Online Ranking with Top-1 Feedback

Sougata Chaudhuri
University of Michigan, Ann Arbor

Ambuj Tewari
University of Michigan, Ann Arbor

Abstract

We consider a setting where a system learns to rank a fixed set of m items. The goal is produce good item rankings for users with diverse interests who interact online with the system for T rounds. We consider a novel top-1 feedback model: at the end of each round, the relevance score for only the top ranked object is revealed. However, the performance of the system is judged on the entire ranked list. We provide a comprehensive set of results regarding learnability under this challenging setting. For PairwiseLoss and DCG, two popular ranking measures, we prove that the minimax regret is $\Theta(T^{2/3})$. Moreover, the minimax regret is achievable using an efficient strategy that only spends $O(m \log m)$ time per round. The same efficient strategy achieves $O(T^{2/3})$ regret for Precision@k. Surprisingly, we show that for normalized versions of these ranking measures, i.e., AUC, NDCG & MAP, no online ranking algorithm can have sublinear regret.

1 Introduction

Consider a system that is learning to rank a fixed set of objects for presentation to users, when different users have varied preferences for the objects. Learning occurs in an online setting: at each round, the system outputs a ranked list of the objects and the quality of ranking is measured by one of several popular ranking measures (like DCG [Järvelin and Kekäläinen, 2000] or MAP [Baeza-Yates and Ribeiro-Neto, 1999]), taking into account the users' preferences encoded as relevance vectors. We work in a game-theoretic setting and do not make any stochastic assumptions on how the relevance vectors are generated. Thus, it is assumed that the relevance vectors are generated by an oblivious, non-stochastic adversary. The objective of the learner is to have a sub-linear (in the number of rounds) regret against the best ranking in hindsight. The idea of ranking for diverse preferences has been motivated from a branch of work, sometimes called "ranking with diversity".

Most of the existing work on"ranking with diversity" [Radlinski et al., 2008, 2009, Agrawal et al., 2009] has focused on learning an optimal ranking of a fixed set of objects with a simple 0-1 loss. The loss in a round is 0 if among the top k (out of m) objects presented to a user, the user finds at least one relevant object. Our model focuses on optimal ranking where the losses considered are practical ranking losses like DCG and MAP. In addition, we consider a novel and challenging feedback model in this paper: *the learner only gets to see the relevance of the object placed at the top* (rank 1), whereas the ranking performance measure, and hence the regret, depends on the full relevance vector. Of course, one can consider a top-c feedback model for a constant $c \geq 1$ that doesn't grow with m. We choose to focus on the most challenging $c = 1$ case. We highlight two practical scenarios motivating the feedback model.

Economic Constraints: A company wants to produce a ranked order of a fixed set of products related to a query. Different products are likely to have varying relevance to different users, depending on user characteristics such as age, gender, etc. In principle, a user can browse through the entire ranked list giving carefully considered ratings, say on a 5 point scale, to each product. In practice, however, it is quite likely that user will scan through all the products and have a rough idea about how relevant each product is to her. But she will likely be reluctant to give thorough feedback on each product, unless the company provides some economic incentives to do so. Though the company needs high quality feedback on each product to keep refining the ranking strategy, it cannot afford to give incentives due to budget constraints. Hence, they require the user to give feedback only on top placed

Appearing in Proceedings of the 18^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2015, San Diego, CA, USA. JMLR: W&CP volume 38. Copyright 2015 by the authors.

product. This allows the user to look at all products but does not burden her with task of providing feedback beyond the top-ranked product. In this scenario, a full relevance vector is implicit in the user's mind but the system (company) gets to see, and possibly pays for, the relevance of only the top placed product.

User Burden Constraints: A medical company wants to build an app to suggest activities (take a walk, meditate, watch relaxing videos, etc.) that can lead to reduction of stress in a certain highly stressed segment of the population. Not all activities are suitable for everyone under all conditions since the effects of the activities vary depending on the user attributes like age & gender and on the context such as time of day & day of week. To satisfy most users, the company wants to produce a useful ordering of the stress reduction activities, but stressed users are unlikely to give feedback on the usefulness (relevance) of every activity because it increases their cognitive burden. So the company can ask for feedback about just the top ranked activity while, as in the previous example, each activity has an implicit relevance score for each user. However, the system will only get to see the relevance of the top ranked suggested activity.

Theoretically, the top-1 feedback model is neither full-feedback nor bandit-feedback since not even the loss (quantified by some ranking measure) at each round is revealed to the learner. The appropriate framework to study the problem is that of *partial monitoring* [Cesa-Bianchi et al., 2006]. A very recent paper shows another practical application of the partial monitoring framework where the feedback is neither full nor bandit [Lin et al., 2014]. Recent advances in the classification of partial monitoring games tell us that the minimax regret, in an adversarial setting, is governed by a property of the loss and feedback functions called *observability* [Bartok et al., 2014, Foster and Rakhlin, 2012]. Observability is of two kinds: *local* and *global*. We instantiate these general observability notions for the top-1 feedback case and prove that, for some ranking measures, namely PairwiseLoss [Duchi et al., 2010], DCG and Precision@k [Liu et al., 2007], global observability holds. This immediately shows that the upper bound on regret scales as $O(T^{2/3})$. Specifically for PairwiseLoss and DCG, we further prove that local observability fails, which shows that their *minimax* regret scales as $\Theta(T^{2/3})$. However, the generic algorithm that enjoys $O(T^{2/3})$ regret for globally observable games maintains an explicit distribution over learner actions. For us, the action set is the exponentially large set of $m!$ rankings over m objects. We therefore provide an *efficient* algorithm that exploits the structure of rankings. It runs in $O(m \log m)$ time per step and achieves a $O(T^{2/3})$ regret bound for PairwiseLoss, DCG and Precision@k. Moreover, the regret of our efficient algorithm has a logarithmic dependence on number of learner's actions (i.e., polynomial dependence on m), whereas the generic algorithm has a linear dependence on number of actions (i.e., exponential dependence on m).

For several measures, their *normalized* versions are considered. For example, the normalized versions of PairwiseLoss, DCG and Precision@k are called AUC [Cortes and Mohri, 2004], NDCG [Järvelin and Kekäläinen, 2002] and MAP respectively. We show an unexpected result for the normalized versions: *they do not* admit sub-linear regret algorithms under top-1 feedback. This is despite the fact that the opposite is true for their unnormalized counterparts! Intuitively, the normalization makes it hard to construct an unbiased estimator of the (unobserved) relevance vector. Surprisingly, we are able to translate this intuitive hurdle into a provable impossibility. Finally, we present some preliminary experiments to explore the performance of our efficient algorithm and compare its regret to its full information counterpart.

2 Notation and Preliminaries

We have a fixed set of m objects numbered $\{1, 2, \ldots, m\}$. A permutation σ gives a mapping from objects to their ranks and its inverse σ^{-1} gives a mapping from ranks back to objects. Thus, $\sigma(i) = j$ means object i is placed at position j while $\sigma^{-1}(i) = j$ means object j is placed at position i. For a binary relevance vector $r \in \{0, 1\}^m$, $r(i)$ indicates relevance level of object i. We denote $\{1, \ldots, n\}$ by $[n]$. The learner can choose from $m!$ actions (permutations) whereas nature/adversary can choose from 2^m outcomes (when relevance levels are restricted to binary) or from n^m outcomes (when there are n relevance levels). We sometimes refer to the ith player action (in some fixed ordering of $m!$ available actions) as σ_i (resp. ith adversary action as r_i). With this convention, $\sigma(i)$ is a number but σ_i is a permutation. Also, a vector can be row or column vector depending on context.

The oblivious adversary chooses the relevance vectors r_t in advance but doesn't reveal them to the learner. At round t, the learner outputs a permutation (ranking) σ_t of the objects (possibly using some internal randomization, based on feedback history so far). The quality of σ_t is judged against r_t by a ranking loss RL. *Crucially, only the relevance of the top ranked object (i.e., $r_t(\sigma_t^{-1}(1))$) is revealed to the learner at end of round t.* Thus, the learner gets to know neither r_t (full information problem) nor $RL(\sigma_t, r_t)$ (bandit problem). The objective of the learner is to minimize the expected regret with respect to best permutation

in hindsight:

$$\mathbb{E}_{\sigma_1,\ldots,\sigma_T}\left[\sum_{t=1}^{T} RL(\sigma_t, r_t)\right] - \min_{\sigma} \sum_{t=1}^{T} RL(\sigma, r_t). \quad (1)$$

When RL is a gain, not loss, we need to negate the quantity above. We consider binary relevance but many of our techniques should easily extend to multi-graded relevance provided the performance measure has the right form. The worst-case regret of a learner strategy is its maximal regret over all possible choices of $r_1, \ldots, r_T$. The **minimax regret** is the minimal worst-case regret over all learner strategies.

3 Ranking Measures

We consider ranking measures which can be expressed in the form $f(\sigma) \cdot r$, where the function $f : \mathbb{R}^m \to \mathbb{R}^m$ is composed of m copies of a univariate, monotonic, scalar valued function. Thus, $f(\sigma) = [f^s(\sigma(1)), f^s(\sigma(2)), \ldots, f^s(\sigma(m))]$, where $f^s : \mathbb{R} \to \mathbb{R}$. Monotonic (increasing) means $f^s(\sigma(i)) \geq f^s(\sigma(j))$, whenever $\sigma(i) > \sigma(j)$. Monotonic (decreasing) is defined similarly. The following popular ranking measures can be expressed in the form $f(\sigma) \cdot r$.

PairwiseLoss & SumLoss: PairwiseLoss is defined as: $PL(\sigma, r) = \sum_{i=1}^{m} \sum_{j=1}^{m} \mathbb{1}(\sigma(i) < \sigma(j)) \mathbb{1}(r(i) < r(j))$. PairwiseLoss cannot be directly expressed in the form of $f(\sigma) \cdot r$. Instead, we consider **SumLoss**, defined as: $SumLoss(\sigma, r) = \sum_{i=1}^{m} \sigma(i)\, r(i)$. SumLoss has the form $f(\sigma) \cdot r$, where $f(\sigma) = \sigma$. It has been shown by Ailon [2014] that SumLoss differs from PairwiseLoss only by an r-dependent constant and hence the regret under the two measures are equal:

$$\sum_{t=1}^{T} PL(\sigma_t, r_t) - \sum_{t=1}^{T} PL(\sigma, r_t) = \sum_{t=1}^{T} SumLoss(\sigma_t, r_t) - \sum_{t=1}^{T} SumLoss(\sigma, r_t). \quad (2)$$

Discounted Cumulative Gain: DCG, which admits non-binary relevance vectors, is defined as: $DCG(\sigma, r) = \sum_{i=1}^{m} \frac{2^{r(i)}-1}{\log_2(1+\sigma(i))}$ and becomes $\sum_{i=1}^{m} \frac{r(i)}{\log_2(1+\sigma(i))}$ for $r(i) \in \{0, 1\}$. Thus, for binary relevance, $DCG(\sigma, r)$ has the form $f(\sigma) \cdot r$, where $f(\sigma) = [\frac{1}{\log_2(1+\sigma(1))}, \frac{1}{\log_2(1+\sigma(2))}, \ldots, \frac{1}{\log_2(1+\sigma(m))}]$.

Precision@k Gain: Precision@k is defined as $Prec@k(\sigma, r) = \sum_{i=1}^{m} \mathbb{1}(\sigma(i) \leq k)\ r(i)$. Precision@$k$ can be written as $f(\sigma) \cdot r$ where $f(\sigma) = [\mathbb{1}(\sigma(1) < k), \ldots, \mathbb{1}(\sigma(m) < k)]$. Our focus is on $k \geq 2$, since for $k = 1$, top-1 feedback is actually the same as full information feedback, for which efficient algorithms exist.

Normalized measures are not of the form $\mathbf{f}(\sigma) \cdot \mathbf{r}$: PairwiseLoss, DCG and Precion@k are unnormalized versions of popular ranking measures, namely, Area Under Curve (AUC), Normalized Discounted Cumulative Gain (NDCG) and Mean Average Precision (MAP) respectively. None of these can be expressed in the form $f(\sigma) \cdot r$.

NDCG: $NDCG(\sigma, r) = \frac{1}{Z(r)} \sum_{i=1}^{m} \frac{2^{r(i)}-1}{\log_2(1+\sigma(i))}$ and becomes $\frac{1}{Z(r)} \sum_{i=1}^{m} \frac{r(i)}{\log_2(1+\sigma(i))}$ for $r(i) \in \{0, 1\}$. Here $Z(r) = \max_{\sigma} \sum_{i=1}^{m} \frac{2^{r(i)}-1}{\log_2(1+\sigma(i))}$ is the normalizing factor ($Z(r) = \max_{\sigma} \sum_{i=1}^{m} \frac{r(i)}{\log_2(1+\sigma(i))}$ for binary relevance). It can be clearly seen that $NDCG(\sigma, r) = f(\sigma) \cdot g(r)$, where $f(\sigma)$ is same as DCG but $g(r) = \frac{r}{Z(r)}$ is non-linear in r.

MAP: MAP is a gain function and is defined as: $MAP(\sigma, r) = \frac{1}{\|r\|_1} \sum_{i=1}^{m} \frac{\sum_{j \leq i} \mathbb{1}(r(\sigma^{-1}(j))=1)}{i} \mathbb{1}(r(\sigma^{-1}(i) = 1)$. It can be clearly seen that MAP cannot be expressed in the form $f(\sigma) \cdot r$.

AUC: AUC is a loss function and is defined as: $AUC(\sigma, r) = \frac{1}{N(r)} \sum_{i=1}^{m} \sum_{j=1}^{m} \mathbb{1}(\sigma(i) < \sigma(j)) \mathbb{1}(r(i) < r(j))$, where $N(r) = (\sum_{i=1}^{m} \mathbb{1}(r(i) = 1)) \cdot (m - \sum_{i=1}^{m} \mathbb{1}(r(i) = 1))$. It can be clearly seen that AUC cannot be expressed in the form $f(\sigma) \cdot r$.

All subsequent results will be for binary valued relevance vectors, unless stated otherwise.

4 Summary of Results

We summarize our main results here before delving into technical details. The regret bounds are over time horizon T, with learner playing against an *oblivious adversary*. Unless otherwise stated, all proofs and extensions are given in the appendix.

Result 1: The minimax regret under DCG and PairwiseLoss (and hence SumLoss) is $\Theta(T^{2/3})$.

Result 2: An efficient algorithm, with running time $O(m \log m)$ per step, achieves the minimax regret under DCG and PairwiseLoss and also has a regret of $O(T^{2/3})$ for Precision@k. *The precise minimax regret under Precision@k, $k \geq 2$, remains an open issue.*

Result 3: The minimax regret for any of the normalized versions – NDCG, MAP and AUC – is $\Theta(T)$. Thus, there is no algorithm that guarantees *sublinear* regret for the normalized measures.

Result 4: The minimax regret rate, as a function of T, both for DCG and NDCG, does not change (i.e., remains $\Theta(T^{2/3})$ and $\Theta(T)$ respectively) when we consider non-binary, multi-graded relevance vectors.

Table 1: Loss matrix L for $m = 3$

Objects	r_1	r_2	r_3	r_4	r_5	r_6	r_7	r_8
123	000	001	010	011	100	101	110	111
$\sigma_1 = 123$	0	3	2	5	1	4	3	6
$\sigma_2 = 132$	0	2	3	5	1	3	4	6
$\sigma_3 = 213$	0	3	1	4	2	5	3	6
$\sigma_4 = 231$	0	1	3	4	2	3	5	6
$\sigma_5 = 312$	0	2	1	3	3	5	4	6
$\sigma_6 = 321$	0	1	2	3	3	4	5	6

Table 2: Feedback matrix H for $m = 3$

Objects	r_1	r_2	r_3	r_4	r_5	r_6	r_7	r_8
123	000	001	010	011	100	101	110	111
$\sigma_1 = 123$	0	0	0	0	1	1	1	1
$\sigma_2 = 132$	0	0	0	0	1	1	1	1
$\sigma_3 = 213$	0	0	1	1	0	0	1	1
$\sigma_4 = 231$	0	1	0	1	0	1	0	1
$\sigma_5 = 312$	0	0	1	1	0	0	1	1
$\sigma_6 = 321$	0	1	0	1	0	1	0	1

5 Relevant Definitions from Partial Monitoring

We develop all results in context of SumLoss. We then extend the results to other ranking measures. Our main results on regret bounds build on some of the theory for abstract partial monitoring games developed by Bartok et al. [2014] and Foster and Rakhlin [2012]. For ease of understanding, we reproduce the relevant notations and definitions in context of SumLoss.

Loss and Feedback Matrices: The online learning game with the SumLoss measure and feedback being relevance of top ranked object, can be expressed in form of a pair of *loss matrix* and *feedback matrix*. The *loss matrix* L is an $m! \times 2^m$ dimensional matrix, with rows indicating the learner's actions (permutations) and columns representing adversary's actions (relevance vectors). The entry in cell (i, j) of L indicates loss suffered when learner plays action i (i.e., σ_i) and adversary plays action j (i.e., r_j), that is, $L_{i,j} = \sigma_i \cdot r_j = \sum_{k=1}^{m} \sigma_i(k) r_j(k)$. The *feedback matrix* H has same dimension as *loss matrix*, with (i, j) entry being the relevance of top ranked object, i.e., $H_{i,j} = r_j(\sigma_i^{-1}(1))$. When the learner plays action σ_i and adversary plays action r_j, the true loss is $L_{i,j}$, while the feedback received is $H_{i,j}$.

Table 1 and 2 illustrate the matrices, with number of objects $m = 3$. In both the tables, the permutations indicate rank of each object and relevance vector indicates relevance of each object. For example, $\sigma_5 = 312$ means object 1 is ranked 3, object 2 is ranked 1 and object 3 is ranked 2. $r_5 = 100$ means object 1 has relevance level 1 and other two objects have relevance level 0. Also, $L_{3,4} = \sigma_3 \cdot r_4 = \sum_{i=1}^{3} \sigma_3(i) r_4(i) = 2 \cdot 0 + 1 \cdot 1 + 3 \cdot 1 = 4$; $H_{3,4} = r_4(\sigma_3^{-1}(1)) = r_4(2) = 1$. Other entries are computed similarly.

Let $\ell_i \in \mathbb{R}^{2^m}$ denote row i of L. Let Δ be the probability simplex in $\mathbb{R}^{2^m}$, i.e., $\Delta = \{p \in \mathbb{R}^{2^m} : \forall\ 1 \leq i \leq 2^m,\ p_i \geq 0,\ \sum p_i = 1\}$. The following definitions, given for abstract problems by Bartok et al. [2014], has been refined to fit our problem context.

Definition 1: Learner action i is called optimal under distribution $p \in \Delta$, if $\ell_i \cdot p \leq \ell_j \cdot p$, for all other learner actions $1 \leq j \leq m!$, $j \neq i$. For every action $i \in [m!]$, probability cell of i is defined as $C_i = \{p \in \Delta :$ action i is optimal under $p\}$. If a non-empty cell C_i is $2^m - 1$ dimensional (i.e, elements in C_i are defined by only 1 equality constraint), then associated action i is called *Pareto-optimal*.

Note that since entries in H are relevance levels of objects, there can be maximum of 2 distinct elements in each row of H, i.e., 0 or 1 (assuming binary relevance).

Definition 2: The *signal matrix* S_i, associated with learner's action σ_i, is a matrix with 2 rows and 2^m columns, with each entry 0 or 1, i.e., $S_i \in \{0, 1\}^{2 \times 2^m}$. The entries of ℓth column of row 1 and 2 of S_i are respectively: $(S_i)_{1,\ell} = \mathbb{1}(H_{i,\ell} = 0)$ and $(S_i)_{2,\ell} = \mathbb{1}(H_{i,\ell} = 1)$.

Note that by definitions of signal and feedback matrices, the 2nd row of S_i (2nd column of $S_i^\top$)) is precisely the ith row of H. The 1st row of S_i (1st column of $S_i^\top$)) is the (boolean) complement of ith row of H.

6 Minimax Regret for SumLoss

The minimax regret for SumLoss will be established by showing that: a) SumLoss satisfies *global observability*, and b) it does not satisfy *local observability*.

6.1 Global Observability

Definition 3: The condition of *global observability* holds, w.r.t. loss matrix L and feedback matrix H, if for every pair of learner's actions $\{\sigma_i, \sigma_j\}$, it is true that $\ell_i - \ell_j \in \oplus_{k \in [m!]} Col(S_k^\top)$ (where Col refers to column space).

The global observability condition states that the (vector) loss difference between any pair of learner's actions has to belong to the vector space spanned by columns of (transposed) signal matrices corresponding to all possible learner's actions. We derive the following theorem on global observability for *SumLoss*.

Theorem 1. *The global observability condition, as per Definition 3, holds w.r.t. loss matrix L and feedback matrix H defined for SumLoss, for any $m \geq 1$.*

Proof. For any σ_a (learner's action) and r_b (adversary's action), we have

$$L_{a,b} = \sigma_a \cdot r_b = \sum_{i=1}^{m} \sigma_a(i) r_b(i) \stackrel{1}{=} \sum_{j=1}^{m} j \; r_b(\sigma_a^{-1}(j)) \stackrel{2}{=} \sum_{j=1}^{m} j \; r_b(\tilde{\sigma}_{j(a)}^{-1}(1)) \stackrel{3}{=} \sum_{j=1}^{m} j \; (S^{\top}_{\tilde{\sigma}_{j(a)}})_{r_b,2}.$$

Thus, we have

$$\ell_a = [L_{a,1}, L_{a,2}, \ldots, L_{a,2^m}] = [L_{\sigma_a,r_1}, L_{\sigma_a,r_2}, \ldots, L_{\sigma_a,r_{2^m}}] = [\sum_{j=1}^{m} j \; (S^{\top}_{\tilde{\sigma}_{j(a)}})_{r_1,2}, \sum_{j=1}^{m} j \; (S^{\top}_{\tilde{\sigma}_{j(a)}})_{r_2,2}, .., \sum_{j=1}^{m} j \; (S^{\top}_{\tilde{\sigma}_{j(a)}})_{r_{2^m},2}] \stackrel{4}{=} \sum_{j=1}^{m} j \; (S^{\top}_{\tilde{\sigma}_{j(a)}})_{:,2}.$$

Equality 4 shows that ℓ_a is in the column span of m of the $m!$ possible (transposed) signal matrices, specifically in the span of the 2nd columns of those (transposed) m matrices. Hence, for all actions σ_a, it is holds that $\ell_a \in \oplus_{k \in [m!]} Col(S_k^{\top})$. This implies that $\ell_a - \ell_b \in \oplus_{k \in [m!]} Col(S_k^{\top}), \; \forall \; \sigma_a, \sigma_b$.

1. Equality 1 holds because $\sigma_a(i) = j \Rightarrow i = \sigma_a^{-1}(j)$.

2. Equality 2 holds because of the following reason. For any permutation σ_a and for every $j \in [m]$, $\exists$ a permutation $\tilde{\sigma}_{j(a)}$, s.t. the object which is assigned rank j by σ_a is the same object assigned rank 1 by $\tilde{\sigma}_{j(a)}$, i.e., $\sigma_a^{-1}(j) = \tilde{\sigma}_{j(a)}^{-1}(1)$.

3. In Equality 3, $(S^{\top}_{\tilde{\sigma}_{j(a)}})_{r_b,2}$ indicates the r_bth row and 2nd column of (transposed) signal matrix $S_{\tilde{\sigma}_{j(a)}}$, corresponding to learner action $\tilde{\sigma}_{j(a)}$. Equality 3 holds because $r_b(\tilde{\sigma}_{j(a)}^{-1}(1))$ is the entry in the row corresponding to action $\tilde{\sigma}_{j(a)}$ and column corresponding to action r_b of H (see Definition 2).

4. Equality 4 holds from the observation that for a particular j, $[(S^{\top}_{\tilde{\sigma}_{j(a)}})_{r_1,2}, (S^{\top}_{\tilde{\sigma}_{j(a)}})_{r_2,2}, \ldots, (S^{\top}_{\tilde{\sigma}_{j(a)}})_{r_{2^m},2}]$ forms the 2nd column of $(S^{\top}_{\tilde{\sigma}_{j(a)}})$, i.e., $(S^{\top}_{\tilde{\sigma}_{j(a)}})_{:,2}$. □

6.2 Local Observability

Definition 4: Two Pareto-optimal (learner's) actions i and j are called *neighboring actions* if $C_i \cap C_j$ is a $(2^m - 2)$ dimensional polytope (where C_i is probability cell of action σ_i). The *neighborhood action set* of two neighboring (learner's) actions i and j is defined as $N_{i,j}^{+} = \{k \in [m!] : C_i \cap C_j \subseteq C_k\}$.

Definition 5: A pair of neighboring (learner's) actions i and j is said to be locally observable if $\ell_i - \ell_j \in \oplus_{k \in N_{i,j}^{+}} Col(S_k^{\top})$. The condition of *local observability* holds if every pair of neighboring (learner's) actions is locally observable.

We now show that local observability condition fails for L, H under SumLoss. First, we present the following two lemmas characterizing Pareto-optimal actions and neighboring actions for SumLoss.

Lemma 2. *For SumLoss, each learner's action i is Pareto-optimal, where Pareto-optimality has been defined in Definition 1.*

Lemma 3. *A pair of learner's actions $\{\sigma_i, \; \sigma_j\}$ is a neighboring actions pair, if there is exactly one pair of objects, numbered $\{a, \; b\}$, whose positions differ in σ_i and σ_j. Moreover, a needs to be placed just before b in σ_i and b needs to placed just before a in σ_j.*

Lemma 2 and 3 lead to following result.

Theorem 4. *The local observability condition, as per Definition 5, fails w.r.t. loss matrix L and feedback matrix H defined for SumLoss, already at $m = 3$.*

6.3 Minimax Regret Bound

We establish the minimax regret for SumLoss by combining results on global and local observability. First, we get a lower bound by combining our Theorem 4 with Theorem 4 of Bartok et al. [2014].

Corollary 5. *Consider the online game for SumLoss with top-1 feedback and $m = 3$. Then, for every online learning algorithm, there is an adversary strategy generating relevance vectors, that guarantees the following*

$$\mathbb{E}\left[\sum_{t=1}^{T} SumLoss(\sigma_t, r_t)\right] - \min_{\sigma} \sum_{t=1}^{T} SumLoss(\sigma, r_t) = \Omega(T^{2/3}). \quad (3)$$

where the expectation is taken w.r.t. randomized learner's actions.

An immediate corollary of Theorem 1 and Theorem 3.1 in Cesa-Bianchi et al. [2006] gives an inefficient algorithm (inspired by the algorithm originally given in Piccolboni and Schindelhauer [2001]) obtaining $O(T^{2/3})$ regret.

Corollary 6. *The algorithm in Figure 1 of Cesa-Bianchi et al. [2006] achieves $O(T^{2/3})$ regret bound for SumLoss.*

The results above establish that the minimax regret for SumLoss, under top-1 feedback model, is $\Theta(T^{2/3})$.

However, the algorithm in Cesa-Bianchi et al. [2006] is intractable in our setting since the number of learner's actions is exponential in number of objects m. The next section tackles the efficiency issue.

7 Efficient Algorithm for Obtaining Minimax Regret under SumLoss

We provide an efficient algorithm for getting an $O(\text{poly}(m)T^{2/3})$ regret bound for SumLoss. The per round running time of the algorithm is $O(m \log m)$.

The key idea that we use in our algorithm is to divide time horizon T into phases. Within each phase, we allot a small number of rounds for pure *exploration* (this lets us estimate the average relevance vector for that phase). The estimated average vector is fed into a full information algorithm to get the distribution over actions for the next phase. Rounds in the next phase choose actions according to the distribution suggested by Follow the Perturbed Leader (FTPL) Kalai and Vempala [2005] (this is *exploitation* of previous experience). One of the key reasons for using FTPL as the full information algorithm, instead of exponential weighing schemes, is that the structure of our problem allows the FTPL update to be implemented via a simple sorting operation on m objects. Exponential weighting schemes would explicitly maintain distribution over $m!$ actions, a prohibitively expensive step.

Our algorithm is motivated by the reduction from bandit-feedback to full feedback given by Blum and Mansour [2007]. However, the reduction *cannot be directly applied to our problem*, because we are not in the bandit setting and hence do not know loss of any action. Further, the algorithm of Blum and Mansour [2007] spends N rounds per phase to try out *each* of the N available actions — this is infeasible in our setting since $N = m!$.

Discussion of Algorithm 1. Our algorithm RTop-1F divides the time horizon into equal sized blocks of size K (lines 2-3). At the beginning of each block, m time points are selected uniformly at random without replacement in that block (lines 8-9). Within each block, if the current time is one of the selected times for exploration, an arbitrary permutation that places a particular object on top is played (lines 12-14). Otherwise, the permutation which minimizes the dot product with the perturbed score vector is played (lines 16-19). Note that the step $\sigma_t = M(\hat{s}_i + p_t)$ requires sorting of the m objects, which takes $O(m \log m)$ time. Our main theorem on regret of Algorithm 1 is as follows.

Theorem 7. *The expected regret of SumLoss, obtained by applying Algorithm 1, with* $K = m^{-1/3}T^{2/3}$ *and* $\epsilon = \sqrt{\frac{1}{mK}}$*, and the expectation being taken over random learner's actions* σ_t*, is*

$$\mathbb{E}\left[\sum_{t=1}^{T} SumLoss(\sigma_t, r_t)\right] \leq \min_{\sigma} \sum_{t=1}^{T} SumLoss(\sigma_t, r_t) + O(m^{8/3}T^{2/3}). \quad (4)$$

Algorithm 1 RankingwithTop-1Feedback(RTop-1F)

1: T = Time horizon, K = No. of (equal sized) blocks,
2: Time horizon divided into blocks $\{B_1, \ldots, B_K\}$,
3: where, $B_i = \{(i-1)(T/K)+1, \ldots, i(T/K)\}$.
4: Randomization parameter ϵ.
5: Initialize $\hat{s}_0 = \mathbf{0} \in \mathbb{R}^m$, $\hat{r}_0 = \mathbf{0} \in \mathbb{R}^m$.
6: **For** $i = 1, \ldots, K$
7: Update $\hat{s}_i = \hat{s}_{i-1} + \hat{r}_{i-1}$.
8: Select m time points $\{i_1, \ldots, i_m\}$ from block B_i,
9: uniformly at random, without replacement.
10: **For** $t \in B_i$
11: **If** $t = i_j \in \{i_1, \ldots, i_m\}$
12: Output any permutation σ_t which places
13: jth object on top.
14: Receive feedback on the jth object $r_{i_j}(j)$.
15: **Else**
16: Sample $p_t \in [0, 1/\epsilon]^m$ from the product
17: of uniform distribution in each dimension.
18: Output permutation $\sigma_t = M(\hat{s}_i + p_t)$
19: where $M(y) = \operatorname{argmin}_{\sigma} \sigma \cdot y$.
20: **end for**
21: Set $\hat{r}_i = [r_{i_1}(1), \ldots, r_{i_m}(m)] \in \mathbb{R}^m$.
22: **end for**

The following simple but useful lemma is required to prove Theorem 7.

Lemma 8. *Let the average of full relevance vectors over the time period* $\{1, 2, \ldots, t\}$ *be denoted as* $r^{avg}_{1:t}$*, that is,* $r^{avg}_{1:t} = \sum_{k=1}^{t} \frac{r_k}{t}$*. Let* $\{i_1, i_2, \ldots, i_m\}$ *be* m *arbitrary time points, chosen uniformly at random, without replacement, from* $\{1, \ldots, t\}$*. At time point* i_j*, only the* j*th component of vector* r_{i_j}*, i.e.,* $r_{i_j}(j)$*, becomes known,* $\forall j \in \{1, \ldots, m\}$*. Then the relevance vector* $\hat{r}_t = [r_{i_1}(1), \ldots, r_{i_m}(m)]$ *is an unbiased estimator of* $r^{avg}_{1:t}$*.*

8 Regret Bounds for PairwiseLoss, DCG and Prec@k

As we saw in Eq. 2, the regret of SumLoss is same as regret of PairwiseLoss. Thus, SumLoss in Cor. 5 and Thm. 7 can be replaced by PairwiseLoss to get exactly same results on regret.

All the results of SumLoss can be extended to DCG. Moreover, the results can be extended even for discrete, non-binary relevance vectors. Thus, the min-

imax regret of DCG, when the adversary can take any discrete valued, non-negative relevance vector is $\Theta(T^{2/3})$, which can be achieved by (a slight variant of) the efficient algorithm of Sec. 7. The main differences between SumLoss and DCG are the following. The former is a loss function, the latter is a gain function. Also, $f(\sigma) \neq \sigma$ in DCG (Def. in Sec.2) and when $r \in \{0, 1, \ldots, n\}^m$, DCG cannot be expressed as $f(\sigma) \cdot r$, as is clear from definition in Sec. 3. Nevertheless, DCG can be expressed as $f(\sigma) \cdot g(r)$, , where $g(r) = [g^s(r(1)), g^s(r(2)), \ldots, g^s(r(m))]$, $g^s(i) = 2^i - 1$ is constructed from univariate, monotonic, scalar valued functions. Thus, there are minor differences in definitions and proofs of theorems for SumLoss and DCG. The structural properties of $f(\cdot)$, $g(\cdot)$ are key in extending results. For binary valued relevance vectors, Algorithm 1 can be applied to DCG as is. For multigraded relevance vector, the only thing that changes is that the relevance feedback is transformed via component functions of $g(\cdot)$.

We provide the extension of Theorem 7 for DCG. Let relevance vectors chosen by adversary be of level $n+1$, i.e., $r \in \{0, 1, \ldots, n\}^m$. In practice, n is almost always less than 5.

Theorem 9. *The expected regret of DCG, obtained by applying Algorithm 1 , with $K = m^{-1/3}T^{2/3}$ and $\epsilon = \sqrt{\frac{1}{(2^n-1)^2 mK}}$, and the expectation being taken over random learner's actions σ_t, is*

$$\mathbb{E}\left[\sum_{t=1}^{T} DCG(\sigma_t, r_t)\right] \geq \max_{\sigma} \sum_{t=1}^{T} DCG(\sigma, r_t) - O((2^n - 1)m^{5/3}T^{2/3}). \quad (5)$$

In case of binary relevance vector, the regret term is $O(m^{5/3}T^{2/3})$. Moreover, since local observability fails, there is a matching $\Omega(T^{2/3})$ lower bound.

The regret upper bounds we proved for SumLoss also easily extend to Precision@k. We have the following extension of Theorem 7.

Theorem 10. *The expected regret of Prec@k, obtained by applying algorithm 1, with $K = m^{-1/3}T^{2/3}$ and $\epsilon = \sqrt{\frac{1}{mK}}$, and the expectation being taken over random learner's actions σ_t, is*

$$\mathbb{E}\left[\sum_{t=1}^{T} Prec@k(\sigma_t, r_t)\right] \geq \max_{\sigma} \sum_{t=1}^{T} Prec@k(\sigma, r_t) - O(km^{2/3}T^{2/3}). \quad (6)$$

However, the value of $Prec@k$ is independent of the order of objects in the top k positions of the ranked list. This changes the neighboring action claims. Therefore, the minimax regret of $Prec@k$ remains an open question, since we do not have local observability failure results for $Prec@k$.

9 Non-Existence of Sublinear Regret Bounds for NDCG, MAP and AUC

As stated in Sec. 3, NDCG, MAP and AUC are normalized versions of measures DCG, Precision@k and PairwiseLoss. We have the following lemma for all these normalized ranking measures.

Lemma 11. *The global observability condition, as per Definition 1, fails for NDCG, MAP and AUC.*

Combining the above lemma with Theorem 2 of Bartok et al. [2014], we conclude that there *cannot exist any algorithm which has sublinear regret for any of the following measures: NDCG, MAP or AUC, with top-1 feedback.*

Theorem 12. *There exists an online game for NDCG with top-1 feedback, such that for every online algorithm, there is an adversary strategy that guarantees the following*

$$\max_{\sigma} \sum_{t=1}^{T} NDCG(\sigma, r_t) - \mathbb{E}\left[\sum_{t=1}^{T} NDCG(\sigma_t, r_t)\right] = \Omega(T). \quad (7)$$

Furthermore, the same lower bound holds if NDCG is replaced by MAP or AUC.

Note: In the NDCG case, allowing the adversary to play multigraded, and not just binary, relevance vectors only makes the adversary more powerful. So the lower bound above continues to apply.

10 Simulation Results

We conducted a simulation study to compare regret rate under the popular DCG metric when feedback is received only for top ranked object (by applying Algorithm 1) with the case when full relevance vector is revealed at end of each round (by applying Follow the Perturbed Leader of Kalai and Vempala [2005]). Relevance vectors were restricted to take binary values. The reason for choosing DCG is that it is a popular metric used in industry and to empirically confirm that Algorithm1 works for DCG, even though the derivations focused on SumLoss.

We simulated relevance vectors for a fixed set of 10 objects ($m = 10$). We initially fixed half of the objects to be relevant and other half irrelevant, as the true relevance vector. Then, binary valued relevance vectors for adversary were simulated by adding small Gaussian noise to the true relevance vector. Thus, there

was mostly small variation among the relevance vectors, simulating the case that, in real world, majority of users might agree on the relevance of most objects, with small differences. A total of $T = 10000$ relevance vectors were generated (simulating number of rounds).

In Algorithm 1, since the average of the relevance vectors per block was estimated by uniform sampling according to Lemma 8, the algorithm was run 10 times, with the same set of relevance vectors, for averaging under the algorithm's randomization. Fig. 1 shows time-normalized regret with top-1 feedback for DCG. Time-normalized means the cumulative regret upto time t was divided by t, for $1 \leq t \leq T$. The figure clearly indicates that after the learning phase of the initial few iterations, the learner outputs mostly correct rankings, with the average regret going down to 0 at rate $O(T^{-1/3})$.

Fig. 2 compares time-normalized regret, between top-1 and full information feedback, for DCG. The comparison was done from 1000 iterations onwards, i.e., roughly after the learning phase of the learner. It can be clearly seen that average regret with full information goes down at rate faster ($\Theta(T^{-1/2})$) than average regret with top-1 feedback ($\Theta(T^{-1/3})$).

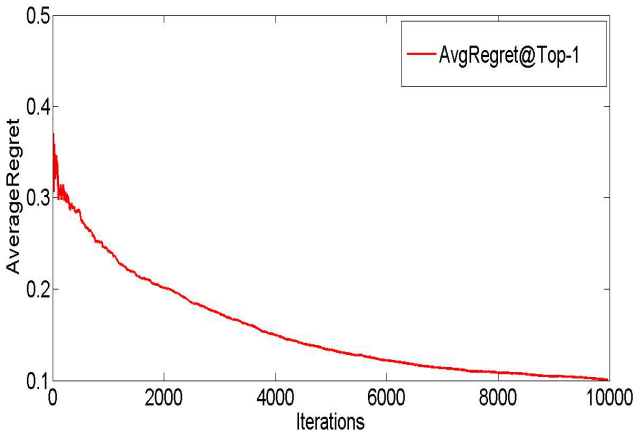

Figure 1: Average regret for DCG with feedback on top ranked object. *Best viewed in color.*

11 Conclusion

We introduced a novel, interesting feedback model for online ranking of a fixed set of objects for users with diverse preferences. Our results are quite comprehensive as far as the T dependence is concerned. The only exception is Precision@k where the possibility of an $O(T^{1/2})$ regret algorithm remains open. Note that Precision@k is really peculiar since top-1 feedback is actually full feedback when $k = 1$.

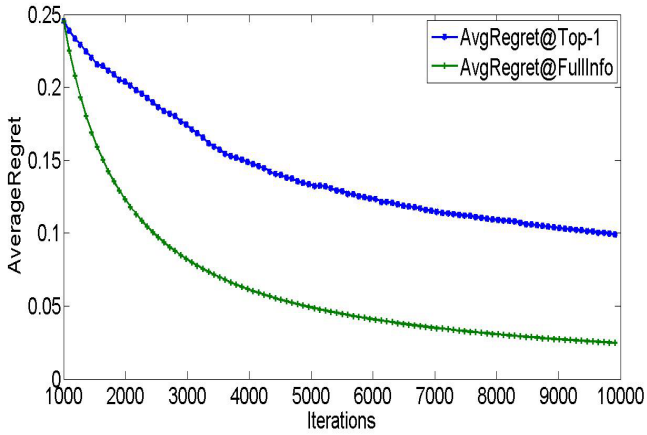

Figure 2: Comparison of average regret over time, for DCG, between top-1 feedback and full relevance vector feedback. *Best viewed in color.* .

The most interesting future extension of this work is to move beyond ranking of a fixed set of objects and considering different document lists associated with queries. This falls under the category of partial monitoring with side information. Very little relevant work has been done in the general setting and our current work can lay the foundations for interesting application in this field. Another extension is investigating whether an algorithm with sublinear regret can be defined for NDCG, MAP or AUC, when the regret is defined relative to some constant factor (larger than 1) times the best performance in hindsight.

Acknowledgement

The authors acknowledge the support of NSF under grant IIS-1319810.

References

Rakesh Agrawal, Sreenivas Gollapudi, Alan Halverson, and Samuel Ieong. Diversifying search results. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining*, pages 5–14. ACM, 2009.

Nir Ailon. Improved bounds for online learning over the permutahedron and other ranking polytopes. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics*, pages 29–37, 2014.

Ricardo Baeza-Yates and Berthier Ribeiro-Neto. *Modern information retrieval*, volume 463. ACM Press, 1999.

Gabor Bartok, Dean P. Foster, David Pal, Alexander Rakhlin, and Csaba Szepesvari. Partial monitoringclassification, regret bounds, and algorithms. *Mathematics of Operations Research*, 39(4):967–997, 2014.

A. Blum and Y. Mansour. Learning, regret minimization, and equilibria. In N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani, editors, *Algorithmic game theory*. Cambridge University Press, 2007.

Nicolo Cesa-Bianchi, Gábor Lugosi, and Gilles Stoltz. Regret minimization under partial monitoring. *Mathematics of Operations Research*, pages 562–580, 2006.

Corinna Cortes and Mehryar Mohri. Auc optimization vs. error rate minimization. *Advances in neural information processing systems*, 16(16):313–320, 2004.

John C. Duchi, Lester W. Mackey, and Michael I. Jordan. On the consistency of ranking algorithms. In *Proceedings of the 27th International Conference on Machine Learning*, pages 327–334, 2010.

Dean P. Foster and Alexander Rakhlin. No internal regret via neighborhood watch. In *International Conference on Artificial Intelligence and Statistics*, pages 382–390, 2012.

Kalervo Järvelin and Jaana Kekäläinen. IR evaluation methods for retrieving highly relevant documents. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 41–48. ACM, 2000.

Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems*, pages 422–446, 2002.

Adam Kalai and Santosh Vempala. Efficient algorithms for online decision problems. *Journal of Computer and System Sciences*, pages 291–307, 2005.

Tian Lin, Bruno Abrahao, Robert Kleinberg, and John Lui. Combinatorial partial monitoring game with linear feedback and its applications. In *Proceedings of the 31th International Conference on Machine Learning*, pages 901–909. ACM, 2014.

Tie-Yan Liu, Jun Xu, Tao Qin, Wenying Xiong, and Hang Li. Letor: Benchmark dataset for research on learning to rank for information retrieval. In *Proceedings of SIGIR 2007 workshop on learning to rank for information retrieval*, pages 3–10, 2007.

Antonio Piccolboni and Christian Schindelhauer. Discrete prediction games with arbitrary feedback and loss. In *Computational Learning Theory*, pages 208–223. Springer, 2001.

Filip Radlinski, Robert Kleinberg, and Thorsten Joachims. Learning diverse rankings with multi-armed bandits. In *Proceedings of the 25th International conference on Machine learning*, pages 784–791. ACM, 2008.

Filip Radlinski, Paul N Bennett, Ben Carterette, and Thorsten Joachims. Redundancy, diversity and interdependent document relevance. In *ACM SIGIR*, pages 46–52. ACM, 2009.