

A reproducibility study on: 'Should I Follow the Crowd? A Probabilistic Analysis of the Effectiveness of Popularity in Recommender Systems' by R. Cañamares & P. Castells

January 30, 2022

Hirzberger Markus
Vienna University of Technology
Vienna, Austria
markus.hirzberger@gmail.com

Milicevic Valentin
Vienna University of Technology
Vienna, Austria
valentin.milicevic@gmail.com

Bogenreiter Dario
Vienna University of Technology
Vienna, Austria
dariobogenreiter@rocketmail.com

ABSTRACT

This paper reviews the experimental setup, experiments, and results of a study [3] by Cañamares and Castells. The original paper focuses on the overarching question of whether popularity is an unwanted bias or a valuable signal in the context of recommender systems. The findings were largely confirmed, but shortcomings in documentation and reproducibility were identified.

KEYWORDS

recommender systems; popularity; evaluation; bias; accuracy; non-random missing data; collaborative filtering

ACM Reference Format:

Hirzberger Markus, Milicevic Valentin, and Bogenreiter Dario. 2022. A reproducibility study on: 'Should I Follow the Crowd? A Probabilistic Analysis of the Effectiveness of Popularity in Recommender Systems' by R. Cañamares & P. Castells January 30, 2022 . In *Proceedings of* . ACM, New

York, NY, USA, 4 pages.

1 SUMMARY OF THE PAPER

This section provides a brief overview of the research problem, related aspects, and the results obtained by the authors.

1.1 The Problem faced

Both modern recommender algorithms as well as information retrieval (IR) methods and metrics (which have recently been widely utilized to evaluate recommender systems (RecSys)) exhibit a strong bias towards popular articles. This could lead to a distortion in the search for the best recommender algorithms, as even a simple ranking by popularity (without any intelligence) often leads to fairly good results that are not to be ignored.

The research problem described above was often solved in the past by trying to measure the bias first and then removing it. But this procedure was never well enough justified, as algorithms could also rightfully recommend popular items if good results are achieved that way. Therefore, this paper raises the question of whether popularity is the fact a valuable signal or not.

This question is not simple to answer because popularity has its advantages and downsides. A frequently mentioned criticism of popularity in the literature is the lack of newness in the recommendation. However, this is often countered by the positive effect of some degree of popularity on pure accuracy.

A RecSys that is strongly driven by popularity has the advantage that it is easy and inexpensive to develop. This system yields good results especially with new users, about whom no data has yet been gathered, or who do not have the time to form their own opinions. Other RecSys, on the other hand, have the advantage that they focus more on personalized user satisfaction and are thus less susceptible to trends that may not have anything to do with the actual tastes of the user.

While these basic considerations are already known, it is still unexplored to what extent and in what circumstances popularity is a good ingredient for a Recsys and if this is measured in a suitable way and manner.

To explore this question, the researchers have employed both theoretical and empirical methods.

1.2 Data sets

In order to perform the experiments with realistic data, the researchers have chosen the following two data sets:

MovieLens

This dataset [2] is quite popular among data scientists and often used in research, especially in connection with recommender systems. It contains various data about movies (e.g. their genres, titles, and publication dates), users (e.g. their gender, age, profession), and the ratings that these users have given on the movie-ranking website 'MovieLens'. The dataset is available in several sizes - currently, you can choose between the sizes 1m, 100k, 50k, 20k, or 10k. The researchers have chosen the largest option, which consists of over 1 million ratings. This has the advantage that the high sample size minimizes the chance of randomness in the experiments.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](https://www.acm.org/permissions).

CM100k

The second dataset [1] which has 100k data records can be considered as a comparatively small dataset that was crowdfunded for the paper. The creation of the data happened in incorporation with the music streaming platform 'deezer.com'. For this, 1000 music songs and 1000 users (which were checked for authenticity) were randomly selected as samples via API access. And then 100 tracks were recommended to users. Afterward, the users were asked if they knew the song and whether they liked it - which generated a total of 100k records. Through this approach, the researchers intended to eliminate the discovery bias that is often found on standard datasets (such as the MovieLens dataset) and thus increase the overall data quality.

1.3 Experimental Setup

The paper under investigation conducts several experiments for recommendation tasks, where the goal is to recommend items to a set of users. These items are then given an observed rating value - which then exists for a subset of users and items.

Using these ratings as input, various recommender algorithms are then trained which should provide the most suitable recommendation. To evaluate the results of most experiments done, the data is shuffled and then split into a train and test set with a ratio of p to $p-1$ (where p is a number higher than 0 and lower than zero). Only in experiments with true metric values, the full data is taken as training data set (with no split) - since these metrics assume unbiased and/or full knowledge.

For the quantitative evaluation of the performance of the algorithms, different accuracy metrics like precision, recall, nDCG, and MAP are applied. A recommended item is considered appropriate for a user if there is a positive rating for it in the test dataset. However, it is important to note that the accuracy measured in this way often deviates from the actual one, since user preferences are often not known. This difference can be measured in common experiments - which is also done in this study. Through these measurements, the researchers obtain a more complete picture of the situation and are able to produce more results.

In the course of the experiments, the popularity of the different items is measured. This is defined as the number of people who rated an item positively or negatively (i.e. interacted with it). However, the authors decide to consider only the positive interaction in their experiments to determine the popularity of an item.

In addition to popularity, the average rating of the items is also measured. The researchers measure this as the proportion of users who liked an item. They justify this decision with the fact that this is better suited for probabilistic analysis.

1.4 Findings

One of the central findings of the paper is that the value of popularity as a signal depends mainly on three circumstances: item relevance, item discovery, and inaction decision made by the user. In addition, it could be found that a recommendation based on popularity can in some cases be completely pointless. Even a situation could occur in which such a recommendation performs worse than a random one. On the other hand, it has been shown that decisions

based on popularity can also yield good results in situations where discovery strongly or weakly relies on relevance.

Another finding from the paper is that the average rating may be a more reasonable signal than the number of ratings. This result is of particular importance since the literature has so far assumed the opposite.

Furthermore, it was found that accuracy can be a misleading metric, and the observed accuracy often does not correspond to the actual true accuracy. This occurs especially if item discovery is widely disconnected from user taste.

2 REPRODUCING WORKFLOW

Capturing provenance in scientific experiments has been a major concern both for comprehension and reproducibility. As a solution, scientists are required to standardize their components, procedures, and workflows. It is recommended that scientists document the complete system set-up across the entire provenance chain.

2.1 Source Code

There was no reference to the source code in this scientific paper, fortunately, we were able to find the source code on the GitHub platform. We, therefore, encourage authors to include in their papers links to the source code for the smooth reproduction of their scientific papers.

2.2 Brief Software Description

The code contains two main modules:

- Module 1: Monte Carlo computation of the integral described in section 5.3 of the paper, producing the results displayed in Figure 3.
- Module 2: Computation of the metrics $P@1$ and $nDCG@10$ (true and observed versions) as reported in sections 6.2 and 6.3 of the paper. For section 6.2, randomized versions of a crowdsourced dataset (CM100k) are generated recreating different independence assumptions, on which non-personalized recommenders are compared: random, popularity, average rating, and the optimal rankings, producing the results displayed in Figure 5 (along with basic results for MovieLens 1M). For section 6.3, normalized and non-normalized kNN variants are run on the MovieLens 1M and CM100k datasets.

2.3 Data

The GitHub repository includes a copy of datasets used in the paper, that are needed for the reproduction of the experiments. There are two datasets included were, the CM100k dataset which was released as part of the contribution of the paper, and the MovieLens 1M dataset.

2.4 Workflow

After finding the GitHub repository which provides the source code and data needed to reproduce the experiments, we were able to rerun the code almost effortlessly. In the first step, we have cloned the GitHub repository of the source code. Secondly, we have imported the cloned project in the IntelliJ IDEA application

January 30, 2022

as a Maven project. IntelliJ IDEA application has automatically downloaded all required dependencies to run the project or source code. After that, we have run the code and got results.txt as an output result file. Finally, we have copied text from the results.txt file to the MS Excel file figures.xlsx which was provided in the GitHub repository to generate similar graphs to the ones displayed in the paper.

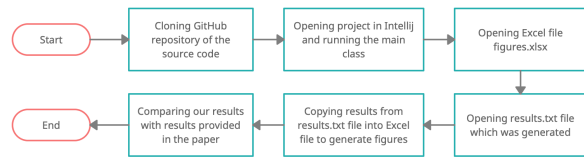


Figure 1: Workflow for reproducing the experiments

3 PRIMAD ANALYSIS

In this section, we will analyze the work of Cañamares and Castells using the PRIMAD model which was introduced by Freire, Fuhr, and Rauber [4]. They introduced this framework to summarize the aspects of an experiment that can be changed or kept the same while trying to reproduce it. We will focus on varying the parameters of the platform as well as the data. Nevertheless, we will discuss the other dimensions and briefly subsume the status of their work in this dimension as well as potential gains of improving along those. It was not within the scope of our work to change all the possible parameters, so we focused on those that we expected to yield the most important information to validate their outcomes. Therefore we were only able to test the potential of generalization to a limited extent.

Platform

As mentioned above, the code of the experiment was available on GitHub. The README file included in this GitHub repository had a clear description of how to run the code and also specifications of the platform the experiment was conducted on. The code was run on Java JDK version 1.8.0_181 and with maven version 3.6.0 Cañamares and Castells did not provide a specification of the hardware nor the operating system they used. Because no virtual machine or docker container was provided, we reran the experiments on three of our computers with the following differences:

- macOS 10.13.6, java version 10.0.2 and maven version 3.3.9
- macOS 12.1, java version 17.0.2 and maven version 3.8.1
- windows10 version 20H2, java version 17.0.2 and maven version 3.8.4

This enabled us to ensure portability is given to the extent we tested it. We kept the parameters the same and could not identify significant differences in the result. Of course, the time efficiency varied among those different settings. With a MacBook with macOS 10.13.6 and 8 GB RAM as well as 2 cores, the runtime was 7:54 minutes. With a MacBook with macOS 12.1 and 16 GB RAM as well as 8 cores, the runtime was 3:31 minutes. The variance of both settings was negligible.

Research Objective

In their paper, Cañamares and Castells wanted to answer the question of whether popularity is a valuable signal or not. However, their approach could be reused to answer other hypotheses as well. As this would exceed the scope of this work we do not propose concrete research objectives. Nevertheless, we want to highlight their implementation of measurement of the ranking algorithms as well as the MovieLens dataset, which has already proven to be reusable for different settings and research questions.

Implementation

As already mentioned their code is available on GitHub. A problem that might occur when working with a remote git repository is, that commits after the experiment might influence results. We went through all of the 20 commits and it turned out that after the initial commit of the code for the experiment only minor changes were made. Those changes included modifications of the output and the README file and can not influence the results. The implementation requires a specific package developed by Saúl Vargas Sandoval. This package was updated several times since the initial experiment was conducted and it is difficult to say if that changed the results. Moreover, we had to rewrite parts of the package, because it did not set the seed, nor did it provide the user with the possibility to do so. We will discuss the effect of that in greater detail below with the data dimension. All the other packages that were necessary to run the code were included with maven and the pom-file specified the versions. A potential future work would be to re-implement the experiment in python to enable higher adoption of the used approach. This should not be done for the sake of doing it in another programming language, but for the advantage of better accessibility for a broader audience. It might also be interesting to evaluate the potential time efficiency differences.

Method

The method was already briefly summarized above and we will not go deeper in this section. We could re-implement parts of the work to run it on the same setup with the same data, but we had no approaches for actually meaningful changes.

Actor

To show the independence of the experiment we could obtain new data from a new set of users. In their setup, they have used CrowdFlower to retrieve the user-generated data. Therefore, one could try to obtain the data from a different platform and test if there is a significant change, but we would not expect that and have therefore not done it, because the expected gain in knowledge, with the expected effort, would not pay off. We do not have other actors involved in the experiment that could affect the results.

Data

As already mentioned above the repository included a copy of the used data. Unfortunately, we were not able to verify if the dataset they created with the help of CrowdFlower was edited to better fit their hypothesis. The MovieLens dataset has already proven to be

a suitable dataset for experiments concerning the recommendation. When it comes to data we rather focused on the parameters and less on the input data. On the one hand, we tried to validate their results and on the other hand, we tried to show that the results can be generalized by changing seeds, cross-validation splits as well as arguments of the recommendation algorithms. In the process of validating the results, we encountered the issue, that they did not set the seed in the process of random number generation. Moreover, in the package developed by Sandoval, which was a critical part of their experiment, it was not even possible to set the seed. As mentioned above, we first re-wrote the relevant parts to enable the setting of the seeds which is a crucial aspect of comparing different approaches. In their work, they used a two-tailed two-sample t-test and so will we when we try to validate their results. We focused on specific parts of their results which showed the smallest still significant difference. We evaluated if those significant differences were still given when we change some of the parameters. As a significance level, we chose $p < 0.01$ as in the paper.

First, we set a seed of "123" and ran the experiment 5 times and performed several two-sample t-test ($df=4$). The results remained the same with respect to the ranked performance of the different recommendation algorithms and the differences were significant. Thus, we were able to validate their results.

Second, we tested a different seed ("357") and performed again several two-sample t-tests ($df=4$) to validate the results one more time. Again the results supported the conclusion in the exact same way as in the experiment. Moreover, exactly the same differences were significant as in the initial experiment.

Third, we rerun the experiment again with a 10-fold cross-validation instead of the 5-fold cross-validation that was initially conducted. We again performed a two-sample t-test ($df=4$) to validate the results. This run also resulted in the same rankings of the algorithms and the differences were again significant as they were in the initial experiment.

4 SUGGESTIONS FOR IMPROVEMENT

In our process of Reproducing the results, the first obstacle that could easily be removed was the fact that the source code on the GitHub repository was not mentioned in the paper. Therefore, we would recommend always including a link in the paper in case the code is available online.

After discussing the approach of Cañamares and Castells considering the aspects of the PRIMAD model, we have some suggestions on how they could enhance reproducibility even further.

Concerning the platform, we have two comments or suggestions respectively. First, to keep the platform as set up in the experiment to just vary the remaining parameters, a virtual machine or docker file would enable the reproducibility of the platform. Second, the package for recommender systems by Saúl Vargas Sandoval was updated several times and we were not able to find out which status of the package they used for their experiment. They could have briefly mentioned that in the README file on GitHub.

Probably our strongest suggestion concerns the setting of seed. Unfortunately, they did not set seed, nor did they provide the possibility to do that with one variable or parameter respectively. Thus, we

recommend always at least setting a seed to ensure reproducibility. The next step for improvement would be to allow ad-hoc parameters to set the seed in one place because we had to search the whole project to set the seed in 12 different places.

One last thing we would suggest relates to changing the implementation. As we discussed above, it might be interesting to reimplement the experiment in python to enable further adoption. Unfortunately, neither the classes nor the methods included in the projects were properly documented. Therefore, documentation of the code would be helpful to especially support the process of improving the implementation.

REFERENCES

- [1] [n. d.]. Crowdsourced 100k. <http://ir.ii.uam.es/cm100k>
- [2] [n. d.]. MovieLense Dataset. <https://grouplens.org/datasets/movielens/latest/>
- [3] Rocío Cañamares and Pablo Castells. 2018. Should I follow the crowd? A probabilistic analysis of the effectiveness of popularity in recommender systems. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 415–424.
- [4] Juliana Freire, Norbert Fuhr, and Andreas Rauber. 2016. Reproducibility of Data-Oriented Experiments in e-Science (Dagstuhl Seminar 16041). *Dagstuhl Reports* 6, 1 (2016), 108–159. <https://doi.org/10.4230/DagRep.6.1.108>