

Topic: Synthesis and Evaluation of various Recommender System approaches against a Movie database dataset

## Table of Contents

<b>1</b>	<b><i>Abstract</i></b> .....	<b>3</b>
<b>2</b>	<b><i>Recommender Systems</i></b> .....	<b>4</b>
2.1	Introduction .....	4
2.2	History .....	4
2.3	Overview .....	5
<b>3</b>	<b><i>Basic principles</i></b> .....	<b>6</b>
3.1	Beyond accuracy .....	6
3.1.1	Netflix, Last.fm and Pandora .....	8
3.1.2	Recommender systems and software engineering .....	8
3.1.3	Recommender systems and Big data .....	8
3.1.4	Future of Recommender systems .....	8
3.2	Approaches .....	8
3.2.1	Collaborative filtering .....	9
3.2.2	Content-based recommender systems .....	11
3.2.3	Knowledge-based recommender systems .....	11
3.2.4	Other approaches .....	11
3.3	Evaluation .....	12
<b>4</b>	<b><i>Methodology</i></b> .....	<b>12</b>
<b>5</b>	<b><i>Bibliography</i></b> .....	<b>14</b>

# 1 Abstract

The idea of this thesis is to implement and evaluate multiple recommender system categories against the publicly available MovieLens dataset. Primary objective is to see how using various implementations and algorithms affects the recommendations and end-user experience.

We will be using both the Collaborative filtering algorithms (both neighborhood and model-based) and Content-based recommender system implementations.

For evaluation metrics, the main focus will be on comparison between predicted and actual outcomes. Nonetheless, an attempt will be made in order to evaluate predicted users experience.

The outcome of this thesis should be a clear overview of which implementation brings the best results and what steps could be taken to further enhance the outcome.

## 2 Recommender Systems

### 2.1 Introduction

Recommender systems have evolved as a natural response to ever growing amount of information available to wide audiences of users. With the rise of Internet and general availability of it to common users, it has become necessary to help users navigate the content and steer them towards the new options they might have not anticipated in the first place.

Strictly technically speaking, Recommender systems represent a subclass of Information filtering systems whose main purpose is to predict the “ratings” or “preferences” a user would give to an item [1].

### 2.2 History

There does not seem to be an official information on when the recommender systems have been mentioned for the first time.

One of the earliest mentions of Recommender systems as such dates to early 1990's where the official term has been mentioned in a technical report written by Jussi Karlgren at Columbia University [2].

Since then, especially with the rise of Cloud computing and Big data, when processing of vast amounts of data became possible, recommender systems have become an important aspect and indispensable commodity of all successful businesses.

It is especially important to mention the efforts of GroupLens working group [3]. GroupLens is a group of scientists from the department of computer science and engineering in University of Minnesota. This group has pushed forward the efforts towards both publishing various research papers on recommender systems and publishing the freely available datasets to be analyzed and used for educational and research purposes. Some of their featured projects include MovieLens – a web site that helps people find movies to watch, Cyclopath – an editable map where anyone can find maps and routes for riding bicycle and LensKit – open source toolkit for building, researching and studying recommender systems. MovieLens is also the dataset that this thesis is using for evaluation of results.

## 2.3 Overview

It has been stated before that one of the main driving factors for introduction of recommender systems was development of Internet and World wide web. Once the data became publicly available and many users gained access to it, there was a need to help users browse the content and steer their attention towards the items that are curated for them.

Another highly important catalyst that has driven this development is the ease with which users are able to express their preferences. Namely, today, users are able to demonstrate their liking or disliking of a certain product by a single click of mouse. This one second of users feedback multiplied by number of users and amount of items they interact with, results in enormous volumes of data which are then used to recommend even more fine-grained items. This loop repeats itself.

Charu C. Aggarwal in his Recommender systems book [4] takes a Netflix as an example. He suggests to take content providers, such as Netflix, as an example. *“In such cases”, he further states, “users are able to easily provide feedback with a simple click of a mouse. A typical methodology to provide feedback is in the form of ratings, in which users select numerical values from a specific evaluation system (e.g., five-star rating system) that specify their likes and dislikes of various items.”.*

The aforementioned approaches for feedback collection are usually referred to as “Explicit ratings” or “Explicit feedback”. This naming stems from the fact that the rating or the feedback of the item being recommended was explicitly specified by user. This also means that user is most likely consent to share his preferences with the service provider and is looking forward towards getting more curated content.

In contrast to explicit ratings, there is a group of so called “Implicit ratings” or “Implicit feedback”. Namely, this is the kind of feedback that can be derived based on the actions and behavior of user in question. Perfect example of such scenario is Amazon.com. As Aggarwal [4] states, a simple act of a user buying or browsing an item may be viewed as an endorsement for that item. Another example would be a YouTube.com. Having a user watch the video from beginning to an end is a perfect example of a positive feedback where users expresses his interest towards the item being watched. On the other hand, user skimming through the video being played is an indirect act of providing negative feedback. This negative feedback should be treated with caution as it can happen that user is not interested only at that moment, but might be looking forward to interacting with same item on another occasion. It is up to the

designer of system to make sure that the feedback is evaluated properly and all relevant things are being taken into consideration.

### 3 Basic principles

Before starting to dig into the basic principles of recommender systems, it is rather important to introduce a basic terminology that will be used throughout this thesis.

Broadly speaking, the entity to whom the recommendation is being made, and based on whose feedback is the decision being based on, is called *user*. User does not have to, necessarily be a human, but can really be any entity that is interacting with system that we are currently predicting the recommendations against.

Product that is being recommended to aforementioned user is called *item*. As mentioned above, what is said for users is also valid for items – this does not have to be a physical item, but can rather be a content, commodity or any other type of service being offered by system in question.

Broadly speaking, no matter which type of recommender system are we referring to, Aggarwal [4] identifies two distinct categories:

1. Prediction version of problem – in this version, we are dealing with incomplete  $m \times n$  matrix, where rows of the matrix  $m$  represent users, and  $n$  columns represent the items. This matrix is incomplete because not all users have specified ratings for all items. The goal here is to find the best-fitting values that would make this matrix complete. This problem is usually referred to as *matrix completion problem*.
2. Ranking version of problem – in contrast to prediction, ranking problems are usually concerned with selection of top-k items that user might be interested in. This sort of systems is usually found in e-commerce websites and online shops.

Whichever category is being used, recommender systems are usually implemented in order to increase the user's engagement with the service provider, with the end result of increasing the sales and overall profit.

#### 3.1 Beyond accuracy

*Accuracy* of the recommendation system is one of the most important evaluation metrics. Bad or inaccurate recommendations would surely lead to user's dissatisfaction and lost profit.

However, even though it is one of the most important metrics, there are others non-directly observable, which have a strong influence on quality of recommendations.

Kaminskas et al. [5] mention the following important metrics:

1. *Relevance* – as mentioned before, this is, indeed, one of the most important metrics that has to be measured. What's more, having training and test sets, this is one of the easiest metrics to be evaluated. However, it's not important to treat this in isolation and influence of other metrics is important as well
2. *Coverage* – refers to a degree to which the recommender system covers the specter of available items. The more the available items are included in recommendations, the higher the coverage is.
3. *Diversity* – if recommender system keeps recommending the most popular items, user might either get overwhelmed or stop liking them, and the risk of negative feedback increases. Diversity refers to a degree on to which the recommender system is able to break apart common recommendations, while still being able to suggest relevant items to the user.
4. *Novelty* – this is an important metric that refers to recommending something that user hasn't seen or experienced in the past
5. *Serendipity* – even though it sounds similar to novelty, this is a metric that, on certain occasions may have even stronger influence than novelty. While the former one refers to a content that user hasn't seen before, it is still somewhat expected. Serendipity, on the other hand, refers to a content that user might even consider as being lucky to have found. This metric is usually related to, so called "latent interests", which user might not even have been aware of in the first place. We'll use Youtube.com as an example here. Let's consider user who mostly enjoys the techno music and occasionally enjoys watching travel channels and videos with nature. Recommending new DJs or new travel videos that he hasn't seen before can be considered as *novelty*. However, combining those two interests and recommending a video with DJ playing at a concert in the nature, would be truly serendipitous experience. On the other hand, as Aggarwal [4] states, serendipitous algorithms often end up recommending irrelevant content. Still, long term benefits of such experiences seem to outweigh the short-term disadvantages.

It is sufficient to say that, with the exception of relevance, all other factors are harder to measure and evaluate, as they are mostly related to users experience and interaction with the system.

Nevertheless, the attention should be put towards trying to increase all of the mentioned metrics.

In the end, it is important to mention that not all services and recommender systems are inclined towards increasing the sales and overall revenue. Instead, there are instances where recommendations are used with sole purpose of increasing users engagement with a product. Let's take Facebook.com as an example. Facebook uses the recommender systems for recommending new friends and groups to the user, which, in turn, directly increases users engagement with the product. This engagement results in user spending more time on the website and more ads being displayed, which finally leads to more revenue. As can be seen, this is an example where recommender system is not directly driving the revenue, but rather passively, by increasing users engagement.

### 3.1.1 Netflix, Last.fm and Pandora

Write a bit about how recommender systems were/are used at these

### 3.1.2 Recommender systems and software engineering

Read [https://www.researchgate.net/profile/Gerald\\_Ninaus/publication/261263565\\_recommender\\_systems\\_future/links/00463533bd544bf3c3000000/recommender-systems-future.pdf](https://www.researchgate.net/profile/Gerald_Ninaus/publication/261263565_recommender_systems_future/links/00463533bd544bf3c3000000/recommender-systems-future.pdf) more in

### 3.1.3 Recommender systems and Big data

Search for some work in this area and write a bit about it ...

### 3.1.4 Future of Recommender systems

Write a bit about how we see the future of Recommender systems, etc.

## 3.2 Approaches

### 3.2.1 Basic models

Generally speaking, there are roughly two distinct kinds of recommender systems – ones that are working with incomplete matrices of users and ratings and ones that are modeling against the attributes of the item being recommended. Former ones are referred to as *Collaborative*



*Filtering* methods, because other users collaborate in order to derive the predicted rating for item being rated. The latter ones are usually referred to as *Content-based filtering* methods because the attributes of the item itself are what drives the recommendation. Content-based filtering methods are usually user-centric as the predictor is built for every single user and usually considers users previous behavior mixed with item's attributes. There is also a third category of recommender systems which is known as *Knowledge-based* systems. These are, as the name suggests, systems that are based solely on *user's requirements* which are explicitly specified by user before the recommendation process is started at all. Finally, there is a group of so called *hybrid systems* which combine multiple approaches in order to derive the best possible rating for user in question.

Some examples of Collaborative-filtering methods include popular services like Netflix.com, Youtube.com and Spotify.com. These are all the services which use ratings of other users in order to drive predictions for the current user. When it comes to Content-based filtering, a perfect example would be a movie recommendation service that learns their users preferences based on their previous feedback. Such an example is a MovieLens project, a movie recommendation service developed by GroupLens research group [3].

### 3.2.2 Collaborative filtering

Collaborative filtering represents a family of recommender systems where recommendations are solely based on so called “collaboration” between users or items. Therefore, the predicted recommendations are based solely on how other users have rated the same item (in case of user-based collaborative algorithms) or how the majority of items was rated (in case of item-based collaborative algorithms). Main challenge with collaborative filtering algorithms seems to be the fact that underlying rating matrices are usually incomplete [4].

We'll take an example of a Movie recommendation engine. In systems like this one, users usually express their liking or disliking of a particular movie, or, in some cases, specify the numerical rating. The main problem that collaborative filtering algorithms suffer from is the fact that, out of the whole universe of available movies, only a handful of them is actually rated by users. The remaining majority is usually either sparsely rated or not rated at all.

The basic premise of collaborative filtering algorithms is that the missing ratings can be imputed by observing the ratings of similar users or products, depending on what is being observed. What's more, one of the main premises that drives these engines is the assumption

that if two users have a similar taste and have similarly rated number of items, then, they must have the similar taste and one can infer the ratings from the other.

Let's take as an example two users Alice and Bob. We will assume that both Alice and Bob have a similar taste in movies and we conclude this by observing the ratings they provided for the same set of items. Let's further assume that Bob is looking for a new movie to watch. Supposing that they have a similar taste and that Alice has liked Terminator movie which Bob hasn't rated yet, we can assume that Bob might be interested in watching Terminator as well.

These algorithms, unfortunately, suffer from so called *cold start* problem. This topic which will be discussed in greater length, generally refers to the case where there are either no previous ratings to suggest, which is a case in newly built systems, or to a case when a new item is added which was never rated before. This matter will be discussed at greater length in later text.

### 3.2.2.1 Types of Collaborative Filtering systems

There are two distinctively different categories of Collaborative filtering systems [4]:

1. *Memory-based methods* and
2. *Model-based methods*

*Memory-based methods*, also referred to as “lazy” methods are the simplest and one of the earliest experimented with and implemented methods of recommendation systems. Their process of working is relatively simple – for any given user, find the set of users who appear to be similar and make recommendations based on what other users have liked.

These can be further split into two categories:

1. *User-based collaborative filtering* and
2. *Item-based collaborative filtering*

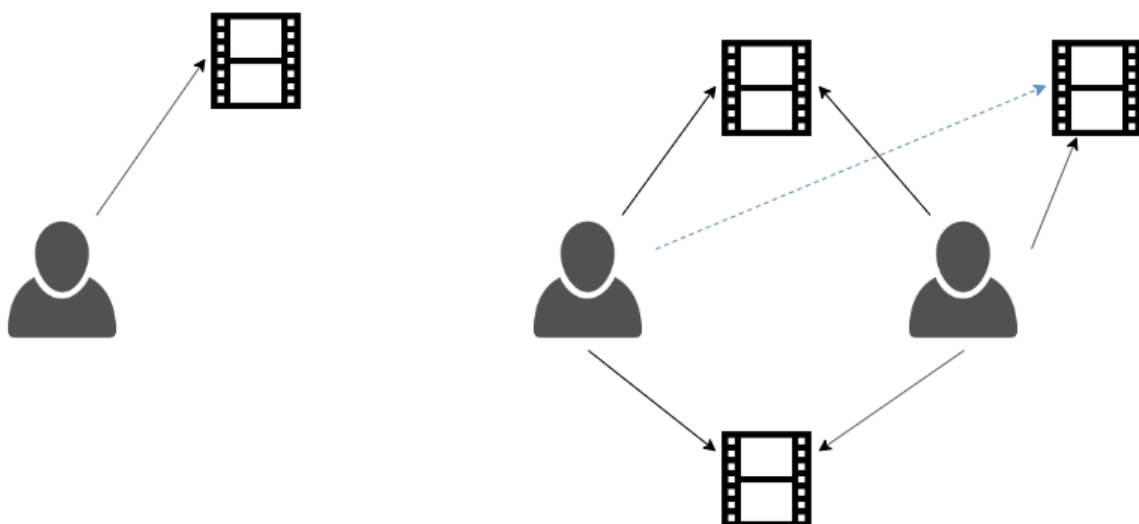
In User-based systems, the idea is to find the like-minded users and base the recommendations on that. As mentioned in previous chapter, if two users, Alice and Bob, have like the same movies in the past, then we can assume that movies that were positively rated by Alice but have not been rated by Bob are going to be positively perceived by him as well. Generally speaking, the underlying assumption is that top-k users who are most similar to Bob can actually predict the movies that Bob will like.

In Item-based systems, the idea is generally similar. In order to predict item B to user A, we first need to find a set of top-k items that are most similar to B and based on that we can conclude and infer the rating of user A for item B.

For example, let's assume that Bob has expressed positive liking towards the movies Thor, Avengers and Captain America. We can make a conclusion that Bob likes Marvel's movies and, based on that, we look for movies that are similar and one of the movies that can be recommended is Guardians of the Galaxy.

As can be seen, both user-based and item-based methods work on one-by-one basis, and that is exactly why they are also referred to as being "lazy". This comes from the fact that the ratings are known only at the time of evaluation and not before that.

The advantages of memory-based techniques is that they are very simple to implement and resulting recommendations are often very easy to explain [4].



*Figure 1 – User 2 and User 3 like the same items and item 4 being recommended based on that*

In the model-based filtering systems,

3.2.3 Content-based recommender systems

3.2.4 Knowledge-based recommender systems

3.2.5 Other approaches

### 3.3 Evaluation

## 4 Methodology

Write a bit about the data set that will be used. Show some data, etc.

Write about how the results will be evaluated.



## 5 Bibliography

- [1] F. R. a. L. R. a. B. Shapira, Introduction to Recommender Systems Handbook, Springer, 2011.
- [2] J. Karlgren, An Algebra for Recommendations, Syslab Working Paper 179, 1990.
- [3] GroupLens research project, <https://grouplens.org/>.
- [4] C. C. Aggarwal, Recommender systems: The Textbook, Springer, 2016.
- [5] M. Kaminskis and D. Bridge, Diversity, Serendipity, Novelty, and Coverage: A Survey and Empirical Analysis of Beyond-Accuracy Objectives in Recommender Systems, ACM Trans Interact Intell Syst, 2016.