why items are recommended is very important in encouraging the users' actions such as actual purchases. Exploiting both ratings and social graph for recommendation, however, is not trivial because of the heterogeneity of the data.

In this paper, we propose UniWalk, an explainable and accurate recommender system that exploits both social network and rating data. UniWalk combines both data into a unified graph, learns latent features of users and items, and recommends items to each user through the features. Importantly, it explains why items are recommended together with the recommendation results. Extensive experiments show that UniWalk provides the best explainability and achieves the state-of-the-art accuracy.

1 Introduction

Given rating and social network data, how can we recommend appropriate items convincingly? Recommending proper items is a crucial task benefiting both suppliers and consumers. Recommendation increases sales for suppliers by capturing latent demands and facilitates searches over massive items for consumers. Providing reasons behind a recommendation is much more important since it enriches user experiences and draws out user participation, which eventually enhances long-term performance of the recommender system.

Most recommender systems are based on rating data only. However, rating data are sparse and expensive to gather. Additional information is required to overcome the sparsity and enhance the recommendation results. One of the most prevalent additional infor-

mation accompanying e-commerce services is social network among users. Social influence is powerful in recommendation, since people trust recommendation through acquaintance more than that from an anonymous one. Social information also provides useful information on tastes of users according to homophily property [17]: neighbors in social networks are likely to share similar preferences. Consequently, it is essential to exploit social networks as well as rating data for explainable and accurate recommendations.

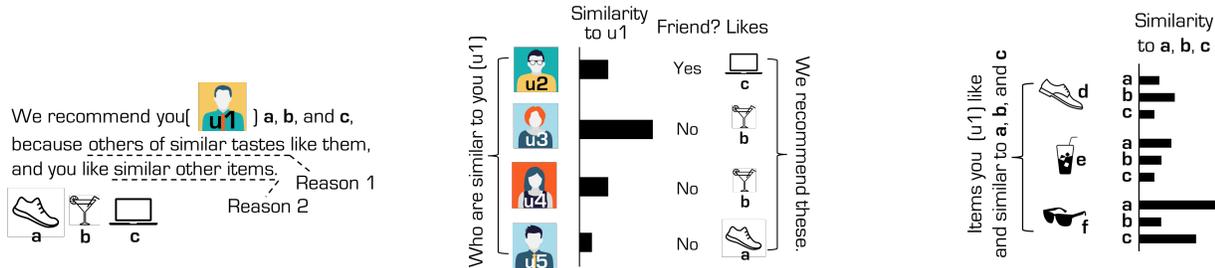
Leveraging both rating and social network data in recommendation and explanation, however, is challenging. Many multi-step relations such as friends of friends or friends' favorite items provide useful information on users' tastes, but there is no straightforward mechanism to translate the multi-step links into rating values. In addition, latent features represented by many collaborative filtering methods are no more than series of numbers to human. Existing methods suffer from those difficulties. For example, recommendation methods such as TrustSVD [6], ASMF and ARMF [13] use social networks partially. TrustSVD utilizes only direct friendships, and ASMF and ARMF use social graphs only to pick out potential items in a pre-processing stage. Also, those methods do not offer any explanations.

In this paper, we propose UniWalk, an explainable and accurate recommendation model that utilizes both ratings and graph data. UniWalk generates a unified graph describing both preference and user similarity. Then, it samples not only quantitative links representing rating data but also qualitative links indicating entities' similarity or dissimilarity from the unified graph with random walks. Next, UniWalk learns entities' latent features with a newly devised objective function consisting of a supervised term for the quantitative links and unsupervised terms for the qualitative links. UniWalk then recommends items of the highest predicted ratings and provides reasons of the recommendation. UniWalk provides the best explainability and accuracy among competitors, as shown in Table 1.

The main contributions of this paper are as follows.

- **Method.** We propose UniWalk, a novel method that exploits ratings and social graph for recom-

*Seoul National University, Seoul, Republic of Korea. {hkpark627, jeon185, kjh900809, elaborate, ukang}@snu.ac.kr



(a) We recommend items ($a, b, \text{ and } c$) to a target user (u_1) with two reasons (Figures 1b and 1c). (b) Reason1: Explain with similar other users ($u_2, u_3, u_4, \text{ and } u_5$) who are similar to the target user. (c) Reason2: Explain with similar items. The target user likes other similar items ($d, e, \text{ and } f$).

Figure 1: UniWalk explains its recommendation using two reasons. More details are described in Section 4.2.

Table 1: Comparison of UniWalk and other methods. UniWalk provides the best explainability and accuracy, and it is the only method that utilizes all types of links. Meta-explanation is sub-explanation for the main explanation.

Methods →		UniWalk (Proposed)	UCF	ICF	TrustSVD	MF
Explainability	Can explain	Yes	Yes	Yes	No	No
	Meta-explanation	Yes	No	No	No	No
	Using similar users	Yes	Yes	No	No	No
	Using friends	Yes	No	No	No	No
	Using similar items	Yes	No	Yes	No	No
Error	RMSE	Lowest (most accurate)	Low	Low	High	High
	MAE	Lowest (most accurate)	Low	Low	High	High
Utilization of links	Ratings	Yes	Yes	Yes	Yes	Yes
	Social links	Yes	No	No	Yes	No
	Multi-step links	Yes	No	No	No	No

recommendations. UniWalk leverages not only quantitative rating information but also qualitative similarity and dissimilarity information for better understanding of users and items.

- **Explainability.** UniWalk explains why items are recommended to a target user (Figure 1). UniWalk recommends items when similar users like the same items or the target user prefers other similar items. UniWalk also provides meta-explanations for better persuasiveness: why the other users and items are determined to be similar to the target user and the recommended item, respectively.
- **Accuracy.** UniWalk achieves the state-of-the-art accuracy for recommendation with ratings and a social network in terms of RMSE and MAE (Section 4.3).

The rest of this paper is developed as follows. We first explain backgrounds of UniWalk including matrix factorization and network embedding in Section 2. Then we propose our method UniWalk in Section 3. After presenting experimental results in Section 4, we list related works in Section 5 and conclude this paper in Section 6. The codes and datasets are available at <http://datalab.snu.ac.kr/uniwalk>.

2 Preliminary

In this section, we describe preliminaries on matrix factorization and network embedding.

2.1 Matrix Factorization with Bias Factors. Matrix factorization (MF) predicts unobserved ratings given observed ratings. MF predicts a rating of an item i given by a user u as $\hat{r}_{ui} = \mu + b_u + b_i + \mathbf{x}_u^T \mathbf{y}_i$, where μ is global average rating, b_u is u 's bias, b_i is i 's bias, \mathbf{x}_u is u 's vector, and \mathbf{y}_i is i 's vector. Biases indicate the average tendency of rating scores given by users or given to items. The bias terms are known to boost the performance of prediction accuracy compared to that by inner product term only.

The objective function is defined in Equation (2.1), where r_{ui} is an observed rating and K is a set of (user, item) pairs for which ratings are observed. The term $\lambda(b_u^2 + b_i^2 + \|\mathbf{x}_u\|^2 + \|\mathbf{y}_i\|^2)$ prevents overfitting by regularizing the magnitude of parameters, where the degree of regularization is controlled by the hyperparameter λ . The objective function is minimized by gradient descent.

$$(2.1) \quad L = \frac{1}{2} \sum_{(u,i) \in K} (r_{ui} - \mu - b_u - b_i - \mathbf{x}_u^T \mathbf{y}_i)^2 + \lambda(b_u^2 + b_i^2 + \|\mathbf{x}_u\|^2 + \|\mathbf{y}_i\|^2)$$

2.2 Network embedding. Network embedding represents each node as a vector that encodes structural information on the network. The task can be formalized as follows: given a graph $G = (V, E)$, find a function $f : V \rightarrow \mathbb{R}^d$ mapping each node $v \in V$ to a d -dimensional vector $f(v)$.

DeepWalk [20] is one of the most representative network embedding methods. It exploits Skipgram with negative sampling (SGNS) to learn node vectors. DeepWalk simulates multiple truncated random walks to apply SGNS on network data. For each node in the network, the model generates random walks of fixed length. Each step of random walk selects next node according to transition probabilities defined by weights on edges. The resulting walk is a sequence of nodes in which more strongly connected node pairs appear in close distance more frequently. Then, Skipgram defines ‘neighborhood’ where a pair of nodes co-occurs in the simulated random walk. Neighborhood node pair is defined by a window sliding on the node sequence. The center node in the window is defined as a *target*, and other nodes in the window are *neighbors* of the *target*. Pairs of co-occurring *targets* and *neighbors* are stored in a set \mathcal{D} . Co-occurrence statistics represents proximity between nodes in the graph, since nodes closer in the graph co-occur more frequently in the random walk.

If the model optimizes over only proximate pairs, the embedding would converge to a single point. SGNS model applies negative sampling to prevent this convergence of embedding process by adding distant pairs into the process. Negative sampling technique randomly selects pairs of nodes and pushes embeddings for those pairs apart from each other. For each proximate node pair $(v, w) \in \mathcal{D}$, n randomly selected nodes w_1, \dots, w_n form n distant node pairs $(v, w_1), \dots, (v, w_n)$.

3 Proposed Method

In this section, we describe UniWalk, our proposed method for rating prediction and explanation on its logic behind recommendations. We solve the following problems.

PROBLEM 1. (RECOMMENDATION WITH RATINGS AND A GRAPH) *Given an explicit feedback rating matrix $\mathbf{R} \in \mathbb{R}^{|U| \times |I|}$ and a social network $G = (V, E)$, predict a rating of an item i given by a user u which is denoted as \hat{r}_{ui} . Ideally, \hat{r}_{ui} is close to the corresponding observed rating $r_{ui} = \mathbf{R}_{ui}$.*

PROBLEM 2. (EXPLANATION OF RECOMMENDATION) *For a user u and predicted rating matrix $\hat{\mathbf{R}} \in \mathbb{R}^{|U| \times |I|}$, recommend items of the highest predicted ratings and explain why they are recommended to u .*

In the remaining part of this section, we first state challenges for explainable and accurate recommendation

and our main ideas to overcome the challenges in Section 3.1. Then, we describe our proposed method UniWalk in detail to solve Problem 1 in Section 3.2 and to solve Problem 2 in Section 3.3.

3.1 Challenges and Solutions. For explainable and accurate recommendation, we take both quantitative and qualitative relationships of users and items into account. Quantitative links represent rating scores, and qualitative links indicate similarity or dissimilarity among users and items. They are stated as follows.

DEFINITION 1. *Link types for our learning process.*

- *Quantitative or score links are edges between users and items that represent observed ratings.*
- *Qualitative links are relationships among users and items to represent similarity or dissimilarity among them. There are two types of qualitative links.*
 - *Similarity links are relationships among similar entities. They include friend links and multi-step links between similar entities such as items rated similarly by common users.*
 - *Dissimilarity links are multi-step links among dissimilar entities such as users and their friends’ unfavorable items.*

Modeling qualitative links is beneficial in two ways. First, the similarity links are used in explaining reasons of recommendation. We recommend items since “similar” other users like them, or the recommended items are “similar” to other observed preferred items. Second, the qualitative links lead to more accurate recommendation. The qualitative links contain additional information on users’ tastes and items’ properties and mitigate data-sparsity problem in rating and social data.

However, there are three key difficulties in using those qualitative links: (1) drawing out proper qualitative links spanning both ratings and a social network, (2) learning from the qualitative links in accordance with the score links, and (3) using qualitative links systematically in explanation. Most real-world graphs show the small-world phenomenon, meaning that the diameters are small, and thus random entities can be paired even with a small number of steps; sampling appropriate pairs of similar and dissimilar entities is essential for the learning process. In addition, qualitative links only imply possible similarities or dissimilarities in properties among users and items, while a quantitative link directly gives numerical values, which we predict.

We provide solutions to those challenges. UniWalk uses efficient random walk sampling to pick similar or dissimilar entities to tackle the first challenge. Proximate entity pairs in a simulated random walk are our desired qualitative connected pairs which potentially contain informative properties of entities. Random walk sampling process is efficient, because each sampled en-

tivity in the sequence of random walk forms multiple connections by being paired with multiple nearby entities. The second challenge is solved by our objective function that covers quantitative and qualitative links. The objective function combines supervised learning for quantitative links and unsupervised learning for qualitative ones in the same framework. We solve the last challenge by measuring similarities of the qualitative links and presenting the numerical values in explanation.

3.2 UniWalk for Rating Prediction. We present details of the process of UniWalk to solve Problem 1. UniWalk comprises three steps. First, it generates a unified graph to leverage rich relations among users and items from the heterogeneous input data. Next, UniWalk applies network embedding on the unified graph to learn bias and vector encoding preference or properties of users and items. Lastly, it calculates predicted ratings \hat{r}_{ui} for each item i and user u .

Step 1: Build a unified graph. In this step, we generate a weighted and undirected graph \tilde{G} that combines ratings and a social network whose weights of edges present similarity of entities. Each rating is represented by a user-item edge in \tilde{G} . Social links are described as user-user edges in \tilde{G} . High ratings convey similarities of the users and items, and friends' similarities are determined by a hyperparameter c . The process of generating $\tilde{G} = (\tilde{V}, \tilde{E})$ is as follows. First, we convert rating matrix \mathbf{R} into a graph structure. For a user u and an item i where a rating r_{ui} is observed, we add u and i into \tilde{V} and (u, i) into \tilde{E} . The added edge (u, i) has a weight of r_{ui} . Next, we add edges in a social network G into \tilde{G} . Weights of the added social edges are set to the hyperparameter c . Users and items are termed as *entities* in the unified graph.

DEFINITION 2. *Entity.*

Entities are users and items in the unified graph of ratings and a social network.

Step 2: Embed nodes of \tilde{G} . UniWalk applies network embedding on the unified graph \tilde{G} to learn biases and latent vectors of entities. There are three goals in our embedding method. First, we aim to reduce differences between observed ratings and predicted ratings corresponding to score links. Second, we increase inner products of vectors of two entities related by similarity links. Third, we decrease inner products of vectors of dissimilar entities connected by dissimilarity links.

We devise a loss function L consisting of terms (3.2) and (3.3). The terms implement the above three goals. Term (3.2) is aimed to minimize the difference between an observed rating r_{ui} and a predicted rating \hat{r}_{ui} for a user u and an item i . The left term in Equation (3.3) drives latent vectors \mathbf{z}_v and \mathbf{z}_w of similar entities v and

w to have a high inner product. The right term in Equation (3.3) is aimed to make latent vectors \mathbf{z}_v and \mathbf{z}_w of dissimilar entities v and w have a negatively high inner product. We call the term (3.2) as supervised term, the left term in Equation (3.3) as positive unsupervised term, and the right one in Equation (3.3) as negative unsupervised term. In the objective function L , a predicted rating \hat{r}_{ui} is calculated as $\mu + b_u + b_i + \mathbf{z}_u^T \mathbf{z}_i$, where μ is the average of rating values, b_v and \mathbf{z}_v are a bias factor and a latent vector of entity v , respectively. λ_b and λ_z are regularization parameters for bias factors and latent embedding vectors, respectively. α and β indicate weight of the positive unsupervised term and the negative unsupervised term, respectively. $\mathbf{b} \in \mathbb{R}^{|\tilde{V}|}$ is the bias vector of all entities, and $\mathbf{Z} \in \mathbb{R}^{|\tilde{V}| \times d}$ is the latent vector matrix of entities where d is the dimension of entities' vector \mathbf{z}_v .

$$(3.2) \quad L = \underbrace{\sum_{(u,i) \in \mathcal{D}^R} \frac{1}{2} (r_{ui} - \hat{r}_{ui})^2}_{\text{Supervised term}} + \underbrace{\frac{\lambda_b}{2} \|\mathbf{b}\|^2 + \frac{\lambda_z}{2} \|\mathbf{Z}\|_F^2}_{\text{Regularization term}}$$

$$(3.3) \quad + \underbrace{\alpha \cdot \sum_{(v,w) \in \mathcal{D}^+} -\mathbf{z}_v^T \mathbf{z}_w}_{\text{Positive unsupervised term}} + \underbrace{\beta \cdot \sum_{(v,w) \in \mathcal{D}^-} \mathbf{z}_v^T \mathbf{z}_w}_{\text{Negative unsupervised term}}$$

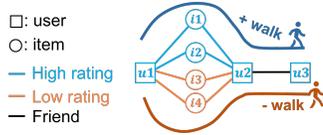
Step 2 mainly consists of two parts: sampling and optimization. We sample node sequences from the unified graph in the sampling phase. We extract entity pairs from the sequences and minimize L in the optimization phase.

In the sampling phase, we generate node sequences with the following random walks on the unified graph.

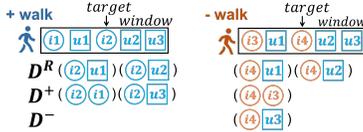
DEFINITION 3. *Random walks for sampling links.*

- *Positive random walk is a random walk whose transition probabilities are proportional to weights on edges.*
- *Negative random walk is a random walk whose transition probabilities are proportional to $(\min R + \max R - \text{weights on edges})$ for score links, and weights on edges for social links, where $\max R$ and $\min R$ are the maximum and the minimum ratings, respectively.*
- *Unweighted random walk is a random walk whose transition probabilities are uniform in all edges.*

The positive random walk generates sequences of similar entities with its transition rule, since high weight of an edge indicates high similarity between the nodes. The negative random walk generates sequence of entities of two groups: users who dislike similar items and the items not preferred by the user group. The groups are dissimilar to each other, and members in a group are similar. The unweighted walk has the effect of sampling



(a) Sampling phase to sample node sequences with random walks. The positive walk (+walk) is likely to walk on the blue high rating links and the black social link. The negative walk (-walk) is likely to walk on the orange low rating links and the black friend link. Unweighted walk is omitted, since it samples sequences randomly.



(b) Extracting node pairs in the optimization phase. \mathcal{D}^R , \mathcal{D}^+ , and \mathcal{D}^- are for entities linked by ratings, similar entities, and dissimilar entities, respectively.

Figure 2: An example of step 2 in UniWalk. (a) shows the sampling phase, and (b) shows extracting pairs in the optimization phase.

more substantial nodes, because higher degree nodes, which are active users or items, are sampled more. Figure 2a shows an example of the sampling phase.

In the optimization phase, we extract node pairs from the sampled node sequences and learn features of the nodes. We define sets of the node pairs as follows.

DEFINITION 4. *Entity pair sets.*

- \mathcal{D}^R is a multiset of node pairs connected by score links for the supervised term (Equation (3.2)).
- \mathcal{D}^+ is a multiset of node pairs connected by similarity links, which is used in the positive unsupervised term (left term in Equation (3.3)).
- \mathcal{D}^- is a multiset of dissimilar node pairs for the negative unsupervised term (right term in Equation (3.3)).

We extract node pairs from node sequences generated by the walks and assign the pairs into the entity pair sets according to the entities' relationships. In a sliding window on each sequence, we appoint a center as *target* and other members as *neighbors*. We pair up the *target* and each *neighbor*. We assign the pairs into \mathcal{D}^R if they are linked by score links, into \mathcal{D}^+ if the paired entities are similar and into \mathcal{D}^- if the paired ones are dissimilar. In the unweighted walk, we assign each pair into \mathcal{D}^R for entities linked by score links and discard it otherwise. In the positive walk, we assign each pair into \mathcal{D}^R for entities of score links and into \mathcal{D}^+ otherwise. In the negative walk, we assign each pair into \mathcal{D}^R for entities linked with score links, into \mathcal{D}^- for a user and an item, and into \mathcal{D}^+ for two users and two items. Figure 2b shows an example of extracting node pairs.

We learn biases and vectors of all node $v \in \tilde{G}$ that minimize L with the extracted node pairs in \mathcal{D}^R , \mathcal{D}^+ and \mathcal{D}^- by gradient descent method with momentum [21] for a faster learning. The gradient update processes are stated in the supplementary material (Section 4.1 of [19]).

Step 3: Predict ratings. For a user u and an item i , we predict rating of i given by u as $\hat{r}_{ui} = \mu + b_u + b_i + \mathbf{z}_u^T \mathbf{z}_i$.

3.3 UniWalk for Explainable Recommendation.

For a user u , we recommend items i whose predicted ratings $\hat{r}_{ui} = \mu + b_u + b_i + \mathbf{z}_u^T \mathbf{z}_i$ are the highest. UniWalk not only predicts rating scores, but also explains why items are recommended to a user. UniWalk recommends items to a *target user* u because (1) other users who are similar to u like the items, and (2) u likes other items that are similar to the recommended items. We reinforce the above two explanations by further examining the *supplementary entities* which are defined as follows.

DEFINITION 5. *Supplementary entity.*

Supplementary entities are entities used in explanation of recommendation such as other users similar to a target user, and items similar to the recommended items.

We explain why the *supplementary entities* are actually similar to u and the recommended items for more understandable explanations. The further explanation is stated as follows.

DEFINITION 6. *Meta-explainability of UniWalk.*

UniWalk provides meta-explanation (explanation for explanation): it explains why the supplementary entities are similar to target user or recommended items.

We use \mathcal{D}^R and \mathcal{D}^+ in our explanation, because we use rating information to denote like (high rating) or dislike (low rating), and similar relationships of entities. We do not use \mathcal{D}^- in explanation, because multi-step dissimilar relationships are not used. The only dissimilar relationship we use is from low observed ratings in \mathcal{D}^R such as common unlikable items in Figure 3a.

The first reason of our recommendations is that other similar users prefer the recommended items. The similar users are defined as ones with the highest similarities to the target user among all candidates who rate at least one of the recommended items. A similarity of distinct entities v and w is defined in Equation (3.4), where $\#(v, w)$ denotes the number of times the pair (v, w) appears in \mathcal{D}^+ , and $\#v = (\sum_{(v, w) \in \mathcal{D}^+} 1) + (\sum_{(w, v) \in \mathcal{D}^+} 1)$.

$$(3.4) \quad sim(v, w) = \frac{\#(v, w)}{\#v\#w}$$

\mathcal{D}^+ is a set of similar entities defined in Definition 4. $\#v\#w$ serves as a normalization term.

The next reason of our recommendations is that u prefers other similar items. The similar items are selected among all candidates that u rated. For each recommended item i , we find out the most similar items j to the recommended item among the candidates with the similarity score $sim(i, j)$ defined in Equation (3.4).

UniWalk also provides meta-explanations. We explain why the supplementary entities are similar to the target user or the recommended items. The supplementary users are determined to be similar because the supplementary users and the target user have common friends, common favorite items, or common unlikable items. The common favorite items are highly rated items by the users, and the common unlikable items are lowly rated ones in observed ratings. The supplementary items are determined to be similar items because common users rate them favorably.

4 Experiment

We present experimental results to answer the following questions.

- **Q1 (Explainability):** How can UniWalk explain its recommendation results to users? (Section 4.2)
- **Q2 (Accuracy):** How accurately does UniWalk predict ratings? (Section 4.3)
- **Q3 (Learning details):** How does accuracy of UniWalk change during iterations? How do hyperparameters affect the accuracy? (Section 4.4)

4.1 Experimental settings.

Machine. All experiments are conducted with a single CPU Intel(R) Xeon(R) CPU E5-2640 v3 @ 2.60GHz with 32 GB memory.

Datasets. We use social rating network data that have both observed ratings and a social network. We summarize datasets in Table 2. These datasets contain explicit ratings and directed or undirected edges between users. We convert directed edges to undirected edges in our experiment.

- *FilmTrust*¹ is a movie sharing and rating website. This dataset is crawled by Guo et al. [5].
- *Epinions*² ([epinions.com](http://www.epinions.com)) is a who-trust-whom online social network of general consumer reviews.
- *Flixster*³ ([flixster.com](http://www.flixster.com)) is also a website where users share movie ratings. It is crawled by Jamali and Etser [9].

¹<https://www.librec.net/datasets.html>.

²<http://www.trustlet.org/epinions.html>

³<https://www.librec.net/datasets.html>

Table 2: Statistics of Datasets.

	<i>FilmTrust</i>	<i>Epinions</i>	<i>Flixster</i>
# of users	1,642	49,289	787,213
# of items	2,071	139,738	48,794
# of rating	35,494	664,824	8,196,077
# of social edges	1,309	381,036	7,058,819

Competitors. We compare UniWalk with explainable methods UCF and ICF. We also compare UniWalk with MF to show if the unsupervised terms in UniWalk improve accuracy. In addition, we choose TrustSVD as a baseline method because it outperforms other social recommender systems.

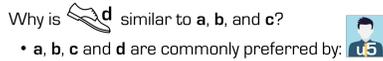
- UCF (User-based collaborative filtering) predicts a rating of a user u with k most similar users of cosine the highest cosine similarity. A predicted rating is calculated as $\hat{r}_{ui} = \frac{1}{k} \sum_{v \in K_k} r_{vi}$, where K_k is a set of the similar users. UCF explains its recommendation: it recommends items that are preferred by the similar users.
- ICF (Item-based collaborative filtering) predicts a rating of an item i with k most similar items in terms of cosine similarity with i . The rating is predicted as $\hat{r}_{ui} = \frac{1}{k} \sum_{j \in K_k} r_{uj}$, where K_k is a set of the similar items. ICF explains its recommendations for a user u : it recommends items that are similar to u 's favorite items.
- MF (Matrix factorization with bias terms) is described in Section 2.1.
- TrustSVD [6] is based on matrix factorization, and the state-of-the-art method among the ones that use both a rating matrix and a social network. TrustSVD outperforms other methods including PMF [18], RSTE [14], SoRec [15], SoReg [16], SocialMF [9], TrustMF [26], and SVD++ [12].

Hyperparameters. The hyperparameters of UniWalk, MF, UCF, and ICF were determined experimentally, and they are reported in supplementary document. We use hyperparameters for UniWalk as follows. In *Filmtrust* dataset, $c=5$, $l=30$, $\alpha=0.05$, $\beta=0.005$, $d=25$, $s=7$, $\lambda_b=0.1$, $\lambda_z=0.1$, $\eta=0.01$, and $\gamma=0.2$. In *Epinions* dataset, $c=6$, $l=50$, $\alpha=0.001$, $\beta=0.0007$, $d=25$, $s=7$, $\lambda_b=0.08$, $\lambda_z=1.3$, $\eta=0.003$, and $\gamma=0.6$. In *Flixster* dataset, $c=5$, $l=50$, $\alpha=0.001$, $\beta=0.001$, $d=25$, $s=7$, $\lambda_b=0.2$, $\lambda_z=0.2$, $\eta=0.005$, and $\gamma=0.6$. We use hyperparameters for TrustSVD as reported in [6].

4.2 Explainability of UniWalk. We show an experiment on our explanation approach with *Filmtrust* dataset. UniWalk explains the reason behind its recommendations in two ways with meta-explanations for each reason. Items are recommended, because they are preferred by other similar users, and the target user



(a) u_5 is determined to be similar to the target user (u_1), because they like or dislike common items. (b) u_2 is determined to be similar to u_1 , because they have common favorite items and friends.



(c) The supplementary item d is similar to the recommended items since they are commonly preferred by other users.

Figure 3: UniWalk’s meta-explanation to further explain why the *supplementary entities* are determined to be similar to target user (u_1) and recommended items (a , b , and c) in Figure 1.

likes other items that are similar to the recommended ones. We generate meta-explanation about the reasons by providing further explanation on supplementary entities. We explain why the *supplementary users* and items are determined to be similar.

Figure 1⁴ illustrates our explanation of recommendation. Figure 1b illustrates the first reason in our recommendation that the items (a , b , and c) are recommended to the target user (u_1): other similar users (u_2 , u_3 , u_4 , and u_5) like them. We describe the similar users more by presenting their similarity and friend relationships with the target user. Figures 3a⁵ and 3b⁶ explain why the similar users are determined to be similar. They like or dislike common items and have common friends. Figure 1c explains the second reason why the items are recommended: the target user likes other similar items (d , e , and f). We explicitly present similarity scores of *supplementary items* with the recommended items. Figure 3c explains why the supplementary item (d) is determined to be similar to the recommended items. It is preferred by common people including the target users’ friends such as u_5 .

4.3 Accuracy. We measure RMSE (Root Mean Square Error) and MAE (Mean Absolute Error) to evaluate accuracy of recommendation models. Lower values indicate accurate predictions. We separate training and test sets under 5-folded cross validation, ensuring that each observed rating is included in a test set only once.

⁴Ids of entities in *Filmtrust* dataset: $u_1=28$, $u_2=1332$, $u_3=814$, $u_4=319$, $u_5=1038$, $a=1803$, $b=2820$, $c=3635$, $d=1859$, $e=1919$, $f=1948$.

⁵ $g=1700$, $h=1908$, $i=1921$.

⁶ $u_6=508$, $j=1930$, $k=1933$.

Table 3: RMSE of UniWalk and other competitors.

Dataset	UniWalk (proposed)	UCF	ICF	MF	TrustSVD
<i>Filmtrust</i>	0.783	0.924	0.914	0.839	0.787
<i>Epinions</i>	1.041	1.200	1.257	1.135	1.043
<i>Flixster</i>	0.913	1.097	1.092	0.951	0.948

Table 4: MAE of UniWalk and other competitors.

Dataset	UniWalk (proposed)	UCF	ICF	MF	TrustSVD
<i>Filmtrust</i>	0.598	0.727	0.713	0.658	0.607
<i>Epinions</i>	0.798	0.907	0.939	0.873	0.804
<i>Flixster</i>	0.711	0.854	0.871	0.739	0.726

Tables 3 and 4 show RMSE and MAE of UniWalk and competitors, respectively. UniWalk shows the lowest RMSE and MAE, outperforming competitors and achieving the state-of-the-art accuracy for recommendation with rating data and a social network. TrustSVD shows the second best accuracy; however, it fails to explain its recommendation results, as we described in Section 4.2. UniWalk is much more accurate than MF, which indicates our extended unsupervised terms enhance accuracy.

4.4 Learning Details. We present details of UniWalk’s learning process and hyperparameter sensitivity.

We first present how the accuracy changes during learning iterations: a iteration performs learning from positive walks, negative walks, and unweighted walks (lines 9-11 in Algorithm 2 of [19]). Figure 4 shows the learning process of UniWalk on *FilmTrust* dataset. We note that the minimum test error (denoted by the dashed line) is achieved after only three iterations, thanks to the momentum term in the gradient descent (Section 4.1 of [19]). Learning with the momentum is 1.45× faster and requires 2.5× less iterations than learning without the momentum in our experiment. We assess our model’s sensitivity to the learning weight α of positive unsupervised term, the learning weight β of negative unsupervised term, and the weight c of social links; sensitivity for other hyperparameters is reported in the supplementary document (Section 6.3 of [19]). Figure 5 shows the UniWalk’s hyperparameter sensitivity on *FilmTrust* dataset. UniWalk is less sensitive to c . UniWalk is sensitive to α and β , but not much.

5 Related Works

We review related works along the following highly related aspects: recommender systems with auxiliary information, explainable recommendation, and network embedding.

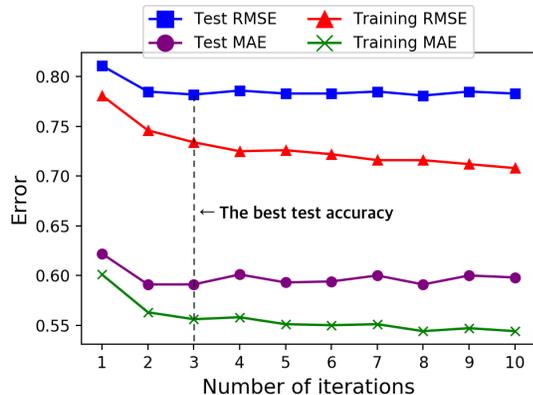


Figure 4: Changing accuracy of UniWalk in training and test set. RMSE and MAE decrease initially and converge through the iterations.

5.1 Recommendation with Additional Data.

Many studies have proved that using auxiliary data in recommendation improves accuracy. Various additional data alleviate rating sparsity problem. Recent studies employ auxiliary information for better performance.

Social Recommendation. Many recommender systems use social networks which contain rich information on influences among users. Studies [8, 10, 15, 27] prove that a social network enhances rating prediction. [31] trains ranking-based model with a social network. Chaney et al. [3] develop a probabilistic model that integrates social network with traditional matrix factorization method.

Heterogeneous Information. Other information is used to improve recommendation performance. Li et al. [13] use both geographical and social information. Another example shows that attributes of entities (e.g. genre) are used in recommending personalized entities [28]. [29] divides heterogeneous data into three categories and extracts features from the categories.

5.2 Explainable Recommendation. An explainable recommender system explains reasons behind its recommendations. A reasonable explanation is beneficial: explainable predictions increase users’ trust [23], and induce the users to inform the system of its wrong predictions [30]. Many methods explain their recommendation results to convince users. We first study explainable recommendation models using users’ reviews and then present other methods using social network.

Review-based explanation of recommendation. User reviews are useful in explaining recommendation because the reviews have keywords of users’ opinions on their preferences. Many explainable recommendation models such as SCAR [22], TriRank [7], and EFM [30] exploit user reviews. SCAR explains its recommendations by listing keywords of an item’s concept and topics with topic analysis, sentiment prediction, and viewpoint

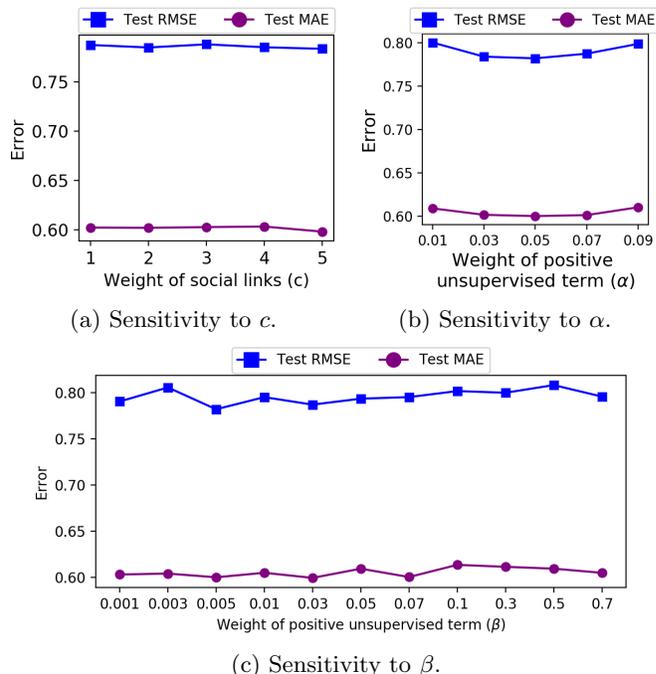


Figure 5: UniWalk’s hyperparameter sensitivity.

regression. TriRank analyzes user-item-aspect tripartite graph, and explains its results using important words or phrases in the reviews to describe items. EFM extracts feature words of products and users’ opinions from reviews. It explains its recommendations by suggesting a set of feature words that users mainly focus on and that describe recommended products.

Friend-based explanation of recommendation. Social relationships are also useful to explain recommendations. Recommendation models such as SPF [3] and LBSN [11] explain reasons of its results with social links. SPF recommends items to a user because the items are favorites of the users’ friends. LBSN recommends locations to a user at a certain time because the locations are visited by the users’ friends and it is likely that the friends will visit the place again soon.

5.3 Network Embedding. Network embedding has been studied extensively and has become a popular way to represent graph data. Recently proposed network embedding methods [1,2,4,20,24,25] try to encode proximity structure. Tian et al. [25] apply autoencoder to reconstruct neighborhood structure of each node. Perozzi et al. [20] propose DeepWalk model which applies language model on sequences of nodes generated by truncated random walks on a graph. Tang et al. [24] directly model first-order and second-order proximities and optimize them. Grover and Leskovec [4] extend DeepWalk by adding parameters in the truncated random walk to customize the random walk on different data.

6 Conclusion

In this paper, we propose UniWalk, a novel recommendation model that exploits both rating data and a social network. UniWalk constructs a unified graph containing users and items where the weights reflect the degree of association between users or user-item pairs. Then, UniWalk performs network embedding on the unified graph to extract latent features of users and items, and predicts unobserved ratings with the embedded features. UniWalk provides the best accuracy for recommendation using rating data and a social network, outperforming existing methods. Furthermore, UniWalk provides explanations of its recommendations. Future works include extending the method for distributed systems for more scalable learning.

References

- [1] S. CAO, W. LU, AND Q. XU, *Grarep: Learning graph representations with global structural information*, in CIKM, 2015.
- [2] S. CAO, W. LU, AND Q. XU, *Deep neural networks for learning graph representations*, in AAAI, 2016.
- [3] A. J. CHANEY, D. M. BLEI, AND T. ELIASSI-RAD, *A probabilistic model for using social networks in personalized item recommendation*, in RecSys, 2015.
- [4] A. GROVER AND J. LESKOVEC, *node2vec: Scalable feature learning for networks*, in KDD, 2016.
- [5] G. GUO, J. ZHANG, AND N. YORKE-SMITH, *A novel bayesian similarity measure for recommender systems*, in IJCAI, 2013.
- [6] G. GUO, J. ZHANG, AND N. YORKE-SMITH, *Trustsvd: Collaborative filtering with both the explicit and implicit influence of user trust and of item ratings*, in AAAI, 2015.
- [7] X. HE, T. CHEN, M.-Y. KAN, AND X. CHEN, *Trirank: Review-aware explainable recommendation by modeling aspects*, in CIKM, 2015.
- [8] J. HUANG, X. CHENG, J. GUO, H. SHEN, AND K. YANG, *Social recommendation with interpersonal influence.*, in ECAI, 2010.
- [9] M. JAMALI AND M. ESTER, *A matrix factorization technique with trust propagation for recommendation in social networks*, in RecSys, 2010.
- [10] M. JIANG, P. CUI, R. LIU, Q. YANG, F. WANG, W. ZHU, AND S. YANG, *Social contextual recommendation*, in CIKM, 2012.
- [11] P. KEFALAS, P. SYMEONIDIS, AND Y. MANOLOPOULOS, *New perspectives for recommendations in location-based social networks: time, privacy and explainability*, in MEDES, 2013.
- [12] Y. KOREN, *Factorization meets the neighborhood: a multifaceted collaborative filtering model*, in KDD, 2008.
- [13] H. LI, Y. GE, R. HONG, AND H. ZHU, *Point-of-interest recommendations: Learning potential check-ins from friends*, in KDD, 2016.
- [14] H. MA, I. KING, AND M. R. LYU, *Learning to recommend with social trust ensemble*, in SIGIR, 2009.
- [15] H. MA, H. YANG, M. R. LYU, AND I. KING, *Sorec: Social recommendation using probabilistic matrix factorization*, in CIKM, 2008.
- [16] H. MA, D. ZHOU, C. LIU, M. R. LYU, AND I. KING, *Recommender systems with social regularization*, in WSDM, 2011.
- [17] M. MCPHERSON, L. SMITH-LOVIN, AND J. M. COOK, *Birds of a feather: Homophily in social networks*, Annual review of sociology, (2001).
- [18] A. MNH AND R. R. SALAKHUTDINOV, *Probabilistic matrix factorization*, in NIPS, 2008.
- [19] H. PARK, H. JEON, J. KIM, B. AHN, AND U. KANG, *Supplementary document (submitted)*, in SDM, 2018.
- [20] B. PEROZZI, R. AL-RFOU, AND S. SKIENA, *Deepwalk: Online learning of social representations*, in KDD, 2014.
- [21] N. QIAN, *On the momentum term in gradient descent learning algorithms*, Neural networks, (1999).
- [22] Z. REN, S. LIANG, P. LI, S. WANG, AND M. DE RIJKE, *Social collaborative viewpoint regression with explainable recommendations*, in WSDM, 2017.
- [23] M. T. RIBEIRO, S. SINGH, AND C. GUESTRIN, *Why should i trust you?: Explaining the predictions of any classifier*, in KDD, 2016.
- [24] J. TANG, M. QU, M. WANG, M. ZHANG, J. YAN, AND Q. MEI, *Line: Large-scale information network embedding*, in WWW, 2015.
- [25] F. TIAN, B. GAO, Q. CUI, E. CHEN, AND T.-Y. LIU, *Learning deep representations for graph clustering.*, in AAAI, 2014.
- [26] B. YANG, Y. LEI, D. LIU, AND J. LIU, *Social collaborative filtering by trust*, in IJCAI, 2013.
- [27] X. YANG, H. STECK, AND Y. LIU, *Circle-based recommendation in online social networks*, in KDD, 2012.
- [28] X. YU, X. REN, Y. SUN, Q. GU, B. STURT, U. KHANDELWAL, B. NORICK, AND J. HAN, *Personalized entity recommendation: A heterogeneous information network approach*, in ICWSM, 2014.
- [29] F. ZHANG, N. J. YUAN, D. LIAN, X. XIE, AND W.-Y. MA, *Collaborative knowledge base embedding for recommender systems*, in KDD, 2016.
- [30] Y. ZHANG, G. LAI, M. ZHANG, Y. ZHANG, Y. LIU, AND S. MA, *Explicit factor models for explainable recommendation based on phrase-level sentiment analysis*, in SIGIR, 2014.
- [31] T. ZHAO, J. J. MCAULEY, AND I. KING, *Leveraging social connections to improve personalized ranking for collaborative filtering*, in CIKM, 2014.