Time-dependent Evaluation of Recommender Systems

Teresa Scheidt [1] and Joeran Beel [2]

[1] *Lund University, Box 117, 221 00 Lund, Sweden*
[2] *Siegen University, Adolf-Reichwein-Straße 2, 57076 Siegen, Germany*

Abstract

Evaluation of recommender systems is an actively discussed topic in the recommender system community. However, some aspects of evaluation have received little to no attention, one of them being whether evaluating recommender system algorithms with single-number metrics is sufficient. When presenting results as a single number, the only possible assumption is a stable performance over time regardless of changes in the datasets, while it intuitively seems more likely that the performance changes over time. We suggest presenting results over time, making it possible to identify trends and changes in performance as the dataset grows and changes. In this paper, we conduct an analysis of 6 algorithms on 10 datasets over time to identify the need for a time-dependent evaluation. To enable this evaluation over time, we split the datasets based on the provided timesteps into smaller subsets. At every tested timepoint we use all available data up to this timepoint, simulating a growing dataset as encountered in the real-world. Our results show that for 90% of the datasets the performance changes over time and in 60% even the ranking of algorithms changes over time.

Keywords

Recommender Systems, Evaluation, Time-dependent Evaluation

1. Introduction

Recommender-system evaluation is an actively discussed topic in the community. Discussions include advantages and disadvantages of evaluation methods such as online evaluations, offline evaluations, and user studies [7, 11] or the ideal metrics to measure recommendation effectiveness [1].

An issue that has received little attention is the question if presenting results as a single number (e.g. recall@10 = 0.64) is sufficient.[2] Typically, researchers present results as single-number metrics, e.g. precision, normalized discounted cumulative gain (nDCG), recall, root mean square error (RMSE). These metrics are based on the whole dataset and hold no information if the algorithm performs the same over the whole time period or if it improved or worsened over time. Consequently, there is some uncertainty how the algorithms will perform in the future when the dataset changes/grows.

When presented with a single number, the only assumption is that the performance is the same over the whole time period the dataset was collected and will stay the same in the future (Figure 1a). It could however be different in reality, e.g. the performance of one algorithm could increase steadily while another algorithms performance decreases (Figure 1b,c). This could lead to 'crossing' of the performance lines, which changes the conclusion of which algorithm performs better. This is especially problematic if the crossing-point is in the future (Figure 1c): the best performing algorithm at the time of evaluation (Alg A) likely won't be the best performing algorithm in the future. Another possibility is that the performance fluctuates over time and two algorithms cross multiple times (Figure 1d). This means the conclusion of which algorithm is better depends strongly on the time point the algorithms are tested.

Perspectives on the Evaluation of Recommender Systems Workshop (PERSPECTIVES 2021), September 25th, 2021, co-located with the 15th ACM Conference on Recommender Systems, Amsterdam, The Netherlands
EMAIL: teresascheidt@gmail.com (A. 1); joeran.beel@uni-siegen.de (A. 2)
ORCID: 0000-0002-4537-5573 (A. 2)

[2] This issue was mentioned by Beel in a research proposal [4], which this work is based on.

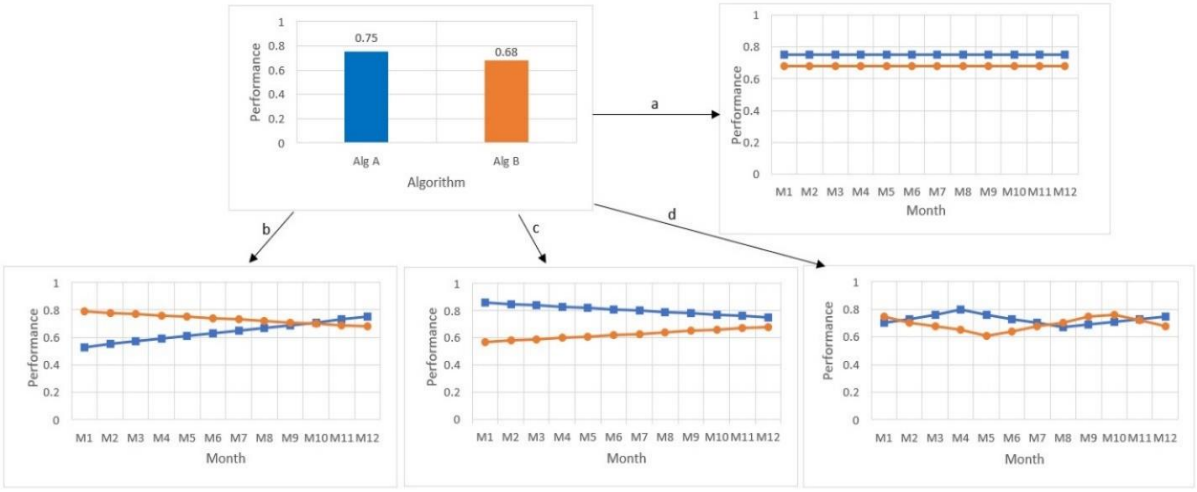


Figure 1: When presented with a single number metric, it is not obvious how different algorithms perform over time. There are several possibilities how the performance changes over time, that might change the conclusion of which algorithm is better: (a) the performance is constant over time (Alg A is better) (b) the performance changes linearly, the crossing point of two algorithms is in the past (Alg A is better) (c) the performance changes linearly, the crossing point of two algorithms is in the future (Alg B is better in the future) (d) the performance fluctuates, which algorithm is better changes often over time.

Sometimes, researchers do report metrics over time intervals. For instance, He et al. [13] report changing performance of algorithms over a 24h interval and Feely et al. [10] evaluate predictions over several weeks. Beel et al. [5] report Click-Through rate on a monthly basis over two years and try to select the best algorithm, among others, based on time [3]. Lathia [14] demonstrates how the performance of algorithms changes over time and shows that the performance can be improved when using different algorithms depending on the timepoint. Barreau and Carlier [2] argued for their new algorithm by showing that their proposed algorithm is more stable over time while others constantly decrease in performance. However, the majority of researchers report single-number metrics. This is evidenced by a small ad-hoc analysis that we conducted for our current paper. We analyzed all full and short papers of the ACM RecSys 2020 conference (n=67). Of those 67 papers, 55 evaluated algorithms, and of these 55, 89% presented single-number metrics, and only 11% presented metrics over time.

While researchers sporadically present metrics over time, there is no comprehensive analysis of how recommender-system evaluation metrics change over time. We found only two partly related studies [14, 16] that, among other things, studied the evolution of performance over time on one or two datasets respectively. They reported some interesting results, however, by studying just one or two datasets, general conclusions on the necessity of evaluation over time can barely be drawn.

We hypothesize that, instead of a single number, recommender system research would benefit from presenting metrics over time, i.e. each metric should be calculated multiple times at different time points, e.g.every week, month or year. This will allow to gain more information about an algorithm's effectiveness over time, identify trends and help choose the best algorithm. With this paper, we systematically evaluate how performance changes over time over several datasets and examine to what extent the community would benefit from evaluation over time. To the best of our knowledge, this is the first study that explicitly focuses on metrics over time, and the first research paper to present such results on a relatively large number of datasets.

2. Methodology

2.1. Algorithms, Metrics and Datasets

Overall, we test six algorithms on four datasets in a total of ten variations, that we split based on the timestamp. We observe the performance at every time-step and evaluate how the performance changes over time.

To identify the effects of time on the performance and differences of common recommendation algorithms, we compare three model-based and three memory-based algorithms. The algorithms used in this paper are from the Lenskit library [9], we chose funkSVD, biasedMF, Bias, UserKNN, ItemKNN, and Most Popular. We evaluate the performance with nDCG, recall and RMSE.

We chose four of the most common datasets [6, 17] in recommender system research, i.e. MovieLens [15], Netflix [8], Amazon[3] and Yelp[4] in their different variations (MovieLens 100k, MovieLens 1M, MovieLens 10M, Amazon books, Amazon Instant Video, Amazon Toys and Games, Amazon Music, Amazon Electronics, Yelp, Netflix) totaling in 10 datasets (Table 1). The choice of datasets for our research question is limited, as the datasets need to have timestamps included so that the data can be spit and evaluated based on time. At every timestep we filter the dataset to only include users with more than 2 ratings to make sure predictions are possible and meaningful.

The code for our evaluation over time can be found here.

Table 1
Overview of used datasets.

Dataset	Number of Ratings	Timespan	Split (# of subsets)
MovieLens 100k	100 thousand	1995-1998	Monthly (8)
MovieLens 1M	1 million	2000-2003	Yearly (4)
MovieLens 10M	10 million	1996-2009	Yearly (14)
Netflix Prize[1]	51 million	1998-2005	Yearly (7)
Amazon Books	14.8 million	1997-2013	Yearly (18)
Amazon Instant Video	135 thousand	2007-2014	Yearly (8)
Amazon Toys and Games	878 thousand	2000-2014	Yearly (14)
Amazon Music	579 thousand	1998-2014	Yearly (17)
Amazon Electronics	3.6 million	2000-2013	Yearly (14)
Yelp	192 thousand	2005-2013	Yearly (8)

[1] only comined_data_1 and combined_data_2 is used

2.2. Evaluation over time

To evaluate how the performance changes over time when more and more data becomes available, the datasets have to be split based on the provided timestamp. We split the datasets on a monthly or yearly basis (see Table 1), leading to 4-18 subsets per dataset. In general, there are two ways to define the subsets, each subset could include only the data of that month or year, or each subset could contain all data up to a certain timepoint. We define the subsets according to the second option, so with advancing time the dataset grows, and the last subset consists of the whole dataset. This splitting is closer to the 'real world', as practitioners probably would rather use all the available data, than only the last month of the available data. The splitting process we implemented is visualized in Figure 2.

[3] Downloaded from Amazon review data (ucsd.edu) [12]
[4] Downloaded from Yelp Dataset

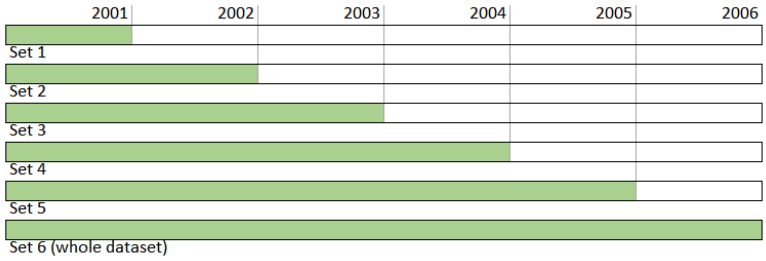


Figure 2: Time-based split of datasets. At every timepoint, all data up to that timepoint is included in the set.

At every timestep, we set aside a test-set consisting of the last 20% of the ratings from each user for evaluation. Each algorithm is then optimized using grid-search with 5-fold cross-validation for 5 iterations[5] on the remaining 80% of the subset. The algorithms are optimized two times, once w.r.t. nDCG and once w.r.t. RMSE. The optimized algorithms are then applied to the subset and evaluated on the corresponding test-set. We do not use any information from previous timesteps for the optimization or model-fitting process and the models are optimized and retrained from scratch at every timestep.

We exclude subsets with less than 500 ratings, as too few ratings lead to very different (and worse) performance or even lead to algorithms not working (e.g. UserKNN can't find enough neighbors). If the first subset has less than 500 ratings, we evaluate the next bigger subset as the first subset.

3. Results

Our analysis shows that the performance of algorithms often changes over time (for 90% of the datasets). For instance, on the MovieLens 10M dataset (Figure 3a), algorithms achieve an RMSE between 0.81 (SVD) and 0.85 (Bias) in the first year. The performance then decreases steadily for five years to RMSE between 0.85 and 0.91 before it increases again to an RMSE between 0.80 and 0.86 in the latest year.

We found no evidence for algorithm following different trends over time, i.e. some algorithms improving while others decrease over time. In most cases (90%) the performance develops roughly the same over time (Figure 4). For instance, nDCG on the Amazon-toys dataset worsens over time for all algorithms (Figure 3e). While all algorithms reach an nDCG in the range of 0.02 to 0.09 in the first year, the nDCG decreases to 0.001-0.01 in 2014. For the Netflix dataset (Figure 3c) the RMSE improves over time for all algorithms from 1.09-1.24 in 1998 to 0.86-0.93 in 2005.

Even though we did not observe algorithms following different trends over time, the ranking of algorithms does change over time. Especially in the beginning of data collection, the ranking of algorithms changes frequently (for 60% of the datasets). At later time-steps (and consequently more data) the ranking however remains more stable. A change of rank can for instance be seen on the Amazon-toys dataset (Figure 3e), where at the first time-step '*Most Popular*' is the best algorithm, measured by nDCG, followed by the '*Bias*' algorithm while at the second time-step '*ItemKNN'* performs best, and '*Bias*' is the second worse performing algorithm. The ranking keeps changing until the 10th year and then stays the same for the last 5 years. A similar behavior can be observed for the Netflix dataset (Figure 3c), where the ranking of algorithm changes in the first 3 time-steps and afterwards stays the same until the end. How often algorithms crossed lines, i.e. the ranking of algorithms changed, can be seen in Table 2.

[5] This relatively short grid-search does not guarantee to find the global optimum, for our purposes it is however sufficient to find a rough optimum, as we just want to compare the evolution over time for different algorithms rather than compare perfectly optimized algorithms.

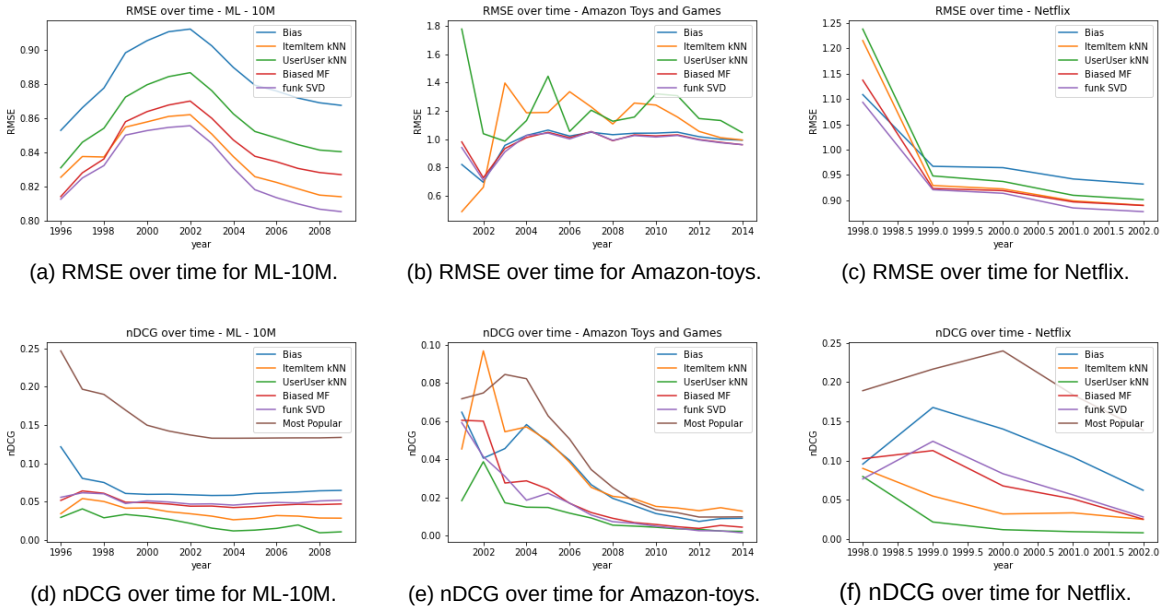
Table 2

Overview of the evolution of the algorithms over time on the datasets MovieLens (ML), Netflix, Amazon (A), Yelp, and their variations. Recall is omitted due to space restrictions, the results are very similar to nDCG. If several trends were observed over time (mixed) the latest trend is named in parentheses. The trend and range are given for the algorithm that performed best on the whole dataset (i.e. the last subset).

Dataset	Changed ranking of algorithms (changed best algorithm)		Trend of best performing algorithm		Range of best performing algorithm (difference in %)	
	nDCG	RMSE	nDCG	RMSE	nDCG	RMSE
ML 100k	1 (N)	2 (Y)	Stable	Mixed (Stable)	0.174-0.21 (17%)	0.975-1.09 (6%)
ML 1M	0 (N)	0 (N)	Stable	Stable	0.138-0.143 (3%)	0.87-0.88 (0.5%)
ML 10M	1 (N)	1 (N)	Stable	Mixed (Improving)	0.132-0.246 (46%)	0.80-0.85 (5%)
Netflix	2 (N)	2 (N)	Mixed (Decrease)	Improving	0.138-0.239 (34%)	0.87-1.09 (20%)
A - Books	5 (Y)	2 (N)	Stable	Improving	0.01-0.016 (37%)	0.98-1.07 (8%)
A - Video	5 (N)	>5 (Y)	Decrease	Mixed (Stable)	0.102-0.284 (64%)	0.95-1.09 (13%)
A - Toys	>5 (Y)	>5 (Y)	Decrease	Decrease	0.012-0.096 (87%)	0.71-1.05 (32%)
A - Music	>5 (N)	>5 (Y)	Decrease	Mixed (Increase)	0.004-0.06 (93%)	0.85-1.26 (32%)
A – Elec.	>5 (N)	>5 (Y)	Decrease	Decrease	0.012-0.09 (86%)	0.77-1.2 (35%)
Yelp	3 (N)	4 (Y)	Decrease	Decrease	0.044-0.09 (53%)	0.95-1.2 (20%)

It should be noted that the results vary based on the metric. While we observed similar evolutions over time for recall und nDCG, the results for RMSE differed. This can be seen in Table 2, where the observed trends differ in 40% depending on which metric is used. When looking at the MovieLens 10M dataset (Figure 3a,d), for example, this becomes evident. While the performance reaches a stable state after a few timesteps when looking at nDCG, the performance measured by RMSE first decreases until 2001 and then starts increasing. Additionally, the observed range of the metrics is different for nDCG/recall compared to RMSE. For nDCG the results sometimes differ up 90% over time (e.g. Amazon-music), while for RMSE the largest observed range is 35% (Amazon-electronics). For all datasets, the range for nDCG is bigger than for RMSE (see Table 2).

We observed distinct differences between datasets, especially between the Amazon and the MovieLens datasets. The MovieLens datasets show a more stable behavior over time, with few changes in ranking of algorithms and small ranges of nDCG and RMSE (less than 10% for all three variations). The Amazon datasets on the other hand have many changes in rankings and a higher decrease over time. A factor contributing to these differences might be the pruning of the datasets, the MovieLens datasets include only users with 20 or more ratings while all other datasets include users with 2 or more ratings. The bigger factor however appears to be the size of the datasets. Generally, the bigger datasets seem to behave more stable over time, which can be seen for Netflix (Figure 3c) or Movie Lens-10M (Figure 3d) for example, where the performance and the ranking of algorithms stays stable after the first two timesteps. Similarly, Amazon books, the biggest set within the Amazon database, has the smallest range of nDCG and RMSE values over time compared to the other Amazon datasets. The smaller datasets have more variation in performance especially in the early subset but also exhibit more stable behavior towards the end with more data (e.g. Amazon Toys and Games in Figure 3b,e), which might also be explained by the dataset-size.



(a) RMSE over time for ML-10M. (b) RMSE over time for Amazon-toys. (c) RMSE over time for Netflix.

(d) nDCG over time for ML-10M. (e) nDCG over time for Amazon-toys. (f) nDCG over time for Netflix.

Figure 3: Examples of the development of the algorithms' performance over time for three datasets: MovieLens 10M (a,d), Amazon Toys and Games (b,e) and Neflix (c,f). Plots over time for all datasets can be seen in the Appendix.

4. Discussion & Outlook

We hope our work initiates a discussion if presenting results of recommender-systems evaluations as single numbers should be changed. Given our analysis, we suggest presenting the performance of algorithms over time. In many cases the performance as well as the ranking of algorithms changes over time, making conclusions time dependent. Our results show that the performance of algorithms change in 90% of the datasets and the ranking of algorithms changes in 60% over time. Especially in the beginning of the data-collection phase the ranking of algorithm changes a lot, which should be considered when evaluating algorithms. For larger datasets, the performance of algorithms still changes over time sometimes, but the ranking is relatively stable. In those cases, single-number metrics might be sufficient to present the results. Nonetheless, the evaluation over time reveals trends and holds more information than a single-number metric. Consequently, for the development of new algorithms, it could be useful to evaluate them over time, to see for example if they follow different trends or behave more stable than the benchmark algorithms.

In the future, it should be further investigated what factors influence the performance over time. While we found that the development over time varies for different datasets and metrics, it remains unclear from our analysis which factors have the biggest influence on the performance. Factors that should be investigated include the data set size, number of users and items over time, data pruning, the number of ratings per user and the impact of the 'cold-start problem'. With deeper understanding what influences the performance over time, new and better algorithms can be developed that consider these changes over time and adapt to them, and more informed decisions can be made about which algorithms to use.

5. References

[1] Aggarwal, C.C. 2016. Evaluating Recommender Systems. *Recommender Systems*. Springer International Publishing. 225–254.

[2] Barreau, B. and Carlier, L. 2020. History-Augmented Collaborative Filtering for Financial

Recommendations. *RecSys 2020 - 14th ACM Conference on Recommender Systems* (2020), 492–497.

[3] Beel, J. et al. 2019. Darwin & goliath: A white-label recommender-system as-a-service with automated algorithm-selection. *RecSys 2019 - 13th ACM Conference on Recommender Systems* (Sep. 2019), 534–535.

[4] Beel, J. 2017. It's Time to Consider "Time" when Evaluating Recommender-System Algorithms [Proposal]. (Aug. 2017).

[5] Beel, J. et al. 2019. Rard II: The 94 million related-article recommendation dataset. *CEUR Workshop Proceedings* (2019).

[6] Beel, J. and Brunel, V. 2019. Data pruning in recommender systems research: Best-practice or malpractice? *CEUR Workshop Proceedings* (2019), 26–30.

[7] Beel, J. and Langer, S. 2015. A comparison of offline evaluations, online evaluations, and user studies in the context of research-paper recommender systems. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (2015), 153–168.

[8] Bennett, J. and Lanning, S. 2007. The Netflix Prize. *KDD Cup and Workshop*. (2007), 3–6.

[9] Ekstrand, M.D. 2020. LensKit for Python: Next-Generation Software for Recommender Systems Experiments. *International Conference on Information and Knowledge Management, Proceedings* (Oct. 2020), 2999–3006.

[10] Feely, C. et al. 2020. Providing Explainable Race-Time Predictions and Training Plan Recommendations to Marathon Runners. *RecSys 2020 - 14th ACM Conference on Recommender Systems* (2020), 539–544.

[11] Gunawardana, A. and Shani, G. 2015. Evaluating recommender systems. *Recommender Systems Handbook, Second Edition*. Springer US. 265–308.

[12] He, R. and Mcauley, J. Ups and Downs: Modeling the Visual Evolution of Fashion Trends with One-Class Collaborative Filtering. DOI:https://doi.org/10.1145/2872427.2883037.

[13] He, X. et al. 2020. Contextual User Browsing Bandits for Large-Scale Online Mobile Recommendation. *RecSys 2020 - 14th ACM Conference on Recommender Systems* (2020), 63–72.

[14] Lathia, N.K. 2010. Evaluating collaborative filtering over time. *Methodology*. (2010), 1–140. DOI:https://doi.org/citeulike-article-id:7853161.

[15] Maxwell, H. and A., K. 2015. The MovieLens Datasets. *ACM Transactions on Interactive Intelligent Systems (TiiS)*. 5, 4 (Dec. 2015). DOI:https://doi.org/10.1145/2827872.

[16] Soto, P.G.C. 2011. Temporal Models in Recommender Systems: An Exploratory Study on Different Evaluation Dimensions. *Time*. (2011).

[17] Sun, Z. et al. 2020. Are We Evaluating Rigorously? Benchmarking Recommendation for Reproducible Evaluation and Fair Comparison. *RecSys 2020 - 14th ACM Conference on Recommender Systems* (2020), 23–32.

6. Appendix

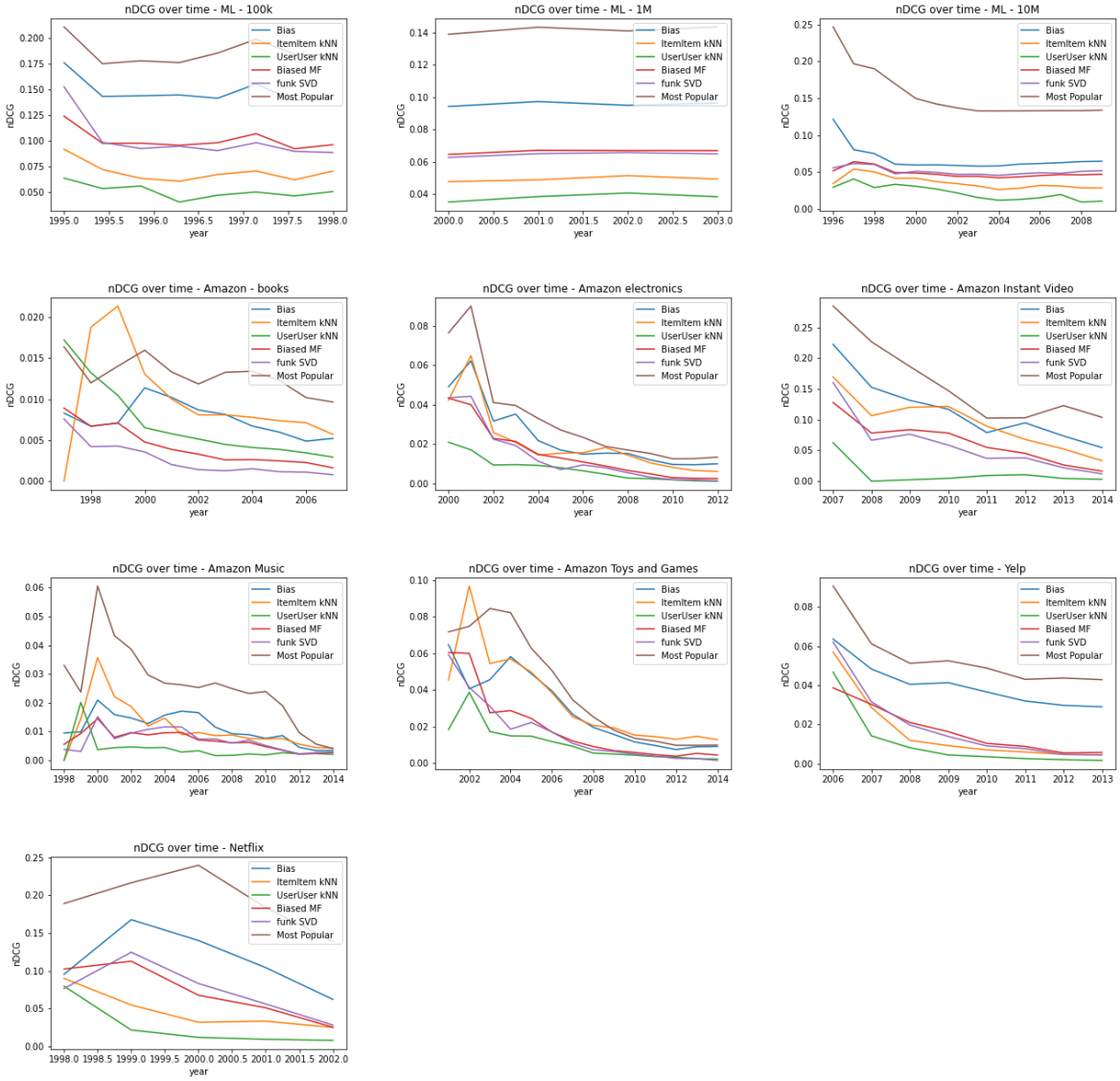


Figure 4: Evolution of nDCG over time for all datasets. (Evolution of recall is very similar and due to space restrictions omitted).

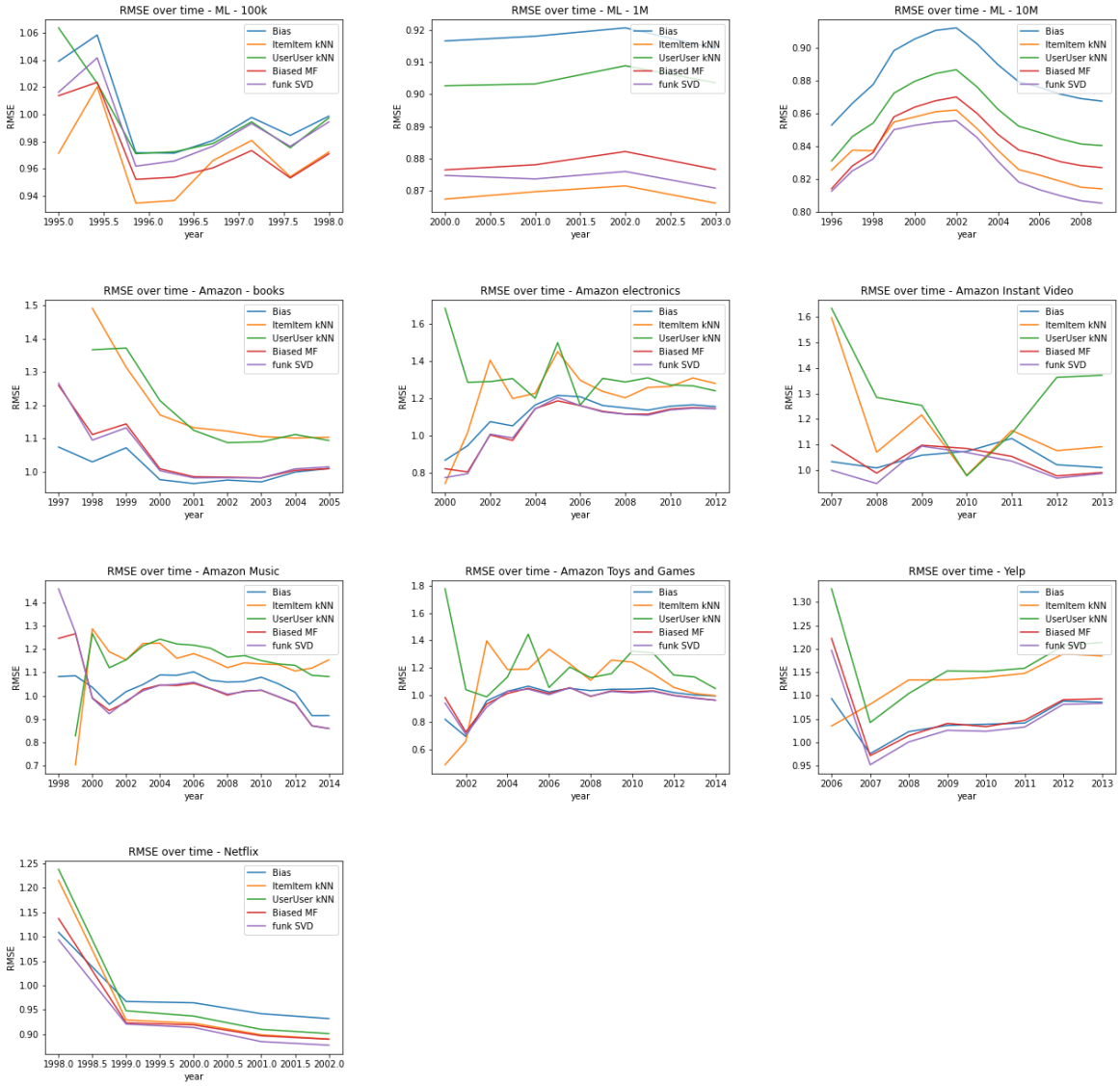


Figure 5: Evolution of RMSE over for all datasets.