

Submission

# **On the Difficulty of Evaluating Baselines A Study on Recommender Systems**

**Marc Feger, B.Sc.**

# Contents

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>Introduction</b>                               | <b>1</b>  |
| <b>2</b> | <b>A Study on Recommender Systems</b>             | <b>1</b>  |
| 2.1      | Recommender Problem . . . . .                     | 1         |
| 2.2      | Content-Based . . . . .                           | 2         |
| 2.3      | Collaborative-Filtering . . . . .                 | 2         |
| 2.4      | Matrix-Factorization . . . . .                    | 2         |
| 2.4.1    | Basic Matrix-Factorization . . . . .              | 3         |
| 2.4.2    | Regulated Matrix-Factorization . . . . .          | 3         |
| 2.4.3    | Weighted Regulated Matrix-Factorization . . . . . | 3         |
| 2.4.4    | Biased Matrix-Factorization . . . . .             | 3         |
| 2.4.5    | Advanced Matrix-Factorization . . . . .           | 3         |
| 2.5      | Optimization and Learning . . . . .               | 4         |
| 2.5.1    | Stochastic Gradient Descent . . . . .             | 4         |
| 2.5.2    | Alternating Least Square . . . . .                | 4         |
| 2.5.3    | Bayesian Learning . . . . .                       | 5         |
| 2.6      | Short Summary of Recommender Systems . . . . .    | 5         |
| <b>3</b> | <b>On the Diffculty of Evaluating Baselines</b>   | <b>5</b>  |
| 3.1      | Motivation and Background . . . . .               | 5         |
| 3.1.1    | Netflix-Prize . . . . .                           | 5         |
| 3.1.2    | MovieLens . . . . .                               | 6         |
| 3.2      | Experiment Realization . . . . .                  | 6         |
| 3.2.1    | Experiment Preparation . . . . .                  | 7         |
| 3.2.2    | Experiment Implementation . . . . .               | 8         |
| 3.3      | Obeservations . . . . .                           | 9         |
| 3.3.1    | Stronger Baselines . . . . .                      | 9         |
| 3.3.2    | Reproducability . . . . .                         | 9         |
| 3.3.3    | Inadequate validations . . . . .                  | 10        |
| <b>4</b> | <b>Conclusion</b>                                 | <b>10</b> |
| <b>5</b> | <b>Critical Assessment</b>                        | <b>11</b> |

# 1 Introduction

Today's use of *recommender systems* finds an increased and yet unconscious access to our everyday life. More and more areas of life are therefore subject to constant optimisation. Companies such as *Netflix*, *Amazon* and *YouTube* adapt their product proposals to the individual wishes of their customers. To make this possible, the various *collaborative-filtering* and *content-based recommender systems* are used.

Since [Karlgrén \(1990\)](#) first presented *recommender systems* as a kind of intelligent bookcase, much effort has been put into the development and research of such systems. The most diverse subject areas were not only illuminated by the industry. A whole new branch of research also opened up for science.

In their work "*On the Difficulty of Evaluating Baselines A Study on Recommender Systems*" [Rendle et al. \(2019\)](#) show that current research on the *MovieLens10M-dataset* leads in a wrong direction. In addition to general problems, they particularly list wrong working methods and misunderstood *baselines* by breaking them by a number of simple methods such as *matrix-factorization*.

They were able to beat the existing *baselines* by not taking them for granted. On the contrary, they questioned them and transferred well evaluated and understood properties of the *baselines* from the *Netflix-Prize* to them.

As a result, they were not only able to beat the *baselines* reported for the *MovieLens10M-dataset*, but also the newer methods from the last five years of research. Therefore, it can be assumed that the current and former results obtained on the *MovieLens10M-dataset* were not sufficient to be considered as a true *baseline*. Thus they show the *community* a critical error on which can be found not only in the evaluation of *recommender systems* but also in other scientific areas.

The first problem the authors point out that, scientific papers whose focus is on better understanding and improving existing *baselines* do not receive recognition because they do not seem innovative enough. In contrast to industry, which tenders horrendous prizes for researching and improving such *baselines*, there is a lack of such motivation in the scientific field. From the authors point of view, the scientific work on the *MovieLens10M-dataset* is misdirected, because *one-off evaluations* leading to *one-hit-wonders*, which are then used as a starting point for further work. Thus [Rendle et al. \(2019\)](#) points out as a second point of criticism, that the need for further basic research for the *MovieLens10M-dataset* is not yet exhausted.

This submission takes a critical look at the topic presented by [Rendle et al. \(2019\)](#). In addition, basic terms and the results obtained are presented in a way that is comprehensible to the non-experienced reader. For this purpose, the submission is divided into three subject areas. First of all, the non-experienced reader is introduced to the topic of *recommender systems* in the section "*A Study on Recommender Systems*". Subsequently, building on the first section, the work in the section "*On the Difficulty of Evaluating Baselines*" is presented in detail. The results are then evaluated in a critical discourse.

## 2 A Study on Recommender Systems

This section explains the basics of *recommender systems* necessary for the essential understanding of the paper presented. Besides the general definition of the *recommender problem*, the corresponding solution approaches are presented. Furthermore, the main focus will be on the solution approach of *matrix-factorization*.

### 2.1 Recommender Problem

The *recommender problem* consists of the entries of the sets  $\mathcal{U}$  and  $\mathcal{I}$ , where  $\mathcal{U}$  represents the set of all *users* and  $\mathcal{I}$  the set of all *items*. Each of the *users* in  $\mathcal{U}$  gives *ratings* from a set  $\mathcal{S}$  of possible scores for the available *items* in  $\mathcal{I}$ . The resulting *rating-matrix*  $\mathcal{R}$  is composed of  $\mathcal{R} = \mathcal{U} \times \mathcal{I}$ . The entries in  $\mathcal{R}$  indicate the *rating* from *user*  $u \in \mathcal{U}$  to *item*  $i \in \mathcal{I}$ . This entry is then referred to as  $r_{ui}$ . Due to incomplete *item-ratings*,  $\mathcal{R}$  may also be incomplete. In the following, the subset of all *users* who have rated a particular *item*  $i$  is referred to as  $\mathcal{U}_i$ . Similarly,  $\mathcal{I}_u$  refers to the subset of *items* that were rated by *user*  $u$ . Since  $\mathcal{R}$  is not completely filled, there are missing values for some *user-item relations*. The aim of the *recommender system* is to estimate the missing *ratings*  $\hat{r}_{ui}$  using a *prediction-function*  $p(u, i)$ . The *prediction-function* consists of  $p : \mathcal{U} \times \mathcal{I} \rightarrow \mathcal{S}$  ([Desrosiers and Karypis, 2011](#)). In the further course of the work different methods are presented to determine  $p(u, i)$ .

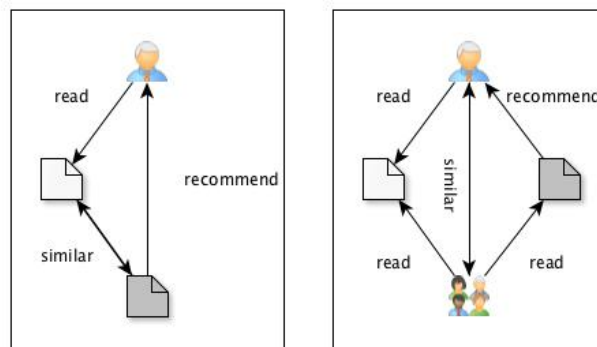
In the following, the two main approaches of *collaborative-filtering* and *content-based recommender systems* will be discussed. In addition, it is explained how *matrix-factorization* can be integrated into the two ways of thinking.

## 2.2 Content-Based

*Content-based recommender systems (CB)* work directly with *feature vectors*. Such a *feature vector* can, for example, represent a *user profile*. In this case, this *profile* contains informations about the *user's preferences*, such as *genres*, *authors*, etc. This is done by trying to create a *model* of the *user*, which best represents his preferences. The different *learning algorithms* from the field of *machine learning* are used to learn or create the *models*. The most prominent *algorithms* are: *tf-idf*, *bayesian learning*, *Rocchio's algorithm* and *neural networks* (Lops et al., 2011; Dacrema et al., 2019b; Desrosiers and Karypis, 2011). Altogether the built and learned *feature vectors* are compared with each other. Based on their closeness, similar *features* can be used to generate *missing ratings*. Figure 1a shows a sketch of the general operation of *content-based recommenders*.

## 2.3 Collaborative-Filtering

Unlike the *content-based recommender*, the *collaborative-filtering recommender (CF)* not only considers individual *users* and *feature vectors*, but rather a *like-minded neighborhood* of each *user*. Missing *user ratings* can be extracted by this *neighbourhood* and *networked* to form a whole. It is assumed that a *missing rating* of the considered *user*  $u$  for an unknown *item*  $i$  will be similar to the *rating* of a *user*  $v$  as soon as  $u$  and  $v$  have rated some *items* similarly. The similarity of the *users* is determined by the *community ratings*. This type of *recommender system* is also known by the term *neighborhood-based recommender* (Desrosiers and Karypis, 2011). The main focus of *neighbourhood-based methods* is on the application of iterative methods such as *k-nearest-neighbours* or *k-means*. A *neighborhood-based recommender* can be viewed from two perspectives: The first and best known problem is the so-called *user-based prediction*. Here, the *missing ratings* of a considered *user*  $u$  are to be determined from his *neighborhood*  $\mathcal{N}_i(u)$ .  $\mathcal{N}_i(u)$  denotes the subset of the *neighborhood* of all *users* who have a similar manner of evaluation to  $u$  and who have rated the unknown *item*  $i$ . The second problem is that of *item-based prediction*. Analogously, the similarity of the *items* are determined by their received *ratings*. This kind of problem considers the *neighborhood*  $\mathcal{N}_u(i)$  of all *items* which were rated by *user*  $u$  and who are similar to  $i$ . The similarity between the objects of a *neighborhood* is determined by *distance functions* such as *mean-squared-difference*, *pearson-correlation* or *cosine-similarity*. Figure 1b shows a sketch of the general operation of *collaborative-filtering recommender*.



(a) Content-Based. (b) Collaborative-Filtering.

Figure 1: Overview of *content-based* (left) and *collaborative-filtering* (right) recommender systems. *Content-based recommender systems* work via *feature vectors*. In contrast, *collaborative filtering recommender systems* work over *neighborhoods*.

## 2.4 Matrix-Factorization

The core idea of *matrix-factorization* is to supplement the not completely filled out *rating-matrix*  $\mathcal{R}$ . For this purpose the *users* and *items* are to be mapped to a joined *latent feature space* with *dimensionality*  $f$ . The *user* is represented by the vector  $p_u \in \mathbb{R}^f$  and the *item* by the vector  $q_i \in \mathbb{R}^f$ . As a result, the *missing ratings* and thus the *user-item interaction* are to be determined via the *inner product*  $\hat{r}_{ui} = q_i^T p_u$  of the corresponding vectors (Koren et al., 2009).

In the following, the four most classical *matrix-factorization* approaches are described in detail. Afterwards, the concrete learning methods with which the vectors are learned are presented. In addition, the *training data* for which a *concrete rating* is available should be referred to as  $\mathcal{B} = \{(u, i) | r_{ui} \in \mathcal{R}\}$ .

#### 2.4.1 Basic Matrix-Factorization

The first and easiest way to solve *matrix-factorization* is to connect the *feature vectors* of the *users* and the *items* using the *inner product*. The result is the *user-item interaction*. In addition, the *error* should be as small as possible. Therefore,  $\min_{p_u, q_i} \sum_{(u, i) \in \mathcal{B}} (r_{ui} - \hat{r}_{ui})^2$  is defined as an associated *minimization problem* (Koren et al., 2009).

#### 2.4.2 Regulated Matrix-Factorization

This problem extends the *basic matrix-factorization* by a *regulation factor*  $\lambda$  in the corresponding *minimization problem*. Since  $\mathcal{R}$  is thinly occupied, the effect of *overfitting* may occur due to learning from the few known values. The problem with *overfitting* is that the generated *ratings* are too tight. To counteract this, the magnitudes of the previous vectors are taken into account. High magnitudes are punished by the factor  $\lambda(\|q_i\|^2 + \|p_u\|^2)$  in the *minimization problem*. Overall, the *minimization problem*  $\min_{p_u, q_i} \sum_{(u, i) \in \mathcal{B}} (r_{ui} - \hat{r}_{ui})^2 + \lambda(\|q_i\|^2 + \|p_u\|^2)$  is to be solved. The idea is that especially large entries in  $q_i$  or  $p_u$  cause  $\|q_i\|$ ,  $\|p_u\|$  to become larger. Accordingly,  $\|q_i\|$  and  $\|p_u\|$  increases the larger its entries become. This value is then additionally punished by squaring it. Small values are rewarded and large values are penalized. Additionally the influence of this value can be regulated by  $\lambda$  (Koren et al., 2009).

#### 2.4.3 Weighted Regulated Matrix-Factorization

The *weighted regulated matrix-factorization* builds on the *regulated matrix-factorization*. Additional *weights*  $\alpha$  and  $\beta$  are introduced to take into account the individual magnitude of a vector. The *minimization problem* then corresponds to  $\min_{p_u, q_i} \sum_{(u, i) \in \mathcal{B}} (r_{ui} - \hat{r}_{ui})^2 + \lambda(\alpha\|q_i\|^2 + \beta\|p_u\|^2)$  (Zhou et al., 2008).

#### 2.4.4 Biased Matrix-Factorization

A major advantage of *matrix-factorization* is the ability to model simple relationships according to the application. Thus, an excellent data source cannot always be assumed. Due to the *natural interaction* of the *users* with the *items*, *preferences* arise. Such *preferences* lead to *behaviour patterns* which manifest themselves in the form of a *bias* in the data. A *bias* is not bad overall, but it must be taken into account when modeling the *recommender system*. The most popular model that takes *bias* into account is called *biased matrix-factorization*. In addition, the *missing rating* is no longer determined only by the *inner product* of the two vectors  $q_i$  and  $p_u$ . Rather, the *bias* is also considered. Accordingly, a *missing rating* is calculated by  $\hat{r}_{ui} = b_{ui} + q_i^T p_u$ , where  $b_{ui}$  is the *bias* of a *user*  $u$  and an *item*  $i$ . The *bias* is determined by  $b_{ui} = \mu + b_u + b_i$ . The parameter  $\mu$  is the *global average* of all *ratings*  $r_{ui} \in \mathcal{R}$ . Furthermore,  $b_u = \mu_u - \mu$  and  $b_i = \mu_i - \mu$ . Here  $\mu_u$  denotes the *average* of all *assigned ratings* of the *user*  $u$ . Similarly,  $\mu_i$  denotes the *average* of all *received ratings* of an *item*  $i$ . Thus  $b_u$  indicates the *deviation* of the *average assigned rating* of a *user* from the *global average*. Similarly,  $b_i$  indicates the *deviation* of the *average rating* of an *item* from the *global average*. In addition, the *minimization problem* can be extended by the *bias*. Accordingly, the *minimization problem* is then  $\min_{p_u, q_i} \sum_{(u, i) \in \mathcal{B}} (r_{ui} - \hat{r}_{ui})^2 + \lambda(\|q_i\|^2 + \|p_u\|^2 + b_u^2 + b_i^2)$ . Analogous to the *regulated matrix-factorization*, the values  $b_u$  and  $b_i$  are penalized in addition to  $\|q_i\|$ ,  $\|p_u\|$ . In this case  $b_u$ ,  $b_i$  are penalized more if they assume a large value and thus deviate strongly from the *global average* (Koren et al., 2009).

#### 2.4.5 Advanced Matrix-Factorization

This section is intended to show that there are *other approaches* to *matrix-factorization*. Thus, *implicit data* can also be included. First of all, it should be mentioned that *temporary dynamics* can also be included. On the one hand, it is not realistic that a *user* cannot change his taste. On the other hand, the properties of an *item* may also not remain constant. Therefore, *missing ratings* can also be determined *time-based*. A *missing rating* is then determined by  $\hat{r}_{ui} = \mu + b_i(t) + b_u(t) + q_i^T p_u(t)$  (Koren et al., 2009). As a second possibility, *implicit influence* can be included. This can involve the *properties* of the *items* a *user* is dealing with. A *missing rating* can be determined by  $\hat{r}_{ui} = \mu + b_i + b_u + q_i^T (p_u + |\mathcal{I}_u|^{-\frac{1}{2}} \sum_{i \in \mathcal{I}_u} y_i)$ .  $y_i \in \mathbb{R}^f$  describes the *feature vectors* of the *items*  $i \in \mathcal{I}_u$  which have been evaluated by *user*  $u$ . The corresponding *minimization problems* can be adjusted as mentioned in the sections above (Koren, 2008).

## 2.5 Optimization and Learning

An important point that does not emerge from the above sections is the question of how the individual components  $p_u, q_i, b_u, b_i$  are constructed. In the following, the three most common methods are presented.

### 2.5.1 Stochastic Gradient Descent

The best known and most common method when it comes to *machine learning* is *stochastic gradient descent* (SGD). The goal of SGD is to *minimize* the *error* of a given *objective function*. Thus the estimators mentioned in section 2.4 can be used as *objective functions*. In the field of *recommender systems*, Funk (2006) presented a *modified* variant of SGD in the context of the *Netflix-Prize*. SGD can be applied to *regulated matrix-factorization* with *bias* as well as without *bias*. This method can be described by the following pseudo code:

---

#### Algorithm 1 SGD of Funk

---

**Require:** training-matrix  $\mathcal{R}_{train}$ , initial mean  $\mu$ , initial standard deviation  $\sigma^2$ , regularization parameter  $\lambda$ , learning rate  $\gamma$ , feature embedding  $f$ , epochs to train  $n_{epochs}$

```

1:  $\mathcal{P} \leftarrow \mathcal{N}(\mu, \sigma^2)^{|\mathcal{U}| \times f}$ 
2:  $\mathcal{Q} \leftarrow \mathcal{N}(\mu, \sigma^2)^{f \times |\mathcal{I}|}$ 
3: for  $epoch \in \{0, \dots, n_{epochs} - 1\}$  do
4:   for  $(u, i) \in \mathcal{R}_{train}$  do
5:      $e_{ui} \leftarrow r_{ui} - \hat{r}_{ui}$ 
6:      $q_i \leftarrow q_i + \gamma(e_{ui}p_u - \lambda q_i)$ 
7:      $p_u \leftarrow p_u + \gamma(e_{ui}q_i - \lambda p_u)$ 
8:      $b_i \leftarrow b_i + \gamma(e_{ui} - \lambda b_i)$ 
9:      $b_u \leftarrow b_u + \gamma(e_{ui} - \lambda b_u)$ 
10:  end for
11: end for
12: return  $\mathcal{P}, \mathcal{Q}$ 
```

---

At the beginning, the matrices  $\mathcal{P}, \mathcal{Q}$  are filled with *random numbers*. According to Funk (2006) this can be done by a *gaussian-distribution*. Then, for each element in the *training set*, the entries of the corresponding vectors  $p_u \in \mathcal{P}, q_i \in \mathcal{Q}$  are recalculated on the basis of the *error* that occurred in an *epoch*. The parameters  $\lambda, \gamma$  are introduced to avoid *over-* and *underfitting*. These can be determined using *grid-search* and *k-fold cross-validation*. For the *optimization* of the parameters  $\lambda$  and  $\gamma$  the so-called *grid-search* procedure is used. A *grid* of possible parameters is defined before the analysis. This *grid* consists of the sets  $\Lambda$  and  $\Gamma$ . The *grid-search* method then trains the algorithm to be considered with each possible pair of  $(\lambda \in \Lambda, \gamma \in \Gamma)$ . The models trained in this way are then tested using a *k-fold cross-validation*. The data set is divided into  $k$ -equally large fragments. Each of the  $k$  parts is used once as a test set while the remaining  $(k - 1)$  parts are used as training data. The average error is then determined via the *k-folds* and entered into the *grid*. Thus the pair  $(\lambda \in \Lambda, \gamma \in \Gamma)$  can be determined for which the *error* is lowest. This approach is also called *Funk-SVD* or *SVD* in combination with section 2.4.2 and 2.4.4 (Rendle et al., 2019). The algorithm shown above can also be extended. Thus procedures like in section 2.4.5 can be solved. The second method from section 2.4.5 is then also called *SVD++*. A coherent SGD approach was given by Koren and Bell (2011).

### 2.5.2 Alternating Least Square

The second method often used is *alternating least square* (ALS). In contrast to SGD, the vectors  $q_i, p_u$  are adjusted in *two steps*. Since SGD  $q_i$  and  $p_u$  are both unknown, this is a *non-convex problem*. The idea of ALS is to capture one of the two vectors and work with one unknown variable each. Thus the problem becomes *quadratic* and can be solved optimally. For this purpose the matrix  $\mathcal{P}$  is filled with *random numbers* at the beginning. These should be as small as possible and can be generated by a *gaussian-distribution*. Then  $\mathcal{P}$  is recorded and all  $q_i \in \mathcal{Q}$  are recalculated according to the *least-square problem*. This step is then repeated in reverse order. ALS terminates if a *termination condition* such as the *convergence* of the error is satisfied for both steps (Zhou et al., 2008).



### 2.5.3 Bayesian Learning

The third approach is known as *bayesian learning*. With this approach the so-called *gibbs-sampler* is often used. The aim is to determine the *common distribution* of the vectors in  $\mathcal{P}$ ,  $\mathcal{Q}$ . For this purpose the *gibbs-sampler* is given an initialization of *hyperparameters* to generate the *initial distribution*. The *common distribution* of the vectors  $q_i \in \mathcal{Q}$ ,  $p_u \in \mathcal{P}$  is approximated by the *conditional probabilities*. The basic principle is to select a variable in a *reciprocal way* and to generate a value dependent on the values of the other variable according to its *conditional distribution*, with the other values remaining unchanged in each *epoch*. The approaches shown in sections 2.4.1 to 2.4.4 in combination with this learning approach are also known as *bayesian probabilistic matrix-factorization (BPMF)*. A detailed elaboration of the *BPMF* and the *gibbs-sampler* was written by [Salakhutdinov and Mnih \(2008\)](#).

## 2.6 Short Summary of Recommender Systems

As the previous section clearly shows, the field of *recommender systems* is versatile. Likewise, the individual approaches from the *CB* and *CF* areas can be assigned to unambiguous subject areas. *CF* works rather with *graph-theoretical-approaches* while *CB* uses methods from *machine learning*. Of course there are *overlaps* between the approaches. Such overlaps are mostly found in *matrix-factorization*. In addition to *classical matrix-factorization*, which is limited to *simple matrix-decomposition*, approaches such as *SVD++* and *BPMF* work with methods from *CB* and *CF*. *SVD++* uses *graph-based information* while *BPMF* uses classical approaches from *machine learning*. Nevertheless, *matrix-factorization* forms a separate part in the research field of *recommender systems*, which is strongly influenced by *CB* and *CF* ways of thinking. Figure 2 finally shows a detailed overview of the different *recommender-systems* and their dependencies.

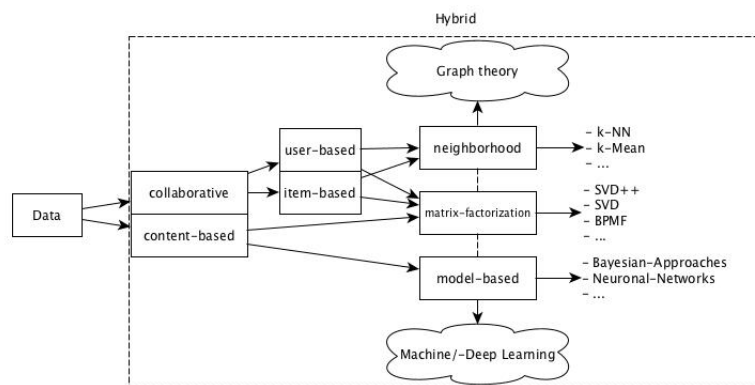


Figure 2: Overview of the entire field of the *recommender system* and their dependencies with each other.

## 3 On the Difficulty of Evaluating Baselines

This section reviews the main part of the work represented by [Rendle et al. \(2019\)](#). In addition to a detailed description and explanation of the experiments carried out and the observations gained from them, a short introduction is given regarding the driving motivation.

### 3.1 Motivation and Background

As in many other fields of *data-science*, a valid *benchmark-dataset* is required for a proper execution of experiments. In the field of *recommender systems*, the best known datasets are the *Netflix-* and *MovieLens-datasets*. This section introduces both datasets and shows the relationship of [Koren](#), one of the authors of this paper, to the *Netflix-Prize*, in addition to the existing *baselines*.

#### 3.1.1 Netflix-Prize

The topic of *recommender systems* was first properly promoted and made known by the *Netflix-Prize*. On *October 2nd 2006*, the competition announced by *Netflix* began with the *goal* of beating the self-developed *recommender system Cinematch* with an *RMSE* of *0.9514* by at least *10%*. In total, the *Netflix-dataset* was divided into three parts that can be grouped into two categories: *training* and *qualification*. In addition to a *probe-dataset* for *training* the algorithms, two further datasets were retained to qualify the winners. The

*quiz-dataset* was then used to calculate the score of the *submitted solutions* on the *public leaderboard*. In contrast, the *test-dataset* was used to determine the actual winners. Each of the pieces had around 1.408.000 elements and similar statistical values. By splitting the data in this way, it was possible to ensure that an improvement could not be achieved by *simple hill-climbing-algorithms*. It took a total of *three years* and *several hundred models* until the team *BellKor's Pragmatic Chaos* was chosen as the *winner* on *21st September 2009*. They had managed to achieve an *RMSE* of *0.8554* and thus an improvement of *0.096*. Such a result is extraordinary excellent, because it took *one year* of work and intensive research to reduce the *RMSE* from *0.8712* (*progress award 2007*) to *0.8616* (*progress award 2008*). The co-author of the present paper, [Koren](#), was significantly involved in the work of this team. Since the beginning of the event, *matrix-factorization methods* have been regarded as promising approaches. Even with the simplest SVD methods, *RMSE* values of *0.94* could be achieved by [Kurucz et al. \(2007\)](#). The *breakthrough* came through [Funk \(2006\)](#) who achieved an *RMSE* of *0.93* with his *FunkSVD*. Based on this, more and more work has been invested in the research of simple *matrix-factorization methods*. Thus, [Zhou et al. \(2008\)](#) presented an *ALS variant* with an *RMSE* of *0.8985* and [Koren \(2009\)](#) presented an *SGD variant* with *RMSE* *0.8998*. *Implicit data* were also used. For example, [Koren \(2009\)](#) could also achieve an *RMSE* of *0.8762* by extending *SVD++* with a *time variable*. This was then called *timeSVD++*.

The *Netflix-Prize* made it clear that even the *simplest methods* are *not trivial* and that a *reasonable investigation* and *evaluation* requires an *immense effort* from within the *community*.

### 3.1.2 MovieLens

In the *non-commercial sector* of *recommender systems* the *MovieLens10M-dataset* is mostly used. It consists of *10.000.054 elements* and was published by the research group *GroupLens* in *2009* ([Harper and Konstan, 2015](#)). In most cases a *global and random 90:10 split* of the data is used to evaluate the *RMSE*. This means that through a *random selection* *90%* of the data is used for *training* and *10%* of the remaining data is used for *testing*. Over the last *five years* a large number of algorithms on this dataset have been evaluated and the results have been published on *well-known conferences* such as *ICML*, *NeurIPS*, *WWW*, *SIGIR* and *AAAI*. *Figure 3* shows the *results obtained* over the last *five years* on the *MovieLens10M-dataset*. It can be clearly stated that the *existing baselines* have been *beaten* and *newer methods* have made *steady progress*.

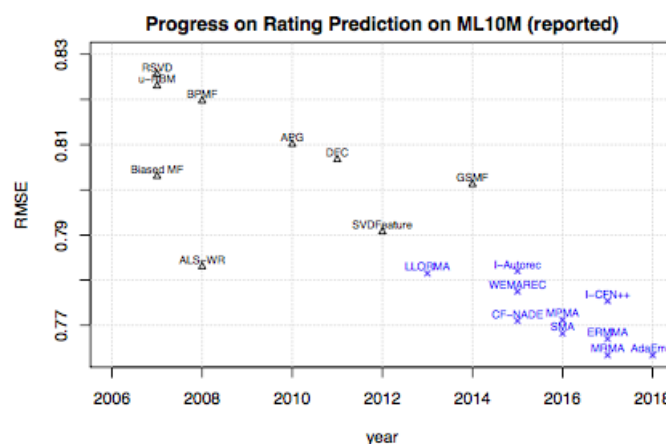


Figure 3: Results obtained on the *MovieLens10M-dataset* over the last *five years*. The *y-axis* shows the corresponding *RMSE* values and the *x-axis* shows the *year* in which the corresponding method was developed. *Blue marked points* show *newer methods* that have *competed* against the points shown in *black*. ([Rendle et al., 2019](#))

### 3.2 Experiment Realization

As the *Netflix-Prize* has shown, *research* and *validation* is *complex* even for very *simple methods*. Not only during the *Netflix-Prize* was intensive work done on researching *existing* and *new reliable methods*. The *MovieLens10M-dataset* was used just as often. With their experiment, the authors *doubt* that the *baselines*



of *MovieLens10M* are *adequate* for the evaluation of new methods. To test their hypothesis, the authors transferred all the findings from the *Netflix-Prize* to the existing baselines of *MovieLens10M*.

### 3.2.1 Experiment Preparation

Before actually conducting the experiment, the authors took a closer look at the given *baselines*. In the process, they noticed some *systematic overlaps*. These can be taken from the *table* below.

| Methods                        | Overlaps   |
|--------------------------------|--|
| <i>Biased MF, RSVD</i>         | Same method with the only difference being a different setup of the hyperparameters. |
| <i>ALS-WR, Biased MF, RSVD</i> | Same models that were learned with other approaches (SGD and ALS).                   |
| <i>BPMF, RSVD, ALS-WR</i>      | Completely different approach of learning but fundamentally the same model.          |

Table 1: *Systematic consistency of the baselines used on MovieLens10M.*

From the three aspects it can be seen that the models are fundamentally similar and that the main differences arise from different setups and learning procedures. Thus, the authors examined the two learning methods *stochastic gradient descent* and *bayesian learning* in combination with *biased matrix-factorization* before conducting the actual experiment. For  $b_u = b_i = 0$  this is equivalent to *regulated matrix-factorization* (RSVD). In addition, for  $\alpha = \beta = 1$  the *weighted regulated matrix-factorization* (WR) is equivalent to RSVD. Thus, the only differences are explained by the different adjustments of the methods. To prepare the two learning procedures they were initialized with a *gaussian-distribution*  $\mathcal{N}(\mu, 0.1^2)$ . The value for the *standard deviation* of 0.1 is the value suggested by the *factorization machine libFM* as the default. In addition, [Rendle \(2013\)](#) achieved good results on the *Netflix-Prize-dataset* with this value. Nothing is said about the parameter  $\mu$ . However, it can be assumed that this parameter is around the *global average* of the *ratings*. This can be assumed because *ratings* are to be *generated* with the *initialization*.

For both approaches the number of *sampling steps* was then set to 128. Since SGD has two additional *hyperparameters*  $\lambda, \gamma$  these were also determined. Overall, the *MovieLens10M-dataset* was evaluated by a *10-fold cross-validation* over a *random global and non-overlapping 90:10 split*. In each step, 90% of the data was used for *training* and 10% of the data was used for *evaluation* without overlapping. In each split, 95% of the *training data* was used for *training* and the remaining 5% for *evaluation* to determine the *hyperparameters*. The *hyperparameter search* was performed as mentioned in section 2.5.1 using the *grid* ( $\lambda \in \{0.02, 0.03, 0.04, 0.05\}, \gamma \in \{0.001, 0.003\}$ ) and a *64-dimensional embedding*. This grid was inspired by findings during the *Netflix-Prize* ([Koren, 2008](#); [Paterek, 2007](#)). In total the parameters  $\lambda = 0.04$  and  $\gamma = 0.003$  could be determined. Afterwards both *learning methods* and their settings were compared. The *RMSE* was plotted against the used *dimension*  $f$  of  $p_u, q_i \in \mathbb{R}^f$ . *Figure 4* shows the corresponding results.

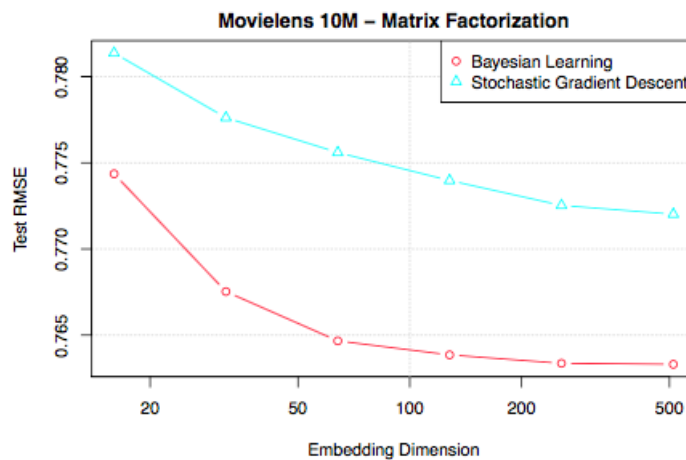


Figure 4: Comparison of *matrix-factorization* learned by *gibbs-sampling* (*bayesian learning*) and *stochastic gradient descent* (SGD) for an *embedding dimension* from 16 to 512 with 128 sampling steps.

As a *first intermediate result* of the preparation it can be stated that both *SGD* and *gibbs-sampler* achieve better *RMSE* values for increasing *dimensional embedding*.

In addition, it can be stated that learning using the *bayesian approach* is better than learning using *SGD*. Even if the results could be different due to more efficient setups, it is still surprising that *SGD* is worse than the *bayesian approach*, although the *exact opposite* was reported for *MovieLens10M-dataset*. For example, figure 3 shows that the *bayesian approach* *BPMF* achieved an *RMSE* of 0.8197 while the *SGD approach* *Biased MF* performed better with 0.803. The fact that the *bayesian approach* outperforms *SGD* has already been reported and validated by Rendle (2013), Salakhutdinov and Mnih (2008) for the *Netflix-Prize-dataset*. Looking more closely at figures 3 and 4, the *bayesian approach* scores better than the reported *BPMF* and *Biased MF* for each *dimensional embedding*. Moreover, it even beats all reported *baselines* and new methods. Building on this, the authors have gone into the detailed examination of the methods and *baselines*.

### 3.2.2 Experiment Implementation

For the actual execution of the experiment, the authors used the knowledge they had gained from the preparations. They noticed already for the two *simple matrix-factorization models* *SGD-MF* and *Bayesian MF*, which were trained with an *embedding* of 512 dimensions and over 128 epochs, that they performed extremely well. Thus *SGD-MF* achieved an *RMSE* of 0.7720. This result alone was better than: *RSVD* (0.8256), *Biased MF* (0.803), *LLORMA* (0.7815), *I-Autorec* (0.782), *WEMAREC* (0.7769) and *I-CFN++* (0.7754). In addition, *Bayesian MF* with an *RMSE* of 0.7633 not only beat the *reported baseline BPMF* (0.8197). It also beat the *best algorithm MRMA* (0.7634). As the *Netflix-Prize* showed, the use of *implicit data* such as *time* or *dependencies* between *users* or *items* could immensely improve existing models. In addition to the two *simple matrix factorizations*, table 2 shows the extensions of the authors regarding the *bayesian approach*.

| Name                 | Feature                                 | Comment   |
|----------------------|---|---|
| Matrix-Factorization | $u, i$                                  | Simple <i>matrix-factorization</i> similar to <i>biased matrix-factorization</i> and <i>RSVD</i> .  |
| timeSVD              | $u, i, t$                               | Based on the <i>matrix-factorization</i> , <i>time dependencies</i> are taken into account.   |
| SVD++                | $u, i, \mathcal{I}_u$                   | Based on the <i>matrix-factorization</i> , the <i>items</i> $\mathcal{I}_u$ that a <i>user</i> has viewed are included.                       |
| timeSVD++            | $u, i, t, \mathcal{I}_u$                | Combination of <i>SVD++</i> and <i>timeSVD</i> .  |
| timeSVD++ flipped    | $u, i, t, \mathcal{I}_u, \mathcal{U}_i$ | Extension of <i>timeSVD++</i> whereby all other <i>users</i> $\mathcal{U}_i$ who have seen a certain <i>item</i> are also taken into account. |

Table 2: *Models* and their *features* created and used by the *authors*.

As it turned out that the *bayesian approach* gave more promising results, the given models were trained with it. For this purpose, the *dimensional embedding* as well as the *number of sampling steps* for the models were examined again. As indicated in section 3.2.1, the *gaussian-distribution* was used for *initialization*. Figure 5 shows the corresponding results.

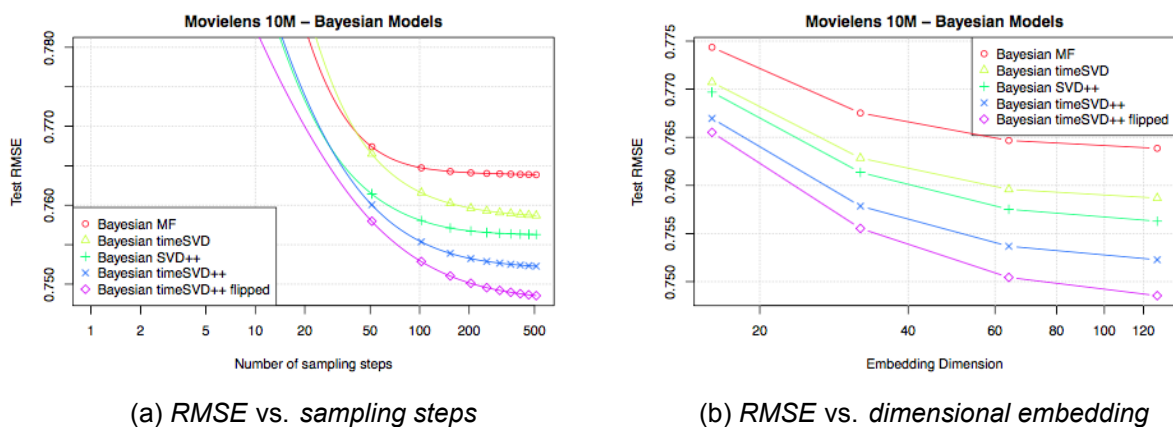


Figure 5: Final evaluation of the *number of sampling steps* and *dimensional embedding* for the designed models. Figure 5a shows the *number of sampling steps* with a *dimensional embedding* of 128 against the corresponding *RMSE*. Figure 5b shows the *RMSE* generated by 512 sampling steps with *variable dimensional embedding*.

### 3.3 Observations

The first observation that emerges from *figure 5a* is that the *increase* in *sampling steps* with a *fixed dimensional embedding* also results in an *improvement* in *RMSE* for all models. Based on this, *figure 5b* also shows that an *increase* in the *dimensional embedding* for *512 sampling steps* also leads to an *improvement* in the *RMSE* for all models. Thus, both the *number of sampling steps* and the size of the *dimensional embedding* are involved in the *RMSE* of *matrix-factorization models* when they are trained using the *bayesian approach*.

#### 3.3.1 Stronger Baselines

As a second finding, the *RMSE* values of the created models can be taken from *figure 5b*. Several points can be addressed. Firstly, it can be seen that the *individual inclusion* of *implicit knowledge* such as *time* or *user behaviour* leads to a significant *improvement* in the *RMSE*. For example, models like *Bayesian timeSVD* (0.7587) and *Bayesian SVD++* (0.7563), which already use *single implicit knowledge*, beat the *simple Bayesian MF* with an *RMSE* of 0.7633. In addition, it also shows that the *combination* of *implicit data* further improves the *RMSE*. *Bayesian timeSVD++* achieves an *RMSE* of 0.7523. Finally, *Bayesian timeSVD++ flipped* can achieve an *RMSE* of 0.7485 by adding *more implicit data*. This results in the third and most significant observation of the experiment. Firstly, the *simple Bayesian MF* with an *RMSE* of 0.7633 already beat the best method *MRMA* with an *RMSE* of 0.7634. Furthermore, the best method *MRMA* could be surpassed with *bayesian timeSVD++* by 0.0149 with respect to the *RMSE*. Such a result is astonishing, as it took *one year* during the *Netflix-Prize* to reduce the leading *RMSE* from 0.8712 (*progress award 2007*) to 0.8616 (*progress award 2008*). Additionally, this result is remarkable as it *challenges* the *last five years* of research on the *MovieLens10M-dataset*. Based on the results obtained, the *authors* see the first problem with the *results* achieved on the *MovieLens10M-dataset* as being that they were *compared against too weak baselines*. From *figure 6* the *improved baselines* and the *results of the new methods* can be examined.

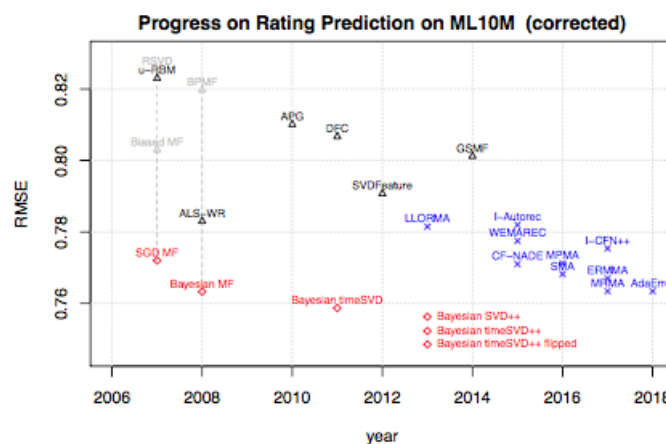


Figure 6: Improved baselines and new methods

#### 3.3.2 Reproducibility

But where do these *weak baselines* come from? In response, the authors see two main points. The first is *reproducibility*. This is generally understood to mean the *repetition* of an *experiment* with the aim of *obtaining* the *specified results*. In most cases, the code of the authors of a paper is taken and checked. Not only during the *Netflix-Prize*, this was a common method to compare competing methods, improve one's own and generally achieve *stronger baselines*. However, the authors do not consider the *simple repetition* of the experiment for the purpose of achieving the same results to be appropriate. Thus, the *repetition* of the experiment only provides information about the results achieved by a specific setup. However, it does not provide deeper insights into the method, nor into its general quality. This is not only a problem of *recommender systems* but rather a general problem in the field of *machine learning*. Thus, *indicators* such as *statistical significance*, *reproducibility* or *hyperparameter search* are often regarded as *proof* of

the quality of an experiment. But they only give information about a certain experiment, which could be performed with *non-standard protocols*. The question of whether the method being used is applied and configured in a meaningful way is neglected. Thus, *statistical significance* is often taken as an *indication* that *method A performs better than method B*.

### 3.3.3 Inadequate validations

The authors do not doubt the relevance of such methods. They even consider them *necessary* but *not meaningful enough* for the *general goodness* of an *experiment*. Thus, their preparation, which takes up the above mentioned methods shows, that they can achieve meaningful results. Therefore the authors see the second point of criticism of the results obtained on the *MovieLens10M-dataset* as the *wrong understanding* of *reliable experiments*. The *main reason* given is the *difference* between *scientific* and *industrial work*. For example, during the *Netflix-Prize*, which represents *industrial work*, *audible sums* were awarded for the best results. This had several consequences. Firstly, a *larger community* was addressed to work on the solution of the *recommender problem*. On the other hand, the high number of *competitors* and the *simplicity* in the formulation of the task encouraged each participant to investigate the *simplest methods* in *small steps*. The *small-step approach* was also driven by the *standardized guidelines* for the *evaluation* of the methods given in *section 2.4* and by the *public competition*. Thus, a better understanding of the *basic relationships* could be achieved through the *miniscule evaluation* of hundreds of models. All in all, these insights led to *well-understood* and *sharp baselines* within a *community* that *continuously* worked towards a *common goal* over a total of *three years*. Such a *motivation* and such a *target-oriented competitive idea* is mostly not available in the *scientific field*. Thus, publications that achieve *better results* with *old methods* are considered *unpublishable*. Instead, experiments are *not questioned* and their *results* are *simply transferred*. In some cases experiments are *repeated exactly as specified* in the instructions. Achieving the *same result* is considered a *valid baseline*. According to the authors, such an approach is *not meaningful* and, by not questioning the *one-off evaluations*, leads to *one-hit-wonders* that *distort* the *sharpness* of the *baselines*. Therefore, the *MovieLens10M-dataset* shows that the main results of the last *five years* were *measured* against *too weak baselines*.

## 4 Conclusion

Overall, [Rendle et al. \(2019\)](#) concludes that the last *five years* of *research* for the *MovieLens10M-dataset* have not really produced any new findings. Although in the presented experiment the *best practice* of the *community* was applied, the *simplest matrix-factorization* methods could clearly beat the reported results. Thus, the authors support the thesis that *finding and evaluating valid and sharp baselines* is *not trivial*. *Empirical data* are collected, since there is *no formal evidence* in the field of *recommender systems* to make the methods comparable. From the *numerical evaluation* the authors identify the *rating of a work* in a *scientific context* as a *major problem*. Here, a *publication* is classified as *not worth publishing* if it achieves *better results with old methods*. Rather, most papers aim to *distinguish themselves* from the others by using new methods that beat the old ones. In this way, *baselines* are *not questioned* and the *community* is steered in the wrong direction, as their work competes against *insufficient baselines*.

This problem was not only solved during the *Netflix-Prize* by the *horrendous prize money*. However, it turns out that the *insights* gained there were more *profound* and can be transferred to the *MovieLens10M-dataset*. Thus *new techniques* but *no new elementary knowledge* could be achieved on the *MovieLens10M-dataset*.



## 5 Critical Assessment

With this paper [Rendle et al. \(2019\)](#) addresses the highly experienced reader. The simple structure of the paper convinces by the clear and direct way in which the problem is identified. Additionally, the paper can be seen as an *addendum* to the *Netflix-Prize*.

The problem addressed by [Rendle et al. \(2019\)](#) is already known from other topics like *information-retrieval* and *machine learning*. For example, [Armstrong et al. \(2009\)](#) described the phenomenon in the context of *information-retrieval systems*, that too *weak baselines* are used. He also sees that *experiments* are *misinterpreted* by giving *misunderstood indicators* such as *statistical significance*. In addition, [Armstrong et al. \(2009\)](#) also sees that the *information-retrieval community* lacks an adequate overview of results. In this context, he proposes a collection of works that is reminiscent of the *Netflix-Leaderboard*. [Lin \(2019\)](#) also observed the problem of *baselines* for *neural networks* that are *too weak*. Likewise, the actual observation that *too weak baselines* exist due to empirical evaluation is not unknown in the field of *recommender systems*. [Ludewig \(2018\)](#) already observed the same problem for *session-based recommender systems*. Such systems only work with data generated during a *session* and try to predict the next *user selection*. They also managed to achieve better results using *session-based matrix-factorization*, which was inspired by the work of [Rendle et al. \(2012\)](#) and [Rendle et al. \(2010\)](#). The authors see the problem in the fact that there are *too many datasets* and *different measures* of evaluation for *scientific work*. In addition, [Dacrema et al. \(2019b\)](#) take up the problem addressed by [Lin \(2019\)](#) and shows that *neural approaches* to solving the *recommender-problem* can also be beaten by simplest methods. They see the main problem in the *reproducibility* of publications and suggest a *rethinking* in the *verification* of results in this field of work. Furthermore, they do not refrain from taking a closer look at *matrix-factorization* in this context. Compared to the listed work, it is not unknown that in some subject areas *baselines* are *too weak* and lead to *stagnant development*. Especially when considering that *information-retrieval* and *machine learning* are the *cornerstones* of *recommender systems* it is not surprising to observe similar phenomena. Nevertheless, the work published by [Rendle et al. \(2019\)](#) stands out from the others. Using the insights gained during the *Netflix-Prize*, he underlines the problem of the *lack of standards* and *unity* for *scientific experiments* in the work mentioned above.

However, the work published by [Rendle et al. \(2019\)](#) also clearly stands out from the above-mentioned work. In contrast to them, not only the problem for the *MovieLens10M-dataset* in combination with *matrix-factorization* is recognized. Rather, the problem is brought one level higher. Thus, it succeeds in gaining a global and reflected but still distanced view of the *best practice* in the field of *recommender systems*. Besides calling for *uniform standards*, [Rendle et al. \(2019\)](#) criticizes the way the *scientific community* thinks. [Rendle et al. \(2019\)](#) recognizes the *publication-bias* addressed by [Sterling \(1959\)](#). The so-called *publication-bias* describes the problem that there is a *statistical distortion* of the data situation within a *scientific topic area*, since only successful or modern papers are published. [Rendle et al. \(2019\)](#) clearly abstracts this problem from the presented experiment. The authors see the problem in the fact that a scientific paper is subject to a *pressure to perform* which is based on the *novelty* of such a paper. This thought can be transferred to the *file-drawer-problem* described by [Rosenthal \(1979\)](#). This describes the problem that many *scientists* do not publish their work and, out of concern about not meeting the *publication standards* such as *novelty* or the question of the *impact on the community*, do not submit their results at all and prefer to *keep them in a drawer*. Although the problems mentioned above are not directly addressed, they can be abstracted due to the detailed presentation. In contrast to the other works, this way a wanted or unwanted abstraction and naming of concrete and comprehensible problems is achieved.

Nevertheless, criticism must also be made of the work published by [Rendle et al. \(2019\)](#). Despite the high standard of the work, it must be said that the problems mentioned above can be identified but are not directly addressed by the authors. The work of [Rendle et al. \(2019\)](#) even lacks an embedding in the context above. Thus, the experienced reader who is familiar with the problems addressed by [Armstrong et al. \(2009\)](#), [Sterling \(1959\)](#) and [Rosenthal \(1979\)](#) becomes aware of the contextual and historical embedding and value of the work. In contrast, [Lin \(2019\)](#) and [Dacrema et al. \(2019b\)](#), published in the same period, succeed in this embedding in the contextual problem and in the previous work. Moreover, it is questionable whether the problem addressed can actually lead to a change in *long-established thinking*. Especially if one takes into account that many scientists are also investigating the *transferability* of new methods to the *recommender problem*. Thus, the call for research into *better baselines* must be viewed from two perspectives. On the



one hand, it must be noted that *too weak baselines* can lead to a false understanding of new methods. On the other hand, it must also be noted that this could merely trigger the numerical evaluation in a competitive process to find the best method, as it was the case with the *Netflix-Prize*. However, in the spirit of [Sculley et al. \(2018\)](#), it should always be remembered that: *"the goal of science is not wins, but knowledge"*.

As the authors [Rendle](#) and [Koren](#) were significantly *involved* in this competition, the points mentioned above are convincing by the experience they have gained. With their results they support the very simple but not trivial statement that finding good *baselines* requires an *immense effort* and this has to be *promoted* much more in a *scientific context*. This implies a change in the *long-established thinking* about the evaluation of scientific work. At this point it is questionable whether it is possible to change existing thinking. This should be considered especially because the scientific sector, unlike the industrial sector, cannot provide financial motivation due to limited resources. On the other hand, it must be considered that the individual focus of a work must also be taken into account. Thus, it is *questionable* whether the *scientific sector* is able to create such a large unit with regard to a *common goal* as *Netflix* did during the competition. It should be clearly emphasized that it is immensely important to use sharp *baselines* as guidelines. However, in a *scientific context* the *goal* is not as *precisely defined* as it was in the *Netflix-Prize*. Rather, a large part of the work is aimed at investigating whether new methods such as *neural networks* etc. are applicable to the *recommender problem*. Regarding the results, however, it has to be said that they clearly support a *rethinking* even if this should only concern a *small part* of the work.

On the website *Papers with Code*<sup>1</sup> the *public leaderboard* regarding the results obtained on the *MovieLens10M-dataset* can be viewed. The source analysis of *Papers with Code* also identifies the results given by [Rendle et al. \(2019\)](#) as leading. In addition, *future work* should be focused on a more *in-depth source analysis* which, besides the importance of the *MovieLens10M-dataset* for the *scientific community*, also examines whether and to what extent *other datasets* are affected by this phenomenon. Due to the recent publication in spring 2019, this paper has not yet been cited frequently. So time will tell, what impact it will have on the *community*. Nevertheless, [Dacrema et al. \(2019a\)](#) was able to base his own work on this article and expand it. According to this, [Rendle](#) seems to have recognized an elementary and unseen problem and made it public.

This is strongly reminiscent of the so-called *Artificial-Intelligence-Winter (AI-Winter)* in which *stagnation* in the *development* of *artificial intelligence* occurred due to too high expectations and other favourable factors. Overall the paper has the potential to *counteract* the *stagnation* in development and thus *prevent* a *winter* for *recommender systems*.

---

<sup>1</sup><https://paperswithcode.com/sota/collaborative-filtering-on-movielens-10m>

## References

- Timothy Armstrong, Alistair Moffat, William Webber, and Justin Zobel. Improvements that don't add up: Ad-hoc retrieval results since. pages 601–610, 11 2009. doi: 10.1145/1645953.1646031.
- Maurizio Ferrari Dacrema, Simone Boglio, Paolo Cremonesi, and Dietmar Jannach. A troubling analysis of reproducibility and progress in recommender systems research. *ArXiv*, abs/1911.07698, 2019a.
- Maurizio Ferrari Dacrema, Cremonesi Paolo, and Jannach Dietmar. Are we really making much progress? A worrying analysis of recent neural recommendation approaches. *CoRR*, abs/1907.06902, 2019b. URL <http://arxiv.org/abs/1907.06902>.
- Christian Desrosiers and George Karypis. A comprehensive survey of neighborhood-based recommendation methods. In P.B. Kantor, F. Ricci, L. Rokach, and B. Shapira, editors, *Recommender Systems Handbook*, pages 107–144. Springer, 01 2011. doi: 10.1007/978-0-387-85820-3\_4.
- Simon Funk. Netflix update: Try this at home. <https://sifter.org/~simon/journal/20061211.html>, 12 2006. Accessed: 2019-12-12.
- F. Maxwell Harper and Joseph A. Konstan. The movielens datasets: History and context. *ACM Trans. Interact. Intell. Syst.*, 5(4):19:1–19:19, December 2015. ISSN 2160-6455. doi: 10.1145/2827872. URL <http://doi.acm.org/10.1145/2827872>.
- Jussi Karlgren. *An algebra for recommendations : Using reader data as a basis for measuring document proximity*. Number 179 in SYSLAB technical reports. Department of Computer and Systems Sciences, Stockholm University, 1990.
- Yehuda Koren. Biography of yehuda koren. <https://ieeexplore.ieee.org/author/37414256700>. Accessed: 2019-12-21.
- Yehuda Koren. Factorization meets the neighborhood: A multifaceted collaborative filtering model. pages 426–434, 08 2008. doi: 10.1145/1401890.1401944.
- Yehuda Koren. The bellkor solution to the netflix grand prize. 09 2009.
- Yehuda Koren and Robert Bell. Advances in collaborative filtering. In P.B. Kantor, F. Ricci, L. Rokach, and B. Shapira, editors, *Recommender Systems Handbook*, pages 145–186. Springer, 01 2011. doi: 10.1007/978-0-387-85820-3\_4.
- Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42:30–37, 08 2009.
- Miklós Kurucz, András Benczúr, and Károly Csalogány. Methods for large scale svd with missing values. *ACM KDDCup 2007*, 01 2007.
- Jimmy Lin. The neural hype and comparisons against weak baselines. *SIGIR Forum*, 52(2):40–51, January 2019. ISSN 0163-5840. doi: 10.1145/3308774.3308781. URL <https://doi.org/10.1145/3308774.3308781>.
- Pasquale Lops, Marco de Gemmis, and Giovanni Semeraro. Content-based recommender systems: State of the art and trends. In P.B. Kantor, F. Ricci, L. Rokach, and B. Shapira, editors, *Recommender Systems Handbook*, pages 74–105. Springer, 01 2011. doi: 10.1007/978-0-387-85820-3\_4.
- Jannach Ludewig. Evaluation of session-based recommendation algorithms. *CoRR*, abs/1803.09587, 2018. URL <http://arxiv.org/abs/1803.09587>.
- Arkadiusz Paterek. Improving regularized singular value decomposition for collaborative filtering. *Proceedings of KDD Cup and Workshop*, 01 2007.
- Steffen Rendle. Papers of steffen rendle. <https://dblp.org/pers/hd/r/Rendle:Steffen>. Accessed: 2020-01-20.

- Steffen Rendle. Scaling factorization machines to relational data. volume 6, pages 337–348, 03 2013. doi: 10.14778/2535573.2488340.
- Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, page 811–820, New York, NY, USA, 2010. Association for Computing Machinery. ISBN 9781605587998. doi: 10.1145/1772690.1772773. URL <https://doi.org/10.1145/1772690.1772773>.
- Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence, UAI 2009*, 05 2012.
- Steffen Rendle, Li Zhang, and Yehuda Koren. On the difficulty of evaluating baselines: A study on recommender systems. *CoRR*, abs/1905.01395, 2019. URL <http://arxiv.org/abs/1905.01395>.
- Robert S. Rosenthal. The file drawer problem and tolerance for null results. 1979.
- Ruslan Salakhutdinov and Andriy Mnih. Bayesian probabilistic matrix factorization using markov chain monte carlo. volume 25, pages 880–887, 01 2008. doi: 10.1145/1390156.1390267.
- D. Sculley, Jasper Snoek, Alexander B. Wiltschko, and Ali Rahimi. Winner’s curse? on pace, progress, and empirical rigor. In *ICLR*, 2018.
- Theodore D. Sterling. Publication decisions and their possible effects on inferences drawn from tests of significance—or vice versa. *Journal of the American Statistical Association*, 54(285):30–34, 1959. ISSN 01621459. URL <http://www.jstor.org/stable/2282137>.
- Yunhong Zhou, Dennis Wilkinson, Robert Schreiber, and Rong Pan. Large-scale parallel collaborative filtering for the netflix prize. pages 337–348, 06 2008. doi: 10.1007/978-3-540-68880-8\_32.