Evaluating Recommendations in a User Interface With Multiple Carousels

Discussion Paper

Maurizio Ferrari Dacrema[1], Nicolò Felicioni[1] and Paolo Cremonesi[1,2]

[1]Politecnico di Milano, Piazza Leonardo da Vinci 32, 20133 Milano, Italy
[2]ContentWise, Via Simone Schiaffino 11, Milano, 20158, Milano, Italy

Abstract

Many video-on-demand and music streaming services provide the user with a page consisting of several recommendation lists, i.e., *widgets* or *swipeable carousels*, each built with a specific criteria (e.g., most recent, TV series, etc.). Finding efficient strategies to select which carousels to display is an active research topic of great industrial interest. In this setting the overall quality of the recommendations of a new algorithm cannot be assessed by measuring solely its individual recommendation quality. Rather, it should be evaluated in a context where other recommendation lists are already available, to account for how they complement each other. The traditional offline evaluation protocol however does not take this into account. To address this limitation, we propose an *offline evaluation protocol for a carousel setting* in which the recommendation quality of a model is measured by how much it improves upon that of an already available set of carousels. Our results indicate that under a carousel setting the ranking of the algorithms changes sometimes significantly. This work is an extended abstract of [1].

Keywords

Recommender Systems, User Interface, Evaluation

1. Introduction

The general goal of a recommendation system is to help the users navigate the large number of options at their disposal by suggesting a limited number of relevant results. Traditionally, the focus of newly developed recommendation systems is to generate the best possible ranked list of results, see [2, 3, 4]. A common assumption in almost all research works is that the recommendations will be provided to the user as a single list which will be explored following its order from the first element to the last. However, many industrial applications provide users with a two-dimensional layout of recommendations. Examples are video on demand (e.g., Netflix, Amazon Prime Video) and music streaming services (e.g., Spotify). In these scenarios the user is provided with an interface composed of multiple rows, each row containing thematically consistent recommendations, e.g., most recent, most popular, editorially curated, see [5, 6, 7, 8, 9]. These rows are referred to as *widgets*, *shelves* or as *carousels*. In a carousel interface scenario the

IIR2022: 12th Italian Information Retrieval Workshop, June 29 - June 30th, 2022, Milan, Italy
maurizio.ferrari@polimi.it (M. Ferrari Dacrema); nicolo.felicioni@polimi.it (N. Felicioni); paolo.cremonesi@polimi.it (P. Cremonesi)
0000-0001-7103-2788 (M. Ferrari Dacrema); 0000-0002-3555-7760 (N. Felicioni); 0000-0002-1253-8081 (P. Cremonesi)

user satisfaction depends both on the entire set of carousels shown to the user, rather than on a single list, and on their relative positions. Finding appropriate combinations of algorithms and ranking them to provide the user with a personalized page is an active research topic of significant industrial interest [9, 10, 5], but the community lacks a standardized evaluation procedure to represent this scenario. Frequently, the recommendation quality of the carousel interface is measured by flattening all recommendation lists into a single one, but this is not a realistic evaluation process. In paper [1] we propose: (*i*) a procedure for the offline evaluation of recommendations under a carousel layout; (*ii*) an extension to the NDCG that accounts for how users navigate a two-dimensional interface and perform actions to reveal hidden parts of the interface; (*iii*) two simple strategies to rank carousels in a page. Several recommendation models are evaluated both independently and as the last carousel in a page containing other recommendation lists. In this scenario the recommendation quality of a model is based on how many *new* correct recommendation it provides compared to those already present in the page. Results indicate that the two evaluation procedures lead sometimes to very different results, highlighting the importance to take into account how the carousels complement each other.

2. Characteristics of a Carousel Interface

The carousel interface layout and the way it is usually generated by video-on-demand and music streaming platforms has important characteristics that distinguish it from a single-list setup, see [11]. While a carousel layout may seem similar to a traditional merge-list ensemble, where multiple recommendation lists are combined into one, this is not the case. In a real scenario multiple constraints play a role and must be taken into account:

Layout Structure: The two dimensional user interface of almost all devices is organized with multiple horizontal carousels, where each carousel is generated according to a certain (often explainable) criteria e.g., most recent, most popular, because you watched, editorially curated.

Recommendation Lists: The lists shown to the users within each carousel are generated with different algorithms or by different providers and independently from each other (i.e., each algorithm or provider is not aware of the existence of the other lists or of their content). Consider for example *content aggregators*, which combine carousels from different providers, e.g., Sky, Youtube, Netflix, Prime Video, etc. Due to either business constraints or the strict real-time requirements, no single post-processing step is applied, e.g., to remove items duplicated across different carousels. Due to this, while each individual recommendation list does not contain duplicates, the same item may be recommended in multiple carousels.

User Behavior: The users will focus on the top-left triangle of the screen rather than exploring the carousels sequentially. This is usually called *golden triangle*. Furthermore, they will explore the recommendations in different ways according to which actions they need to perform in order to reveal them. Usually users tend to navigate more easily with simple swipes rather than repeated mouse clicks, hence their behavior, as it is known, will change according to the device (e.g., personal computer, smartphone, tablet, Smart TV).

3. Experimental Methodology

While the traditional evaluation assesses the recommendation quality of a single recommendation model, in a carousel scenario the goal is to assess the recommendation quality of a certain layout composed of recommendation lists. Once it is possible to evaluate the overall recommendation quality of a single layout it is possible to compare different layouts in order to select the best one. For example, one may wish to select the optimal carousel ranking or to choose which recommendation model should generate a specific carousel.

Layout generation: The layout will contain a fixed number of carousels, or recommendation lists, of a given length. If some of the carousels are generated with recommendation models, the first step is to ensure that all models are adequately optimized. Since the specific layout structure that will be shown is, in general, not known in advance each recommendation model should be optimized independently. The recommendations that will be shown to the users are the sequence of all the recommendation lists in the layout, without any centralized postprocessing.

Evaluation metrics: The recommendations provided to the user will be displayed in a two-dimensional pattern. A frequent simplification is to concatenate all lists in a single one and remove duplicate recommendations. While this allows to use traditional metrics (e.g., NDCG, MAP), it makes assumptions that are not consistent with a carousel layout: (*i*) the user explores the lists sequentially; (*ii*) the recommendation lists are centrally collected and postprocessed. In a real scenario we must ensure that a correct recommendation is only counted once and with the correct ranking, if it appeared multiple times in different positions. The correct recommendation should be counted where the user would *see it first*, according to the user's navigation pattern, which requires to define a new ranking discount function to compute ranking metrics.[1]

4. Results and Discussion

In a realistic carousel scenario, several recommendation lists (or carousels) are available generated with different algorithms or editorial rules. In order to mimic this setting we include in the evaluation 16 algorithms that are both simple, well-known and competitive [12]. The algorithsm are: TopPopular, Global Effects, SLIM ElasticNet and SLIM BPR [13], EASE^R [14], P$^3\alpha$ [15] and RP$^3\beta$ [16], PureSVD [3], FunkSVD [12], Non-negative matrix factorization [17], MF BPR [18] and IALS [19], ItemKNN content-based and ItemKNN CF-CBF [20]. The evaluation includes three datasets: *MovieLens 20M* [21], *Netflix Prize* [22] and *ContentWise Impressions* [7]. The data is split in 80% training, 10% validation and 10% test with a random holdout. The hyperparameter search is conduced as in [12] with Bayesian Search by exploring 50 cases.

Due to space reasons, we will describe in particular one experiment comparing the model ranking for Movielens20M. The goal is to choose which model to add as the *last* carousel in an interface that contains an increasing number of carousels: TopPopular, ItemKNN CF and, for the Movielens 20M dataset, ItemKNN CBF. The models are first evaluated individually with the traditional evaluation protocol and then with the proposed carousel evaluation protocol.

[1] See the full paper for a detailed description on how to compute evaluation metrics [1].

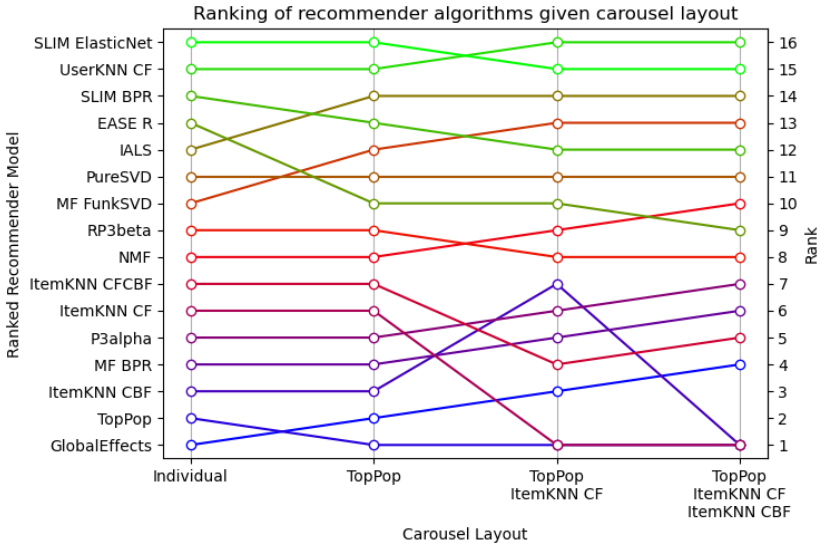

Figure 1: Visualization of how the ranking of several recommendation models changes for Movielens20M when they are evaluated independently or as the last recommendation list in a carousel interface of increasing complexity.

All recommendation lists have a length of 10 and are evaluated with NDCG. As a general trend we can see that the relative effectiveness of the models differ, resulting in changes to the ranking of the algorithms in the two evaluation modes, see Figure 1. Some models such as GlobalEffects and PureSVD are always ranked in the same position. Others, in this case all other matrix factorization algorithms, gain 2 or 3 positions. On the other hand item-based machine learning models tend to consistently lose some positions, with EASE^R being the worst affected and losing 4 positions. As a result, while in the individual evaluation the best algorithms are SLIM ElasticNet, UserKNN CF, SLIM BPR and EASE^R, in the carousel evaluation the best algorithms are UserKNN CF, SLIM ElasticNet, IALS and FunkSVD. Since the recommendation lists generated by all algorithms are identical for both evaluation procedures, the difference in the ranking lies in how those recommendations intersect. Algorithms which will tend to recommend popular items will be penalized in this carousel evaluation because popular items will already be present in the TopPopular carousel, whereas algorithms providing accurate recommendation but involving less popular items will be advantaged. These results are similar for the Netflix Prize and ContentWise Impressions datasets, although the affected models change. For example, on the Netflix Prize dataset a carousel layout with TopPopular and ItemKNN CF causes two sharp changes in ranking: EASE^R falls by 6 positions while FunkSVD gains 7.

The results indicate that accounting for how multiple recommendation lists complement each other can produce substantially different results compared to evaluating each model independently and therefore it is an aspect that should be taken into account when developing recommendation models aimed to domains that use carousel interfaces. The carousel evaluation protocol also opens new research directions, such as how to combine the strength of various models to provide the user with ever more accurate and interesting recommendations.

References

[1] M. Ferrari Dacrema, N. Felicioni, P. Cremonesi, Offline evaluation of recommender systems in a user interface with multiple carousels, Frontiers Big Data 5 (2022) 910030. URL: https://doi.org/10.3389/fdata.2022.910030. doi:10.3389/fdata.2022.910030.

[2] J. L. Herlocker, J. A. Konstan, L. G. Terveen, J. Riedl, Evaluating collaborative filtering recommender systems, ACM Trans. Inf. Syst. 22 (2004) 5–53. URL: https://doi.org/10.1145/963770.963772. doi:10.1145/963770.963772.

[3] P. Cremonesi, Y. Koren, R. Turrin, Performance of recommender algorithms on top-n recommendation tasks, in: X. Amatriain, M. Torrens, P. Resnick, M. Zanker (Eds.), Proceedings of the 2010 ACM Conference on Recommender Systems, RecSys 2010, Barcelona, Spain, September 26-30, 2010, ACM, 2010, pp. 39–46. URL: https://doi.org/10.1145/1864708.1864721. doi:10.1145/1864708.1864721.

[4] M. Sanderson, W. B. Croft, The history of information retrieval research, Proc. IEEE 100 (2012) 1444–1451. URL: https://doi.org/10.1109/JPROC.2012.2189916. doi:10.1109/JPROC.2012.2189916.

[5] C. Wu, C. V. Alvino, A. J. Smola, J. Basilico, Using navigation to improve recommendations in real-time, in: S. Sen, W. Geyer, J. Freyne, P. Castells (Eds.), Proceedings of the 10th ACM Conference on Recommender Systems, Boston, MA, USA, September 15-19, 2016, ACM, 2016, pp. 341–348. URL: https://doi.org/10.1145/2959100.2959174. doi:10.1145/2959100.2959174.

[6] E. Elahi, A. Chandrashekar, Learning representations of hierarchical slates in collaborative filtering, in: R. L. T. Santos, L. B. Marinho, E. M. Daly, L. Chen, K. Falk, N. Koenigstein, E. S. de Moura (Eds.), RecSys 2020: Fourteenth ACM Conference on Recommender Systems, Virtual Event, Brazil, September 22-26, 2020, ACM, 2020, pp. 703–707. URL: https://doi.org/10.1145/3383313.3418484. doi:10.1145/3383313.3418484.

[7] F. B. Pérez Maurera, M. Ferrari Dacrema, L. Saule, M. Scriminaci, P. Cremonesi, Contentwise impressions: An industrial dataset with impressions included, in: M. d'Aquin, S. Dietze, C. Hauff, E. Curry, P. Cudré-Mauroux (Eds.), CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19-23, 2020, ACM, 2020, pp. 3093–3100. URL: https://doi.org/10.1145/3340531.3412774. doi:10.1145/3340531.3412774.

[8] A. Gruson, P. Chandar, C. Charbuillet, J. McInerney, S. Hansen, D. Tardieu, B. Carterette, Offline evaluation to make decisions about playlistrecommendation algorithms, in: J. S. Culpepper, A. Moffat, P. N. Bennett, K. Lerman (Eds.), Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, WSDM 2019, Melbourne, VIC, Australia, February 11-15, 2019, ACM, 2019, pp. 420–428. URL: https://doi.org/10.1145/3289600.3291027. doi:10.1145/3289600.3291027.

[9] W. Bendada, G. Salha, T. Bontempelli, Carousel personalization in music streaming apps with contextual bandits, in: R. L. T. Santos, L. B. Marinho, E. M. Daly, L. Chen, K. Falk, N. Koenigstein, E. S. de Moura (Eds.), RecSys 2020: Fourteenth ACM Conference on Recommender Systems, Virtual Event, Brazil, September 22-26, 2020, ACM, 2020, pp. 420–425. URL: https://doi.org/10.1145/3383313.3412217. doi:10.1145/3383313.3412217.

[10] W. Ding, D. Govindaraj, S. V. N. Vishwanathan, Whole page optimization with global

constraints, in: A. Teredesai, V. Kumar, Y. Li, R. Rosales, E. Terzi, G. Karypis (Eds.), Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019, ACM, 2019, pp. 3153–3161. URL: https://doi.org/10.1145/3292500.3330675. doi:10.1145/3292500.3330675.

[11] N. Felicioni, M. Ferrari Dacrema, P. Cremonesi, A methodology for the offline evaluation of recommender systems in a user interface with multiple carousels, in: J. Masthoff, E. Herder, N. Tintarev, M. Tkalcic (Eds.), Adjunct Publication of the 29th ACM Conference on User Modeling, Adaptation and Personalization, UMAP 2021, Utrecht, The Netherlands, June 21-25, 2021, ACM, 2021, pp. 10–15. URL: https://doi.org/10.1145/3450614.3461680. doi:10.1145/3450614.3461680.

[12] M. Ferrari Dacrema, S. Boglio, P. Cremonesi, D. Jannach, A troubling analysis of reproducibility and progress in recommender systems research, ACM Trans. Inf. Syst. 39 (2021). URL: https://doi.org/10.1145/3434185. doi:10.1145/3434185.

[13] X. Ning, G. Karypis, SLIM: Sparse linear methods for top-n recommender systems, in: Proceedings of the 11th IEEE International Conference on Data Mining (ICDM '11), 2011, pp. 497–506.

[14] H. Steck, Embarrassingly shallow autoencoders for sparse data, in: L. Liu, R. W. White, A. Mantrach, F. Silvestri, J. J. McAuley, R. Baeza-Yates, L. Zia (Eds.), The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019, ACM, 2019, pp. 3251–3257. URL: https://doi.org/10.1145/3308558.3313710. doi:10.1145/3308558.3313710.

[15] C. Cooper, S. Lee, T. Radzik, Y. Siantos, Random walks in recommender systems: exact computation and simulations, in: C. Chung, A. Z. Broder, K. Shim, T. Suel (Eds.), 23rd International World Wide Web Conference, WWW '14, Seoul, Republic of Korea, April 7-11, 2014, Companion Volume, ACM, 2014, pp. 811–816. URL: https://doi.org/10.1145/2567948.2579244. doi:10.1145/2567948.2579244.

[16] B. Paudel, F. Christoffel, C. Newell, A. Bernstein, Updatable, accurate, diverse, and scalable recommendations for interactive applications, ACM Trans. Interact. Intell. Syst. 7 (2017) 1:1–1:34. URL: https://doi.org/10.1145/2955101. doi:10.1145/2955101.

[17] A. Cichocki, A. H. Phan, Fast local algorithms for large scale nonnegative matrix and tensor factorizations, IEICE Trans. Fundam. Electron. Commun. Comput. Sci. 92-A (2009) 708–721. URL: https://doi.org/10.1587/transfun.E92.A.708. doi:10.1587/transfun.E92.A.708.

[18] S. Rendle, C. Freudenthaler, Z. Gantner, L. Schmidt-Thieme, BPR: bayesian personalized ranking from implicit feedback, in: J. A. Bilmes, A. Y. Ng (Eds.), UAI 2009, Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, Montreal, QC, Canada, June 18-21, 2009, AUAI Press, 2009, pp. 452–461. URL: https://dslpitt.org/uai/displayArticleDetails.jsp?mmnu=1&smnu=2&article_id=1630&proceeding_id=25.

[19] Y. Hu, Y. Koren, C. Volinsky, Collaborative filtering for implicit feedback datasets, in: Proceedings of the 8th IEEE International Conference on Data Mining (ICDM 2008), December 15-19, 2008, Pisa, Italy, IEEE Computer Society, 2008, pp. 263–272. URL: https://doi.org/10.1109/ICDM.2008.22. doi:10.1109/ICDM.2008.22.

[20] B. Mobasher, X. Jin, Y. Zhou, Semantically enhanced collaborative filtering on the web, in: B. Berendt, A. Hotho, D. Mladenic, M. van Someren, M. Spiliopoulou, G. Stumme (Eds.), Web Mining: From Web to Semantic Web, First European Web Mining Forum, EMWF 2003, Cavtat-Dubrovnik, Croatia, September 22, 2003, Revised Selected and Invited

Papers, volume 3209 of *Lecture Notes in Computer Science*, Springer, 2003, pp. 57–76. URL: https://doi.org/10.1007/978-3-540-30123-3_4. doi:10.1007/978-3-540-30123-3_4.

[21] F. M. Harper, J. A. Konstan, The movielens datasets: History and context, ACM Trans. Interact. Intell. Syst. 5 (2016) 19:1–19:19. URL: https://doi.org/10.1145/2827872. doi:10.1145/2827872.

[22] J. Bennett, S. Lanning, et al., The netflix prize, in: Proceedings of KDD cup and workshop, volume 2007, New York, 2007, p. 35.