

# An Automated Advice Seeking and Filtering System

Reham Alskireen<sup>1</sup>, Dr. Said Kerrache<sup>2</sup>, Dr. Hafida Benhidour<sup>3</sup>  
Computer Science Department, College of Computer and Information Sciences  
King Saud University, Riyadh, Saudi Arabia

**Abstract**—Advice seeking and knowledge exchanging over the Internet and social networks became a very common activity. The system proposed in this work aims to assist the users in choosing the best possible advice and allows them to exchange advice automatically without knowing each other. The approach used in this work is based on a newly proposed dynamic version of the hidden metric model, where the distance between each couple of users is computed and used to represent the users in a  $d$  dimensional Euclidean space. In addition to the position, a degree is also assigned to each user, which represents his/her popularity or how much he/she is trusted by the system. The two factors, distance and degree, are used in selecting advice providers. Both the positions of the users and their degrees are adjusted according to the feedback of the users. The proposed feedback algorithm is based on a Bayesian framework and has a goal of obtaining more accurate advice in the future. The system evaluated and tested using simulation. In the applied experiment, the mean square error was measured for different parameters. All parts of the experiments are performed on a varying number of users (100, 500 and 1000 users). This shows that the system can scale to a large number of users.

**Keywords**—Recommender system; hidden metric; advice; Bayesian framework

## I. INTRODUCTION

We are continuously faced with choice making problems in our professional and daily lives. For example, which conference to submit a given paper, which doctor to visit to treat a certain health problem; which school should our children attend, or even more simply, what kind of shoes or clothes should we buy for the season! Faced with such choices, we often turn to our acquaintances, whether real or virtual, for information and advice. We consult our family members and friends or seek information and advice on forums and mailing lists. In doing so, we select individuals who are more likely to give us the best advice, because of their knowledge about the subject matter, but also because we trust their intentions. Once we have the opinions of many individuals, we filter those that seem trustworthy and make our decision based on them.

Experience shows that seeking advice in the traditional way takes effort and time, and we are not out of risk of making a bad decision because of bad advice. The problem addressed in this work is to automatize the advice seeking and filtering process by a system that can be used over the Internet or mobile phones. The typical use case is that a user initiates a request for an advice to the system. The system selects a number of users to whom the request is sent. Then, advice is formulated by combining the replies given by the advice providers into one advice. Then, this advice is sent back to the advice seeker who initiated the request.

Several issues need to be addressed in order to achieve this goal. First, how to represent the request for advice? Second, to whom should the request sent? Finally, how to synthesize a final advice from all the advices received? In this work, a system is proposed that helps the user in seeking and filtering advice in an intelligent way. The approach used is based on the *hidden metric model*, which procures the system with the structure of a complex network. First, an initial distance between each couple of users is computed based on their profiles and used to assign positions to them in a Euclidean space. A proposed seeking algorithm is applied in order to select the users that would receive the advice request. A filtering algorithm combines the replies given by the advice providers into one advice and sent it back to the advice seeker. A feedback algorithm that updates the positions of the users penalizes ill-behaved users and rewards active users. These changes are decided based on a Bayesian framework.

## II. RECOMMENDER SYSTEMS

Seeking advice can be processed through different information processing systems such as Recommender Systems. Recommender Systems are defined as information processing systems that provide suggestions for users by gathering various kinds of data using special tools and techniques. Deshpande and Karypis in [2] define a recommender system as a technique that filters personalized information in order to predict whether one user will like one item.

Recommender systems have a significant role in many famous websites such as Amazon, and Netflix. Methods and techniques of recommender systems can be used for the automation of advice seeking and filtering process. Data that are used by the recommender systems usually refers to items and users who will receive recommendations, where items are the entities that are suggested to users [1]. Basically, recommender systems consist of three important parts: background data, input data, and a special algorithm that gathers background and input data to build recommendations. Background data refers to knowledge that a system has before starting recommendation process. Input data are data provided by users to obtain recommendations.

The main approaches used to combine background and input data can be categorized as collaborative filtering approach, content-based approach [2,3], and hybrid approach [4,6,7]. Hybrid approach is a combination of collaborative and content-based techniques where the combination process can be performed in different forms.

A. Content-Based Approach

In content-based approach, system aims to predict similar items to the previously liked by the user [4,5,6]. The measure of similarity relies on the features associated with each item. Recommendation process in content-based approach works by matching features of user profile with features associated with each item where user profile is built by analyzing a group of documents and/or descriptions of items that the user rated before. User profile is considered as a structured representation of user preferences that relies on the features associated with item rated by this user [7]. Accuracy of user profile reflects interests of that user and as a result, recommendations would be more effective. Accurate recommendations can have an influence on the behavior of user to access information [9,10]. For example, recommender systems would filter searching results in order to determine if the user will like a certain web page or not. The predictions of items in content-based approach are made in three steps. Each step is directed by a separated component; content analyzer, profile learner, and filtering component [11,12].

B. Collaborative Filtering Approach

Collaborative filtering is a very common approach of recommender systems [2,3,4,5,6,7,9,10,11,12]. It is considered as a knowledge dependent approach as it depends on the knowledge about what other users like in the past not only what the user himself liked in the past. The system recommends items that are interesting to users with similar taste, this similarity is calculated based on users ratings in the past. Collaborative filtering approach has two main classes; neighborhood-based and model-based approaches [9,14].

III. PROPOSED SYSTEM

The system proposed in this work allows the users to exchange advice automatically, without knowing each other. Upon receiving an advice request, the system selects automatically the users to whom this request is sent. It then combines the replies from these users and sends it back to advice seeker. The feedback from the users is used to improve the performance of the system and provide better recommendations to the advice seekers. The approach used in this work is based on a newly proposed dynamic version of the hidden metric model [13].

A. Protocol

The users of the system can play either two of the following roles:

- Advice seeker: This is the user that initiates the interaction by sending a request for advice to the system.
- Advice provider: Who is a user selected by the system to answer to a request for advice.

The system is a centralized agent that manages the interactions between the users. Fig. 1 shows interaction phases between the system, advice seeker and advice providers. The protocol for using the system is as follows:

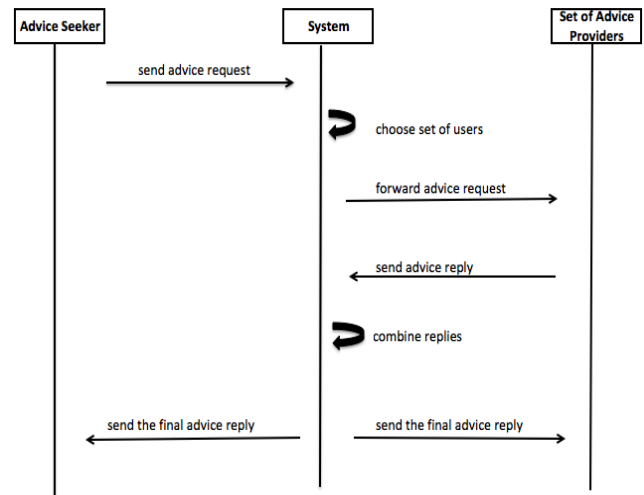


Fig. 1. System Interaction.

**Advice Request**

1- Domain of Question: .....

2- Background Information: .....

3- Question: .....

4- Choices:

a)

b)

c)

Fig. 2. Advice Request Form.

**Advice Reply**

1- Choices: (Suggested Order)

b)

a)

c)

2- Comments: .....

.....

Fig. 3. Advice Reply Form.

1) The advice seeker sends a request of advice to the system with a format as in Fig. 2. Where the users selects from a pre-specified set of domains of questions. This tag helps the system make a better choice of the advice providers. Background information is a text explaining the situation of the user and any relevant information that may help the advice providers to give a more helpful advice. Choices are a list of the alternatives that is available to the user. A choice list format is somewhat restrictive, but it can be efficiently handled computationally.

2) The system chooses a set of users, the advice providers, to whom the request is sent.

3) Each user who receives a request for advice can either refuse to participate or send a reply with a format as in Fig. 3. Where the list of choices initially in the advice request ordered as seen fit by the advice provider. Comments allow the user provide any useful information for the advice seeker.

4) The system collects the replies from all users and combines them to obtain a reply that is sent to the advice seeker as well as the advice providers. The replies of the advice providers are also made available to all participating users. The advice seeker and providers have access to the comments sent by each user. The users of the system can give two types of feedback:

a) An advice seeker can specify the choice he made and his/her opinion about this choice, which for simplicity is assumed to be either positive or negative.

b) Any user who interacts with another user can give a feedback about him/her. This feedback can be either positive or negative.

#### B. Internal Representation

The system contains initially  $n$  users. The initial distance between each couple of users is computed based on their profiles [15]. Multidimensional scaling [17] is then used to assign locations to the users in a  $d$  dimensional Euclidean space. The dimensionality  $d$  is chosen so that the Euclidean distance between the users is as similar as possible to the distance computed from profiles. In addition to the position, a degree  $k_i$  is also assigned to each user. When a user  $i$  initiated a request for advice a set of users is selected to send advice reply. Two factors participate in selecting advice providers: their distance (dissimilarity) [16] from the advice seeker and their degree, which represents their popularity, or how much they are trusted by the system. These two factors are updated based on feedback from the users.

The seeking algorithm automatically filters users with bad reputation from providing advice. This is done thanks to the degree parameter. The filtering algorithm then combines the replies given by the providers into one advice that is sent back to the advice seeker. There are many ways in which the rankings given by the provider can be combined as weighted vote and simple vote. The feedback from the system users is used to update their positions and also penalize malicious advice providers. Updating the position has as a goal obtaining a more accurate estimation of the preferences of the user, so that future advice requests will be sent to more relevant

providers. Adjusting the degree of the user according to his/her acceptance rate is necessary to avoid overloading or underloading users with requests. Finally, Penalizing malicious users allows the system to direct requests to only serious users who can provide useful advice. The first task of the system is to translate this feedback to a dissimilarity measure between user  $i$  and the advice providers. If the feedback is positive, the providers who recommended the choice have more similarity to the provider than those who did not recommended it. The situation is of course reversed if the feedback is negative. The degree is updated in two cases:

1) When a user accepts or refuses a request for advice.

2) When a user gives a feedback, positive or negative, about another user.

The user degree reflects his/her willingness to provide advice to other users; hence it must adjust to reflect the decisions of the user.

#### IV. SYSTEM EVALUATION

The behavior of the proposed model studied using simulation. In fact, the applied simulation aims to estimate the performance of the system algorithm and to locate any weaknesses in the proposed system and helps to test system actions even when the number of user is huge. Also, system simulation helps to understand why a particular event happens, and so we can re-simulate the same event with different parameters. The simulation model used in this work is similar to some extent to the model used in [8].

##### A. Simulation Model

The simulation performed on different number of users and with a different number of iterations. As the program starts, users instances and resources created. Users' hidden preferences are distributed as uniform distribution. Bernoulli distribution is used to simulated whether the user will seek an advice or not, and power law to generate the hidden degree of each user. Each user assigned a degree where the degrees of users are fixed at the first iteration of the system. Furthermore, users also assigned a probability to send an advice request, which is fixed also in the first iteration, and a list of user experience that represents his experience towards the used resources in the system.

##### B. Experimental Settings

The simulation performed on different number of users and with a different number of iterations. As the program starts, users instances and resources created. Users' hidden preferences are distributed as uniform distribution. Bernoulli distribution is used to simulated whether the user will seek an advice or not, and power law to generate the hidden degree of each user. Each user assigned a degree where the degrees of users are fixed at the first iteration of the system. Furthermore, users also assigned a probability to send an advice request, which is fixed also in the first iteration, and a list of user experience that represents user's experience towards the used resources in the system. In the next step, system iterations start where users with high probability are able to send an advice request more than others. However, users with low probability choose one of the existing resources randomly where this resource is added to user's list of experience.

When a user is able to send an advice request, the advice request is formulated. In order to formulate an advice request, a number  $m$  of resources are chosen such that one of these resources has been used before where the other resources are not used. After that advice request is sent to similar users in the system. Similarity between users is calculated where each user is assigned a weight. Then a random number is generated and compared with sum of the weights of users to decide which user will reply. Users can provide a reply for advice otherwise this user is ignored. After that users replies are stored in order to formulate the final reply that would be sent to the initiating user.

When a user receives an advice request, user's list of experience and utilities are checked to determine whether he/she liked the resources listed in the request or not. If a user like one resource he/she gives a vote to this option. The formulation of the final reply is done by counting number of votes for each choice. Then choices are ordered in ascending order. When the initiating user receives the final reply, the system is provided with positive or negative feedback. Providing the system with feedback is the main factor to improve the system in learning process. Based on the provided feedback users positions on the system are updated. The described iterations are repeated many times up to 10,000 iterations. Finally, the experiment is applied in order to measure system performance.

### C. Performance Measures

In order to evaluate how the proposed algorithm behaved, the mean square error is calculated. Using mean square error we can measure the ability of the proposed algorithm to group users with similar preferences and estimate the distance between the provider and the users. Furthermore, the average minimum utility is used to measure whether users gained the best resources or not.

## V. EXPERIMENTAL RESULTS

Three experiments performed on the system for evaluation. Each experiment done on two parts. The goal of the first part is to show that if the distance is 100% accurate then the system can find the real distances. In order to apply the measurement, mean square error is calculated. In the second part we aim to measure the average minimum utility of system users and analyze whether it is decreasing or not. All parts of the experiments performed on varying number of users (100, 500 and 1000 users).

### A. First Experiment

The parameters used to perform this experiment are 100 users and 300 resources where the system executed 1000 iterations on the data. The mean square error decreases as the number of iterations increases which means that users positions (users preferences) in the system are changing towards the best position. Fig. 4 and Fig. 5 show average (mean) error and average minimum utility of this experiment. The experiment shows that average minimum utility is decreased as the number of iteration increase. This fact shows that users gained the best resources for their preferences.

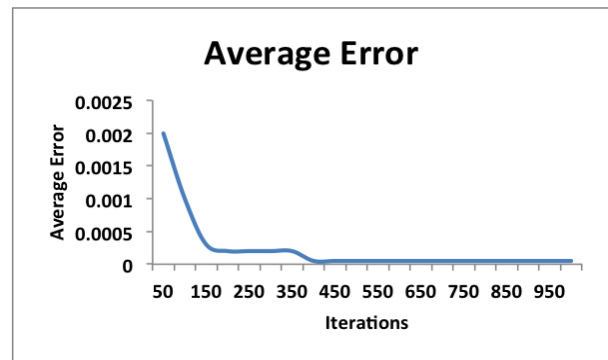


Fig. 4. Average Error-Experiment 1.

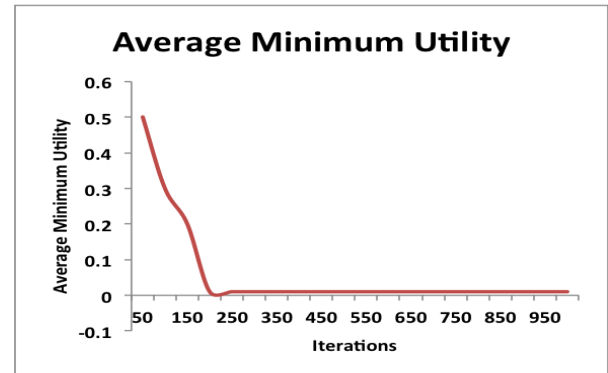


Fig. 5. Average Minimum Utility-Experiment 1.

### B. Second Experiment

In the second experiment, the number of users increased to 500 users where resources also increased to 800 and iterations performed 5000 times. Fig. 6 and Fig. 7 show average error and average minimum utility of the second experiment. Results show good results for average error and average minimum utility which very close to zero as in the first experiment.

### C. Third Experiment

A big number of users (1000 users) tested with 800 resources and 10,000 iterations. Fig. 8 and Fig. 9 show average error and average minimum utility of the third experiment. Results were similar to previous experiments. This proves that the system can scale to a large number of users.

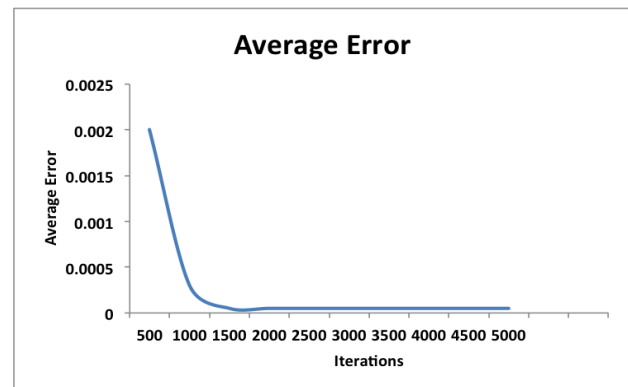


Fig. 6. Average Error-Experiment 2.

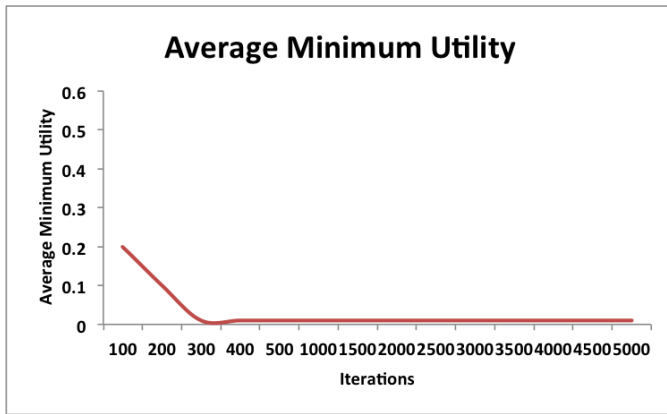


Fig. 7. Average Minimum Utility–Experiment 2.

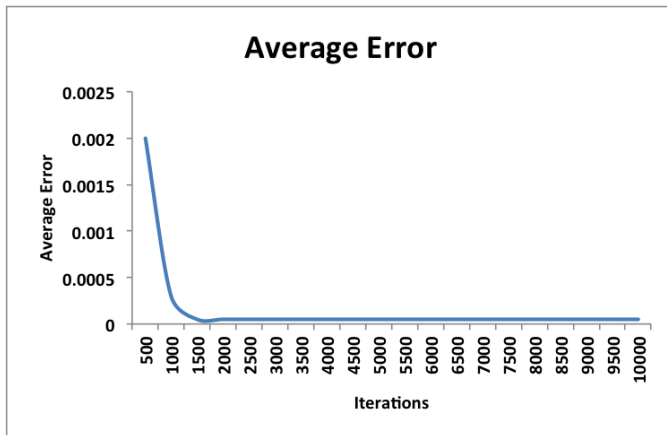


Fig. 8. Average Error-Experiment 3.

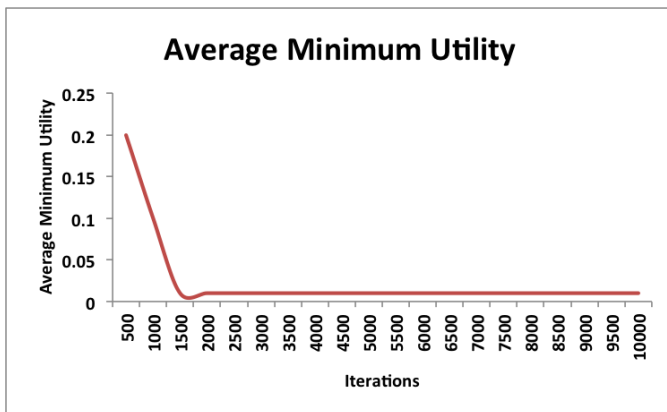


Fig. 9. Average Minimum Utility–Experiment 3.

## VI. CONCLUSION AND FUTURE WORK

The proposed model aims to support efficient, fast, and accurate advice seeking system. The approach used in this work is based on a dynamic version of the hidden metric model where the distance between each couple of users is computed and presented in a d dimensional Euclidean space where the result of this step is a set of positions of system users. Results show that when users seek advice from other users and then

their position on the system adapted in response to the quality of advice received, users can have better advice next time. Results also show that users gained the best resources for their preferences. All parts of the experiments performed on a varying number of users (100, 500 and 1000 users). This shows that the system can scale to a large number of users. In the future, we aim to simulate the proposed model in order to measure the error of user's degrees and study its effect on system behavior.

## REFERENCES

- [1] Weiss, G. Distributed reinforcement learning. Robotics and Autonomous Systems, vol.15, pp.135-142.1995.
- [2] D.Almazro, G.Shahatah, L.Albdkarim, M.Kherees, R.Martinez, W.Nzoukou.A Survey Paper on Recommender Systems.CoRR journal.vol.1006.5278. 2010.
- [3] B.Robin.Hybrid Recommender Systems: Survey and Experiments. User Modeling and User-Adapted Interaction Journal.vol. 12(4). pp. 331 - 370. 2002.
- [4] G.Adomavicius, A.Tuzhilin. Towards the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. IEEE Transactions on Knowledge and Data Engineering Journal.vol. 17 (6). pp.734-749. 2005.
- [5] S.Perugini, M.Andr Goncalves, and Edward A. Fox. Recommender Systems Research: A Connection-Centric Survey. Journal of Intelligent Information Systems. vol.23 (2). pp.107 - 143. 2004.
- [6] K. Tumer and D. Wolpert, "A survey of collectives." In Collectives and the Design of Complex Systems, Eds, K. Tumer and D. Wolpert, pp. 1-42, 2004.
- [7] Francesco Ricci and Lior Rokach and Bracha Shapira, Introduction to Recommender Systems Handbook, Recommender Systems Handbook, Springer.pp. 1-35. 2011.
- [8] Rezaei, G., Pfau, J.,& Kirley, M. Distributed Advice-Seeking on an Evolving Social Network. Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010 IEEE/WIC/ACM International conference. vol.2.pp. 24 – 31. doi:10.1109/WI-IAT.2010.78. 2010.
- [9] Prem Melville and Vikas Sindhwani. Recommender Systems. In Encyclopedia of Machine Learning, Claude Sammut and Geoffrey Webb (Eds), Springer, 2010.
- [10] Shlomo Berkovsky, Tsvi Kuflik, Francesco Ricci. The impact of data obfuscation on the accuracy of collaborative filtering. Expert Systems with Applications, Vol. 39 (5). pp. 5033–5042. 2012.
- [11] Linden, G., Smith, B., & York, J. Amazon.com recommendations: Item-to-item collaborative filtering. IEEE Internet Computing. vol. 7(1), pp. 76-80. 2003.
- [12] Karatzoglou, A., Smola, A., and Weimer, M. Collaborative filtering on a budget. Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS) 2010, Chia Laguna Resort, Sardinia, Italy. Volume 9 of JMLR: W&CP 9. 2010.
- [13] Marian Boguna, Dmitri Krioukov, and kcclay. Navigability of complex networks. Nature Physics Journal. vol.5. pp.74-80. 2009.
- [14] D. J. Watts and Steven Strogatz (June 1998). "Collective dynamics of 'small-world' networks". Nature 393 (6684): 440–442. Bibcode 1998Natur.393.440W. doi:10.1038/30918. PMID 9623998.
- [15] Rocchio, J.Relevance Feedback Information Retrieval. In: G. Salton (ed.) The SMART retrieval system - experiments in automated document processing, pp. 313–323. Prentice-Hall, Englewood Cliffs, NJ. 1971.
- [16] Herlocker, L., Konstan, J.A., Terveen, L.G., Riedl, J.T.: Evaluating Collaborative Filtering Recommender Systems. ACM Transactions on Information Systems.vol. 22(1),pp. 5–53.2004.
- [17] Joseph B. Kruskal and Myron Wish. Multidimensional Scaling. Quantitative Applications in the Social Sciences. SAGE Publications, January 1978.