DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Informatics

# Development of a Recommender System that addresses the diversity principle to improve user satisfaction.

Ozan Pekmezci

DEPARTMENT OF INFORMATICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Informatics

# Development of a Recommender System that addresses the diversity principle to improve user satisfaction.

# Entwicklung eines Empfehlungdienst unter Berücksichtigung der Benutzerzufriedenheit zur Diversität der Empfehlungen.

| | |
|---|---|
| Author: | Ozan Pekmezci |
| Supervisor: | Prof. Dr-Ing. Klaus Diepold |
| Advisor: | Julian Wörmann, M.Sc. |
| Submission Date: | 15.05.2019 |

I confirm that this master's thesis in informatics is my own work and I have documented all sources and material used.

Munich, 15.05.2019 Ozan Pekmezci

# Acknowledgments

# Abstract

# Contents

# 1 Introduction

Recommender Systems (RSs) are software tools and techniques that provide suggestions for items that are most likely of interest to a particular user [RRS15].

Suggestions and items depend on the field that recommder system is applied. For example, for the topic of news article recommenders, the aim will most likely be suggesting news to the readers. In the field of job recommenders though, these suggestions can be bidirectional. Meaning that, job postings can be suggested to applicants or resumes can be recommended to the human resources team of a company.

This chapter of the thesis will focus on explaining the basic terminology of recommender systems so that readers who are new to the topic can understand the rest easily.

## 1.1 Background of Classical Recommender Systems

Although the history of the recommender systems go back to mid-1990s [Par+12], the real boom happened after e-commerce services became mainstream [1]. Since there were too many items to choose from for users, such service was needed. Users of websites were becoming overloaded with the information and the developers of recommender systems had aim of reducing the information to be only relevant to users.

This section contains brief information about classical recommender systems.

### 1.1.1 Why Recommender Systems?

As mentioned in the last paragraph, recommender systems have the general function of suggesting items to users. However, why do recommender systems get developed? What kind of benefit do they have for both companies and users?

First of all, recommender systems increase the number of items sold [RRS15]. Also, most recommenders suggest personalized results, which means that users will see content that fits their desires. This will also increase users buying more items.

Recommenders also increase the coverage of items that user see. Coverage denotes number of recommended unique items divided by the number of all items. Therefore, users can iteract with items that they wouldn't even see without recommenders, that improves chances of buying more items.

Another important point is definitely increasing the user satisfaction. This function of recommenders is the foundation of the thesis at hand. Unfortunately, most of the researchers don't take into account that the user satisfaction does not solely depend on simple evaluation metrics like accuracy, precision or recall but it can also depend on privacy, data security, diversity, serendipity, labeling, and presentation [Bee+16]. When recommender systems take those factors into account, they clearly increase user satisfaction. [Gerekirse iki madde daha var]

### 1.1.2 Functions

To achieve the goals in the previous subsection, recommender systems need to fulfill some functions. Common functions include but not limited to [RRS15]:
[kopi - peyst]

- Find Some Good Items: Recommend to a user some items as a ranked list along with predictions of how much the user would like them (e.g., on a scale of one-to-five stars). This is the main recommendation task that many commercial systems address (see, for instance, Chap. 11). Some systems do not show the predicted rating.

- Find all good items: Recommend all the items that can satisfy some user needs. In such cases it is insufficient to just find some good items. This is especially true when the number of items is relatively small or when the RS is mission-critical, such as in medical or financial applications. In these situations, in addition to the benefit derived from carefully examining all the possibilities, the user may also benefit from the RS ranking of these items or from additional explanations that the RS generates.

- Recommend a sequence: Instead of focusing on the generation of a single recommendation, the idea is to recommend a sequence of items that is pleasing as a whole. Typical examples include recommending a TV series, a book on RSs after having recommended a book on data mining, or a compilation of musical tracks

- Recommend a bundle: Suggest a group of items that fits well together. For instance, a travel plan may be composed of various attractions, destinations, and accommodation services that are located in a delimited area. From the point of view of the user, these various alternatives can be considered and selected as a single travel destination

[kopi - peyst]

### 1.1.3 Applications

[kopi - peyst]

- Entertainment—recommendations for movies, music, games, and IPTV.
- Content—personalized newspapers, recommendation for documents, recommendations of webpages, e-learning applications, and e-mail filters.
- E-commerce—recommendations of products to buy such as books, cameras, PCs etc. for consumers.
- Services—recommendations of travel services, recommendation of experts for consultation, recommendation of houses to rent, or matchmaking services.
- Social—recommendation of people in social networks, and recommendations of content social media content such as tweets, Facebook feeds, LinkedIn updates, and others.

### 1.1.4 Objects

Data used by typical recommender systems refer to three types of objects [RRS15]: users, items and interactions.

[kopi - peyst] Items Items are the objects that are recommended. Items may be characterized by their complexity and their value or utility. The value of an item may be positive if the item is useful to the user, or negative if the item is not appropriate and the user made the wrong decision when selecting it. We note that when a user is acquiring an item, one will always incur in a cost which includes the cognitive cost of searching for the item and the real monetary cost eventually paid for the item.

Users Users of an RS, as mentioned above, may have very diverse goals and characteristics. In order to personalize the recommendations and the human- computer interaction, RSs exploit a range of information about the users. This information can be structured in various ways, and again, the selection of what information to model depends on the recommendation technique.

Transactions We generically refer to a transaction as a recorded interaction between a user and the RS. Transactions are log-like data that store important information generated during the human-computer interaction and which are useful for the recommendation generation algorithm that the system is using. For instance, a transaction log may contain a reference to the item selected by the user and a description of the context (e.g., the user goal/query) for that particular recommen- dation. If available, that transaction may also include explicit feedback that the user has provided, such as the rating for the selected item. Implicit, explicit unutma ve uzat, ornekle

[kopi - peyst]

### 1.1.5 Types

[gelistir, buyut]

- Collaborative-Filtering: Recommendations based on how other users rated items. Similar users are identified and the items that they rated well are recommended. Has cold-start, sparsity and scalibility issues. First research paper released in the mid-1990s, still used widely [8].
- Content-based filtering: The requirements for this approach are features and the ratings of the items by users. A classifier for users' 'profile' is built and similar items are recommended based on it. It has the problem of recommending only very similar results that the user already is aware of. It was first mentioned on an academic paper on 1998 [8].
- Knowledge-based filtering: This approach also requires features of the items and explicit description of what the user needs or wants. Then, items that match those needs are recommended. The advantage of this approach is the fact that the system doesn't need data from different users. Unfortunately, the suggestion ability is rather static [4]. Research about this topic was first released on 1999.
- Hybrid Recommender Systems: Since all of the methods have some drawbacks, applications have shifted to combining more than one of the previous approaches. Hybrid recommender systems are still widely used by big companies like Amazon [1],Spotify [2] and Netflix [3].

### 1.1.6 Evaluation Metrics

Evaluation metrics are bla

#### Offline Evaluation

- Accuracy
- Precision
- Recall

#### Online Evaluation

Off-line experiments can measure the quality of the chosen algorithm in fulfilling its recommendation task. However, such evaluation cannot provide any insight about the user satisfaction, acceptance or experience with the system. The algorithms might be very accurate in solving the core recommendation problem, i.e., predicting user ratings, but for some other reason the system may not be accepted by users, for example, because the performance of the system was not as expected. Therefore, a user-centric evaluation is also required. It can be performed online after the system has been launched, or as a focused user study. During on-line evaluation, real users interact with the system without being aware of the full nature of the experiment running in the background. It is possible to run various versions of the algorithms on different groups of users for comparison and analysis of the system logs in order to enhance system performance. In addition, most of the algorithms include parameters, such as weight thresholds, the number of neighbors, etc., requiring constant adjustment and calibration.

[kopi - peyst]

More information in research part

### 1.1.7 User Satisfaction

[proposaldan]

- The recommendation performance is mainly evaluated in terms of accuracy (i.e. difference between true and predicted rating). - This might not be a desired goal of the talent/project recommendations (e.g. simply recommending a 'top-N' list of talents that best match the requirements might result in similar talents whose skills only vary in a small extent). - Besides the desired diversity, there might be other properties necessary for adequate recommendations (e.g. privacy, data security, diversity, serendipity, labeling, and presentation, and group recommendation).

## 1.2 Motivation

- Thus, the aim of this thesis is to investigate how these properties transfer to the talent/project matching (i.e. are they necessary or not) and how they can be algorithmically achieved.

[Also explain job recommender if we do that]

## 1.3 N/a

In recent years, the interest in recommender systems has dramatically increased, as the following facts indicate: 1. Recommender systems play an important role in highly-rated Internet sites such as Amazon.com, YouTube, Netflix, Spotify, LinkedIn, Facebook, Tripadvisor, Last.fm, and IMDb. Moreover many media companies are now developing and deploying RSs as part of the services they provide to their subscribers. For example, Netflix, the online provider of on-demand streaming media, awarded a million dollar prize to the team that first succeeded in substantially improving the performance of its recommender system [31]. 2. There are conferences and workshops dedicated specifically to the field, namely the Association of Computing Machinery's (ACM) Conference Series on Recommender Systems (RecSys), established in 2007. This conference stands as the premier annual event in recommender technology research and applications. In addition, sessions dedicated to RSs are frequently included in more traditional conferences in the area of databases, information systems and adaptive systems. Additional noteworthy conferences within this scope include: ACM's Special Interest Group on Information Retrieval (SIGIR); User Modeling, Adaptation and Personalization (UMAP); Intelligent User Interfaces (IUI); World Wide Web (WWW); and ACM's Special Interest Group on Management Of Data (SIGMOD). 3. At institutions of higher education around the world, undergraduate and graduate courses are now dedicated entirely to RSs, tutorials on RSs are very popular at computer science conferences, and a book introducing RSs techniques has been published as well [27]. Springer is publishing several books on specific topics in recommender systems in its series: Springer Briefs in Electrical and Computer Engineering. A large, new collection of articles dedicated to recommender systems applications to software engineering has also recently been published [46]. 4. There have been several special issues in academic journals which cover research and developments in the RSs field. Among the journals that have dedicated issues to RSs are: AI Communications (2008); IEEE Intelligent Systems (2007); International Journal of Electronic Commerce (2006); International Journal of Computer Science and Applications (2006); ACM Transactions on Computer Human Interaction (2005); ACM Transactions on Information Systems (2004); User Modeling and User-Adapted Interaction (2014, 2012); ACM Transactions on Interactive Intelligent Systems (2013); and ACM Transactions on Intelligent Systems and Technology (2015). [RRS15]

# 2 Review of Literature and Research

## 2.1 Types of Recommender Systems

[kopi peyst] Item recommendation approaches can be divided in two broad categories: per- sonalized and non-personalized. Among the personalized approaches are content-based and collaborative filtering methods, as well as hybrid techniques combining these two types of methods. The general principle of content-based (or cognitive) methods [4, 8, 42, 54] is to identify the common characteristics of items that have received a favorable rating from a user, and then recommend to this user new items that share these characteristics. Recommender systems based purely on content generally suffer from the problems of limited content analysis and over- specialization [63]. Limited content analysis occurs when the system has a limited amount of information on its users or the content of its items. For instance, privacy issues might refrain a user from providing personal information, or the precise content of items may be difficult or costly to obtain for some types of items, such as music or images. Another problem is that the content of an item is often insufficient to determine its quality. Over-specialization, on the other hand, is a side effect of the way in which content-based systems recommend new items, where the predicted rating of a user for an item is high if this item is similar to the ones liked by this user. For example, in a movie recommendation application, the system may recommend to a user a movie of the same genre or having the same actors as movies already seen by this user. Because of this, the system may fail to recommend items that are different but still interesting to the user.

Instead of depending on content information, collaborative (or social) filtering approaches use the rating information of other users and items in the system. The key idea is that the rating of a target user for a new item is likely to be similar to that of another user, if both users have rated other items in a similar way. Likewise, the target user is likely to rate two items in a similar fashion, if other users have given similar ratings to these two items. Collaborative approaches overcome some of the limitations of content-based ones. For instance, items for which the content is not available or difficult to obtain can still be recommended to users through the feedback of other users. Furthermore, collaborative recommendations are based on the quality of items as evaluated by peers, instead of relying on content that may be a bad indicator of quality. Finally, unlike content-based systems, collaborative filtering ones can recommend items

with very different content, as long as other users have already shown interest for these different items.

Collaborative filtering approaches can be grouped in the two general classes of neighborhood and model-based methods. In neighborhood-based (memory-based [10] or heuristic-based [2]) collaborative filtering [14, 15, 27, 39, 44, 48, 57, 59, 63], the user-item ratings stored in the system are directly used to predict ratings for new items. This can be done in two ways known as user-based or item-based recommendation. User-based systems, such as GroupLens [39], Bellcore video [27], and Ringo [63], evaluate the interest of a target user for an item using the ratings for this item by other users, called neighbors, that have similar rating patterns. The neighbors of the target user are typically the users whose ratings are most correlated to the target user's ratings. Item-based approaches [15, 44, 59], on the other hand, predict the rating of a user for an item based on the ratings of the user for similar items. In such approaches, two items are similar if several users of the system have rated these items in a similar fashion.

In contrast to neighborhood-based systems, which use the stored ratings directly in the prediction, model-based approaches use these ratings to learn a predictive model. Salient characteristics of users and items are captured by a set of model parameters, which are learned from training data and later used to predict new ratings. Model-based approaches for the task of recommending items are numerous and include Bayesian Clustering [10], Latent Semantic Analysis [28], Latent Dirich- let Allocation [9], Maximum Entropy [72], Boltzmann Machines [58], Support Vector Machines [23], and Singular Value Decomposition [6, 40, 53, 68, 69]. A survey of state-of-the-art model-based methods can be found in Chap. 3 of this book.

Finally, to overcome certain limitations of content-based and collaborative filtering methods, hybrid recommendation approaches combine characteristics of both types of methods. Content-based and collaborative filtering methods can be combined in various ways, for instance, by merging their individual predictions into a single, more robust prediction [8, 55], or by adding content information into a collaborative filtering model [1, 3, 51, 65, 71]. Several studies have shown hybrid recommendation approaches to provide more accurate recommendations than pure content-based or collaborative methods, especially when few ratings are available [2].

## 2.2 Neighborhood-Based Approach

[kopi peyst] While recent investigations show state-of-the-art model-based approaches superior to neighborhood ones in the task of predicting ratings [40, 67], there is also an emerging understanding that good prediction accuracy alone does not guarantee users an effective and satisfying experience.

Model-based approaches excel at characterizing the preferences of a user with latent factors. For example, in a movie recommender system, such methods may determine that a given user is a fan of movies that are both funny and romantic, without having to actually define the notions "funny" and "romantic". This system would be able to recommend to the user a romantic comedy that may not have been known to this user. However, it may be difficult for this system to recommend a movie that does not quite fit this high-level genre, for instance, a funny parody of horror movies. Neighborhood approaches, on the other hand, capture local associations in the data. Consequently, it is possible for a movie recommender system based on this type of approach to recommend the user a movie very different from his usual taste or a movie that is not well known (e.g. repertoire film), if one of his closest neighbors has given it a strong rating. This recommendation may not be a guaranteed success, as would be a romantic comedy, but it may help the user discover a whole new genre or a new favorite actor/director. -> Model based suck at serendipity

- Simplicity: Neighborhood-based methods are intuitive and relatively simple to implement. In their simplest form, only one parameter (the number of neighbors used in the prediction) requires tuning.

- Justifiability: Such methods also provide a concise and intuitive justification for the computed predictions. For example, in item-based recommendation, the list of neighbor items, as well as the ratings given by the user to these items, can be presented to the user as a justification for the recommendation. This can help the user better understand the recommendation and its relevance, and could serve as basis for an interactive system where users can select the neighbors for which a greater importance should be given in the recommendation [6].

- Efficiency: One of the strong points of neighborhood-based systems are their efficiency. Unlike most model-based systems, they require no costly training phases, which need to be carried at frequent intervals in large commercial applications. These systems may require pre-computing nearest neighbors in an offline step, which is typically much cheaper than model training, providing near instantaneous recommendations. Moreover, storing these nearest neighbors requires very little memory, making such approaches scalable to applications having millions of users and items.

- Stability: Another useful property of recommender systems based on this approach is that they are little affected by the constant addition of users, items and ratings, which are typically observed in large commercial applications. For instance, once item similarities have been computed, an item-based system can

readily make recommendations to new users, without having to re-train the system. Moreover, once a few ratings have been entered for a new item, only the similarities between this item and the ones already in the system need to be computed.

While neighborhood-based methods have gained popularity due to these advantages, they are also known to suffer from the problem of limited coverage, which causes some items to be never recommended. Also, traditional methods of this category are known to be more sensitive to the sparseness of ratings and the cold-start problem, where the system has only a few ratings, or no rating at all, for new users and items. Section 2.5 presents more advanced neighborhood-based techniques that can overcome these problems.

# 3 Evaluation Metrics

## 3.1 Introduction

Recommender Systems (RSs) are software tools and techniques that provide suggestions for items that are most likely of interest to a particular user [RRS15].

Suggestions and items depend on the field that recommder system is applied. For example, for the topic of news article recommenders, the aim will most likely be suggesting news to the readers. In the field of job recommenders though, these suggestions can be bidirectional. Meaning that, job postings can be suggested to applicants or resumes can be recommended to the human resources team of a company.

See Table 3.1, Figure 3.1, Figure 3.2, Figure 3.3.

Table 3.1: An example for a simple table.

| A | B | C | D |
|---|---|---|---|
| 1 | 2 | 1 | 2 |
| 2 | 3 | 2 | 3 |

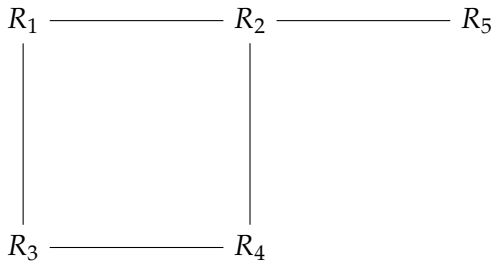


Figure 3.1: An example for a simple drawing.

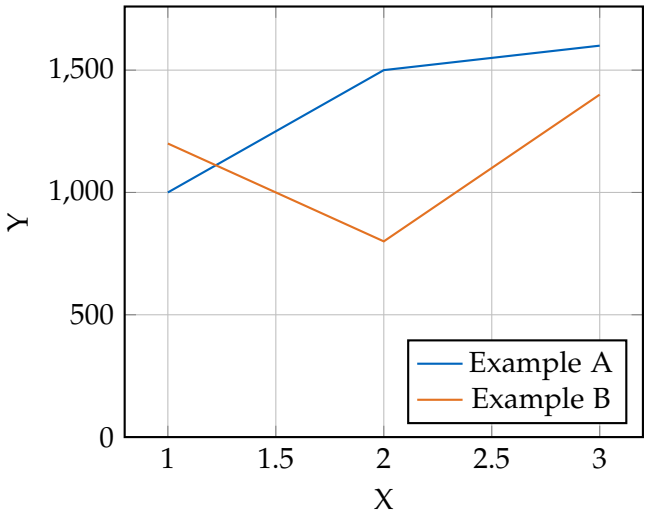


Figure 3.2: An example for a simple plot.

```
SELECT * FROM tbl WHERE tbl.str = "str"
```

Figure 3.3: An example for a source code listing.

# List of Figures

# List of Tables

# Bibliography

[Bee+16] J. Beel, B. Gipp, S. Langer, and C. Breitinger. "Research-paper recommender systems: a literature survey." In: *International Journal on Digital Libraries* 17.4 (Nov. 2016), pp. 305–338. ISSN: 1432-1300. DOI: 10.1007/s00799-015-0156-0.

[Par+12] D. H. Park, H. K. Kim, I. Y. Choi, and J. K. Kim. "A literature review and classification of recommender systems research." In: *Expert Systems with Applications* 39.11 (2012), pp. 10059–10072. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2012.02.038.

[RRS15] F. Ricci, L. Rokach, and B. Shapira. "Recommender Systems: Introduction and Challenges." In: *Recommender Systems Handbook*. Ed. by F. Ricci, L. Rokach, and B. Shapira. Boston, MA: Springer US, 2015, pp. 1–34. ISBN: 978-1-4899-7637-6. DOI: 10.1007/978-1-4899-7637-6_1.