Supplementary Material for Learning to Select for a Predefined Ranking

Aleksei Ustimenko [*,1,2,3] **Aleksandr Vorobev** [*,1] **Gleb Gusev** [1,4] **Pavel Serdyukov** [1]

1. Analysis of Optimal Selection Algorithm

Now we closely study how the optimal decision for a given context depends on the vector $((x_i^0, r_i), i = 1..n)$ in the case of an objective measure Q satisfying the following requirements:

R1. For any lists X, Y and items d, d', we have

$$Q(X \frown (d, d') \frown Y) \geq Q(X \frown (d', d) \frown Y) \iff \iff r(d) \geq r(d'), \quad (1)$$

where "$\frown$" means concatenation. That is, permutation of two neighboring items in a list changes its relevance accordingly to the order of relevances of these items.

R2. If $Q(X \frown Y) < Q(X \frown Y^F)$ for some lists X, Y and selection algorithm F, then $Q(X^{F'} \frown Y) < Q(X^{F'} \frown Y^F)$ for any selection algorithm F'. That is, the fewer items are located above a given tail of a list, the more useful applying a given selection algorithm to this tail is. In the most practical case of an additive measure $Q(d_1, \dots, d_n) = \sum_{i=1}^n w_i r_i, w_i > 0$, it is equivalent to non-strict logarithmic convexity of w_i as a function of the position i, i.e., $w_i/w_{i+1} \geq w_{i+1}/w_{i+2}\ \forall i \in \mathbb{N}$, and covers cases of DCG and RBP with $w_i = \frac{1}{\log_2(i+1)}$ and $w_i = (1-p)p^{i-1}$ respectively.

Under these requirements, the following proposition is valid.

Proposition 1 *The optimal selection for a list $L = (d_1, \dots, d_n)$ has the form $F(d_i) = \mathbb{1}\{r_i > t(i)\}$, where $t(\cdot)$ is a* ***non-increasing*** *function.*

Proof 1 *For a fixed position i, obviously, the optimal decision $F_{opt}(d_i)$ on the item d_i is a threshold-based function of its relevance r_i, i.e., $F(d_i) = \mathbb{1}\{r_i > t(i)\}$ for some function $t(\cdot)$. Assume it is not a non-increasing one. Then, there exist such lists A, B, C and items d_1, d_2 with $r_1 \leq r_2$, and a list $L = A \frown (d_1) \frown B \frown (d_2) \frown C$ such, that the optimal selection for the list L provides the list $L^* = A' \frown (d_1) \frown B' \frown C'$, where the item d_2 is not selected and X' denotes the list X after selection. Assume w.l.o.g. that d_1 is the lowest item in L^* located above d_2 in L and having relevance $r \leq r_2$. Then, for each item $d \in B'$ its relevance r satisfies $r > r_2 \geq r_1$. It allows us to apply the property R1 to d_1 and each $d \in B'$ from top to bottom consequently and yield*

$$Q(L^*) = Q(A' \frown (d_1) \frown B' \frown C') \leq \leq Q(A' \frown B' \frown (d_1) \frown C') \quad (2)$$

Further,

$$Q(A' \frown B' \frown (d_1) \frown C') < Q(A' \frown B' \frown (d_2) \frown C') \leq \leq Q(A' \frown B' \frown C'), \quad (3)$$

where the first inequality is the consequence of increasing monotonicity of Q w.r.t. the relevance of each item and the second follows from the inequality

$$Q(A' \frown (d_1) \frown B' \frown (d_2) \frown C') < L^* = = Q(A' \frown (d_1) \frown B' \frown C')$$

and the property R2 applied to $X = A' \frown (d_1) \frown B'$, $Y = (d_2) \frown C'$, $X^{F'} = A'$, $Y^F = C'$. Combining the inequalities (2) and (3), we obtain

$$Q(L^*) < Q(A' \frown B' \frown C')$$

, what contradicts the optimality of the selected list L^*, since $A' \frown B' \frown C'$ is obtained from L by selection.

Why position i is generally unknown Note that the position i of an item is not commonly known at search engines of the first stage, since it is a global feature of the item. Similarly to selection, even its calculation on the basis of a considerable number of items is highly undesirable from the point of view of efficiency. Of course, we anyway should choose top-m items (for some fixed m) by the dedicated

[*]Equal contribution [1]Yandex, Moscow, Russia [2]Skoltech University, Moscow, Russia [3]Faculty of Computer Science, Higher School of Economics, Moscow, Russia [4]Department of Innovation and High Technology, Moscow Institute of Physics and Technology, Dolgoprudny, Russia. Correspondence to: Aleksei Ustimenko <austimenko@yandex-team.ru>, Aleksandr Vorobev <alvorobyev88@gmail.com>, Gleb Gusev <gleb57@gmail.com>, Pavel Serdyukov <pavser@yandex-team.ru>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019. Copyright 2019 by the author(s).

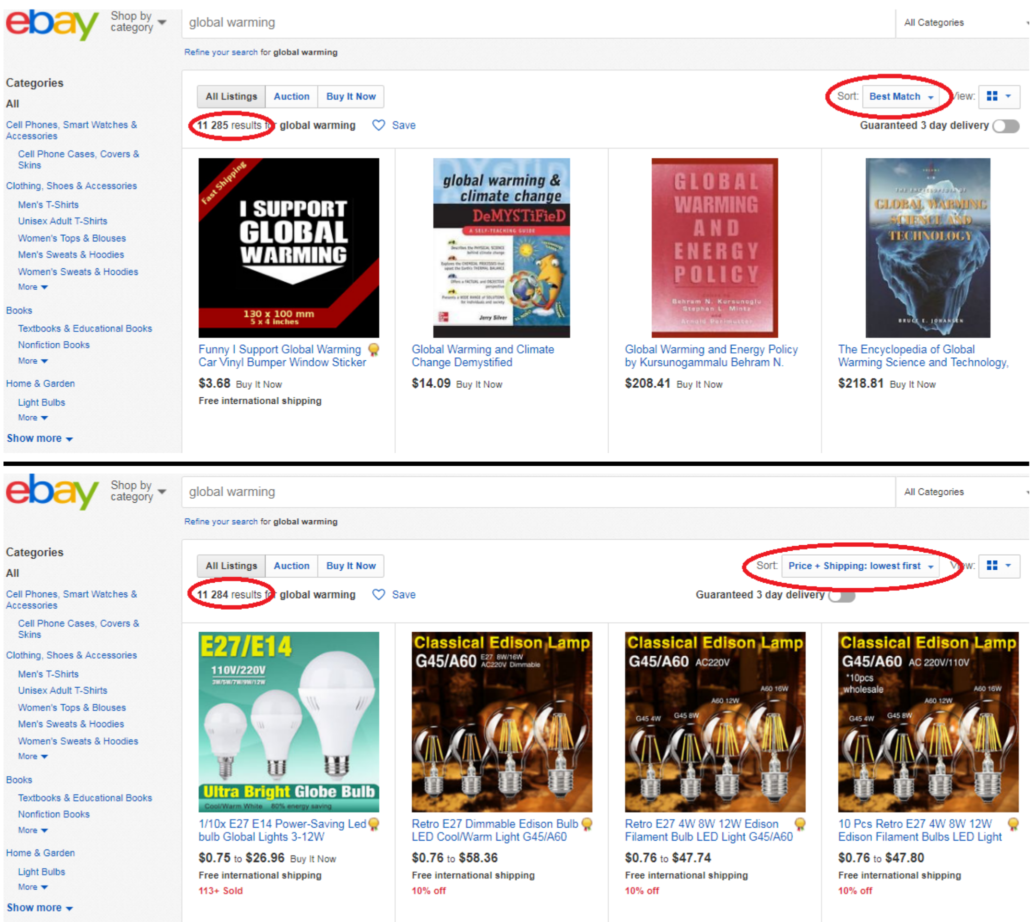

Figure 1. Result pages for the query "global warming" submitted to `www.ebay.com` for ordering by relevance (top) and price (bottom)

attribute to send them to the meta-search level. However, choosing tom-m is the partial sorting problem, which could be solved in time $O(n'+m \cdot \log n')$ (Martınez, 2004), where n' is the number of candidate items considered by the search engine. This complexity is significantly lower than one of the sorting procedure, $n' \cdot \log n'$.

2. Proof of Theorem 1

Proof 2 *1) In the case of logarithmically convex w_i, consider the following example:*
$L = (d_1, d_2, d_3)$, $r_1 = r_3((w_1 - \frac{w_2}{w_1} - \epsilon)$,
$r_2 = r_3(\frac{w_2 - w_3}{w_2} + \epsilon)$ *and* $\epsilon = \frac{w_1 w_3 - w_2^2}{2(2w_1 - w_2)w_2}$.
Let evaluate the lists that are candidates to be the best selected ones:
$Q(d_1, d_2, d_3) = r_3(w_1 - \epsilon(w_1 - w_2))$
$Q(d_1, d_3) = r_3(w_1 - \epsilon w_1)$
$Q(d_2, d_3) = r_3(w_1 - \frac{w_1 w_3}{w_2} + w_2 + \epsilon w_1)$
$Q(d_3) = r_3 w_1$

It is easy to check that

$$Q(d_3) > Q(d_1, d_2, d_3) > Q(d_1, d_3) \text{ and } Q(d_2, d_3) \quad (4)$$

Now, we note that both Q_{smooth} and Q^{low}_{smooth} are monotone functions of each model prediction $f(x_j)$. Indeed, the sign of both derivatives $\frac{\partial Q^{low}_{\text{smooth}}}{\partial f(x_j)}$ (see Equation 6 in the main text) and $\frac{\partial Q_{\text{smooth}}}{\partial f(x_j)}$ (Equation 5 in the main text) does not depend on $f(x_j)$, what is a sufficient condition for monotonicity. For the latter derivative, it is easy to see that by presenting the derivative as

$$\frac{\partial Q_{\text{smooth}}}{\partial f(x_j)}(F, L) =$$
$$= \sigma'|_{f(x_i)}(\mathbb{E}_F\, Q(L^F | p_j = 1) - \mathbb{E}_F\, Q(L^F | p_j = 0))$$

Next, according to inequalities (4), excluding any one item cannot improve the list L in terms of Q, which coincides with Q_{smooth} and Q^{low}_{smooth} on deterministic selection algorithms (i.e., ones with $p_i \in \{0, 1\}$). In combination with *monotonicity of these metrics, it means that L is a local extremum point for G_1 and G_2, while $L^* = (d_3)$ is the global extremum.*

In the case of logarithmically concave w_i, the following example can be checked by analogy with the first case:
$L = (d_1, d_2, d_3)$, $r_1 = r_3((w_1 - \frac{w_2}{w_1} + \epsilon)$,
$r_2 = r_3(\frac{w_2 - w_3}{w_2} - \epsilon)$ *and* $\epsilon = \frac{w_2^2 - w_1 w_3}{4 w_1 w_2}$.

2) Assume the contrary, there is a point L_1 of local but not global maximum. According to monotonicity proved above this point presents a deterministic selection algorithm. Consider the last item d_j with the decision p_j on it different from the optimal decision p_j^{opt}. Consider w.l.o.g. $p_j^{opt} = 0$. Then, the list L_1 and the optimal selected list can be presented as $L_1 = (A \frown (d_j) \frown B)$ and $L^* = (A' \frown B)$ for some lists A and B. Since, due to a geometric form of weights w_i, differences $Q(A \frown (d_j) \frown B) - Q(A \frown B)$ and $Q(A' \frown (d_j) \frown B) - Q(A' \frown B)$ have the same sign, we have $Q(L_1) = Q(A \frown (d_j) \frown B) < Q(A \frown B)$, what contradicts the local optimality of L_1.

3. Features and Their Importance

In our experiments, we utilize all the features (in all methods including baselines) used in the production ranker of the search engine under study, besides additional features described in Section 5.1. These features include *price features* (offer price, price aggregated over different categories the offered product belongs to), *features of query-offer pairs* (e.g., text-based features indicating relevance of the offer to the query, click-based statistics), *offer features* (e.g., features describing the offered product, the shop, shipping, payment).

In order to analyze importance of different groups of features described in Section 5 of the main paper, in LSO task and compare it with their importance for the prediction of the production relevance, we sum the values of feature importance over all the features in each group. The feature importance is defined as the sum of the reductions in the loss functions over all splits based on a given feature (Diaz et al. (Diaz et al., 2010)). The resulting statistics for $OFP + PG$ and for the production relevance prediction is presented in Table 1. Expectedly, price-based features (both production ones and $AvPricePred$) are much more important for LSO. Besides, query features remain important for LSO, since they contain the signal about relevance and price distribution over offers w.r.t. the query.

Besides, for $OFP+PG$, we evaluate the influence on DCG-RR of features $AvPricePred$ and $AvRelPred$ constructed specifically for the LSO task. The relative changes w.r.t. the quality of $WeakCutoff$ are presented in Table-2. These features remarkably increase the quality of the selection and confirm our conclusions about the importance of features encoding distributions of the dedicated attribute and the relevance over the original list of items.

Table 1. Feature importances

Feature groups	$OFP + PG$	Relevance prediction
$AvPricePred$	12.0	1.0
$AvRelPred$	3.5	2.8
Price	31.6	3.3
Query	19.0	20.7
QueryOffer+Offer	34.5	71.4

Table 2. Influence on DCG-RR, relative Δ to WeakCutoff, %

Features (for $OFP + PG$)	DCG-RR
$AvPricePred + AvRelPred$	4.33
$AvPricePred$	3.27
$AvRelPred$	3.25
-	2.55

4. CatBoost Parameters

For training each model, we use the following setting of CatBoost parameters: $depth = 6$, $leaf\_estimation\_iterations = 100$, $border\_count = 32$, $random\_strength = 0$, $bagging\_temperature = 0$, $iterations = 1000$, $loss\_function = UserQuerywiseMetric$. The best iteration (upper limited by 1000) and the $learning\_rate$ parameter were fitted in combination with specific hyperparameters of the algorithms by maximization of DCG-RR, the rest of the CatBoost parameters have default values. All the features are treated as numerical.

5. Relevance Labels

Production relevance of an item is its score used by the chosen system for ranking without ordering by price. It is calculated as a combination of a predicted relevance and a machine-learned click-based component normalized in the interval [0,1]. The first reason for combining is the well-known fact that clicks provide a strong signal about item relevance (Joachims et al., 2017), what motivates combining relevance labels with clicks as targets in LTR (Svore et al., 2011). The second reason is that the number of clicks directly influence the system revenue in Cost Per Mille (CPM) payment scheme (Fain & Pedersen, 2006). In this scheme, the shop pays to the system a fixed price per each thousand of exposures of its offer, and an exposure occurs when the user clicks on the corresponding item (offer snippet) on the result page.

Table 3. Descriptive statistics for selection algorithms

Approach	# of hard items per list, N_{hard}	# of excluded per list	threshold ($t^{\mathrm{rel}}_{\mathrm{const}}$, t^{rel} or $t^{\mathrm{prob}}_{\mathrm{const}}$)
$ConstCutoff$	-	3.4	0.058
$QueryCutoff$	-	41.3	[0.03, 0.67]
OFP	16.9	68.4	0.547
LBO	62.6	0	0.033
PG	58.9	0.2	0.703
$OFP + LBO$	6.1	73.1	0.552
$OFP + PG$	10.6	73.1	0.561
$Oracle$	0	78.9	-

Thus, we use the same combination of predictions (referred as *production relevance* below) as a true relevance label r_i for each item d_i from our data D for training and testing. We use just relevance predictions instead of human relevance judgments, since it is too expensive to collect the latter for hundreds of items per query needed for the result page optimization and evaluation. Indeed, in the case of e-commerce search engines, a user looks through much more items per query issue on average than in the case of web search engines. It is also the case of social media services, whose owners are highly interested in increasing user engagement, that is, in users looking through deep feeds.

Both the model of assessor relevance prediction and the click-based component were trained on hold-out datasets collected from user sessions preceding ones from data D. The first model uses for targets human relevance judgments collected with the 4-grade scale for top-24 items by relevance scores of the preceding production ranker. The second model is trained on targets based on the user behavior on the result page.

References

Diaz, F., Metzler, D., and Amer-Yahia, S. Relevance and ranking in online dating systems. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pp. 66–73. ACM, 2010.

Fain, D. C. and Pedersen, J. O. Sponsored search: A brief history. *Bulletin of the Association for Information Science and Technology*, 32(2):12–13, 2006.

Joachims, T., Granka, L., Pan, B., Hembrooke, H., and Gay, G. Accurately interpreting clickthrough data as implicit feedback. In *ACM SIGIR Forum*, volume 51, pp. 4–11. Acm, 2017.

Martınez, C. Partial quicksort. In *Proc. 6th ACMSIAM Workshop on Algorithm Engineering and Experiments and 1st ACM-SIAM Workshop on Analytic Algorithmics and Combinatorics*, pp. 224–228, 2004.

Svore, K. M., Volkovs, M. N., and Burges, C. J. Learning to rank with multiple objective functions. In *Proceedings of the 20th international conference on World wide web*, pp. 367–376. ACM, 2011.

Table 4. Performance, absolute for $WeakCutoff$ and relative Δ to $WeakCutoff$, $\%$ for others

Approach	DCG-RR	DCG@5	DCG@10	p@10	stup@12
$WeakCutoff$	0.52	0.69	1.07	0.73	0.06
$ConstCutoff$	0.05	2.91	2.94	1.55	-11.66
$QueryCutoff$	0.56	6.6	6.26	4.29	-17.49
OFP	3.86	23.8	20.3	10.71	-33.17
$OFP + LBO$	4.17	25.95	**22.35**	11.95	**-37.61**
$OFP + PG$	**4.33**	**26.04**	**22.35**	**12.23**	-36.78
$Oracle$	14.44				