Efficiency-Effectiveness Trade-offs in Recommendation Systems

Iulia Paun
University of Glasgow
Glasgow, UK
iulia.paun@glasgow.ac.uk

ABSTRACT

Throughout the years, numerous recommendation algorithms have been developed to address the information filtering problem by leveraging users' tastes through implicit or explicit feedback. In this paper, we present the work undertaken as part of a PhD thesis focused on exploring new evaluation dimensions centred around the efficiency-effectiveness trade-offs present in state-of-the-art recommendation systems. Firstly, we highlight the lack of efficiency-oriented studies and we formulate the research problem. Then, we propose a mapping of the design space and a classification of the recommendation algorithms/models with respect to salient attributes and characteristics. At the same time, we explain why and how assessing the recommendations on an accuracy versus training cost curve would advance the current knowledge in the area of evaluation, as well as open new research avenues for exploring parameter configurations within well-known algorithms. Finally, we make the case for a comprehensive methodology that incorporates predictive efficiency-effectiveness models, which illustrate the performance and behaviour of the recommendation systems under different recommendation tasks, while satisfying user-defined quality of service constraints and goals.

CCS CONCEPTS

• **Information systems** → **Recommender systems**; *Evaluation of retrieval results.*

KEYWORDS

Recommendation systems; efficiency/effectiveness methodology; benchmark and evaluation framework.

ACM Reference Format:
Iulia Paun. 2020. Efficiency-Effectiveness Trade-offs in Recommendation Systems. In *Fourteenth ACM Conference on Recommender Systems (RecSys '20), September 21–26, 2020, Virtual Event, Brazil.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3383313.3411452

1 INTRODUCTION

In the past decade, machine learning models have become a critical part of large scale analytics frameworks. This has also been extended to retrieval and recommendation systems. Nevertheless, the constant growth of the input data impacts the efficiency of the models, as more training resources are required to attain a good level of effectiveness. While most of the current work is contributing improved models that achieve high accuracy, limited research has addressed their performance and scalability, which remain open problems in the field.

The evaluation of recommendation systems spans across multiple dimensions, such as accuracy, relevance, novelty, diversity, as well as user and content providers' satisfaction. Over the years, the literature has established how one can evaluate the recommendations along these dimensions and what tools/metrics can be used. However, the diversity of the approaches and their corresponding implementations make it difficult to compare across them and against new models. Baselines are often used to alleviate this problem, but how these are selected is not always trivial due to a large design space with many variables (e.g., recommendation task, algorithms and data structures, datasets, etc.). For example, if someone proposes a new highly scalable algorithm, what baselines will be used to prove its efficiency. The same holds true for solutions that aim to improve effectiveness/accuracy.

This research focuses on defining and addressing the efficiency-effectiveness trade-offs within popular state-of-the-art recommendation systems, through the following goals and contributions: *(a) devise a benchmark and evaluation framework* that compares various implicit, explicit, and deep learning based recommendation algorithms; *(b) build a cost model* that quantifies the efficiency of the aforementioned approaches; *(c) provide an evaluation methodology* that augments current effectiveness metrics to include quality of service constraints, as well as training costs.

Recommendation systems (RS) come in different forms and implement various approaches. The three main categories of classifying them are collaborative filtering (CF), content-based or cognitive filtering (CBF), and hybrid (i.e., CF combined with CBF). Alternatively, RS can be distinguished with respect to the type of feedback (implicit or explicit) used in the input data. Furthermore, RS use a number of different techniques to produce recommendations, including, but not limited to, matrix factorisation, K-nearest neighbours, deep learning, etc. Each of these methods spans different implementations and optimisation schemes making it difficult to decide which version to include as a baseline. Consequently, the need for a *benchmark and evaluation framework* that includes popular state-of-the-art recommendation algorithms arises. This framework will highlight how existing solutions perform across several dimensions, making the baseline selection process easier when a new RS is developed, since researchers will know which algorithms to select for comparison. Additionally, this framework can also be augmented with well-known datasets and evaluation metrics.

As previously described, the efficiency of the recommendation systems is often omitted in the evaluation studies. Beel et al. [1] highlighted that many papers do not report the runtimes of the

RecSys '20, September 21–26, 2020, Virtual Event, Brazil

ACM ISBN 978-1-4503-7583-2/20/09.
https://doi.org/10.1145/3383313.3411452

proposed algorithms and often the computational complexity is not discussed. These aspects play a critical role for the content providers, who deal with large volume of data and need to produce recommendations for their users in near-real-time. Therefore, we propose to capture the efficiency of the algorithms included the aforementioned framework through a *cost model* that reports on the algorithmic complexity and estimates the resource consumption (e.g., runtime, memory footprint) with respect to the size and the characteristics (e.g., sparsity) of the input data.

There is an ongoing discussion in the research community regarding the resource consumption of certain approaches (e.g., deep learning) with respect to their accuracy [20, 30]. This motivates the last main contribution of the PhD, namely an *evaluation methodology* that exploits the efficiency-effectiveness trade-offs within machine learnt models for recommendation systems. As part of this thread of research, we aim to provide accuracy versus training cost curves for the algorithms included in the benchmark framework, and answer questions similar to the following ones: Given an algorithm A, in the family of recommendation systems B, how long does it take to train on input C to achieve a fixed effectiveness D (e.g., RMSE of 0.5, precision@K of 80%, etc.)? Given X hours of CPU time or a memory footprint Y, what is the maximum accuracy D that algorithm A, in the family of recommendation systems B, can achieve on input C? This would allow researchers and practitioners to plan their experimental work, while maximising the available computational resources. Additionally, with the proposed methodology, content providers would be able to know when a model has achieved a satisfactory level of training to provide adequate recommendations or how much data is required to meet a predefined accuracy value.

2 RELATED WORK

Recommendation systems have been traditionally used to address the information overload problem, in which users have to make a decision while they are being faced with a multitude of choices/options. Suggesting relevant items to users has often been described as a recommendation task [27], which has been further expanded as an optimisation problem. In these settings, success is usually defined as finding the set of items that maximises the interest of a user or helps him/her in the decision process. This has been achieved by leveraging users' preferences [25], stated either explicitly, through ratings, or implicitly through views, clicks, etc., and by recommending items that are either (a) similar to other items (CBF) [32] or (b) items that have been consumed by similar users (CF) [29]. There are also studies [18, 23, 24] which combined these two approaches to improve the quality of the recommendations.

The literature shows that recommendation algorithms can be evaluated using several approaches, methodologies, and metrics. One of the first dimensions for assessing the quality of a recommendation is accuracy [13], defined as the number of relevant items recommended to the user. Accuracy increases or decreases based on the users' perceived usefulness of the recommendations. This is often a challenge for modelling the users' interests and needs, since they may have different backgrounds or level of expertise, look for information in different ways [28], or are placed in various contexts [6]. Additionally, it has been demonstrated that accuracy is often linked with the level of coverage in the items collection [10].

The second dimension on which we can assess recommendations is related to the users' satisfaction. Given the abundance of choices that users face every day, it is often incorrect to assume that an accurate recommendation will also be a satisfactory one. Aspects such as serendipity [9], novelty and diversity [4] of the recommended items play a critical role in the users' satisfaction. Furthermore, under the same recommendation task, different demographic groups may attain various levels of satisfaction [2]. The time and effort users spend to get their recommendations also impacts their satisfaction, since in some platforms they need to build a profile with goals and interests, while in other systems these are inferred automatically.

The third dimension for evaluating recommendation systems is the satisfaction of the content/recommendation providers. While it is generally agreed that providers are satisfied when their customers/users are happy, there are other aspects that can impact the providers' satisfaction. One of the main challenges faced by content/recommendation providers is the cost versus the available resources [27], such as labour, CPU usage, memory, disk space, etc. This is why most providers will look into algorithms optimisation schemes to achieve high computational efficiency, while the effectiveness of the recommendations is not impacted too much. To the best of our knowledge there is no algorithm/solution that does well in all three evaluation dimensions. This is why we believe that exploring and addressing the efficiency-effectiveness trade-offs within popular recommendation systems is a very interesting problem.

In order to compare across approaches using the three dimensions described above, we need to be able to quantify them. The efficiency of a system is often measured through metrics such as runtime (CPU usage), memory footprint, disk I/O, page faults, bandwidth, response time, latency [7], while its effectiveness is weighted through accuracy, novelty, precision, etc. In the area of recommendation systems, there are three main evaluation techniques: (a) user studies, (b) online evaluations, and (c) offline evaluations.

For user studies, users actively provide feedback, often quantified as ratings, on the recommendations produced by various algorithms. Then, each solution is assessed through its rating, where a higher rating means a better approach [27]. In online evaluations, the recommendations are presented to users in real-time, and their engagement with the recommended items is measured through the number of clicks. In these settings, the performance of the systems/models is compared through click-through rates, where higher means better [27]. Lastly, in offline evaluations algorithms are assessed against offline datasets using the following technique: the models/algorithms are trained on one part of the data, while some information is being withheld (e.g., K-fold cross-validation, leave-one-out), often described as testing data. Then, the trained models/algorithms are evaluated based on how well they perform in a recommendation task using the testing data [27]. While all three evaluation approaches ((a), (b), and (c)) have their advantages and limitations, offline studies are the most common, due to the large availability of offline data, as well as the low cost setup. User studies and online evaluations provide more insights into how and why

a solution is good/bad/satisfactory/cumbersome, but also require more expensive resources (e.g., user participation and availability). This is why their prevalence is lower in the literature and research community.

3 RESEARCH METHODOLOGY AND DIRECTIONS

This section describes the datasets and evaluation metrics that will be used in the PhD work and illustrates what RS algorithms we have investigated so far, as well as which ones will be used as part of future directions. Then, the scope of the thesis is described, focusing on the main contributions and research goals. Finally, we present a number of research questions that will be addressed as part of the experimental work.

3.1 Datasets and Evaluation Metrics

As part of the experimental apparatus, we firstly included the well-known *MovieLens* collection [12], which contains scores in range of 1 to 5 that describe users' preferences with respect to a suite of items/films. Based on the number of ratings, MovieLens comes in four different datasets: the biggest 20M - containing 20M ratings, followed by 10M, 1M, and the smallest - 100K ratings. Since explicit data is less frequent in real-life scenarios, we have also looked into implicit datasets, such as the *Last FM* dataset [3], which contains implicit feedback about songs in the form of play-count, but also friendship relationships among users. Another dataset which we used is the *GoodBooks 10K* [33], which combines both implicit and explicit information. Similarly to Movielens, GoodBooks 10K contains explicit ratings of books, but also implicit interactions, such as books assigned to shelves by users. Lastly, we will also consider using the *Yelp* collection [17], containing information and reviews about businesses across 11 metropolitan areas in 4 countries.

The evolution of recommendation systems and models has also brought new insights into how they are being evaluated. Initially, error-based metrics, such as RMSE and MAE, were used to determine the quality of the recommendations. These have been highly used in rating prediction tasks. Then, the relevance of the recommended content was assessed through Information Retrieval (IR) metrics@K, such as precision, recall, F1 score, and normalised discounted cumulative gain (NDCG), where K is the number of recommended items. These metrics are commonly found in ranking tasks. Other research avenues pursued additional evaluation dimensions, such as novelty, diversity, robustness, and serendipity [27]. Due to the large number of metrics and their variety, it has been agreed that a model/algorithm cannot attain good scores for every evaluation approach [11].

For the work undertaken in this PhD, we will focus on offline evaluation tasks, using a combination of the aforementioned metrics, depending on the nature of the recommendation task. For measuring the effectiveness of the studied algorithms, when a metric does not have a value in a finite interval/range (e.g., RMSE, MAE), we will use normalisation techniques to ensure that the reported results are (a) reproducible and (b) comparable with other works. Additionally, we aim to advance the current knowledge about the recommendation systems' efficiency, by carrying out exhaustive performance evaluation studies, which capture metrics such as CPU time, memory footprint, disk I/O, etc. Where possible, for both effectiveness and efficiency, we will compare our findings with the results reported in the literature.

3.2 Recommendation Algorithms and Frameworks

Due to the multitude of recommendation systems and approaches, we started mapping out the design space, and we identified three main areas: *explicit* algorithms, *implicit* algorithms, and *deep learning* based algorithms/models. For each category, we aim to select state-of-the-art representatives that will be studied and included in the benchmark and evaluation framework. So far, we have explored a number of popular explicit recommendation algorithms, such as *(i)* baseline, *(ii)* K-nearest neighbours based, *(iii)* variants of matrix factorisation, *(iv)* slope based, and *(v)* co-clustering approaches. For the explicit feedback recommendations, we used the algorithms' implementations provided by the Surprise framework [16].

As part of the implicit algorithms, we considered well-known representatives, including, but not limited to Alternating Least Squares (ALS) [14], Bayesian Personalized Ranking (BPR) [26], and Logistic Matrix Factorization (LMF) [21]. The implementation of these algorithms is available through the Implicit Python library [8]. For deep learning recommendation systems, we would like to start with the models included in the Spotlight framework [22]. Given the large number of recommendation frameworks and libraries, prior to including any algorithm in the proposed benchmark and evaluation framework, we will seek guidance from a panel of RS and IR experts on what are the most used or the state-of-the-art implementations.

3.3 Research Agenda

The scope of this thesis is threefold: firstly, we aim to set the design space by including a large spectrum of well-known recommendation systems, which will be classified and grouped by non-obvious characteristics, such as algorithmic complexity, internal representation of input data, sets of computations, etc.; secondly, we aim to advance the knowledge in the efficiency of the recommendations, by studying the performance of the models, under different constraints and various inputs sizes; finally, we would like to provide a mapping of the aforementioned recommendation algorithms on an accuracy versus cost of training curve, by exploiting the efficiency-effectiveness trade-offs and limitations present in the implementations, exploring new parameter configurations, and formalising potential improvements/optimisations.

The first step towards achieving a good understanding of the recommendation systems' performance is to conduct an *exhaustive efficiency study of the algorithms*, which analyses their features, characteristics, and behaviour with respect to different recommendation tasks and contexts. The outcome of this goal will consist in a *benchmark and evaluation framework* for popular state-of-the-art recommendation algorithms. We will start by classifying the selected candidates into "families" of recommendation algorithms, by taking into consideration aspects such as the filtering approach (e.g., CF, CBF, hybrid), how the data is structured

and represented internally (e.g., matrices, feature vectors, etc.), as well as the main method for producing the recommendations (e.g., matrix factorisation, neighbourhood based, slope schemes, etc.). Then, we will extend this classification using other dimensions such as computational complexity, growth rate of training cost based on different input sizes, and memory consumption. This will be achieved by exploring the algorithms' performance on datasets with various properties (e.g., different users/items sizes, sparsity), allowing us to gain fresh insights into their efficiency, and highlight potential biases that may arise from different configurations (e.g., neighbourhood sizes, number of feature vectors, convergence rates, number of neurons and layers in deep learning models, etc.).

The second part of the thesis will focus on *building cost models that quantify the efficiency of recommendation systems*, and *predict their training overhead* on different types of inputs. We believe this is a major ongoing problem in the research community, as often the runtimes of the algorithms can differ by a significant factor [1], with extreme cases of algorithms requiring 600 times more CPU time than others [15]. This problem is directly impacting the content providers, who need to produce recommendations for a large number of users using an increasing collection of items. Therefore, the choice of the recommendation algorithms is critical, as their efficiency can determine whether they will be used in production environments. As part of the efficiency cost model, we already investigated the performance of easily explainable recommendation algorithms, such as explicit collaborative filtering (CF) representatives. As expected, we observed that some approaches yielded similar effectiveness (i.e., RMSE) results, while their training cost was significantly different. Additionally, we devised *expected runtime prediction models* that learn the hidden factors in the complexity of the aforementioned algorithms by computing curve fitting primitives. To achieve this, we have incorporated the big-O (worst case) complexity [5] into the prediction models, which provided initial insights into the consumed CPU time. However, in some cases, using the big-O complexity did not produce accurate results for predicting the expected training runtimes. We aim to address this limitation by refining the complexity equations to match big-Θ [5], which asymptotically bounds a function from above and below. This approach has proven to be more reliable for predicting the expected runtime of the algorithms, but it is also harder to model, since the big-Θ equations are not always trivial to derive, and sometimes are bound to specific implementation details. We aim to augment this work by also looking into automatic complexity analysis tools that can be used for runtime modelling and prediction. We envision to wrap up these contributions as a suite of *efficiency cost models* that are based on *approximating probabilistic analysis*, through *adaptive sampling strategies*, which predict the expected runtime of a given recommendation algorithm over an input/dataset.

Another goal that ties into the benchmark and evaluation framework, as well as the efficiency cost models is related to devising a *dynamic sampling methodology*. In order to develop reliable runtime prediction models, we asked ourselves how many samples and how large should they be in order to get a representative measure of the algorithms' runtimes. For drawing the samples, we propose a strategy similar to bootstrapping [19] (without replacement) and Monte Carlo rejection sampling [31]. Furthermore, the number of samples we draw should be adaptive and based on *(a) a user-defined upper limit* and *(b) a user-defined accuracy/error threshold*. Ideally, we would like to have a number of samples that provides a good accuracy for the runtime cost models. However, we should not use too many samples, such that their total runtime would be larger than the runtime of training the algorithms on the full dataset. As part of this goal, we aim to contribute an *adaptive sampling algorithm*, which computes the optimal bound on the number of samples needed in the aforementioned *(a)* and *(b)* scenarios. The proposed algorithm will also incorporate "smart" outlier detection mechanisms, and will be able to handle predefined constraints (e.g., maximum time quota).

Finally, the last part of the PhD will cover the *efficiency-effectiveness evaluation methodology* that will provide fresh insights into the usability of the studied recommendation algorithms on a larger scale, by classifying them using new dimensions, such as their computational performance, and plotting them on accuracy versus training cost curves. We would like to augment the current evaluation techniques for recommendation systems, by combining effectiveness metrics (e.g., RMSE, precision, recall, etc.) with complementary efficiency metrics (e.g., CPU time, memory footprint, disk I/O), while also addressing and incorporating quality of service goals and constraints (e.g., responses per second). By doing so, we will be able to understand how/why some algorithms scale better than others and how does their accuracy change with respect to different data sizes. Additionally, as a result, we will be able to explore different parameter configurations and tuning approaches, propose improvements/optimisations to current solutions, as well as devise new algorithms.

To crystallise the contributions described above, we will conduct experimental work fuelled by the following research questions:

- Can we formulate complexity equations that describe the efficiency of a given algorithm A, belonging to a family of recommendation systems B, on an input C?
- Given inputs of different sizes and a class of recommendation algorithms (e.g., matrix factorisation based), how do these algorithms scale?
- Does the type of input, and therefore the type of the recommendation algorithm used, impact the efficiency of the system? For example, does it take longer to train a traditional/deep learning based recommendation algorithm?
- Given a set of resource constraints, can we model the quality of service (i.e., responses per second) of a recommendation system?
- Given the efficiency models for different families of recommendation systems, and a fixed set of resource constraints/goals, can we predict the quality/relevance/effectiveness of the recommendations?
- Given an effectiveness goal (i.e., we want a precision@10 of 0.8), can we identify configurations that improve the efficiency of the algorithm (i.e., different parameter tuning approaches, distribute the training to lower the CPU usage)?
- What are the steps that need to be followed in order to maximise the effectiveness/efficiency of a recommendation

system, given a set of available resources, and the quality of service goals/constraints? Can we precompute features of the input that would speed up the training runtime or provide a better accuracy for the recommendations produced?

4 CONCLUSIONS

This paper presented the current status and the research goals of a PhD thesis focused on exploring and analysing the inherent efficiency-effectiveness trade-offs within popular recommendation algorithms in offline evaluation settings. Firstly, we described the motivation for these performance assessment studies, driven by the constantly growing data and the long-term feasibility of the algorithms used by content providers. Secondly, we identified the need for a benchmark and evaluation framework that would allow new advances in recommendation models to be compared against state-of-the-art solutions on multiple dimensions. Thirdly, we have proposed an efficiency cost model that predicts the expected runtime of an algorithm on a given input, through approximating probabilistic analysis using dynamic sampling and complexity equations. Lastly, we made the case for extending current evaluation metrics to include efficiency aspects and quality of service constraints, by mapping the algorithms on accuracy versus training cost curves. This would lead towards more comprehensive evaluation approaches, by quantifying both the relevance/quality and the performance of the recommendations on different dimensions, as well as exploring new configurations.

While the preliminary results on the efficiency-effectiveness trade-offs encountered in explicit recommendation systems validate some of our hypotheses, we identified that further research and experiments are required to improve the efficiency cost models that predict the expected runtimes. As part of the short-term future work, we would like to pursue novel avenues for computing the hidden factors in the complexity equations. Then, we aim to expand our studies, models, and methodologies to implicit and deep learning based approaches, as well as formalise the efficiency metrics that will complement the current/traditional ways of evaluating the recommendations.

ACKNOWLEDGMENTS

This work is conducted under the supervision of Dr Nikos Ntarmos, Dr Craig Macdonald, and Dr Yashar Moshfeghi. I would also like to thank Dr Richard McCreadie and Dr Simon Rogers for their feedback during the annual progression meetings.

The doctoral studies are co-funded by the Erasmus+ Programme of the European Union under the PRIMES project (no. 2016-1-UK01-KA201-024631) and by the UK Government through an EPSRC grant (no. 509668).

REFERENCES

[1] Joeran Beel, Stefan Langer, Marcel Genzmehr, Bela Gipp, Corinna Breitinger, and Andreas Nürnberger. 2013. Research paper recommender system evaluation: a quantitative literature survey. In *Proceedings of the International Workshop on Reproducibility and Replication in Recommender Systems Evaluation*. ACM RecSys, Hong Kong, 15–22.

[2] Joeran Beel, Stefan Langer, Andreas Nürnberger, and Marcel Genzmehr. 2013. The impact of demographics (age and gender) and other user-characteristics on evaluating recommender systems. In *International Conference on Theory and Practice of Digital Libraries*. Springer, Valletta, Malta, 396–400.

[3] Thierry Bertin-Mahieux, Daniel P.W. Ellis, Brian Whitman, and Paul Lamere. 2011. The Million Song Dataset. In *Proceedings of the 12th International Conference on Music Information Retrieval (2011)*. ISMIR, Florida, USA, 591–596.

[4] Pablo Castells, Neil J Hurley, and Saul Vargas. 2015. Novelty and diversity in recommender systems. In *Recommender systems handbook*. Springer, Boston, MA, USA, 881–918.

[5] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. 2009. *Introduction to Algorithms, Third Edition* (3rd ed.). The MIT Press, Cambridge, Massachusetts.

[6] Toon De Pessemier, Simon Dooms, and Luc Martens. 2014. Context-aware recommendations through context and activity recognition in a mobile environment. *Multimedia Tools and Applications* 72, 3 (2014), 2925–2948.

[7] Paul Fortier and Howard Michel. 2003. *Computer systems performance evaluation and prediction*. Elsevier, Cambridge, MA, USA.

[8] Ben Frederickson. 2016. Fast Python Collaborative Filtering for Implicit Feedback Datasets. https://github.com/benfred/implicit.

[9] Mouzhi Ge, Carla Delgado-Battenfeld, and Dietmar Jannach. 2010. Beyond accuracy: evaluating recommender systems by coverage and serendipity. In *Proceedings of the fourth ACM conference on Recommender systems*. ACM, Barcelona, Spain, 257–260.

[10] Nathaniel Good, J Ben Schafer, Joseph A Konstan, Al Borchers, Badrul Sarwar, Jon Herlocker, John Riedl, et al. 1999. Combining collaborative filtering with personal agents for better recommendations. *AAAI/IAAI* 439 (1999), 1–8.

[11] Asela Gunawardana and Guy Shani. 2009. A survey of accuracy evaluation metrics of recommendation tasks. *Journal of Machine Learning Research* 10, Dec (2009), 2935–2962.

[12] F Maxwell Harper and Joseph A Konstan. 2015. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)* 5, 4 (2015), 1–19.

[13] Jonathan L Herlocker, Joseph A Konstan, Loren G Terveen, and John T Riedl. 2004. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)* 22, 1 (2004), 5–53.

[14] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative filtering for implicit feedback datasets. In *2008 Eighth IEEE International Conference on Data Mining*. IEEE, Pisa, Italy, 263–272.

[15] Wenyi Huang, Saurabh Kataria, Cornelia Caragea, Prasenjit Mitra, C Lee Giles, and Lior Rokach. 2012. Recommending citations: translating papers into references. In *Proceedings of the 21st ACM International Conference on Information and Knowledge management*. ACM New York, Hawaii, USA, 1910–1914.

[16] Nicolas Hug. 2017. Surprise, a Python library for recommender systems. http://surpriselib.com.

[17] Yelp Inc. 2013. Yelp Dataset Challenge. https://www.yelp.com/dataset.

[18] Amir H Jadidinejad, Craig Macdonald, and Iadh Ounis. 2019. Unifying Explicit and Implicit Feedback for Rating Prediction and Ranking Recommendation Tasks. In *Proceedings of the 2019 ACM SIGIR International Conference on Theory of Information Retrieval*. ACM, Santa Clara, CA, USA, 149–156.

[19] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. 2013. *An introduction to statistical learning*. Vol. 112. Springer, New York, USA.

[20] Peter H Jin, Qiaochu Yuan, Forrest Iandola, and Kurt Keutzer. 2016. How to scale distributed deep learning? *arXiv* abs/1611.04581 (2016), 1–16.

[21] Christopher C Johnson. 2014. Logistic matrix factorization for implicit feedback data. *Advances in Neural Information Processing Systems* 27 (2014), 1–9.

[22] Maciej Kula. 2017. Spotlight. https://github.com/maciejkula/spotlight.

[23] Nathan N Liu, Evan W Xiang, Min Zhao, and Qiang Yang. 2010. Unifying explicit and implicit feedback for collaborative filtering. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*. ACM, Toronto, Canada, 1445–1448.

[24] Siping Liu, Xiaohan Tu, and Renfa Li. 2017. Unifying explicit and implicit feedback for top-N recommendation. In *2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA)(*. IEEE, Beijing, China, 35–39.

[25] István Pilászy, Dávid Zibriczky, and Domonkos Tikk. 2010. Fast als-based matrix factorization for explicit and implicit feedback datasets. In *Proceedings of the fourth ACM conference on Recommender systems*. ACM, Barcelona, Spain, 71–78.

[26] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2012. BPR: Bayesian personalized ranking from implicit feedback. *arXiv* abs/1205.2618 (2012), 452–461.

[27] Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B. Kantor. 2010. *Recommender Systems Handbook* (1st ed.). Springer, Boston, Massachusetts.

[28] Daniel E Rose and Danny Levinson. 2004. Understanding user goals in web search. In *Proceedings of the 13th International Conference on World Wide Web*. ACM, New York, USA, 13–19.

[29] J Ben Schafer, Dan Frankowski, Jon Herlocker, and Shilad Sen. 2007. Collaborative filtering recommender systems. In *The adaptive web*. Springer, New York, USA, 291–324.

[30] Shaohuai Shi, Qiang Wang, Pengfei Xu, and Xiaowen Chu. 2016. Benchmarking state-of-the-art deep learning software tools. In *2016 7th International Conference*

on Cloud Computing and Big Data (CCBD). IEEE, Chengdu, China, 99–104.
[31] Zhiqiang Tan. 2006. Monte Carlo integration with acceptance-rejection. *Journal of Computational and Graphical Statistics* 15, 3 (2006), 735–752.
[32] Robin Van Meteren and Maarten Van Someren. 2000. Using content-based filtering for recommendation. In *Proceedings of the Machine Learning in the New Information Age: MLnet/ECML2000 Workshop*, Vol. 30. ECML, Barcelona, Spain, 47–56.
[33] Zygmunt Zajac. 2017. Goodbooks-10k: a new dataset for book recommendations. http://fastml.com/goodbooks-10k.