Large-Scale Recommender Systems at bol.com

Barrie Kersbergen
bol.com
bkersbergen@bol.com

Sebastian Schelter
AIRLab, University of Amsterdam & Ahold Delhaize
s.schelter@uva.nl

Abstract. E-commerce companies use recommender systems to enable customers to find items they might like. With rapidly growing data sizes, the predictive performance, processing efficiency and scalability of machine learning-based recommendations systems and their underlying computations becomes a major concern. We discuss the design of a large-scale recommender system handling billions of interactions on a European e-commerce platform and will present the results from our two studies on improving the predictive performance of this system with both algorithmic and systems-related approaches.

Architecture of our recommender system. We will first provide a high-level overview of the architecture of our recommender system, as illustrated in Figure 1. We describe our nearest neighbors-based algorithms underlying our recommendation models and subsequently discuss how to implement the model training in a scalable distributed dataflow system and how to serve low-latency recommendations in response to user-requests. We will share how we handle the bulk update of pre-computed recommendations into a Bigtable database by automatically adjusting the insertion rate to guarantee a low serving latency during ingestion.

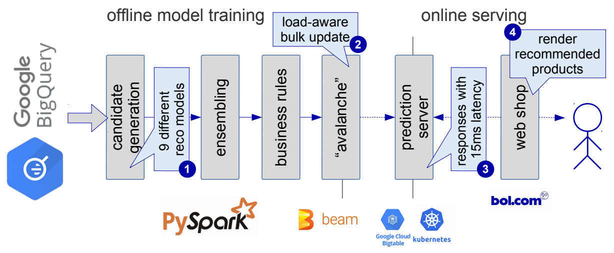

Fig. 1. Architecture overview of the offline training and online serving components of our retail recommendation system, based on Apache Spark and Apache Beam, and backed by cloud infrastructure from the Google Cloud Platform.

Comparison against neural-based algorithms. We explore the benefits of neural-based approaches in comparison to nearest neighbor techniques (such as our system) for our e-commerce scenario as well. The academic setup that is closest to our production use case is session-based recommendation, where the goal is to predict the next item (or the set of next items) that a user will interact with, given the current items of her session. Interestingly, recent academic research [3], [4] indicates that neural-based approach do not outperform classical nearest neighbor approaches in this scenario. We evaluate the predictive performance of all methods on our five data samples. We find that the nearest neighbor approach VS-KNN consistently outperforms all neural-based approaches. The time required to train the neural approaches is three to four orders of magnitude larger than the training time of the neighbor based approaches. Additionally, two of the three neural based methods require much more time for inference than the nearest neighbor methods. These are important properties for a production systems that have to conduct regular retraining of their models, and adhere to strict time constraints for that.

Impact of serving latency on recommendation performance. Motivated by our production setup, our second study focuses on the impact of orthogonal, systems-related improvements. These are in general hard to study for academic researchers without access to real world systems. Work from [5], [6] indicates that a reduction of the response latency has a positive impact on the acceptance rate of a search engine. Therefore we decide to investigate the impact of response latency on our recommender system as well and conduct an A/B test on the live platform with more than 19 million user sessions, which confirms that the latency reduction by 17ms@p90 of the recommender system correlates with a significant increase in important order and revenue based business metrics. We ran this study during an ongoing data center migration, which gave us the unique opportunity to investigate the effect of improving the latency, instead of making it artificially worse as done in previous studies for search engines [7], [6].

Future work. In the future, we will explore how to scale-up well-scoring algorithms for session-based recommendation (in particular VS-KNN) to a full production workload with several billion interactions and low latency. We think that our studies also outlined interesting research directions for improving the suitability of the neural- based recommendation algorithms for production settings. There are ongoing efforts to decrease the training time of neural networks, especially for hyperparameter selection workloads [8], which should include the session-based models in their experiments. We also think that the impact of response latency should be investigated more thoroughly, especially with regards to potential trade-offs with model pruning or precision lowering for neural networks. These techniques usually decrease performance in offline evaluation, but their potential for reducing latency could have a positive impact on their predictive performance in online settings.

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *NeurIPS*, pp. 1097–1105, 2012.
[2] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *ICLR*, 2014.
[3] M. Ludewig, N. Mauro, S. Latifi, and D. Jannach, "Performance comparison of neural and non-neural approaches to session-based recommendation," *RecSys*, pp. 462–466, 2019.
[4] D. Jannach and M. Ludewig, "When recurrent neural networks meet the neighborhood for session-based recommendation," *RecSys*, pp. 306–310, 2017.
[5] R. Kohavi, A. Deng, B. Frasca, T. Walker, Y. Xu, and N. Pohlmann, "Online controlled experiments at large scale," *KDD*, pp. 1168–1176, 2013.
[6] I. Arapakis, X. Bai, and B. B. Cambazoglu, "Impact of response latency on user behavior in web search," *SIGIR*, pp. 103–112, 2014.
[7] B. Hidasi and A. Karatzoglou, "Recurrent neural networks with top-k gains for session-based recommendations," *CIKM*, pp. 843–852, 2018.
[8] S. Nakandala, Y. Zhang, and A. Kumar, "Cerebro: A data system for optimized deep learning model selection," *VLDB*, 2020.