

석사학위논문  
Master's Thesis

상품 평가 품질 조작에 견고한 추천 시스템

A Robust Recommendation System  
Against Review Quality Manipulation

2017

심동진 (沈東鎭 Sim, Dongjin)

한국과학기술원

Korea Advanced Institute of Science and Technology

석 사 학 위 논 문

상품 평가 품질 조작에 견고한 추천 시스템

2017

심 동 진

한 국 과 학 기 술 원

전산학부

# 상품 평가 품질 조작에 견고한 추천 시스템

심 동 진

위 논문은 한국과학기술원 석사학위논문으로  
학위논문 심사위원회의 심사를 통과하였음

2016년 12월 19일

심사위원장 이 윤 준 (인)

심 사 위 원 김 명 호 (인)

심 사 위 원 유 신 (인)

# A Robust Recommendation System Against Review Quality Manipulation

Dongjin Sim

Advisor: Yoon-Joon Lee

A dissertation submitted to the faculty of  
Korea Advanced Institute of Science and Technology in  
partial fulfillment of the requirements for the degree of  
Master of Science in Computer Science

Daejeon, Korea  
2016, December

Approved by

---

Yoon-Joon Lee  
Professor of School of Computing

The study was conducted in accordance with Code of Research Ethics<sup>1</sup>.

---

<sup>1</sup> Declaration of Ethical Conduct in Research: I, as a graduate student of Korea Advanced Institute of Science and Technology, hereby declare that I have not committed any act that may damage the credibility of my research. This includes, but is not limited to, falsification, thesis written by someone else, distortion of research findings, and plagiarism. I confirm that my thesis contains honest conclusions based on my own careful research under the guidance of my advisor.

MCS  
20153338

심동진. 상품 평가 품질 조작에 견고한 추천 시스템. 전산학부 . 2017  
년. 22+ii 쪽. 지도교수: 이윤준. (영문 논문)

Dongjin Sim. A Robust Recommendation System Against Review Quality  
Manipulation. School of Computing . 2017. 22+ii pages. Advisor: Yoon-  
Joon Lee. (Text in English)

### 초 록

추천 시스템은 사용자들의 구매 결정에 주요한 영향을 주기 때문에 악의적인 조작의 표적이 되어왔다. 대표적인 추천 방법은 상품에 대한 평가들이 정직하다는 가정하에 사용자의 평점을 분석하는 협업 필터링 방법이다. 하지만, 협업 필터링 방법은 거짓된 평점을 주입하는 실링 공격에 의해 조작될 위험이 있다. 이로 인해 실링 공격의 영향력을 낮추는 여러 방법이 제안되었다. 한편 많은 추천 시스템은 사용자들에게 다른 사용자의 리뷰 유용성을 평가하길 권장한다. 이러한 과정을 통해 거짓된 리뷰는 사용자들에 의해 분류될 수 있을 거라는 가정하에 연구자들은 리뷰의 유용성을 고려한 추천을 통해 실링 공격을 극복하고자 하였다. 하지만 이러한 추천 방법은 거짓된 리뷰의 유용성 평가 데이터가 주입 된다면 오히려 조작의 정도가 심해질 위험이 있다. 본 논문에선 리뷰의 거짓 평점 데이터의 영향력을 낮추어 진실한 유용성을 추정함으로써 더욱 견고한 추천 방법을 제안한다. 실제 데이터를 토대로 한 실험 결과를 통해 제안하는 방법의 견고함을 보였다.

핵심 낱말 견고한 추천 시스템, 협업 필터링, 추천 시스템 조작, 리뷰 유용성 측정

### Abstract

Recommendation systems influence decision making, and hence have become attractive targets of manipulation. Collaborative filtering, widely adopted in recommendation systems, exploits the observed ratings given to items by users to provide personalized recommendations under the assumption that all users honestly rate items. Unfortunately, however, shilling attacks which inject multiple fake reviews can easily manipulate recommendation systems with the naive assumption. There are several approaches to mitigate the effect of shilling attack. Recently, some researchers have been interested in the fact that most recommendation systems encourage users to write reviews, as well as rate the usefulness of reviews written by other users based on their review content. With the assumption that fake reviews are rated as not helpful by users, they suggest recommendation systems considering the helpfulness of reviews. However, their recommendations are vulnerable to attacks that inject fake helpfulness ratings with intent to boost the helpfulness of fake reviews. In this paper, we propose a robust recommendation system to overcome such attacks. The proposed system estimates the true helpfulness of reviews, thereby preventing such attacks from manipulating recommendation results. Experimental results on a real-world dataset indicate the robustness of our method.

Keywords Robust recommendation system, Collaborative filtering, Shilling attack, Review helpfulness

# Contents

Contents . . . . .	i
List of Tables . . . . .	ii
List of Figures . . . . .	iii
<b>Chapter 1. Introduction</b>	<b>1</b>
<b>Chapter 2. Background</b>	<b>3</b>
2.1 Notation . . . . .	3
2.2 Matrix Factorization Based Collaborative Filtering . . . . .	3
2.3 Shilling Attack . . . . .	4
<b>Chapter 3. Related Work</b>	<b>6</b>
3.1 Robust Recommendation . . . . .	6
3.2 Review Helpfulness . . . . .	6
<b>Chapter 4. Problem Definition</b>	<b>8</b>
4.1 Attack Model . . . . .	8
4.1.1 Fake Item Rating Injection . . . . .	8
4.1.2 Fake Helpfulness Rating Injection . . . . .	8
4.2 Problem Definition . . . . .	9
<b>Chapter 5. Proposed Method</b>	<b>10</b>
5.1 User2Vec . . . . .	10
5.2 Robust Review Quality Measure . . . . .	11
<b>Chapter 6. Experiment</b>	<b>14</b>
6.1 Experimental Setting . . . . .	14
6.2 Metrics . . . . .	14
6.3 Results and Analysis . . . . .	15
<b>Chapter 7. Conclusion</b>	<b>18</b>
<b>Bibliography</b>	<b>19</b>
<b>Acknowledgments in Korean</b>	<b>21</b>
<b>Curriculum Vitae in Korean</b>	<b>22</b>

## List of Tables

6.1	Statistics of CiaoDVD dataset . . . . .	14
6.2	Review helpfulness results. The range of helpfulness is from 0 to 5 . . . . .	16
6.3	Prediction shift on the target items . . . . .	16
6.4	MAE on test set . . . . .	17

## List of Figures

1.1	An item review example . . . . .	2
1.2	A helpfulness rating example . . . . .	2
2.1	Matrix factorization example . . . . .	4
2.2	Matrix factorization in the presence of shilling attack . . . . .	5
3.1	Weighted matrix factorization with well assigned weight matrix . . . . .	7
3.2	Weighted matrix factorization with badly assigned weight matrix . . . . .	7
6.1	Visualization of the User2Vec result . . . . .	15



# Chapter 1. Introduction

Since recommender systems influence purchase decisions, their positive recommendation can lead to significant monetary benefit for product sellers. According to [1], increasing overall rating of business by one star on Yelp can increase its revenue by 9%. Unfortunately, however, the strong impact of recommender systems has attracted malicious attackers who try to bias recommendation result to increase overall rating of their target items.

Matrix factorization(MF) model based collaborative filtering(CF), one of the most common approaches for the recommendation, infers users' tastes and items' attributes based on the observed ratings of users for items and recommends products whose attributes match a user's taste. Standard MF assumes that all the observed ratings are conducted by honest users. However, this assumption is easily violated in practice due to the presence of attackers. Due to the open nature of recommendation systems, attackers can inject multiple fake users (Shillers) and fake ratings to increase overall rating of their target items on the recommendation system. Such injections with intent to bias recommendation are called, shilling attacks. There are a number of studies about shilling attack strategies, shilling attack detection and robust recommendation to prevent shilling attack.

Recent studies [9, 8] focused on the following dual roles of users in recommender systems. In real-world recommender systems, users play as reviewers that write reviews about items in the form of a numeric rating score (such as a 1-5-star rating) accompanied by review text, and they also play as helpfulness raters that rate the helpfulness of reviews based on review content by giving numeric rating score. Figure 1.1 contains a review of *Captain America: Civil War (DVD)* written by a user *lmmysasi29* with a 5-star rating. Figure 1.2 represents how other users rate the helpfulness of the review in Figure 1.1.

Suhang et al. [9] treat helpfulness rating as users' implicit feedback about items, and hence incorporate helpfulness rater role of users in recommendation systems to mitigate the data sparsity and cold-start problems. Sindhu et al. [8] suggest recommendation taking into account review quality to improve the performance in the presence of spurious reviews. They assume that if the number of helpfulness ratings is sufficiently high, spurious reviews get negative helpfulness ratings. They measure the quality of a review by aggregating the helpfulness ratings about the review and mitigate the impact of low-quality reviews on optimizing parameters of recommendation model. These studies show that incorporating review helpfulness information has potential benefits of improving the performance and robustness of recommendation systems.

However, these studies do not deal with manipulating review helpfulness attack. After injecting fake reviews, malicious attackers can easily inject many fake helpfulness ratings to promote the helpfulness of fake reviews. The helpfulness measure vulnerable to injected fake helpfulness ratings results in amplifying the negative effect of fake reviews rather than mitigating it.

Therefore, in this paper, we propose a robust recommendation in the presence of fake reviews and helpfulness rating via unbiased review helpfulness measure. Our approach to a robust recommendation system consists of three stages. The first stage involves the task of mapping users to a representation vector space such that users who are similar in terms of behaviors related to shilling attacks are located in close proximity to one another in the space. In the second stage, the helpfulness of each review is measured by the weighted mean of the helpfulness ratings associated with each review. A helpfulness

rating is “down-weighted” if the similarity between the representation vectors of the helpfulness rater and the writer of the associated review is above the predetermined threshold. In the final stage, the helpfulness of each review measured in the previous stage is used to collaborative filtering. We adopt the cost function suggested by Sindhu et al. [8]. We demonstrate the effectiveness of the proposed method by measuring the effect of various attacks on a real dataset. Compared with other methods, proposed method mitigates the effect of manipulating review helpfulness.

The rest of paper is organized as follows. Chapter 2 presents background of this paper. Related work are described in Chapter 3. In Chapter 4, we formally define our attack model. Chapter 5 presents the proposed method. In Chapter 6 presents the experimental methodology used to evaluate the robustness of our approach and results of our experiment. Finally, we conclude in Chapter 7.

[Captain America: Civil War - Film Review](#)  
**Advantages:** Performance, Visual effects, cast, story, cinematography  
**Disadvantages:** NONE!  
 ...The Russo brothers come back to the big screen with a bang with Marvel's star actors Chris Evans and Robert Downey Jr back in lead roles and facing off against each other in the year's most anticipated superhero showdown, Captain America: Civil War. CAST ...  
 ★★★★★ Immyvasi29 14.09.2016 · [Read full review](#)  
 Ciao members have rated this review on average: ██████████ very helpful  
 Review of [Captain America: Civil War \(DVD\)](#)

Figure 1.1: An item review example

**Review Ratings »**  
 This review of [Captain America: Civil War \(DVD\)](#) has been rated:  
 "exceptional" ██████████ by (14%):  
 1. ♦ [danielclark691](#)  
 2. ♦ [mumsymary](#)  
 3. ● [sellerleygirl](#)  
 "very helpful" ██████████ by (86%):  
 1. ♦ [Pointress](#)  
 2. ● [DodoRabbit](#)  
 3. ● [euphie](#)  
 and [15 other members](#)

Figure 1.2: A helpfulness rating example

## Chapter 2. Background

### 2.1 Notation

Throughout this paper, sets are denoted as italic capital letters. Let  $U = \{u_1, u_2, \dots, u_n\}$  and  $I = \{i_1, i_2, \dots, i_m\}$  be a set of users and items where  $n$  and  $m$  are the number of users and items, respectively. In recommendation systems, users can rate items in the form of a numeric rating score accompanied by review text. Unless otherwise mentioned, the term ‘review’ of a user for an item indicates numeric rating the user give to the item. We use the matrix  $\mathbf{R} \in \mathbb{R}^{n \times m}$  to denote the user-item rating matrix where an entry  $R_{i,j}$  indicates the rating score of the  $i$ -th user for the  $j$ -th item. Note that the rating matrix  $\mathbf{R}$  is sparse since users usually rate a small set of items. If the  $i$ -th user did not rate the  $j$ -th item, then we assign “?” to the missing rating  $\mathbf{R}_{i,j}$ . We use  $IR = \{(u, i, ir) | u_a \in U, i_b \in I, r = \mathbf{R}_{a,b}, r \neq ?\}$  to represent item rating dataset where  $(u, i, ir)$  means user  $u$  rates item  $i$  with score  $ir$ .

Some recommendation systems allow users to rate the helpfulness of other users’ reviews with intent to improve the user experience. In such systems, after reading review text of another user, users give the review helpfulness score in the form of a numeric rating score. For example,  $(u_a, u_b, i_c, hr)$  represents user  $u_a$  gives helpfulness score  $hr$  to the review of user  $u_b$  for item  $i_c$ . We use the tensor  $\mathbf{H} \in \mathbb{R}^{n \times n \times m}$  to denote the user-user-item helpfulness rating tensor where an entry  $\mathbf{H}_{a,b,c}$  indicates the review helpfulness score that the  $a$ -th user gives to the review given by the  $b$ -th user about the  $c$ -th item. Similarly with item rating dataset  $IR, HR = \{(u_a, u_b, i_c, h) | u_a, u_b \in U, i_c \in I, hr = \mathbf{H}_{a,b,c}, hr \neq ?\}$  represents helpfulness rating dataset.

Since we consider an attacker who injects fake users and fake ratings, we use  $U^g$  and  $U^f$  to denote the set of genuine users and fake users, respectively. Genuine item rating and helpfulness rating dataset are denoted by  $IR^g = \{(u, i, r) | u \in U^g\}$  and  $HR^g = \{(u, v, i, h) | u \in U^g\}$ , respectively. Similarly, fake item rating and helpfulness rating dataset are denoted by  $IR^f = \{(u, i, r) | u \in U^f\}$  and  $HR^f = \{(u, v, i, h) | u \in U^f\}$ , respectively. Unless otherwise noted, the range of item rating score is from 1 to 5, and helpfulness rating score ranges from 0 to 5.

### 2.2 Matrix Factorization Based Collaborative Filtering

In the context of collaborative filtering, matrix factorization infers latent features of user and items based on observed ratings of users for items. It decomposes a user-item rating matrix  $\mathbf{R}$  into two latent matrices  $\mathbf{U} \in \mathbb{R}^{n \times d}$  and  $\mathbf{V} \in \mathbb{R}^{d \times m}$  corresponding latent features of user and item, respectively where the  $d$  is the number of latent features. In specific, given an observed rating matrix  $\mathbf{R}$ , matrix factorization optimizes two matrices  $\mathbf{U}$  and  $\mathbf{V}$  by minimizing the following cost function which is the sum of prediction error terms and regularization terms.

$$Cost(\mathbf{U}, \mathbf{V} | \mathbf{R}) = \sum_{\mathbf{R}_{i,j} \neq ?} (\mathbf{R}_{i,j} - (\mathbf{UV})_{i,j})^2 + \lambda(\|\mathbf{U}\|_F^2 + \|\mathbf{V}\|_F^2) \quad (2.1)$$

After obtaining the optimized  $\mathbf{U}$  and  $\mathbf{V}$ , the missing ratings in the rating matrix  $\mathbf{R}$  are predicted via the dense matrix which is the product of  $\mathbf{U}$  and  $\mathbf{V}$ . The figure 2.1 indicates a toy example. Observed rating matrix (Figure 2.1 (a)) represents user-movie rating matrix. The item set of the rating matrix

consists of two romance movies, two horror movies, and one bad movie. The user set of the rating matrix consists of two romance movie lovers and two horror movie lovers. By applying matrix factorization on the rating matrix  $R$ , prediction matrix captures the tastes of users and judges that users will not like the 5-th movie (bad movie).

		Romance Movies		Horror Movies		Bad Movie
Romance Lovers		5	5		2	
		4		2		1
Horror Lovers		2		4	4	1
			2	5	5	

**Rating matrix  $R$**

		Romance Movies		Horror Movies		Bad Movie
Romance Lovers		5	5	2.1	2	1.2
		4	3.8	2	2	1
Horror Lovers		2	1	4	4	1
		3.7	2	5	5	1.4

**Prediction matrix  $UV$**

Figure 2.1: Matrix factorization example

## 2.3 Shilling Attack

Shilling attacks generate fake users (shillers) which give fake ratings to particular items with the intent to bias recommendation results of recommendation system. Shilling attacks are categorized into two categories: push attacks inject fake users which give high ratings to particular items to promote the recommendation score for the particular items, while nuke attacks inject fake users which give low ratings to particular items aiming at decreasing the popularity of the items. The goal of Shilling attacks is to manipulate the recommendation results for the particular items for other normal users. However, injecting fake rating associated with target items only is not enough to manipulate collaborative filtering based recommendation systems. Due to the principle of collaborative filtering, collaborative filtering recommends normal users to the direction shilling attacks wish if it recognizes that normal users are similar to fake users. In order to exploit the principle of collaborative filtering fully, fake users of shilling attack have to mimic rating behaviors of normal users. There are various attack models about how to mimic rating behavior of normal users: Random attack, Average attack, Bandwagon attack. In the context of push attack, Random attack injects fake users who give the highest rating to their target items and rate the randomly chosen items around the overall mean. Average attack generates fake user that give the highest rating to their target items and the mean rating of each item to randomly chosen items. Bandwagon attack consists of fake users whose ratings for their target items and popular items are maximum.

The figure 2.2 shows an example of shilling attack and its effect. We inject two shillers whose aims are boosting the prediction score of the bad movie. Shillers rate the bad movie with the highest possible rating value, i.e. 5, and rate other movies in a similar way to other genuine users. Matrix factorization lowers the error of fake ratings and therefore misjudges the prediction scores of the bad movie. Compared with figure 2.1, figure 2.2 contains high predicted rating of genuine users for the bad movie.

		Romance Movies		Horror Movies		Bad Movie
Romance Lovers		5	5		2	
		4		2		1
Horror Lovers		2		4	4	1
			2	5	5	
Shiller			3		3	5
		3		3		5

Rating matrix **R**

		Romance Movies		Horror Movies		Bad Movie
Romance Lovers		5	5	5	2	5
		3	1	3	4	1
Horror Lovers		3	1	3	4	1
		5	2	5	5	3.9
Shiller		4.3	3	4.3	3	5
		3	3.2	3	1	5

Prediction matrix **UV**

Figure 2.2: Matrix factorization in the presence of shilling attack

## Chapter 3. Related Work

Our paper is related to the following topics: robust recommendation and review helpfulness. We discuss the two topic in the following subsections.

### 3.1 Robust Recommendation

Bhaskar et al. [3] propose Robust Matrix Factorization (RMF) using M-estimators to bound the effect of outliers and noisy data. [10, 11, 4] apply PCA-based variable selection to detect suspicious users in unsupervised setting.

There are studies leveraging social networks to defend against Shilling attacks. Corresponding to [], these studies try to label users in the recommendation system either ‘untrustworthy’ or ‘trustworthy’, or bound the advantage an attacker can gain by injecting fake users. The common assumption of these studies is that a fake user cannot establish an arbitrary number of links to genuine users.

Many researchers take supervised approaches to detect fake reviews based on the textual or meta features in review. However, supervised learning based fake review detections have several drawbacks. In general, High class imbalance in the number of genuine data and attack label data degrades the performance of supervised learning. Furthermore, Review text-based approach is not proper to be adopted in general due to its domain-specific property.

### 3.2 Review Helpfulness

Many E-commerce sites encourage users to rate the helpfulness of reviews and place helpful reviews in a position where many users access to improve the overall user experience. Users can examine the helpfulness of reviews with statistics such as “90 (out of 100) people found this review helpful” or “40 members have rated this review on average (somewhat helpful)”.

Motivated by this, some researchers [8] propose a collaborative filtering method incorporating the helpfulness of reviews as the weight of reviews in the training phase. Under the assumption that spam reviews receive bad helpfulness ratings, they lower the weight of unhelpful reviews prediction, thereby reducing the contribution of unhelpful reviews to missing rating prediction. Kim et al. [5] propose the measure below to quantify helpfulness for an item review by aggregating helpfulness ratings for the item review.

$$Helpfulness(review(u, i)) = \frac{1}{N_{\mathbf{H}_{v,u,i}}} \sum_{\mathbf{H}_{v,u,i} \neq ?} \mathbf{H}_{v,u,i} \quad (3.1)$$

where  $N_{\mathbf{H}_{v,u,i}}$  is the number of helpfulness ratings for item rating  $\mathbf{R}_{u,i}$ .

They use the following cost function proposed by [17] to treat the helpfulness of a review as the weight of the review.

$$Cost(\mathbf{U}, \mathbf{V} | \mathbf{WR}) = \sum_{\mathbf{R}_{i,j} \neq ?} \mathbf{W}_{i,j} (\mathbf{R}_{i,j} - (\mathbf{UV})_{i,j})^2 + \lambda (\|\mathbf{U}\|_F^2 + \|\mathbf{V}\|_F^2) \quad (3.2)$$

In this cost function, prediction error term changes from sum of squared errors to weighted sum of squared errors. In other words, this cost function measures the error of a weighted matrix factorization.

Weight matrix <b>W</b>					Rating matrix <b>R</b>					Prediction matrix <b>UV</b>				
10	10		10		5	5		2		5	5	2.2	2	1.3
10		10		10	4		2		1	4	3.8	2	1.9	1.1
10		10	10	10	2		4	4	1	2	1	4	4	1
	10	10	10			2	5	5		3.8	2	5	5	1.5
	1		1	1		3		3	5	4.6	3.5	4	3.9	1.5
1		1		1	3		3		5	3.7	2.5	3.6	3.6	1.3

Figure 3.1: Weighted matrix factorization with well assigned weight matrix

Weight matrix <b>W</b>					Rating matrix <b>R</b>					Prediction matrix <b>UV</b>				
1	1		1		5	5		2		4.5	5	2.4	2.2	5
1		1		1	4		2		1	1.7	1.9	1.3	1.2	2.8
1		1	1	1	2		4	4	1	1	1	3.8	4	1.8
	1	1	1			2	5	5		2.9	2.1	4.9	5	5
	10		10	10		3		3	5	3	3	3	3	5
10		10		10	3		3		5	3	3	3	3	5

Figure 3.2: Weighted matrix factorization with badly assigned weight matrix

Therefore, the weighted sum of errors allows unhelpful reviews to have a significant error but penalizes error of helpful reviews heavily. They demonstrate considering review helpfulness could improve the overall performance of recommendation in the presence of spam review. Figure 3.1 shows an example where the weights of genuine reviews are bigger than that of fake reviews. With well-assigned weight matrix, latent features of users and items be able to explain genuine review, and hence shillers fail to manipulate prediction of genuine users for the bad movie.

The review helpfulness measure relies on the naive assumption which trusts all helpfulness rating. However, this assumption is easily violated if an attacker injects fake helpfulness ratings to promote the helpfulness of their fake reviews. From an adversarial perspective, the cost of fake helpfulness rating injection is not much more expensive than the cost of fake item rating injection. Figure 3.2 shows the case an attacker successes to manipulate the helpfulness of fake reviews. The fake ratings for the bad movie have a larger weight than other normal users' ratings as shown in Figure 3.2. Due to the manipulated weight matrix, collaborative filtering outputs prediction matrix weighted toward fake reviews.

## Chapter 4. Problem Definition

### 4.1 Attack Model

Since push attack more directly promotes monetary benefit than nuke attack does, this paper only focuses on push attack. We leave nuke attack case for future work. The objective of our attack model is to increase overall users' predicted ratings of target items. Our attack model involves in injecting fake item rating and helpfulness rating dataset,  $IR^f$  and  $HR^f$ .

#### 4.1.1 Fake Item Rating Injection

Since Random attack is known to be not effective, this paper focuses on Average attack and Bandwagon attack for injecting fake item rating dataset. Similarity with [], from the view of each fake user, the item set  $I$  is partitioned into 4 groups,  $I^{target}$ ,  $I^{popular}$ ,  $I^{filler}$  and  $I^{none}$  where  $I^{target}$  is a set of target items;  $I^{popular}$  is a set of popular items;  $I^{filler}$  is a set of randomly chosen items which are referred to filler items; and  $I^{none}$  is the set of unrated items, i.e.  $I^{none} = I - I^{target} - I^{popular} - I^{filler}$ . Each fake user rates items of  $I^{target}$ ,  $I^{popular}$  and  $I^{filler}$  as follows.

First, each fake user rates all target items with the highest possible rating score. Secondly, with the aim of being similar to other users, each fake user gives rating to each filler item of  $I^{filler}$  and its item rating is the average of all item rating given to the filler item. Finally, to increase the probability of being similar to a large number of other users, each fake users give the maximum item ratings for all popular items of  $I^{popular}$ . In summary, general form of fake item rating dataset  $IR^f$  is defined as follows.

$$IR^f = \bigcup_{u^f \in U^f} \{(u^f, i^t, ir_{max}) | i^t \in I^{target}\} \cup \{(u^f, i^f, ir_{avg}(i^f)) | i^f \in I^{filler}\} \cup \{(u^f, i^p, ir_{max}) | i^p \in I^{popular}\} \quad (4.1)$$

$ir_{max}$  is the maximum rating on the item rating scale and  $ir_{avg}(i^f)$  is the average item rating of item  $i^f$ .

#### 4.1.2 Fake Helpfulness Rating Injection

An attack whose goal is promoting the helpfulness of fake reviews involves in fake helpfulness rating injection. Such injection makes our study differ from existing attack models. Each fake user gives the highest helpfulness rating value to all the fake reviews about target items. Additionally, each fake user generates random helpfulness ratings for normal reviews to avoid being outlier. We call such helpfulness ratings as camouflage helpfulness ratings. Formally, fake helpfulness rating dataset  $HR^f$  is defined as follows:

$$HR^f = \bigcup_{u^f \in U^f} \{(u^f, v^f, i^t, hr_{max}) | v^f \in U^f, i^t \in I^{target}\} \cup \{(u^f, v, i, hr_{random}) | v \in U^g, i \in I, \mathbf{R}_{v,i} \neq ?\} \quad (4.2)$$

$hr_{max}$  is the maximum rating on the helpfulness rating scale and  $hr_{random}$  is a random rating. In the presence of such a fake helpfulness rating dataset, computing the average helpfulness rating incurs a biased measurement and therefore amplifies the negative effect of fake reviews.



## 4.2 Problem Definition

We define our problem as: given an item rating and review helpfulness rating dataset in the presence of our attack model, estimate the true review helpfulness which is incorporated into the weighted matrix factorization based CF so that predictions for the missing ratings are insensitive to injected fake data.

## Chapter 5. Proposed Method

This section describes our method in detail. The following sections describe how to capture suspicious helpfulness ratings and how to estimate true quality of reviews.

### 5.1 User2Vec

Recently, various prediction tasks [REF] have improved performance by learning the desirable features themselves, instead of manually determining domain-specific features. Skip-gram model [13] is a popular model proposed for natural language processing task by Mikolov et al. The goal of the Skip-gram model is to capture semantic relationships between words. With the hypothesis that words which frequently appear together in sentences have semantic relationships, the Skip-gram model takes large real-world text corpus as training data and learns feature representations for words. In specific, it map words to a feature space such that words frequently appear together in sentences have similar feature vectors. The Skip-gram model is widely adopted for natural language processing task due to the efficiency and ability to capture useful relationships in the text data. Inspried by the success of the Skip-gram model, some researchers apply the Skip-gram model to learning a mapping of vertices of a network to vectors which encode social relation [12, 15]. With the assumption random walk traces contain social relation between vertices, they generate samples of random walk traces as sequences of vertices and feed them into the Skip-gram model.

In this paper, we propose User2Vec, an algorithm for learning feature representations for users in recommendation system to detect such suspicious relationships between users. [13] uses real-world sentences as sequence of semantically related words [12, 15] generates random walk traces as sequence of socially related vertices to obtain useful features for various prediction tasks. Similarty to such approaches, we generate sequences of attack-related users and feed them into the Skip-gram model to obtain feature representations of user which is useful to detect fake users.

To generate sequences of attack-related users, we focus on behaviors of fake users. Several studies [10, 11] reported that fake users need to work together to maximize the effect of their attack. This strategy is referred to as group attack. In our attack model, fake users equally give the highest item rating to target items and the highest review helpfulness rating to their fake reviews ( $(IR^{target}, HR^{target})$ ). Taking this into consdieration, we regard following relationships between users as clues to the group attack

1. Both user X and Y give  $ir_{max}$  for an item
2. Both user X and Y give  $hr_{max}$  for a review
3. User X gives  $hr_{max}$  for a review written by User Y

We refer to the users rate an item with  $ir_{max}$  as enthusiasts for the item. Similarly, we refer to the users rate the helpfulness of a review with  $hr_{max}$  as supporters for the review. The first (second) relationship represents the pair of enthusiasts (supporters) whose opinions about some item (review) are same. If user  $u_c$  and  $u_d$  always rate in the same way items or reviews, it is reasonable to suspect that  $u_c$  and  $u_d$  are performing group attack. The last relationship indicates the pair of a reviewer and a supporter. If user  $u_a$  always assigns the maximum helpfulness rating to all reviews written by another

user  $u_b$ , then one can doubt that user  $u_a$  intentionally promote the influence of user  $u_b$ . Note that we only target the ratings with the highest score only since we focus on push attack. Of course, pairs of normal users could reveal clues to the group attack due to the coincidence of opinions about items or reviews. However, all fake users have to involve in many connections through the relationships associated with group attack as a necessity to maximize the degree of manipulation. With this in mind, we suspect the truthfulness of a helpfulness rating if the rater and the reviewer are frequently connected through the mentioned relationships

User2Vec consists of two steps. In the first step, we sample user pairs that reveal clues to the group attack. Sampling user pairs corresponding to the first relationship proceeds as follows. Among the items having at least two reviewers who rate the item with the highest possible rating, we first sample an item with the probability proportional to the cardinality of the users associated with the item and choose two reviewers for the sampled item uniformly at random. Sampling user pairs corresponding to the second and last relationship associated with the group attack involves in sampling reviews. For a review to be a sample, it should receive at least two highest ratings. The probability of sampling a review is proportional to the number of the helpfulness raters for the review. We choose two helpfulness raters who rate the review with the highest possible rating for the sample related to the second relationship. With a sampled review, we sample one supporter for the review and produce a pair of reviewer and supporter. In the last step, we feed the sampled user pairs into the Skip-gram model and obtain feature vectors of users. We expect the obtained feature vectors encode group attack patterns. In other words, fake users are very closely located to each other in the feature space, while normal users are scattered. Note that User2Vec, which places the fake users very close to each other in the feature space, does not guarantee that normal users are positioned away from each other in the feature space. However, if the dimension of the feature space is moderately high, the probability that the similarity of two arbitrarily selected users is high is very small. Therefore, although there is a risk of judging false positives, we judge that the relationship between two users with very high similarity is not trustful and define the suspiciousness of a helpfulness rating as a function of the similarity between the feature vectors of the helpfulness rater and the reviewer.

## 5.2 Robust Review Quality Measure

We assume that the true quality of a review can be estimated by the mean of authentic helpfulness ratings. However, in the presence of fake helpfulness ratings, we need to estimate the true quality of a review by the weighted mean of helpfulness ratings where the fake helpfulness ratings have very low weight. If the amount of helpfulness ratings is sufficiently high, then this estimation get high confidence. However, for reviews that have few helpfulness ratings and reviews having only fake helpfulness ratings, the weighted mean is not robust estimator for such reviews. Say a fake review which received only fake helpfulness ratings, then the weighted mean output is biased toward fake helpfulness ratings even if the weight of fake helpfulness ratings is almost zero. With all of these things in mind, we define the following review quality measure to estimate the true quality of a review.

$$Quality(u, i) = \frac{w_{prior}Q_{default} + \sum_{v \in \{x | \mathbf{H}_{v,u,i} \neq ?\}} T(v, u) \times \mathbf{H}_{v,u,i}}{w_{prior} + \sum_{v \in \{x | \mathbf{H}_{v,u,i} \neq ?\}} T(v, u)} \quad (5.1)$$

We take Bayesian average approach [] that incorporates both a prior belief and a weighted mean of review helpfulness ratings associated with a review  $r$ . The prior quality  $Q_{default}$  works as prior belief.  $w_{prior}$  is

the weight given to the prior belief ( $Q_{default}$ ). In this work, we set  $Q_{default}$  as the mean of helpfulness rating range (e.g. 2.5 in range from 0 to 5), and  $w_{prior}$  as 1. The weight of a review helpfulness rating in review quality estimation is determined by the following function  $T : U \times U \rightarrow R$ .

$$T(v, u) = \begin{cases} \exp(-\mu \times (\cosine(userVec_v, userVec_u) - \theta)) & \text{if } \cosine(userVec_v, userVec_u) \geq \theta \\ 1 & \text{otherwise} \end{cases} \quad (5.2)$$

The function  $T$  takes a rater and a reviewer, and output the trustfulness of the relationship between them.  $T$  penalizes the trustfulness if the cosine similarity between the two feature vectors of the reviewer and the helpfulness rater is larger than the threshold  $\theta$ . As mentioned eariler, this policy might lower the weight of authentic helpfulness ratings, but the likelihood of making such a misjudgment is low in the moderately high dimensional feature spaces.  $\mu$  is a constant for amplification of similairty In this work, we set the  $\theta$  as 0.8 and the  $\mu$  as 100

With the above mentioned robust measure, the quality of reviews with few helpfulness ratings will be close to the default quality  $Q_{default}$ , while reviews with many helpfulness ratings given by the users whose similarity to the reviewer is not that high will have a quality score close to its average helpfulness rating. Most importantly, reviews with many helpfulness ratings from the users similar to reviewer will have a helpfulness score close to the default quality  $Q_{default}$ . In other words, our measure prevent fake helpfulness ratings from manipulating the quality of their fake review.

---

**Algorithm 1** User2Vec algorithm

---

**function** USER2VEC(dimensions  $d$ , num\_samples  $n$ , item rating matrix  $\mathbf{R}$ , helpfulness rating tensor  $\mathbf{H}$ )

Initialize  $clues$  to *empty*

**for**  $iter = 1$  to  $n$  **do**

append EnthusiastPair( $\mathbf{R}$ ) to  $clues$

append SupporterPair( $\mathbf{H}$ ) to  $clues$

append ReviewerSupporterPair( $\mathbf{H}$ ) to  $clues$

**end for**

$userVec = \text{Skip-Gram}(clues, d)$

**return**  $userVec$

**end function**

**function** ENTHUSIASTPAIR(item rating matrix  $\mathbf{R}$ )

let  $Enthusiast(item)$  be  $\{u | \mathbf{R}_{u,item} = ir_{max}\}$

let  $EI$  be  $\{item | |Enthusiast(item)| \geq 2\}$

sample item  $i$  from  $EI$  with the prob. proportional to the cardinality of  $Enthusiast(i)$

sample user  $u, v$  from  $Enthusiast(i)$  uniformly at random

**return**  $(u, v)$

**end function**

**function** SUPPORTERPAIR(review helpfulness rating tensor  $\mathbf{H}$ )

let  $Supporter(u, i)$  be  $\{v | \mathbf{H}_{v,u,i} = hr_{max}\}$

let  $SU$  be  $\{(u, i) | |Supporter(u, i)| \geq 2\}$

sample review  $(u, i)$  from  $SU$  with the prob. proportional to the cardinality of  $Supporter(u, i)$

sample user  $u_a, u_b$  from  $Supporter(u, i)$  uniformly at random

**return**  $(u_a, u_b)$

**end function**

**function** REVIEWERSUPPORTERPAIR(review helpfulness rating tensor  $\mathbf{H}$ )

let  $Supporter(u, i)$  be  $\{v | \mathbf{H}_{v,u,i} = hr_{max}\}$

let  $SU$  be  $\{(u, i) | |Supporter(u, i)| \geq 1\}$

sample review  $(u, i)$  from  $SU$  with the prob. proportional to the cardinality of  $Supporter(u, i)$

sample user  $v$  from  $Supporter(u, i)$  uniformly at random

**return**  $(u, v)$

**end function**

---

---

**Algorithm 2** Robust recommendation system

---

**function** RRS(dimensions  $d$ , num\_samples  $n$ , item rating matrix  $\mathbf{R}$ , helpfulness rating tensor  $\mathbf{H}$ )

$userVec = \text{User2Vec}(d, n, \mathbf{R}, \mathbf{H})$

**for** all review  $(u, i)$  **do**

$$\mathbf{W}_{u,i} = \text{Quality}(u, i) = \frac{w_{prior}Q_{default} + \sum_{v \in \{x | \mathbf{H}_{v,u,i} \neq ?\}} T(v, u) \times \mathbf{H}_{v,u,i}}{w_{prior} + \sum_{v \in \{x | \mathbf{H}_{v,u,i} \neq ?\}} T(v, u)}$$

**end for**

Optimize  $\text{Cost}(\mathbf{U}, \mathbf{V} | \mathbf{W}, \mathbf{R}) = \sum_{\mathbf{R}_{i,j} \neq ?} \mathbf{W}_{i,j} (\mathbf{R}_{i,j} - (\mathbf{UV})_{i,j})^2 + \lambda (\|\mathbf{U}\|_F^2 + \|\mathbf{V}\|_F^2)$

**return**  $\mathbf{U}$  and  $\mathbf{V}$

**end function**

---

## Chapter 6. Experiment

### 6.1 Experimental Setting

We use the publicly available dataset provided by [16], namely CiaoDVD. The CiaoDVD dataset contains users' review and helpfulness rating information from [ciao.dvd.co.uk](http://ciao.dvd.co.uk) where users rate DVDs and others' reviews. In the CiaoDVD dataset, users can rate items with a score from 1 to 5 as reviewer, and rate the helpfulness of reviews with a score from 0 to 5 as helpfulness rater. From the original dataset, we filter out reviewer who rated less than five items and item that received less than five ratings. The statistics of the resulting dataset are shown in Table 6.1.

Table 6.1: Statistics of CiaoDVD dataset

Features	CiaoDVD
Reviewers	1822
Items	2069
Reviews	28374
Helpfulness Rater	27900
Helpfulness Rating	661040

We assume the original data is authentic. To this data, we inject fake users, item ratings(reviews) and helpfulness ratings as mentioned in the Chapter 4. Attack size, i.e. the number of injected fake users, ranges from 1% to 3% of total users. Filler size is the number of  $I^{filler}$  and popular size is the number of  $I^{popular}$ . We restrict the sum of filler size and popular size from exceeding 1% of the total number of items. We assume attacks with larger attack size and filler/popular size would be detected easily, so we decide to exclude them in our experiment setting. For performance evaluation, we perform 10-fold cross validation. In each fold, the test set contains random 10% original reviews, and the training set contains the remaining 90% original reviews and all the fake reviews. We choose a set of target items  $I^{target}$  as items which have been rated by at least 1% users with below the median of the rating scale (3 in our rating scale [1,5]).  $I^{popular}$  consists of items rated items by at least 1% users with the  $ir_{max}$ . Note that  $I^{filler}$  is randomly selected for each fake user

### 6.2 Metrics

**Average helpfulness of reviews** measures the average helpfulness of reviews belonging to each category. We compare the average helpfulness values of fake reviews and authentic reviews to find out the robustness of a helpfulness measure. A robust helpfulness measure would produce the small average helpfulness of fake reviews even in the presence of fake helpfulness ratings.

$$AverageHelpfulness(IR) = \frac{1}{N_{IR}} \sum_{(u,i) \in IR} helpfulness(u,i) \quad (6.1)$$

where  $N_{IR}$  is the number of reviews in the set of review  $IR$

**Prediction shift on the target items** measures the average of change in the prediction of genuine users for the attacked items before and after a shilling attack. In other words, this metric measures the degree of success of an attack. The smaller the value of this metric, the more robust the recommendation method is.

$$PredictionShift(U, V, U', V') = \frac{1}{|U^g||I^{target}|} \sum_{u \in U^g} \sum_{i \in I^{target}} (U'V')_{u,i} - (UV)_{u,i} \quad (6.2)$$

where  $(U'V')_{u,i}$  is the predicted rating value of user  $u$  for item  $i$  after an attack

**Mean Average Error(MAE) on test set** is the overall prediction error on ratings in the test set which contains 10% of all item ratings of original users in the dataset. MAE is commonly used to compare the predictive accuracy of recommendation algorithms. In this paper we use MAE to measure the accuracy loss that is sacrificed to improve robustness.

$$MAE(U, V) = \frac{1}{|testset|} \sum_{R_{u,i} \in testset} |R_{u,i} - (UV)_{u,i}| \quad (6.3)$$

### 6.3 Results and Analysis

We first visualize the results of User2Vec to show User2Vec’s ability to capture fake users. We use learned users’ feature vectors as the input to the visualization tool-t-SNE [?]. The users are mapped to the 2-D space. Diamond-shaped points represent fake user, while x-shaped green colored points represent normal users. We observed that feature vectors of fake users are very close to each other.

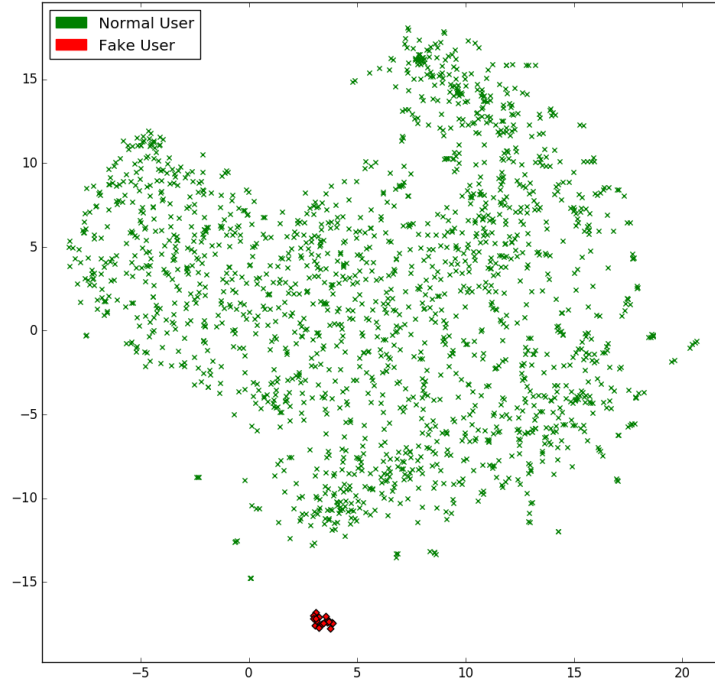


Figure 6.1: Visualization of the User2Vec result

We compute the average of the helpfulness of fake reviews and authentic reviews. We inject fake review through attacks with 1% attack size, 0.5% filler size, and 0.5% popular size. We set dimensions

of feature vectors to 32, which is a parameter needed in our helpfulness measure. As shown in table 6.2, the naive helpfulness measure results in fake reviews have a higher helpfulness than authentic reviews, whereas our helpfulness measure yields the opposite. In specific, while the naive helpfulness measure computes the helpfulness of fake reviews at a value close to  $hr_{max}$ , our helpfulness measure computes the helpfulness of fake reviews at a value close to the default helpfulness  $H_{default}$ .

Table 6.2: Review helpfulness results. The range of helpfulness is from 0 to 5

Attack Size	Naïve Helpfulness Measure		Our Helpfulness Measure	
	Fake Reviews	Authentic Reviews	Fake Reviews	Authentic Reviews
1%	5.0	3.5388	2.5	3.45

To compare the robustness, we compute the prediction shift on the target items of algorithms using different review helpfulness measures under attacks in variety of attack sizes, filler sizes, and popular sizes. From Table 6.3, we observe that our helpfulness measure leads to the lowest prediction shift on the target items in all conditions. The reason for this is that fake item ratings have the least impact on prediction when applying our helpfulness measure rather than applying other methods. Since fake helpfulness ratings increase the influence of fake item ratings on prediction when applying the naive helpfulness measure than when ignoring helpfulness, the naive helpfulness measure causes larger prediction shift on the target items than basic MF in the presence of fake helpfulness ratings. Therefore we argue that our method resistant to review helpfulness manipulations adds robustness to WMF.

Table 6.3: Prediction shift on the target items

Attack size	Filler Size	Popular Size	Base	Naive	Ours
1%	1%	0%	1.01465	1.7805	<b>0.092686</b>
	0.50%	0.50%	1.48154	1.90081	<b>0.239489</b>
	0%	1%	1.72337	1.79765	<b>0.246163</b>
2%	1%	0%	1.5178	2.28872	<b>0.172815</b>
	0.50%	0.50%	1.91454	2.30772	<b>0.387328</b>
	0%	1%	2.08557	2.1092	<b>0.469354</b>
3%	1%	0%	1.79902	2.5618	<b>0.307992</b>
	0.50%	0.50%	1.89458	2.49592	<b>0.580003</b>
	0%	1%	1.99723	2.24161	<b>0.659305</b>

We also investigated the predictive performance of relying on each review helpfulness measure. We compute MAE on the test set when the training set contains fake item ratings. According to 6.4, basic matrix factorization performs better than others. Since MAE gives equal importance to the errors of all the ratings in the test set, it is not ideal metric to measure the performance of weighted matrix factorizations which allow unimportant ratings to have significant errors. Even though MAE is not proper metric for weighted matrix factorization, the performances are not significantly different from basic matrix factorization. We compute Cohen’s  $d$  for the effect size based on means of predictive errors between basic MF and weighted MF using our helpfulness measure. The value of  $d$  is near 0.02, which is a small value according to [18]. Therefore, weighted matrix factorization using our helpfulness measure



provides robustness at a not significant additional cost of predictive accuracy.

Table 6.4: MAE on test set

Attack Size	Filler Size	Popular Size	Base	Naive	Ours
1%	1%	0%	0.821944	0.831673	<b>0.843252</b>
	0.5%	0.5%	0.825909	0.836473	<b>0.842159</b>
	0%	1%	0.825078	0.837706	<b>0.84074</b>
2%	1%	0%	0.820305	0.826343	<b>0.840074</b>
	0.5%	0.5%	0.826043	0.834851	<b>0.836148</b>
	0%	1%	0.822775	0.847235	<b>0.843203</b>
3%	1%	0%	0.816267	0.828714	<b>0.840653</b>
	0.5%	0.5%	0.825463	0.846773	<b>0.841859</b>
	0%	1%	0.829685	0.85275	<b>0.841454</b>

## Chapter 7. Conclusion

This paper proposes a robust review helpfulness measure so that collaborative filtering using review helpfulness information successfully restricts the effect of shilling attacks. Specifically, given an item rating matrix and a review helpfulness rating tensor, we learn representations of users in recommendation system. These representations encode behaviors associated with shilling attack. Armed with user representations, we estimate the true helpfulness of reviews by the Bayesian weighted average of review helpfulness ratings. We penalize the weight of a helpfulness rating if the helpfulness rater and reviewer are suspected of having a relationship of self-rating. Experimental results on a real-world dataset demonstrate the robustness of our method against review helpfulness manipulation. Future research directions include developing robust measures against nuke attacks and more elaborate attacks.

## Bibliography

- [1] M. Luca. Reviews, Reputation, and Revenue: The Case of Yelp.com. Harvard Business School Working Papers, 2011. <http://www.hbs.edu/research/pdf/12-016.pdf>
- [2] Koren, Yehuda, Robert Bell, and Chris Volinsky. "Matrix factorization techniques for recommender systems." *Computer* 42.8 (2009): 30-37.
- [3] B. Mehta, T. Hofmann, and W. Nejdl, "Robust Collaborative Filtering," *RecSys*, p. 49, 2007.
- [4] B. Mehta and W. Nejdl, "Attack resistant collaborative filtering," *Proc. 31st Annu. Int. ACM SIGIR Conf. Res. Dev. Inf. Retr. SIGIR 08*, p. 75, 2008.
- [5] S. Kim, P. Pantel, T. Chklovski, and M. Pennacchiotti. Automatically assessing review helpfulness. *EMNLP*, 2006
- [6] Liu, Jingjing, et al. "Low-Quality Product Review Detection in Opinion Summarization." *EMNLP-CoNLL*. Vol. 7. 2007.
- [7] Ott, Myle, et al. "Finding deceptive opinion spam by any stretch of the imagination." *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*. Association for Computational Linguistics, 2011.
- [8] S. Raghavan, S. Gunasekar, and J. Ghosh. Review quality aware collaborative filtering. In *Proceedings of the sixth ACM conference on Recommender systems*, pages 123–130. ACM, 2012.
- [9] S. Wang, J. Tang, and H. Liu, "Toward Dual Roles of Users in Recommender Systems," *Cikm*, pp. 1651–1660, 2015.
- [10] Mehta, Bhaskar, Thomas Hofmann, and Peter Fankhauser. "Lies and propaganda: detecting spam users in collaborative filtering." *Proceedings of the 12th international conference on Intelligent user interfaces*. ACM, 2007.
- [11] Mehta, Bhaskar. "Unsupervised shilling detection for collaborative filtering." *AAAI*. 2007.
- [12] B. Perozzi, R. Al-Rfou, and S. Skiena. DeepWalk: Online learning of social representations. In *KDD*, 2014.
- [13] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. In *ICLR*, 2013.
- [14] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, 2013.
- [15] Grover Aditya and Leskovec Jure. "node2vec: Scalable Feature Learning for Networks." *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2016.

- [16] Guo, Guibing, et al. "Etaf: An extended trust antecedents framework for trust prediction." *Advances in Social Networks Analysis and Mining (ASONAM), 2014 IEEE/ACM International Conference on*. IEEE, 2014.
- [17] Hu, Yifan, Yehuda Koren, and Chris Volinsky. "Collaborative filtering for implicit feedback datasets." *2008 Eighth IEEE International Conference on Data Mining*. Ieee, 2008.
- [18] Cohen, Jacob (1988). *Statistical Power Analysis for the Behavioral Sciences*. Routledge. ISBN 1-134-74270-3.
- [19] L. Van der Maaten and G. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(2579-2605):85, 2008

## Acknowledgments in Korean

감사합니다.

## Curriculum Vitae in Korean

이 름: 심 동 진  
생 년 월 일: 1992년 8월 4일  
출 생 지: 경기도 송탄시  
전 자 주 소: djsim@kaist.ac.kr

### 학 력

2008. 3. – 2011. 2. 거창고등학교  
2011. 3. – 2014. 8. 성균관대학교 소프트웨어학과 (학사)  
2015. 3. – 2017. 2. 한국과학기술원 전산학부 (석사)

### 경 력

2015. 9. – 2016. 8. 한국과학기술원 전산학부 조교