Master's Thesis

Rating Scheme Analysis for Recommendation System

Youngchul Sung (성 영 철)

Department of Computer Science and Engineering

Pohang University of Science and Technology

2013

추천 시스템을 위한 평점 정보 분석 및 활용 방법

Rating Scheme Analysis for Recommendation System

Rating Scheme Analysis for Recommendation System

by
Youngchul Sung

Department of Computer Science and Engineering
Pohang University of Science and Technology

A thesis submitted to the faculty of Pohang University of Science and Technology in partial fulfillment of the requirements for the degree of Master of Science in the Department of Computer Science and Engineering

Pohang, Korea
06. 12. 2013
Approved by

____
Major Advisor

Rating Scheme Analysis for Recommendation System

Youngchul Sung

The undersigned have examined this thesis
and hereby certify that it is worthy of acceptance for a master's
degree from POSTECH

06/12/2013

Committee	Chair	Hwanjo Yu	(Seal)
	Member	Jong-Hyeok Lee	(Seal)
	Member	Hee-Kap Ahn	(Seal)

M C S E 성 영 철 Youngchul Sung
20111077 Rating Scheme Analysis for Recommendation System, 추천 시스템을 위한 평점 정보 분석 및 활용 방법
Department of Computer Science and Engineering, 2013, 100 P,
Advisor : Hwanjo Yu. Text in English.

ABSTRACT

Abstract

With the growth of Internet, the huge and ever increasing amount, complexity and heterogeneity of available digital information overwhelm the human processing capabilities in a wide array of information seeking and several tasks. To cope with information overload recommendation systems have been introduced to filter those items that are of low relevance or utility for the user, and present only a small selection better suiting the user's tastes and interests. Usually they used rating information which is given a user to develop methods or improve accuracy.

Although the rating system is close to recommendation system, they do not verify whether original rating scheme is appropriate to their methods. In this paper, we suggest various forms of rating scheme for recommendation system to verify the effect of them. The proposed schemes include three concepts: filtering, scaling and biasing. To validate the idea, we run experiments under existing CF approaches for recommendation consisting of k-nearest neighbor, ItemRank and matrix factorization with various parameters which are matched to each method. The results are explained as three evaluation measures: RMSE, Precision-recall and Kendall's τ. Analyzing results, we confirmed that matrix factorization and RMSE are sensetive to biasing scheme, and filtering helps relative ordering or choosing top-k items which are correlated to two methods, KNN and ItemRank, and two measures, F1-score

and Kendall's τ. Future researches will be able to extend these results to various directions by optimizing methods and parameters.

Contents

1 Introduction **1**

2 Related Work **3**

2.1 Collaborative Filtering Apporoach 4

2.2 K-Nearest Neighbors 5

2.3 Matrix Factorization 6

2.4 ItemRank 8

3 Rating Scheme Candidates **10**

3.1 Scaling 11

3.2 Biasing 12

3.3 Filtering 13

4 Experimental Environment **14**

4.1 Dataset 15

4.2 Recommendation Method 16

4.2.1 K-Nearest Neighbors 16

4.2.2 Matrix Factorization 17

4.2.3 ItemRank 18

4.3 Evaluation Measure 19

4.3.1 Root mean squared errors 19

4.3.2 Precision-recall 20

4.3.3 Kendall's tau 21

5 Result **22**

5.1 K-Nearest Neighbors 23

5.2 ItemRank 25

5.3 Matrix Factorization 26

6 Discussion 27

7 Conclusion 29

List of Figures

1 scaling models 11
2 example of KNN 17

List of Tables

1 Rating Scheme 22
2 User-knn Result (knn_o, k: 9) 23
3 Item-knn Result (knn_o, k: 9) 24
4 ItemRank Result (ϵ: 0.01) 25
5 Matrix Factorization Result (num_{lf}: 10) 26
6 Effects of Biasing to Methods (RMSE) 27
7 Effects of $filter_p$ to Methods 28
8 Effects of exp_m to Methods 28
9 Effects of exp_l to Methods 28

1 Introduction

Due to the immense amount of information available, recommendation has become a main challenge in data mining. Because users face the problem of choosing the right products or services, a variety of recommender systems have been introduced to help users select relevant products or services. We have still real-world problems which needs recommendation, for example, the music [1], TV programs [2], social networks [3] [4] and movie [5]. The systems use profiles or feedback from users to recommend items which are similar to their preferences.

The rating system is closely related to the recommendation methods. It helps to diversify researches for recommendation because rating gives numerical resources which mathmatical approaches can access. We can explain user's preference using rating scores easier than binary system, whether he/she watched the item, also. However, we have no previous researches which studied rating system until now. [6] introduced multicriteria ratings and [7] analyzed histograms of user ratings, but even these works did not concentrate on the nature of rating scheme.

With the motivation that there are few previous researches for rating schemes, we suggest several rating schemes which are novel or existing candidates in this paper. We implement well-known recommendation methods and run experiments under various parameters to analyze the relationship between rating schemes and recommendation methods, or between rating schemes and evaluation measures. Additionally, the direction of this study will be predicted as conclusion.

This paper is organized as following. Section 2 introduces existing works which are related to this research, Section 3 presents variations of proposed rating schemes. Section 4 presents recommendation methods and evaluation measures which are used in the experiment, Section 5 shows the summarized results and Section 6 discusses them. Finally Section 6 concludes our whole steps.

2 Related Work

The study what we explain in the paper is based on several researches for recommendation. In the area of data mining, recommendation is a sub-problem of information filtering system which seeks to predict the rating or preference that the user would give to an item, such as music, books, or movies they had not yet considered. The system uses a model built from the characteristics of an item, called content-based approaches, or the user's social environment, called collaborative filtering approaches (CF). We will introduce existing studies for CF, which is used for our experiment in this section. Additionally, introductions for three representative methods for CF, k-nearest neighbors, matrix factorization and itemrank, are followed.

2.1 Collaborative Filtering Apporoach

Unlike content-based approaches, which use the content of items previously rated by a user, CF approaches rely on the ratings of users as well as those of other users in the system [9]. Because it is well-known topic for recommendation, [10] [11] presented a survey of CF techniques. The motivation for CF comes from the concept that a user often get the best recommendation form someone with similar preference. CF explores many techniques to match persons with similar interests and to make recommendation on this basis. To realize this concept, CF requires three components: users' active participations, an simple way to represent users' interests to the system, and algorithms which are able to match people with similar interests.

The whole process of general CF can be summerized as three following steps. (1) The system obtains dataset which a user graded his/her preferences by rating items. We refers these ratings as an approximate representation of users' interests in the corresponding domain. (2) Then, the system matches these ratings with other users' and finds the people who has the most similar tastes. (3) Finally we recommend the items which have rated highly by other users but not rated by target user because they would be likely to have high scores according to similarity. Based on this process, growing challanges are how to combine and weight the preferences of user neighbors. We introduce three methods which propose the solution for these CF challanges at the following sections.

2.2 K-Nearest Neighbors

Among CF approaches, methods based on nearest-neighbors still enjoy a huge amount of popularity, due to their simplicity, their efficiency, and their ability to produce accurate and personalized recommendations [8]. K-nearest neighbors (KNN) algorithm is a non-parametric method to classify target objects based on nearest training examples in the area of pattern recognition. KNN is called as a type of instance-based learining or lazy learining because the function is only approximated locally and all computation is deferred until starting classification. The process is that an object is assigned by a majority vote of its neighbors, with the object being assigned to the class most common amongst its k nearest neighbors. For intance, if k is equal to 1, then the object is simply assigned to the class of its nearest neighbor.

In the paper, we used KNN for regression, a statistical technique for estimating the relationships among variables. We simply assign the property value for the object to be the average of the values of its k nearest neighbors. It can be useful to weight the contributions of the neighbors, so that the nearer neighbors contribute more to the average than the more distant ones. Many existing researches have studied weighting schemes for regression using nearest neighbors. One common scheme of various schemes is a generalization of linear interpolation, which gives each neighbor a weight which is inversely proportional to the distance from a target object.

2.3 Matrix Factorization

In contrast to neighborhood mothods like KNN, there is another primary approach of CF, latent factor models (LF). LF characterizes both items and users on several factors inferred from the rating patterns to try to explain the rating values. For example, the discovered factors might measure obvious dimensions such as comedy versus drama, amount of action, or orientation to children in the case of movie. Matrix factorization (MF) is one of the most successful realizations of LF. MF characterizes both items and users by vectors of factors inferred from item rating patterns. High correspondence between item and user factors leads to recommendation. MF has become popular in recent years by combining good scalability with predictive accuracy and it offers much flexibility for modeling various real-world situations.

MF maps both users and items to a joint latent factor space of dimensionality f, such that user-item interactions are modeled as inner products in that space. Accordingly, each item i is associated with a vector $q_i \in \Re^f$, and each user u is associated with a vector $p_u \in \Re^f$. Then, the resulting dot product, $q_i^T p_u$, captures the interaction between user u and item i, the user's overall interest in the item's characteristics. It can be denoted by r_{ui}, leading to the estimate

$$\hat{r}_{ui} = q_i^T p_u. \tag{1}$$

Now, the major challenge is computing the mapping of each item and user to factor vectors q_i and p_u. When the recommender system completes this mapping, it can easily estimate the rating a user will give to any item by using Equation (1).

Such a model is closely related to singular value decomposition (SVD), a well-established technique for identifying latent semantic factors in information retrieval. Applying SVD in the CF domain requires factoring the user-item rating matrix. This often raises difficulties due to the high portion of missing values caused by sparseness in the user-item ratings matrix. In the paper, we use simple SVD for MF approaches and propose biasing, which performs the experiment to consider these difficulties for CF methods.

2.4 ItemRank

Based on the fact that a recommender system deals with a set of users and items, its goal consists of computing a score that measures the expected interest of users for items on the basis of a knowledge base containing a set of preferences expressed by some users about items. So we need a scoring algorithm to rank items for every given user according to its expected preferences, then a recommender system will suggest to a user top-ranked items with respect to personalized ordering. In a view of this idea, [14] proposed novel CF methods for recommendation, ItemRank.

ItemRank is based on the idea that we can use the model expressed by the Correlation Graph to forecast user preferences. The correlation graph is a valuable graphical model useful to exploit correlation between movies, weights associated to links provide an approximate measure of movie/movie relative correlation, according to information extracted from ratings expressed by users in the training set. The original study of this concept is PageRank [18], which has both propagation and attenuation properties we need, furthermore thanks to significant research efforts we can compute PageRank in a very efficient way [19]. The classic PageRank says that an importance score $\mathbf{PR}$ is defined as:

$$\mathbf{PR} = \alpha \cdot \mathbf{M} \cdot \mathbf{PR} + (1 - \alpha) \cdot \mathbf{d} \tag{2}$$

where $\mathbf{M}$ is a stocahstic matrix, its non-negative entries has to sum up to 1 for every column, and vector $\mathbf{d}$ has non-negative entries summing up to 1.

ItemRank equation can be easily derived from equation 3. A ItemRank value $\mathbf{IR}_{u_i}$ can be expressed as the equation:

$$\mathbf{IR}_{u_i} = \alpha \cdot \mathbf{C} \cdot \mathbf{IR}_{u_i} + (1 - \alpha) \cdot \mathbf{d}_{u_i} \tag{3}$$

where $\mathbf{C}$ is Correlation Matrix, $\mathbf{d}_{u_i}$ has been built according to user u_i preferences as recorded in training set. The interpretation of $\mathbf{IR}_{u_i}$ scorevector for user u_i is straightforward, ItemRank scores induce a sorting of movies according to their expected liking for a given user. The higher is the ItemRank for a movie, the higher is the probability that a given user will prefer it to a lower score movie.

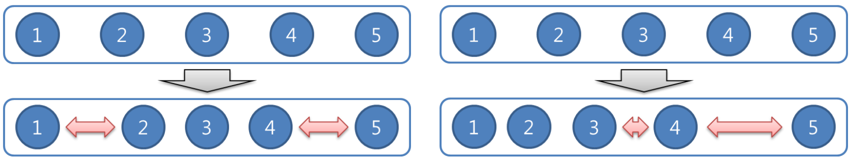
3 Rating Scheme Candidates

The rating scheme of existing dataset is linearly-ordered integers. For example, This paper used movieLens dataset has five levels, from 1 to 5, to grade preference of an item from a user. The higher rating means that a user feels greater or more meaningful to the item, the lower one means the item gives an user worse or meaningless feeling. In this environment, we can say that an item which got score 1 is the worst item and what we do not need to recommend it, the other way, we should recommend items which got score 5 because the user is likely to love it. In this view, intermediate values, 2–4, show suitable preferences matched to each value. Most of existing recommendation methods used this scheme for their experiments.

However, a few reasearches raise objections about using basic scheme to run their algorithms intactly. Does each interval between two discrete scores reflect a preference gap of a user? Is the higher rating more valuable than the lower one to recommend items? How do we control unseen items conceptually? Based on many questions about the original scheme, we designed three categories: scaling, biasing and filtering to generate variations of rating scheme. Briefly, scaling considers to change interval values by applying some exponential translations and biasing controls ratings for unseen items like missing value problems. Finally we filter out some ratings which satisfy given conditions in filtering category. The following steps will explain details for each category.

3.1 Scaling

The motivation of scaling is the validity of each interval between two discrete scores. For instance, an interval between 3 and 4 is the same with one between 4 and 5 numerically. But if an user gives 5 to any item, then it has significant meaning because it is the highest score among five levels. In this view, we may argue that the difference between 4 and 5 needs to have more potential gap than the one between 3 and 4. Thus, We prepare two type of exponential translation to confirm these intuitions. The first candidate is an exponential translation from the medium (exp_m), that is, emphasize end-values, 1 and 5, relatively. We fix the medium value 3 and each ratings are translated to new value which is proportional to the squared distance from the fixed value. The other one is an exponential translation from the lowest value (exp_l), which gives more weight to bigger ratings. This fix the lowest value 1 and other ratings obtain weights proportional to the squared distance from it. These two scaling models are illustrated in Fig. 1.



(a) exp_m (b) exp_l

Figure 1: scaling models

3.2 Biasing

Each user has unique charactoristics for scoring items and each item also has unique one for obtaining scores. The most representative property among them is bias, which means how partial a rating of target user or item is. The bias has been actively researched on the area of matrix factorization because matrix factorization is the closest to missing value problem, which estimates unknown ratings, among recommendation techniques. When we encounter unseen data, one simple answer is just 3 ($bias_m$) because it is medium value of rating levels. Another candidates is the average value of all ratings ($bias_a$) in a view that entire dataset is possible to be biased to any directions. Addltionally, each user or each item tends to give or take a biased score usually. We design the personalized bias ($bias_p$), an average value of each user or each item respectively, to reify this concept.

3.3 Filtering

We may raise a real-world question whether low values is valuable to analyze user's preference to the item. General recommender system considers only what items will be prefered by users. Therefore we can say that low ratings are not useful and suitable to applying it to simulate recommendation algorithms. Specifically, when we face high ratings, 4 or 5, we need to separate 5 from 4 because 5 is the most appropriate to recommend items . But items which got 1 or 2 are not sensitive to consider about recommendation because users are not likely to give high ratings to both of them. Reflecting this concept, we design filtering idea which consider only positie ratings and compare them with original one by evaluating their results ($filter_p$).

4 Experimental Environment

The study of rating scheme what we have introduced until now is a pre-processing step of entire work for recommendation. We need to choose the method and evaluation measure to analyze the efficiency from setting rating scheme. Furthermore, It will be additional study what effect each rating scheme gives methods and evaluation measures. We would find the best combination among them based on the result of this study. In this section, we will explain the dataset we used, recommendation methods and evaluation measures what we used at these experiments.

4.1 Dataset

We used MovieLens dataset which GroupLens Research has collected and made available rating data sets from the MovieLens web site [1]. It was collected over various periods of time, depending on the size of the set. The chosen dataset is MovieLens 100k dataset, consists of 100,000 ratings (1-5) from 943 users on 1682 movies. Each user has rated at least 20 movies and they support simple demographic infomation for the users, also.

[1] http://movielens.umn.edu

4.2 Recommendation Method

Many researches for recommendation have developed various methods to make accuracy higher and satisfy the preference of the user. Among them, the well-known approaches we chose are k-nearest neighbor (KNN) [12], matrix factorization (MF) [13] and ItemRank [14]. Following sections introduce these methods and their parameter variations what we need to consider.

4.2.1 K-Nearest Neighbors

Recommendation using KNN is to estimate the rating by combining ratings of k nearest neighbors in a various way, so we need to tune parameters for KNN to optimize and generalize the result. The first question is how many neighbors we should refer as nearest neighbors, that is, what the value of k is. The problem how to fix the value of k is very sensitive to the accuracy of designed model. For example, the test sample, circle, should be classified either to the first class of squares or to the second class of triangles in Fig 2. If k is equal to three, it is assigned to the second class because there are two triangles and only one square inside the inner circle. If k is five, it is assigned to the first class because of similar calculation. Although this example is based on classification problem, regression using KNN what we used has the same problem. We set various values for k: 3, 5, 7 and 9, to validate the results.

An another topic for KNN is a case that we cannot find nearest neighbors which satisfy the condition. For example, it is possible that nearest neighbors in terms of specific similarity does not have rating information for target item because of sparsity, unfortunately. Then these missed values will be marked as default or random values because there is no clues to guess them. We propose a variation of KNN (knn_r), which considers users which graded ratings for target items firstly where original KNN (knn_o) calculated similarity between

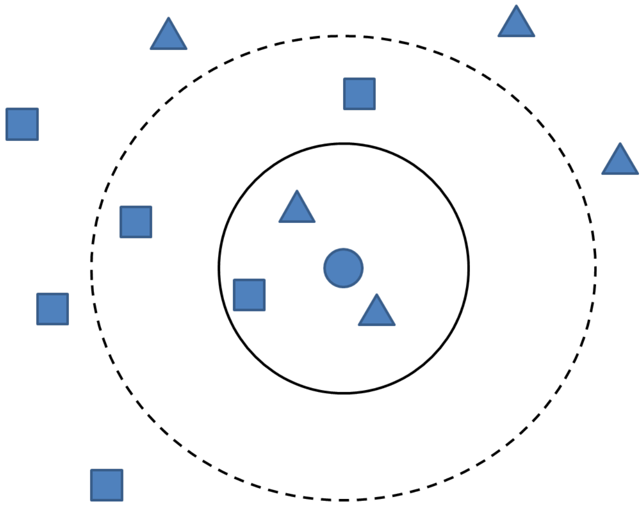


Figure 2: example of KNN

all users firstly.

Finally, We have the choice what domain is for calculating neighbors between users and items. They are called as user-knn and item-knn, respectively. Existing researches run their process after choosing one of them arbitrarily, But there is no explanation why they chose that domain. To confirm the validity of each choice, we deploy our process for both of two domains.

4.2.2 Matrix Factorization

General latent factor models have the common issue how much latent factors are needed for optimizing the result. When we set the number of factors bigger, it fits on the training dataset tightly but it may cause the overfitting problem to test dataset. Otherwise, the small number of factors makes underfitting problem. Due to this problem that is so sensitive to the accuracy, the existing works have used usual technique which checks results from setting several number of factors heuristically because it depends on various environment like recommendation domain, dataset and scale, etc. We used this tuning idea to obtain the confidence of our experiments, the number of latent factors (num_{lf}) is set as 3, 5, 10 and 20.

4.2.3 ItemRank

ItemRank estimates **IR** scores using Correlation Graph iteratively. In applying the learning method using iterative and updating approaches, we proposes two challenging issues: how to stop iterating process and how to fix the portion of current phase to calculated results. However these parameters do not affect the quality of results seriously because they are closely correlated to time-efficiency, not accuracy. So we define the stopping rule as an epsilon value ϵ, which means the difference sum between current values and previous values. The decay factor α, which means reflecting ratio of current change, is fixed as the well-known value, 0.85 [14].

4.3 Evaluation Measure

Evaluation work is the structured interpretation and gives the meaning of predicted or actual impacts of proposals or results. We focused on rating scheme and recommendation methods until now, the quality of result depends on evaluation measures, also. For instance, we suggest a situation that method A is developed to find only the highest rating, where method B is optimized to obtain top-k. Then if current measure requires the quality of result by only top score, it will decide A is the better recommender than B. But if a measure scans the accuracy of more ratings, it will choose method B. In a view of this concept, we can reach the curiousity which combination between methods, rating schemes and evaluation measures are the best or well-suitable. In this section, we introduce representative evaluation measures which we used for experiment.

4.3.1 Root mean squared errors

The root-mean-squared error (RMSE) is a frequently used measure of the differences between values predicted by a model or an estimator and the values actually observed [20]. Because all methods which we used in the experiment can compute predicted ratings, it is easy to obtain RMSE values for each of them. The formula is:

$$\mathbf{RMSE} = \sqrt{\frac{\sum_{t=1}^{n}(r_t - \hat{r}_t)^2}{n}}, \tag{4}$$

where n is the number of samples, $\hat{r}$ is a predicted rating and r is a predicted rating. RMSE serves to aggregate the magnitudes of the errors in predictions for various times into a single measure of predictive power. It is a good measure of accracy, but only to compare forecasting errors of different models for a particular variable and not between variables, as it is dependent to scale.

4.3.2 Precision-recall

In the area of pattern recognition and information retrieval, precision, positive predictive value, is the fraction of retrieved instances that are relevant, while recall, known as sensitivity, is the fraction of relevant instances that are retrieved. Both of them are based on an understanding and measure of relevance [21] [22]. Simply, high recall means that an algorithm returned most of the relevant results, while high precision means that an algorithm returned substantially more relevant results than irrelevant. Equations for precision and recall are defined as:

$$precision = \frac{tp}{tp + fp}, \tag{5}$$

$$recall = \frac{tp}{tp + fn}, \tag{6}$$

where tp is true positive which means corrent result, fp is false positive which means unexpected result and fn is false negative, missing result. But precision and recall is not discussed in isolation usually because they have the trade-off each other. Instead of using individual measure, they are replaced into other one or combined into a single measure, such as F-measure, the weighted harmonic mean of precision and recall. The typical F-measure, F_1, is:

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}. \tag{7}$$

4.3.3 Kendall's tau

Kendall's tau rank distance (τ) is a metric that counts the number of pairwise disagreements between two ranking lists [23]. Specifically, it is for rank correlation, that is, the similarity of the orderings of the data when ranked by each of the quantities. For example, if we have a set of observations $x_1 > x_2 > x_3$, then we have three pairs $x_1 > x_2$, $x_1 > x_3$ and $x_2 > x_3$. If the method guessed $x_1 > x_3 > x_2$, then it generates two correct pairs, $x_1 > x_3$ and $x_1 > x_2$, and one incorrent pairs, $x_3 > x_2$. The exact definition of τ is:

$$\tau = \frac{number\ of\ correct\ pairs}{\frac{1}{2}n(n-1)}. \tag{8}$$

5 Result

In this section, we will explain the summary of results of all experiments because they have too many variations to write here. For each recommendation method, we selected one basic combination of parameters. Table 1 shows the meaning of each rating scheme.

Table 1: Rating Scheme

Rating Scheme	Meaning
$original$	given dataset without modification
$filter_p$	using only positive ratings
exp_m	scaling to exponential from medium value
exp_l	scaling to exponential from lowest value
$bias_a$	biasing to the mean of all dataset
$bias_p$	biasing to each personalized mean

5.1 K-Nearest Neighbors

The results of KNN consist of two types, user-knn and item-knn, they are listed as Table 2 and Table 3, respectively. Two tables are under knn_o, which calculate similarities for all neighbors first, and k is the highest value of our experiment, 9. Analyzing results, $filter_p$ improves F1-score and Kendall's τ slightly for both of two domains. In a view of scaling, exp_m shows a little lower RMSE value, where exp_l makes worse result. We have two options for biasing, $bias_a$ and $bias_p$, they show obviously different results each other. $bias_a$ has no improvement for all cases, where $bias_p$ improves results significantly in terms of RMSE and Kendall's τ.

Table 2: User-knn Result (knn_o, k: 9)

Rating Scheme	RMSE	Precision	Recall	F1	Kendall's τ
original	1.142	0.661	0.751	0.703	0.639
$filter_p$	1.137	0.638	0.821	**0.718**	**0.677**
exp_m	1.135	0.657	0.754	0.702	0.630
exp_l	1.436	0.712	0.515	0.597	0.595
$bias_a$	1.153	0.698	0.609	0.651	0.628
$bias_p$	**1.114**	0.616	0.532	0.571	0.656

Table 3: Item-knn Result (knn_o, k: 9)

Rating Scheme	RMSE	Precision	Recall	F1	Kendall's τ
original	1.108	0.667	0.753	0.707	0.617
$filter_p$	1.156	0.642	0.793	**0.710**	0.618
exp_m	1.092	0.665	0.750	0.705	**0.645**
exp_l	1.329	0.734	0.598	0.659	0.615
$bias_a$	1.103	0.711	0.663	0.686	0.613
$bias_p$	**1.075**	0.647	0.634	0.640	0.642

5.2 ItemRank

ItemRank result is listed as Table 4. It is under the minimal terminating option ϵ=0.01. Filtering scheme, $filter_p$, improves F1-score and Kendall's τ a little but makes RMSE value worse. Otherwise Scaling schemes did not affect the result vastly. exp_m is very similar to *original*, exp_l shows slightly better RMSE value but no difference to the other measures. Finally, $bias_a$ result has no huge difference from *original*, where $bias_p$ improves accuracy significantly in terms of RMSE and Kendall's τ excepting F1-score.

Table 4: ItemRank Result (ϵ: 0.01)

Rating Scheme	RMSE	Precision	Recall	F1	Kendall's τ
original	1.144	0.604	0.875	0.715	0.604
$filter_p$	1.154	0.619	0.864	**0.721**	0.629
exp_m	1.144	0.604	0.874	0.714	0.604
exp_l	**1.139**	0.606	0.871	0.715	0.604
$bias_a$	1.345	0.605	0.872	0.714	0.604
$bias_p$	1.272	0.578	0.843	0.686	**0.671**

5.3 Matrix Factorization

The evaluation for matrix factorization is under the condition that the number of latent factors, num_{lf}, is 10. $filter_p$ shows slightly worse results in terms of RMSE and F1-score, but improves accuracy in terms of Kendall's τ. Observing scaling schemes, exp_m improves only Kendall's τ, where exp_l has no positive effect on all values. Biasing results which are from $bias_a$ and $bias_p$ reduce RMSE values by 0.047 and 0.160, respectively. Additionally, $bias_p$ makes a good result in terms of Kendall's τ.

Table 5: Matrix Factorization Result (num_{lf}: 10)

Rating Scheme	RMSE	Precision	Recall	F1	Kendall's τ
$original$	1.127	0.653	0.906	0.759	0.718
$filter_p$	1.130	0.582	0.968	0.727	0.744
exp_m	1.149	0.658	0.888	0.756	0.745
exp_l	1.198	0.767	0.620	0.686	0.686
$bias_a$	1.080	0.729	0.753	0.741	0.683
$bias_p$	**0.967**	0.664	0.699	0.681	**0.753**

6 Discussion

We account for the summarized result of the previous section and investigate hidden factors of it in this section. First of all, Table 6 displays variations respect to methods in terms of RMSE. Results show that biasing scheme, especially $bias_p$, gives positive effects to matrix factorization significantly when comparing it with results from other methods. The reason of this phenomenon is that matrix factorization needs all values, so it is very sensitive to missing values when the dataset tends to be sparse. Therefore, filling missing values by a suitable technique is the most important step in the method, $bias_p$ is faithful to that role. Additionally, biasing scheme is effective to a method which emphasizes RMSE because it has a normalization effect by predicting a suitable value. This explanation is closely correlated to the fact that matrix factorization is optimized to reduce RMSE score.

Table 6: Effects of Biasing to Methods (RMSE)

Method	$bias_a$	$bias_p$
User-knn	-0.011	0.028
Item-knn	0.004	0.033
ItemRank	-0.201	-0.128
Matrix Factorization	**0.047**	**0.160**

Filtering scheme we proposed is to use only positive ratings. Table 7 lists differences of $filter_p$ results from original ones. $filter_p$ gave slightly positive effects to KNN and ItemRank in terms of F1 and Kendall's τ. It means that filtering helps to obtain top-k items and check ordering of positive ratings because F1 is specialized to top-k recommendation and Kendall's τ is for releative ordering. Contrary to matrix factorization which should consider all values, KNN and ItemRank requires only surrounding ratings. $filter_p$ is well-combinated with these methods due to the fact that it supports only positive things.

Table 7: Effects of $filter_p$ to Methods

Method	RMSE(-)	F1	Kendall's τ
User-knn	0.005	**0.015**	**0.038**
Item-knn	-0.049	0.003	0.001
ItemRank	-0.010	0.006	**0.025**
Matrix Factorization	-0.003	-0.032	0.026

We prepared two scaling schemes, exp_m and exp_l, to verify the confidence of original scale for rating system. However, Examining two result tables, Table 8 and Table 9, we could not find variations which are to consistent directions. It needs to be studied as several variations and creative ideas.

Table 8: Effects of exp_m to Methods

Method	RMSE(-)	F1	Kendall's τ
User-knn	0.007	-0.001	-0.009
Item-knn	0.016	-0.002	0.028
ItemRank	0.000	0.000	0.000
Matrix Factorization	-0.022	-0.003	0.028

Table 9: Effects of exp_l to Methods

Method	RMSE(-)	F1	Kendall's τ
User-knn	-0.293	-0.106	-0.044
Item-knn	-0.222	-0.048	-0.001
ItemRank	0.004	0.000	0.000
Matrix Factorization	-0.071	-0.073	-0.032

7 Conclusion

In this paper, we studied the effect of rating schemes on existing recommendation methods and evaluation measures. Proposed rating schemes are categorized as three kinds: filtering, scaling and biasing. Filtering scheme considers using ratings which match specific conditions, only positive values in the experiment. The motivation of scaling is the suggestion that original rating scale does not explain the nature of rating system. We prepared two candidates, exponential from medium and from lowest value. Finally, biasing means that ratings can be biased respect to the domain, dataset, etc. Two ideas, the average value of all ratings and average values which are personalized to each user and item, are introduced for biasing scheme.

Recommendation methods which are used in the experiment are KNN, ItemRank and Matrix Factorization. For each methods, we gave variations which have been considered in various existing researches. The other issue for recommendation research is what measure we should use for evaluating results. We evaluated rating schemes in terms of three measures: RMSE, Precision-recall and Kendall's τ and checked appropriate combinations by analyzing products.

Firstly, focusing on the relationship between rating schemes and evaluation measures, the experiment shows that biasing, especially $bias_p$, affects to improve RMSE values because it is closely related to missing values. Filtering scheme gave better F1-score due to its property that it concentrates to higher values. Examining effects of rating schemes on recommendation methods, biasing is so sensitive to matrix factorization because it needs all values, $bias_p$ reduced RMSE and improved Kendall's τ especially. Contrary to matrix factorization, ItemRank shows no great change respect to rating schemes totally. KNN results obtained intermediate values between two other methods.

This analysis for rating schemes started from the reason that there is few studies for rating system in recommendation area. Although we did simple process based on basic types of methods because of scarcity to the topic, this work will be extended to various versions by optimizing methods and supporting elaborate setting.

요 약 문

추천 시스템을 위한 평점 정보 분석 및 활용 방법

인터넷의 발달과 함께 디지털 정보가 점차 방대해지고 복잡한 구조로 나아가면서, 사람들이 원하는 정보를 직접 찾고 활용하는 것이 점차 힘들어지고 있다. 이러한 정보량 증가에 따른 어려움을 해결하기 위해, 불필요한 정보를 제거해주고 사용자의 취향에 적합한 아이템들을 제시해주는 추천 시스템이 나타나게 되었다. 이러한 추천 시스템들은 일반적으로 유저로부터 받은 평점 정보를 활용하여 새로운 알고리즘을 개발하고 그 성능을 향상시킨다.

이렇게 평점 정보가 추천 시스템과 밀접하게 연관되어 있음에도 불구하고, 그 동안의 연구들은 주어진 평점 체계가 과연 그들의 방법론에 적합한지에 대해서는 고려하지 않았다. 따라서 본 논문에서는 추천 시스템에서의 평점 정보 체계에 따른 효과를 분석하기 위해 여러 형태의 평점 체계 변화를 제안한다. 우리는 filtering, scaling, biasing의 평점정보 변화 방법론을 그 후보로 결정하였다. 그리고 이 제안을 검증하기 위한 실험에는 Collaborative Filtering 추천 방식에서 대표적인 k-nearest neighbor, ItemRank, matrix factorization 등의 3가지를 여러가지 고유 파라미터들과 함께 활용한다. 또한, 그 결과를 비교하기 위해서 평가 수단이 필요한데 기존 추천 시스템에서 많이 활용되는 RMSE, precision-recall, kendall's τ를 이용하여 평가했다. 그 결과를 분석하여, biasing 방법은 전체 점수의 활용을 중시하는 matrix factorization과 그에 상응하는 RMSE에 큰 영향을 끼치는 것으로 파악하였다. 또한 filtering은 상위 아이템들에 대한 순서 및 몇 개의 아이템을 추출하는 경우에 좋은 결과를 보여주었는데, 이는 KNN, ItemRank 등의 추천 방법과 F1-score, Kendall's τ 등의 평가 수단에 적합함을 보여주었다. 본 연구는 추천 시스템 연구에 있어서 다양한 추천 방식들과 파라미터 설정을 위한 초석이 될 것이라 생각한다.

References

[1] Chen, Hung-Chen and Chen, Arbee LP 2001. A music recommendation system based on music data grouping and user interests. *Conference on Information and Knowledge Management: Proceedings of the tenth international conference on Information and knowledge management*, Vol. 5, No. 10, pp. 231–238, 2001.

[2] Martinez, A and Arias, J and Vilas, A and Garcia Duque, Jorge and Lopez Nores, Martin 2009. What's on TV tonight? An efficient and effective personalized recommender system of TV programs. *Consumer Electronics, IEEE Transactions on*, Vol. 55, No. 1, pp. 286–294, 2009.

[3] Walter, Frank Edward and Battiston, Stefano and Schweitzer, Frank 2008. A model of a trust-based recommendation system on a social network. *Autonomous Agents and Multi-Agent Systems*, Vol. 16, No. 1, pp. 57–74, 2008.

[4] Debnath, Souvik and Ganguly, Niloy and Mitra, Pabitra 2008. Feature weighting in content based recommendation system using social network analysis. *Proceedings of the 17th international conference on World Wide Web*, pp. 1041–1042, 2008.

[5] Mukherjee, Rajatish and Sajja, Neelima and Sen, Sandip 2003. A movie recommendation system–an application of voting theory in user modeling. *User Modeling and User-Adapted Interaction*, Vol. 13, No. 1–2, pp. 5–33, 2003.

[6] Adomavicius, Gediminas and Kwon, YoungOk 2003. New recommendation techniques for multicriteria rating systems. *Intelligent Systems, IEEE*, Vol. 22, No. 3, pp. 48–55, 2007.

[7] Lorenz, Jan 2009. Universality in movie rating distributions. *The European Physical Journal B*, Vol. 71, No. 2, pp. 251–258, 2009.

[8] Desrosiers, Christian and Karypis, George 2011. A comprehensive survey of neighborhood-based recommendation methods. *Recommender systems handbook*, pp. 107–144, 2011.

[9] Deshpande, Mukund and Karypis, George 2004. Item-based top-n recommendation algorithms. *ACM Transactions on Information Systems (TOIS)*, Vol. 22, No. 1, pp. 143–177, 2004.

[10] Sandvig, Jeff J and Mobasher, Bamshad and Burke, Robin 2008. A survey of collaborative recommendation and the robustness of model-based algorithms. *IEEE Data Engineering Bulletin*, Vol. 31, No. 2, pp. 3–13, 2008.

[11] Su, Xiaoyuan and Khoshgoftaar, Taghi M 2009. A survey of collaborative filtering techniques. *Advances in Artificial Intelligence*, Vol. 2009, page. 4, 2009.

[12] Campos, Pedro G and Bellogín, Alejandro and Díez, Fernando and Chavarriaga, J Enrique 2010. Simple time-biased KNN-based recommendations. *Proceedings of the Workshop on Context-Aware Movie Recommendation*, pp. 20–23, 2010.

[13] Koren, Yehuda and Bell, Robert and Volinsky, Chris 2009. Matrix factorization techniques for recommender systems. *IEEE Computer*, Vol. 42, No. 8, pp. 30–37, 2009.

[14] Gori, Marco and Pucci, Augusto and Roma, V and Siena, I 2007. Itemrank: A random-walk based scoring algorithm for recommender engines. *Proceedings of the 20th international joint conference on Artifical intelligence*, pp. 2766–2771, 2007.

[15] Brand, Matthew 2005. A random walks perspective on maximizing satisfaction and profit. *SIAM International Conference on Data Mining*, pp. 12–19, 2005.

[16] Gori, Marco and Pucci, Augusto 2006. A random-walk based scoring algorithm with application to recommender systems for large-scale e-commerce. *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 127–146, 2006.

[17] Hofmann, Thomas 2003. Collaborative filtering via gaussian probabilistic latent semantic analysis. *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pp. 259–266, 2003.

[18] Page, Lawrence and Brin, Sergey and Motwani, Rajeev and Winograd, Terry 1998. The PageRank citation ranking: bringing order to the web. *Technical report, Stanford University*, 1998.

[19] Kamvar, Sepandar D and Haveliwala, Taher H and Manning, Christopher D and Golub, Gene H 2003. Extrapolation methods for accelerating PageRank computations. *Proceedings of the 12th international conference on World Wide Web*, pp. 261–270, 2003.

[20] Hyndman, Rob J and Koehler, Anne B 2006. Another look at measures of forecast accuracy. *International Journal of Forecasting*, Vol. 22, No. 4, pp. 679–688, 2006.

[21] Powers, David MW 2007. Evaluation: From precision, recall and f-factor to roc, informedness, markedness & correlation. *School of Informatics and Engineering, Flinders University, Adelaide, Australia, Tech. Rep. SIE-07-001*, 2003.

[22] Hjørland, Birger 2010. The foundation of the concept of relevance. *Journal of the American Society for Information Science and Technology*, Vol. 61, No. 2, pp. 217–237, 2010.

[23] Kendall, Maurice G 1938. A new measure of rank correlation. *Biometrika*, Vol. 30, No. 1/2, pp. 81–93, 1938.

Acknowledgments

감 사 의 글

2004년 가을에 수시면접 보겠다고 포항에 첫 걸음을 한 후로 9년에 가까운 세월이 흘렀습니다. 처음 입학하고 2년을 방황하다가 과연 졸업할 수 있을까 싶었는데 어느덧 군대를 갔다오고, 학부를 졸업하고, 대학원 석사과정까지 마치게 되었네요. 포항에서의 삶을 함께하면서 저에게 소중한 인연이 되어주신 분들께 이렇게나마 감사의 말을 전합니다.

먼저 저에게 대학원의 길을 열어주시고 성심성의껏 지도해주신 유환조 지도교수님께 감사드립니다. 교수님께서 몸소 보여주신 연구자로서의 마인드와 가르쳐주신 여러가지 삶의 지혜 잊지 않겠습니다. 기꺼이 졸업논문 심사에 응해주시고 미숙한 발표에도 졸업을 허하여 주신 이종혁 교수님, 안희갑 교수님 두 분께도 이 글을 빌어 감사를 전합니다.

2년 동안 함께 지내온 연구실 식구들, 참 여기다 적기에는 쓸 말이 너무 많네요. 먼저 같은 날 다른 학위로 디펜스하신 박사(진) 진하형님, 항상 밥먹을 때마다 '무엇이든 물어보세요'를 보는듯이 분야불문 척척 대답해주시는 모습이 인상적이었는데 요새는 딸바보 놀이에 심취해 계셔서 덩달아 즐거워집니다. 형수님과 함께 딸 이쁘게 키우시고, 20년 뒤를 노리는 승호는 멀리하시길... 그리고 제 모든 석사생활을 함께 해준 진오형, 이거는 appendix를 따로 만들어야 될 것 같은데 그럴 여유가 없는게 아쉽네요. 2년 내내 못난 후배 뒤치닥거리 해주신 덕분에 졸업도 하게 되어 고맙지만 다이아도.. 아.. 아무튼 이래저래 재밌었어요. 프로포절도 끝났는데 다음에 디펜스 꼭 성공하시길, 2년 내내 온/오프라인으로 타박만 했는데 이젠 온라인으로만 해야하는게 안타깝지만! 다음으로 유일하게 학부때부터 알고지낸 성철이형. 귀찮음에 쩔어있는 연구실에서 이거저거 이벤트 챙기고 귀찮은거 솔선수범하는 연구실의 중심이시죠. 그런데 요즘은 연애한다고 주말마다 그를 볼 수가 없다는... 형도 빨리 무사히 졸업하길 바라고 덤으로 저한테 좋은 형 한번 좀.. 그리고 뭔지는 모르지만 항상 바빠보이

는 태훈이형, 누구보다 건강 건강 하면서 연구실 의자에서 매일같이 주무시는거 보고 신기했는데, 운동 하나 취미로 하는게 좋을 것 같아요. 앞으로 하시는 연구 다 잘 되시길. 여기에 큰 공백이 있네요. 도무지 행동반경을 예측할 수가 없는 우리 승걸이, 바로 후배라고 형이 오다보니 되려 연구실 막내같았는데 이제는 어느새 제법 짬도 되고 랩장일 하면서 나름 책임감도 있어 보인다. 맨날 여기서 저기서 구박만 했지만 형이 다 애정이 있어서 그런거니까 용서하렴. 근래에 힘든 일이 많이 있었지만 앞으로도 잘 극복하리라 믿는다. 그리고 연구실의 차세대 기둥이자 6분반 라인 동현이, 내 동생보다 더 동생같이 잘 따라주고 챙겨줬는데 딱히 내가 뭘 해준게 없어서 미안하고 고맙다. 공부도 열심히 하고 연구도 열심히 하고 연애도... 일곱살.. 하아... 열심히 하고, 이제 남은 하나만 열심히 하면 되겠네. 혹시 중국 못가게 되면 형이 백금 캐러 가는거 도와줄께. 또, 악의 축이라고 놀려댔지만 누구보다도 예의바르고 깍듯이 형대접 해준 규동이, 지금은 떠나있지만 우리 연구실에 남아있기엔 너가 가진 재능이 너무 아까웠을 뿐이라고 생각한다. 잠시 방황의 시기를 겪고 있지만 충분히 딛고 어디서든 성공할거라 믿어. 그리고 잠시 휴재 중인 작품활동 빨리 재개하길... 다음은 뜬금없이 우리 연구실로 표류한 또 하나의 6분반 후배이자 연구실의 귀요미를 담당하는 예찐이, 2년차인데도 여러가지가 다르다보니 도와줄 사람이 없어서 이거저거 혼자 고민하고 고군분투하는게 안타깝지만 힘든 소리 안하고 잘 헤쳐나가는 것 같아서 다행이다. 어떤 선택을 할지 모르겠지만 빨리 역마살 끊고 하고싶은 연구 맘껏 하길 바란다. 그리고 유일한 86라인 동은이, 신입생이라길래 파릇파릇한 애들 기대했더니 웬 동년배가 와서 당황했는데 QE도 한방에 뚫고 ML도 별탈없이 듣는 것 보고 존경심마저 든다. 연구실 잘 적응하는건 좋은데 너무 seungkeolization하지는 말고 열심히 해서 연구실 빛내주길. 또 우리 패턴의 에이스를 담당하는 유진이, 어쩌다 여학생이 이렇게 먼 곳까지 와서 귀양살이 하고 있는지부터가 신기하지만 같이 지내다보니 그럴 만하다.. 라는 생각이 들더라. 때론 엉뚱해 보이기도 하고 혼자놀기도 잘하는 것 같아서 크게 걱정은 안하지만 혹시나 힘든 일 있으면 언제든지 연구실 사람들에게 도움 청하렴. 겪어보면 알겠지만 나빼고 다들 착한 사람들임. 마지막으로

어딘가에 유랑하고 있을 홍창이, 사회경험이 풍부해서인지 지나치게 적응을 잘해서 처음엔 부담스러웠는데 그래서인지 빈자리도 큰 것 같다. 어디서든 적응 잘할테니 잘 자리잡고 성공했으면 좋겠다.

지금은 멀리 미국에 계시지만 연구실 적응하는데 큰 도움을 주셨던 영대 형님, 떠나신다고 가기 3달전부터 수십번 송별회한게 잊혀지지가 않네요. 99 학번임에도 마치 바로 윗학번인양 친근하게 대해주시고 많은 경험 시켜주신 것 감사합니다. 그리고 저한테 그렇게 무섭게 했던 삼성맨 재용이형, 얼굴 본지 세번만에 그 무서운 얼굴로 육두문자 날리던게 생생하지만 소고기 사준 걸로 퉁쳐드림. 저한테도 SSAT 떨어지고 삼성가는 비법좀... 사실 연구실에서 공유한 시간은 얼지만 몇년은 본 듯한 붙임성좋은 일환이형, 요즘은 저한테 취직자리 소개해주는 좋은 형이시죠. 서울가서 꼭 연락드릴께요. 또 다사다난 했던 ETRI 프로젝트를 함께한 선이누나, 그것때문에 잠시 삐뚤어지긴 했지만 얼마전에 면접본다니까 먼저 연락해줘서 고마워요. 근데 떨어져서 못감.. 아.. 마지막으로 석사 4호기 전선배, 누구보다도 착하고 부지런하지만 그 이면에 무서운 놈이 숨어있다는 것을 본 적이 있지. 좋은 전례를 남긴 덕분에 졸업이 힘들 줄 알았지만 그 바톤은 다음 석사 6호기에게 넘기는 걸로. 그리고 이번에 포항와서 말했던 약속 꼭 지켜서 연말에 보고하도록. 아, 도대체 언제 다이아 캐러올꺼야?

그리고 우리 컴퓨터공학과의 소중한 인연들에도 감사합니다. 학교에서 가장 오래 알고지낸 05 첫 유부남 삼세훈, 카이스트의 전설 태중이, 05의 입을 담당하는 정재윤, 그립기만 한 전 방돌이 정봉이, 대표 돌아이 여동훈, 곧 끌려가는 한얼이, 앞방 주민 종재혁, 노라키 등등 그리고 미처 언급하지 못한 05학번 동기들 다 다양한 길을 가고 있지만 각자가 뜻하는 바를 이루길 빈다. 또한 여기에 쓰지 못한 컴퓨터공학과 선후배님들 모두 성공하시길 빕니다.

또 하나의 인연 우리 6분반, 덕분에 대학 생활 즐겁게 보냈습니다. 늘그막에 군대간다고 타박했던 태성이형, 항상 둘이 술먹는데 옵션으로 불러준 충재형과 병갑이형, 컴공 중심의 6분반을 형성에 기여하신 03컴공 트리오 효형이형, 수하형, 현석이형, 싸가지없는걸 자랑으로 여기게 해준 효림이형 등등 선배님들

감사합니다. 그리고 분반의 실세 재우, 부산남자 도창이, 4차원의 삶을 사는 진하, 공식 분반장 태경이, 여자사람 술친구 해준 늬주, 써머세션에 아픈 추억을 공유한 양용이 등 우리 05들 덕분에 8년간 즐거웠다. 너무도 다른 길들을 가고 있지만 다들 각자 분야에서 성공해서 한번 만나자.

마지막으로 부모님 밑에서 자립해보고 싶다고 어린마음에 먼 곳으로 뛰쳐나온 아들놈 8년간 뒷바라지 해주신 부모님께 큰 감사를 전합니다. 28년을 살았는데 멀리 학교 다니느라 여유 없다는 핑계로 전화한통 효도한번을 제대로 못해서 죄송합니다. 받은 은혜 꼭 잊지않고 되돌려 드리겠습니다. 형보다 앞서 사회나간 주영이도 늦게나마 취직 축하한다. 그리고 작년부터 건강이 급하게 안좋아지신 외할아버지, 외할머니, 쾌차하시라는 말은 제 욕심이겠지만 남은 나날들 행복하게 사셨으면 좋겠습니다. 전주에 계신 할머니도 건강하시구요.

이 밖에도 말 한마디, 얼굴 한번이라도 마주한 많은 분들이 있지만 차마 여기에 다 쓰지 못한 점 죄송합니다. 학부, 대학원 생활을 하면서 공부도 배우고 다양한 경험도 했지만, 개인적으로 가장 뜻깊게 생각하는 것은 역시 여러 사람들과 맺은 인연입니다. 항상 느꼈지만 누군가를 기억하고, 누군가에게 기억되어 진다는 것만큼 즐거운 일은 없다고 생각합니다. 비록 학업에 치이고, 연구가 부담스럽고, 환경이 불편하더라도 누군가와 밥한끼 같이 먹으면서 잠시 웃을 수 있었다는 것이 저에게는 버틸 힘이 되었기에, 여러분도 여기서 작게나마 즐거움을 찾을 수 있었으면 좋겠습니다. 떠나서 잠시 혼자가 된다니 섭섭하기도 하지만, 저에게 도움을 주셨던 모든 분들이 뜻하는 바를 이루시길 빕니다.

2013년 6월 12일.

성영철 드림

Curriculum Vitate

Name : Youngchul Sung (성 영 철)

Education

2005.3 - 2011.8 : 포항공과대학교 컴퓨터공학과 (B.S.)

2011.9 - 2013.8 : 포항공과대학교 컴퓨터공학과 (M.S.)