



Improving search engine efficiency through contextual factor selection

Anxiang Zeng^{1,2,3} | Han Yu¹ | Qing Da³ | Yusen Zhan³ | Yang Yu^{4,5} |
Jingren Zhou³ | Chunyan Miao^{1,2}

¹ School of Computer Science and Engineering, Nanyang Technological University (NTU), Singapore

² Alibaba-NTU Singapore Joint Research Institute, Singapore

³ Alibaba Group, Hangzhou, China

⁴ School of Artificial Intelligence, Nanjing University, China

⁵ National Key Laboratory for Novel Software Technology, Nanjing University, China

Correspondence

Qing Da, Alibaba Group, Hangzhou, China.
Email: daqing.dq@alibaba-inc.com

Chunyan Miao, School of Computer Science and Engineering, Nanyang Technological University (NTU), Singapore.
Email: ascymiao@ntu.edu.sg

Abstract

Learning to rank (LTR) is an important artificial intelligence (AI) approach supporting the operation of many search engines. In large-scale search systems, the ranking results are continually improved with the introduction of more factors to be considered by LTR. However, the more factors being considered, the more computation resources required, which in turn, results in increased system response latency. Therefore, removing redundant factors can significantly improve search engine efficiency. In this paper, we report on our experience incorporating our Contextual Factor Selection (CFS) deep reinforcement learning approach into the Taobao e-commerce platform to optimize the selection of factors based on the context of each search query to simultaneously maintain search result quality while significantly reducing latency. Online deployment on Taobao.com demonstrated that CFS is able to reduce average search latency under everyday use scenarios by more than 40% compared to the previous approach with comparable search result quality. Under peak usage during the Single's Day Shopping Festival (November 11th) in 2017, CFS reduced the average search latency by 20% compared to the previous approach.

INTRODUCTION

Search engines often need to retrieve and rank a large set of data items in response to user requests. Take the Taobao.com search engine as an example (Figure 1), it needs to select and rank items from a large inventory in the billions. Under everyday use conditions, it serves around 200 million users per day. This number more than doubles under peak usage during the November 11 Shopping Festival (Figure 2). Two issues are the most important to the success of a search engine: 1) effectiveness, concerning how accurate and suitable the search results in the final ranking list are, and 2) efficiency, concerning how fast the

search engine can respond to the user queries. Improving effectiveness often requires a search algorithm to take more factors concerning the searched items into account, while efficiency depends on the computational overhead involved in the ranking operation. It is a challenge to address both issues in large-scale search engines. To address the high computational cost of involving more factors into the ranking process while maintaining efficiency, a search engine often has to degrade the service level by reducing the number of items recalled and deactivating some unnecessary services (Li et al. 2009).

To improve the trade-off between effectiveness and efficiency in the large-scale e-commerce search applications,



FIGURE 1 An example of the search engine on Taobao.com

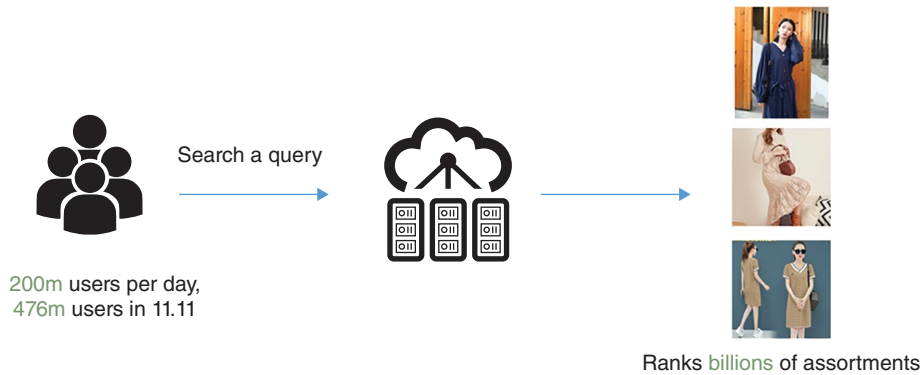


FIGURE 2 Taobao search engine usage

the cascading ranking model has been proposed (Liu et al. 2017). It reduces the number of items in the ranking process according to specific strategies. This inspires us to take a closer look at the factors involved in the ranking algorithm from our Taobao search engine. By analyzing search engine operations, we discovered that many factors involved in the computation are not necessary (Figure 3). On the one hand, there exist redundant factors. On the other hand, conversion rates vary on items under different contexts. For instance, for the users with higher purchasing power or for long-tail queries, computationally less costly factors are enough to produce high quality search results. The above observation shows the possibility of simultaneously improving effectiveness and efficiency by selecting a proper subset of factors under given circumstances, which can be translated into a context-aware combinatorial optimization problem.

In this paper, we report on our experience deploying an innovative model via reinforcement learning based approach—Contextual Factor Selection (CFS)—into Taobao.com to improve search engine efficiency. The algo-

rithm outperforms the previous algorithm used by Taobao since deployment in January 2017. CFS has been shown to be able to reduce average search latency under everyday use scenarios by more than 40% compared to the previous approach with comparable search result quality. Under peak usage during the Single's Day Shopping Festival (November 11th) in 2017, CFS reduced the average search latency by 20% compared to the previous approach.

METHOD

System workflow

In a typical e-commerce search engine, there is a set of all available items in a database. Users with diverse profiles can make a wide range of queries. The ranking problem in e-commerce search engines can be generally defined as a task to generate a permutation function which is a one-to-one mapping from the user-query pairs to the properly ranked selected list of items. The goal is to maximize the

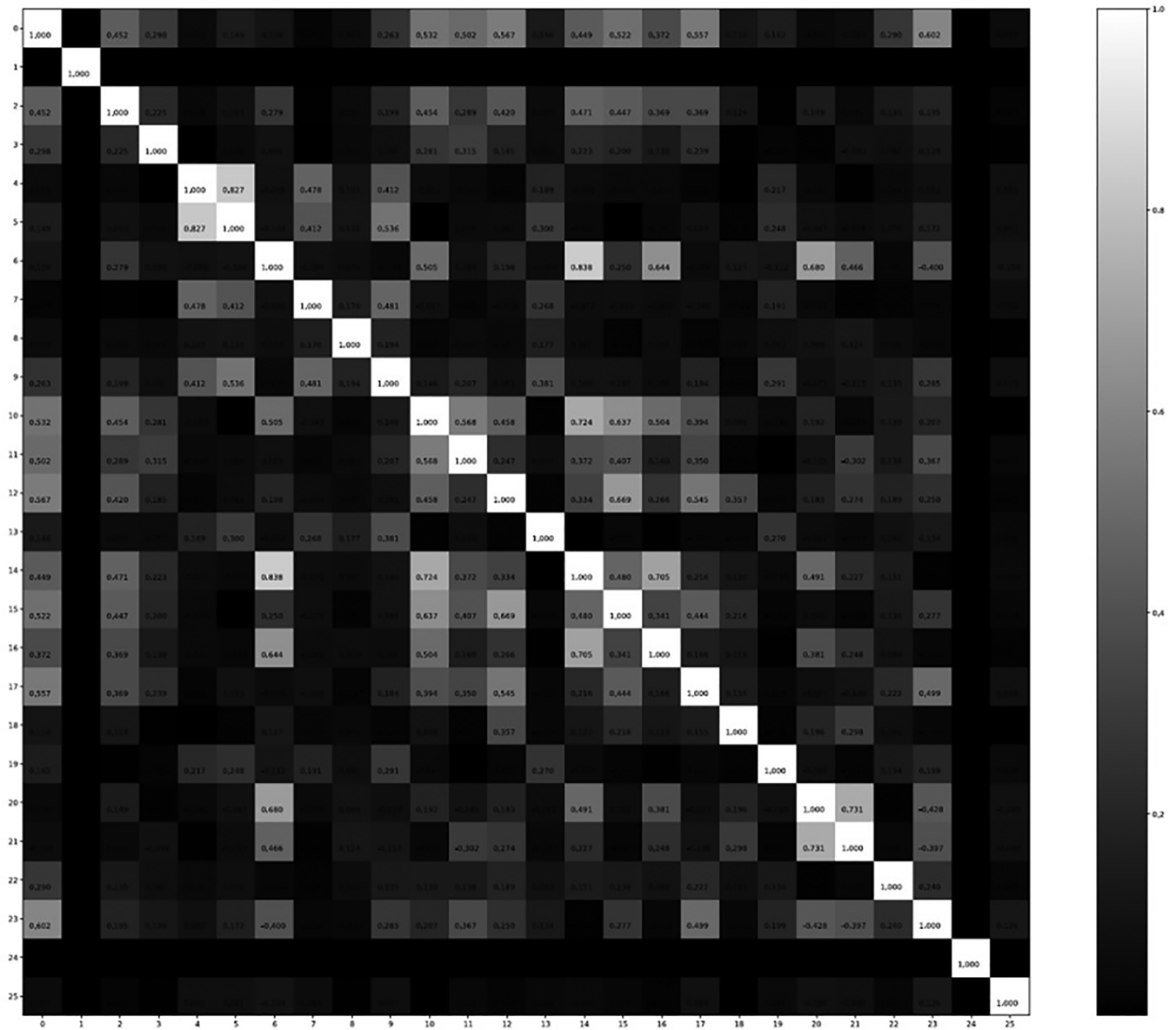


FIGURE 3 Correlation coefficient matrix among some factors considered by the Taobao search engine

probability of purchase under the resulting item ranking list.

The general workflow of search query processing in Taobao.com is illustrated in Figure 4. The Search Planner plays a central role to coordinate this workflow. When a search query is received, it is analyzed by the Query Planner to interpret what the user is looking for. The Indexer then retrieves up to 1 million candidate items, and narrows them down into around 10,000 based on relevance. Finally, the Rank module takes these 10,000 candidate items, and ranks them according to a holistic consideration of relevance be business objectives. The top 50 highly ranked items will be returned to the Search Planner to be displayed to the user. The numbers given here are just to illustrate the scale of operation. They are not the actual numbers.

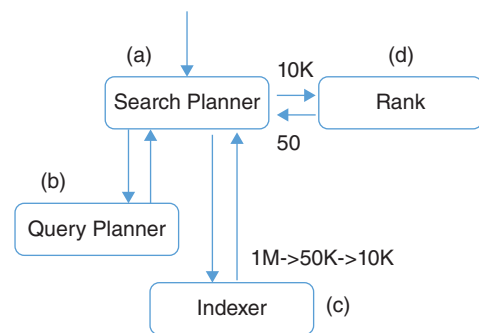


FIGURE 4 The search result ranking workflow in Taobao.com

A wide range of indicators are used to measure business objectives in Taobao.com. Some examples include

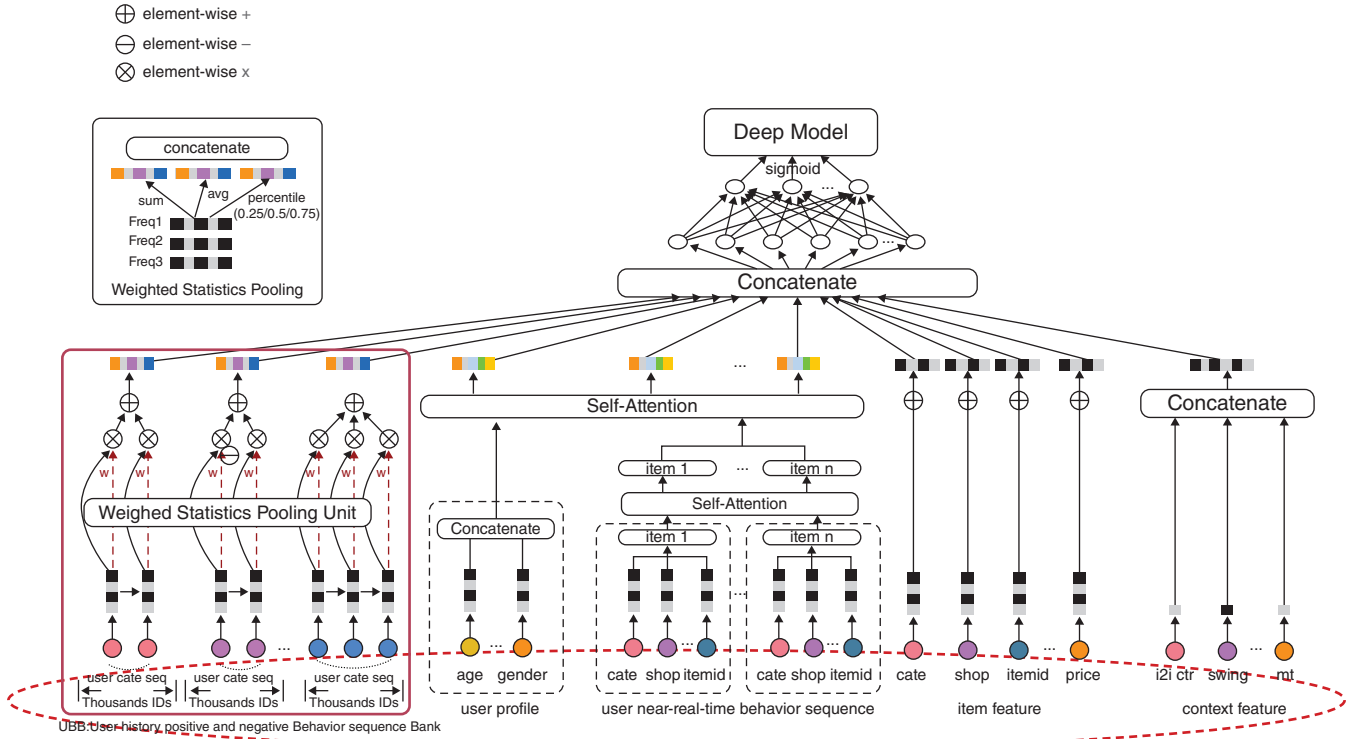


FIGURE 5 An example deep neural network used for click through rate prediction in Taobao.com

conversion rate (CVR), click through rate (CTR), gross merchandize volume (GMV), add-to-cart rate (ATC) and AC diversity which is measured by the average number of product categories viewed by each user. When returning the query results, the search engine needs to optimize a collection of these business performance metrics. The model for predicting each of these metrics when generating the final search result ranking is highly complex. Figure 5 shows the architecture of the deep neural network used by Taobao.com to predict the CTR of a given query result. It can be observed that a large number of factors about user, the item, the context of the search and the user behavior trajectory in the platform are taken into account by the model. This results in high computational overhead and reduces the efficiency of the search engine. If we can dynamically select the most useful subset of factors for a given query, we can improve the performance of the search engine.

Contextual factor selection for item ranking

In this subsection, we describe the proposed Contextual Factor Selection (CFS) approach which can help a search engine achieve both high effectiveness and high efficiency. Each item in the inventory is assigned a factor vector. Each dimension of the factor vector is calculated in an online

manner, which incurs computational costs. From a practical point of view, some of factors are not necessary for ranking. Thus, we defined an indicator function, $\mathbb{I}_S(k) \in \{Skip: 0, Keep: 1\}$, to denote if a factor k is selected into the set of factors CFS will consider when handling a search query. Therefore, the computational cost function can be written as $\sum_k \mathbb{I}_S(k) c_k$, where c_k is the computational cost incurred by calculating factor k . A ranking model which considers all factors, F_Ω , can be regarded as the optimal model in terms of the quality of the recommended item list. Thus, given a user-query pair $\langle u, q \rangle$, the objective can be written as:

$$\min_{S \subseteq \Omega} D(F_\Omega || F_S) + \lambda n \sum_k \mathbb{I}_S(k) c_k$$

where $D(F_\Omega || F_S)$ is the distance between the set of all factors F_Ω and the set of selected factors F_S , which could be any distance between two functions (e.g., Kullback-Leibler divergence [Kullback and Leibler 1951]). The second term is the computational cost involved in calculating the selected factors. $\lambda > 0$ is the trade-off parameter and n denotes the number of items in the given query. By minimizing the objective function, we aim reduce the cost of utilizing factors (i.e. by using low cost factors and reducing the number of factors involved), while approximating the optimal ranking function performance as closely as possible.

However, this objective function is intractable even for a single search request as it is an *NP-hard* problem (Davis, Mallat, and Avellaneda 1997; Natarajan 1995). On top of this, we still need to perform contextual factor selection. To overcome this challenge, we generalize the solution of at the contextual level. That is, we do not directly search for the optimal subset. In this way, CFS reduces the solution space from the original multiple optimal subset selection problems based on the assumption that similar user-query pairs should have similar optimal subset structures. Thus, our goal becomes to search for the general parameter vector to minimize the objective function over all the $\langle u, q \rangle$ requests.

To solve the optimization problem defined in the objective function, CFS learns the indicator function for each factor based on given user-query types using reinforcement learning (RL) (Sutton and Barto 1998). We follow a variant of RL called policy gradients (Bello et al. 2016; Kober and Peters 2011). It is a class of RL useful for solving complex robotic problem. The key idea, in our case, is to transform the assignment of each element in the indicator vector as a sequential decision-making problem. It is possible to learn a selected factor, by a reinforcement learning policy, instead of approximating the indicator vector directly.

To make the action space more manageable, we introduce a fixed factor sequence so that a policy can determine the utility of the factors in a sequential manner. For each user-query pair, the indicator function vector entries can be determined in p steps, where in the k -th step ($1 \leq k \leq p$), we need to decide whether this factor shall be included in the ranking function or not, which is the action taken at step k . After the factors are selected, we can directly calculate the loss to evaluate the effectiveness of selection actions, which can be further used to define the total reward of the actions during the episode. The key issues here are how to design the states, the rewards and the optimization method for this reinforcement learning problem.

There are two issues that make learning the optimal policy for our reinforcement learning problem challenging. Firstly, the reward is sparse over states. This is known as the *sparse feedback problem* (Kulkarni et al. 2016). Secondly, the rewards are distributed widely in a continuous space, making the model convergence hard. To address these challenges, we adjust the state and reward designs based on shaping techniques (Ng, Harada, and Russell 1999). Instead of just using the user-query pair to describe the states, each state also includes the decisions made during the previous episode. For the effectiveness part, we issue a large constant penalty if the ranking loss under the selected set of factors exceeds a pre-defined threshold. For the reward, we decompose it into the part reflecting effectiveness and another part reflecting the efficiency. For the efficiency part, we add a penalty whenever a new factor is

included. With this design, we enable the critic module to distinguish good and bad ranking results more easily and avoid generating poor rankings.

After designing our state and reward space, we leverage the well-known policy gradient method with actor-critic models as described in (Mnih et al. 2016) to obtain solutions. Note that the difficulty of the original optimization problem does not decrease with the introduction of reinforcement learning techniques. The RL-based approach here acts as a solver whose solution space contains the optimal solutions, and provides an efficient search path to the optimal solutions through trial-and-error. More details of the proposed algorithm can be found in (Zeng et al. 2020).

APPLICATION DEVELOPMENT AND DEPLOYMENT

Before the actual deployment of the technique, we evaluated the proposed approach on a dataset from Taobao on an through comparison with the norm elimination method, the l_1 -based feature selection method, the tree-based feature selection method and the F-test feature selection method in an offline setting. This is to ascertain if we shall move forward to deploy RankCFS online.

For l_1 -based feature selection, tree-based feature selection and F-test feature selection, we adopt their implementations in scikit-learn (Pedregosa et al. 2011). RankCFS is implemented with Tensorflow (Abadi et al. 2016). For the ranking model in this offline evaluation, we select one of the linear ranking models from Taobao.com and treat it as a black box so that only the inputs and the outputs of the ranking model are utilized during the evaluation. We select the default ExtraTreeRegressor in scikit-learn package as the tree model. The actor and critic are implemented as two deep neural networks (DNNs) with three fully connected layers each. The DNN structure of the actor is $266 \times 128 \times 20$, and that of the critic is $266 \times 128 \times 128 \times 1$. We adopt ReLU as the activation functions for the hidden layers, and use Adam as the optimizer. The learning rates of actor and critic are set as 0.0001 and 0.001, respectively.

We use a dataset with 100,000 samples.¹ We train the l_1 -based, Tree-based and F-test approaches on 50,000 samples and then test them on the remaining 50,000 samples. It is not necessary to train Norm Elimination method since it only removes those factors whose absolute values of weights are less than a positive constant. For the l_1 -based, Tree-based and F-test methods, feature selection is performed after the training. That is, we use a fixed feature selection policy during the testing stage. Since our method considers the computational costs of factors, the computational cost vector is obtained from the online operational environment of Taobao.com. We compare

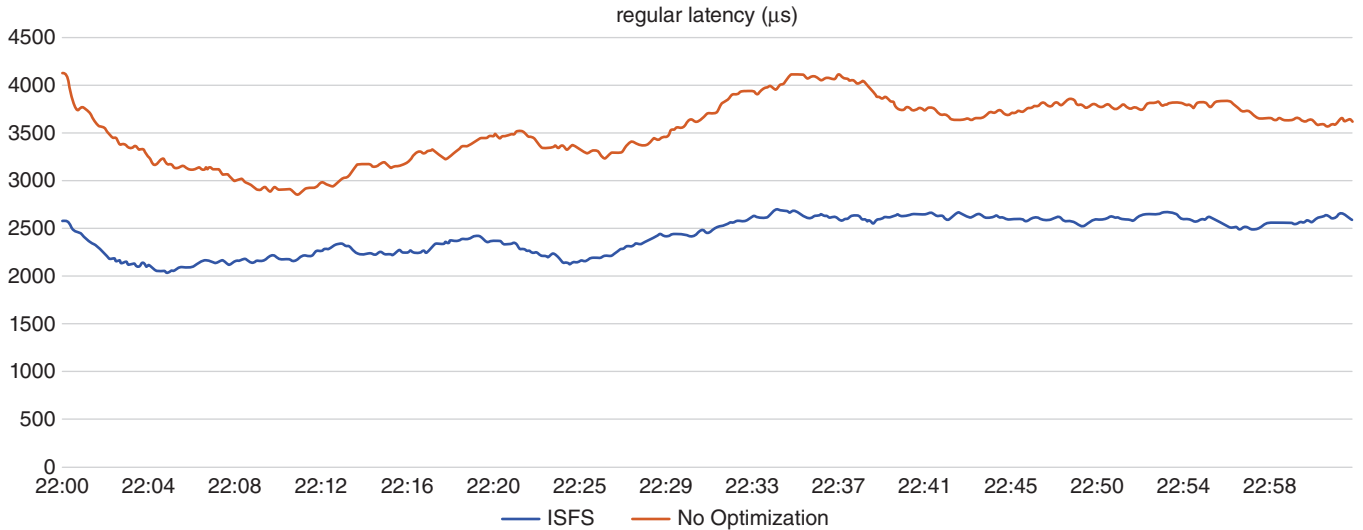


FIGURE 6 Comparison of search latency under normal usage conditions

TABLE 1 Offline evaluation comparing different approaches

	Averaged Pairwise Loss (APL)	Averaged Factor Usage (AFU)	Weighted
Norm Elimination	0.35	7.41	78.84
l_1 -based Feature Selection	0.31	8.00	71.16
F-Test Feature Selection	0.40	11.00	84.63
Tree-based Feature Selection	0.30	7.00	63.00
RankCFS	0.21	7.01	51.06

the approaches on 5,000 page views and each page view contains 10 items so that there are 50,000 testing examples. Then, we evaluate the averaged pairwise loss and factor usage over page views.

The offline evaluation results show that the proposed RankCFS approach successfully explored the function space and found high quality approximations to the optimal ranking function. We summarized the complete experimental results in Table 1 under different parameter settings, which shows that RankCFS significantly outperforms the other methods in terms of APL and WFU. In terms of AFU, it achieves comparable performance with the Tree-based Feature Selection approach. Based on these considerations, RankCFS was deployed in Taobao.com to replace the previous search algorithm.

IMPACT ON THE TAOBAO PLATFORM

CFS has been deployed on the Taobao platform since January 2017. To demonstrate the payoff generated, we

conduct the experiment in the real-world large-scale operational environment of Taobao.com with a standard A/B test setting. The training was conducted with more than 10^9 training samples on a distributed streaming system in an on-line learning fashion.

In our A/B test, we selected 6% of the users from Taobao.com at random to be served by CFS. We adopt search latency as the evaluation metric. Figure 6 shows the performance of CFS and the previous search method in Taobao under normal usage conditions. It can be observed that CFS reduces the average search latency by approximately 40% compared to the control group.

Alibaba Singles' Day shopping festival on November 11 of every year is one of the biggest shopping extravaganzas around the world. The level of activities involved can be thought of as similar to Black Friday. In 2017, by the end of, sales hit a new record of US\$25.3 billion, more than 40% higher than sales on the same day in 2016. It attracted over hundreds of millions of users from more than 200 different countries. The infrastructure handled 0.325 million orders per second during peak usage conditions.² The proposed CFS search approach played a crucial role in this event.

On November 11th, the search traffic burden of the e-commerce search engine abruptly increases by multiple times compared to a regular day. On the one hand, the e-commerce search engine faces the high traffic challenge, which might lead to system degradation. On the other hand, it is still crucial to provide accurate search results.

During this event, the proposed approach was deployed on top of the CLOES approach (Liu et al. 2017) in order to perform optimization at the search engine system level. CLOES enables the system to optimize the number of items

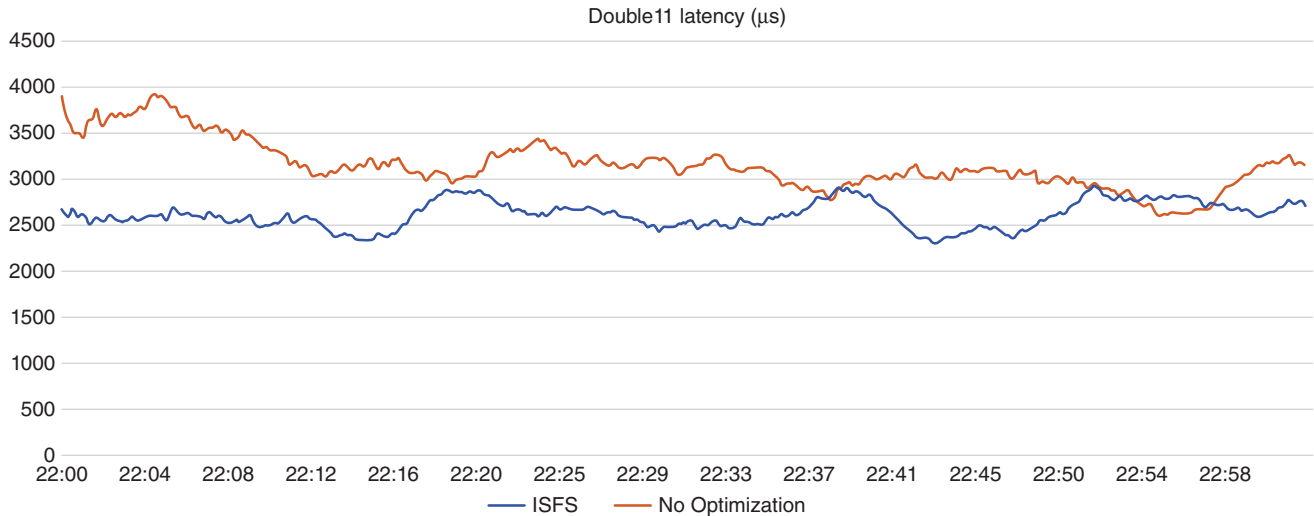


FIGURE 7 Comparison of search latency under peak usage conditions

in the ranking process in a cascading manner. CFS focuses on optimizing the set of selected ranking factors during the ranking process. Thus, both approaches can be applied on the search engine. We use previous the performance under CLOES without CFS as the baseline for comparison in Figure 7, which depicts the average latency change during the course of the experiments under the Single's Day festival peak usage conditions. It can be observed that CFS reduced search latency by more than 20%.

CONCLUSIONS AND FUTURE WORK

In this paper, we report on our experience deploying contextual factor selection to improve search latency performance of a real-world large-scale e-commerce search system – Taobao.com. We leveraged insight into the different relevance and computational cost of the factors in relation to user-query pairs, and framed the problem as a reinforcement learning problem. We then proposed an efficient approach to select the optimal set of factors to be considered based on the context of each search query in order to enhance efficiency. The algorithm has been deployed in Taobao which is China's largest e-commerce platform. Actual usage data demonstrate that our method is a practical solution for real-world large-scale e-commerce search systems which significantly outperformed the previous algorithm in the platform.

In future research, we will enhance the optimization approach to incorporate responsible AI practices such as the consideration for fairness (Yu et al. 2018), persuasiveness (Liu et al. 2019) and privacy preservation (Kairouz et al. 2019; Yang et al. 2019) to further enhance user experience.

ACKNOWLEDGMENTS

This research is supported, in part, by Alibaba Group through Alibaba Innovative Research (AIR) Program and Alibaba-NTU Singapore Joint Research Institute (JRI), Nanyang Technological University, Singapore; Joint SDU-NTU Centre for Artificial Intelligence Research (C-FAIR) (NSC-2019-011); the National Research Foundation, Singapore under its AI Singapore Programme (AISG Award No: AISG2-RP-2020-019); the RIE 2020 Advanced Manufacturing and Engineering (AME) Programmatic Fund (No. A20G8b0102), Singapore; the National Research Foundation, Singapore, Prime Minister's Office under its NRF Investigatorship Programme (NRFI Award No: NRF-NRFI05-2019-0002); and Nanyang Technological University, Nanyang Assistant Professorship (NAP). Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore.

ENDNOTES

¹ 10,000 page views and 10 items in each page view.

² <https://techcrunch.com/2017/11/11/alibaba-smashes-its-singles-day-record/>.

REFERENCES

- Abadi, M., P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, et al. 2016. "Tensorflow: A system for Large-Scale Machine Learning." In *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI'16)*, pp. 265–83.
- Bello, I., H. Pham, Q. V. Le, M. Norouzi, and S. Bengio. 2016. "Neural Combinatorial Optimization with Reinforcement Learning." In *CoRR*, arXiv:1611.09940.
- Benbouzid, D., R. Busa-Fekete, and B. Kegl. 2012. "Fast Classification Using Sparse Decision Dags." In *Proceedings of the 29th International Conference on Machine Learning (ICML'12)*, pp. 747–54.

- Bourdev, L., and J. Brandt. 2005. "Robust Object Detection Via Soft Cascade." In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, pp. 236–43.
- Davis, G., S. Mallat, and M. Avellaneda. 1997. "Adaptive Greedy Approximations." *Constructive Approximation* 13(1): 57–98.
- Gao, D., Y. Liu, A. Huang, C. Ju, H. Yu, and Q. Yang. 2019. "Privacy-Preserving Heterogeneous Federated Transfer Learning." In *Proceedings of the 2019 IEEE International Conference on Big Data (IEEE BigData'19)*, pp. 2552–9.
- Geng, X., T.-Y. Liu, T. Qin, and H. Li. 2007. "Feature Selection for Ranking." In *Proceedings of the 30th Annual International ACM SIGIR Conference (SIGIR'07)*, pp. 407–14.
- Guyon, I., and A. Elisseeff. 2003. "An Introduction to Variable and Feature Selection." *Journal of Machine Learning Research* 3: 1157–82.
- Kairouz, P., H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, et al. 2019. "Advances and Open Problems in Federated Learning." *CoRR arXiv:1912.04977*.
- Kira, K., and L. A. Rendell. 1992. "The Feature Selection Problem: Traditional Methods and a New Algorithm." In *Proceedings of the 10th National Conference on Artificial Intelligence (AAAI-92)*, pp. 129–34.
- Kober, J., and J. Peters. 2011. "Policy Search for Motor Primitives in Robotics." *Machine Learning* 1(84): 171–203.
- Kulkarni, T. D., K. Narasimhan, A. Saeedi, and J. Tenenbaum. 2016. "Hierarchical Deep Reinforcement Learning: Integrating Temporal Abstraction and Intrinsic Motivation." In *Proceedings of the 30th Annual Conference on Neural Information Processing Systems (NeurIPS'16)*, pp. 3675–83.
- Kullback, S., and R. A. Leibler. 1951. "On Information and Sufficiency." *The Annals of Mathematical Statistics* 22(1): 79–86.
- Li, B., H. Yu, Z. Shen, and C. Miao. 2009. "Evolutionary Organizational Search." In *Proceedings of the 8th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'09)*, pp. 1329–30.
- Li, H. 2011. *Learning to Rank for Information Retrieval and Natural Language Processing*. Morgan & Claypool Publishers.
- Liu, L.-P., Y. Yu, Y. Jiang, and Z.-H. Tefe Zhou. 2008. "A time-efficient approach to feature extraction." In *Proceedings of the 8th IEEE International Conference on Data Mining (ICDM'08)*, pp. 423–32.
- Liu, S., F. Xiao, W. Ou, and L. Si. 2017. "Cascade Ranking for Operational E-Commerce Search." In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'17)*, pp. 1557–65.
- Liu, C., Y. Dong, H. Yu, Z. Shen, Z. Gao, P. Wang, C. Zhang, P. Ren, X. Xie, L. Cui, and C. Miao. 2019. "Generating Persuasive Visual Storylines for Promotional Videos." In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management (CIKM'19)*, pp. 901–10.
- Mnih, V., A. P. Badia, M. Mirza, A. Graves, T. Harley, T. P. Lillicrap, D. Silver, and K. Kavukcuoglu. 2016. "Asynchronous Methods for Deep Reinforcement Learning." In *Proceedings of the 33rd International Conference on Machine Learning (ICML'16)*, pp. 1928–37.
- Natarajan, B. K. 1995. "Sparse Approximate Solutions to Linear Systems." *SIAM Journal on Computing* 24(2): 227–34.
- Ng, A. Y., D. Harada, and S. J. Russell. 1999. "Policy Invariance under Reward Transformations: Theory and Application to Reward Shaping." In *Proceedings of the 16th International Conference on Machine Learning (ICML'99)*, pp. 278–87.
- Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, et al. 2011. "Scikitlearn: Machine Learning in Python." *Journal of Machine Learning Research* 12: 2825–30.
- Schneiderman, H. 2004. "Feature-centric Evaluation for Efficient Cascaded Object Detection." In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'04)*, <https://doi.org/10.1109/CVPR.2004.1315141>.
- Sutton, R. S., and A. G. Barto. 1998. *Introduction to Reinforcement Learning*. Cambridge, MA, USA: MIT Press.
- Vinyals, O., M. Fortunato, and N. Jaitly. 2015. "Pointer Networks." In *Proceedings of the 29th Annual Conference on Neural Information Processing Systems (NeurIPS'15)*, pp. 2692–700.
- Viola, P., and M. Jones. 2003. "Rapid Object Detection Using a Boosted Cascade of Simple Features." In *Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'03)*, pp. I-511–I-518.
- Wang, L., J. Lin, and D. Metzler. 2010. "Learning to Efficiently Rank." In *Proceedings of the 33rd Annual International ACM SIGIR Conference (SIGIR'10)*, pp. 138–45.
- Wang, L., D. Metzler, and J. Lin. 2010. "Ranking under Temporal Constraints." In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management (CIKM'10)*, pp. 79–88.
- Yang, Q., Y. Liu, Y. Cheng, Y. Kang, T. Chen, and H. Yu. 2019. *Federated Learning*. San Rafael, CA, USA: Morgan & Claypool Publishers.
- Yu, H., Z. Shen, C. Miao, C. Leung, V. R. Lesser, and Q. Yang. 2018. "Building Ethics into Artificial Intelligence." In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI'18)*, pp. 5527–33.
- Zeng, A., H. Yu, Q. Da, Y. Zan, and C. Miao. 2020. "Accelerating Ranking in E-Commerce Search Engines through Contextual Factor Selection." In *Proceedings of the 32nd AAAI Conference on Innovative Applications of Artificial Intelligence (IAAI-20)*, pp. 13212–9.

AUTHOR BIOGRAPHIES

Anxiang Zeng is a Senior Staff Algorithm Engineer & Director of Alibaba. He is Head of the Globalization Search and Recommendation of Alibaba. He is pursuing his PhD in Nanyang Technological University, Singapore. He has been working in the search and recommendation field for more than ten years. His research focuses on search, recommendation and reinforcement learning. He has published more than ten research papers in leading international conferences and journals.

Han Yu is a Nanyang Assistant Professor (NAP) at the School of Computer Science and Engineering (SCSE), Nanyang Technological University (NTU), Singapore. He held the prestigious Lee Kuan Yew Post-Doctoral Fellowship (LKY PDF) from 2015 to 2018. He obtained his PhD from the School of Computer Science and Engineering, NTU. His research focuses on federated learn-

ing, and algorithmic fairness. He has published over 150 research papers in book chapters, leading international conferences and journals. His research work has won multiple awards from conferences and journals.

Qing Da received the BSc and MSc degrees in computer science from Nanjing University, China, in 2010 and 2013, respectively. He is currently a senior algorithm expert in the search algorithm team of Department of International AI at Alibaba Group. His research interests include reinforcement learning and applications of machine learning.

Yusen Zhan is currently a PhD student in Washington State University, USA, majoring in Computer Science. His research focuses on reinforcement learning and transfer learning. He was involved in this work when he was working in Alibaba.

Yang Yu received the PhD degree in Computer Science from Nanjing University in 2011, and then joined the LAMDA Group in the Department of Computer Science and Technology of Nanjing University as an Assistant Researcher from 2011, and as an Associate Professor from 2014. He joined the School of Artificial Intelligence of Nanjing University as a Professor from 2019. Prof Yu's research interest is in machine learning, a sub-field of artificial intelligence. Currently, Prof Yu is working on reinforcement learning in various aspects, including optimization, representation, transfer, etc. Prof Yu was a recipient of the National Outstanding Doctoral Dissertation Award, the China Computer Federation Outstanding Doctoral Dissertation Award, the PAKDD'08 Best Paper Award, the GECCO'11 Best Paper (Theory Track) and the Microsoft Research Asia Collaborative Research

Award. He is an associate editor for *Frontiers of Computer Science* and an area chair of ACML'17, IJCAI'18, and ICPR'18.

Jingren Zhou is a Vice President in the Alibaba Group. He manages several core technical divisions at Alibaba to drive data intelligent infrastructure and applications in e-commerce and cloud business, including big data and AI infrastructure, search & recommendation, and advertising platform. Before joining Alibaba, he used to be a Partner Development Manager at Microsoft and a researcher at Microsoft Research. His research focuses on cloud computing, database systems, machine learning, search, and advertising. He received my PhD in Computer Science at Columbia University and BS at University of Science and Technology of China. He is a fellow of the IEEE.

Chunyan Miao is the President's Chair Professor and Chair of the School of Computer Science and Engineering (SCSE), Nanyang Technological University (NTU), Singapore. She received her BSc Degree from Shandong University, and her M.Eng. and PhD from NTU. She is the Founding Director of the LILY Research Centre and the Alibaba-NTU Joint Research Institute. Her research focuses on humanized artificial intelligence.

How to cite this article: Zeng, A., Y. Han, D. Qing, Z. Yusen, Y. Yang, Z. Jingren, and M. Chunyan. 2021. "Improving Search Engine Efficiency Through Contextual Factor Selection." *AI Magazine* 42: 50–58.
<https://doi.org/10.1609/aaai.12005>.



Visit AAAI on LinkedIn™

Did you know that AAAI is on *LinkedIn™*?
 If you are a current AAAI member, please join us and connect with the AAAI community!

We welcome your feedback at info21@aaai.org