Online Ranking with Top-1 Feedback

Sougata Chaudhuri and Ambuj Tewari
{sougata,tewaria}@umich.edu

October 7, 2014

Abstract

We consider a setting where a system learns to rank a fixed set of \( m \) items. The goal is to produce a good ranking for users with diverse interests who interact with the system for \( T \) rounds in an online fashion. We consider a novel top-1 feedback model for this problem: at the end of each round, the relevance score for only the top ranked object is revealed to the system. 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 popular ranking measures such as PairwiseLoss and DCG, we prove that the minimax regret is \( \Theta(T^{2/3}) \). Moreover, the minimax regret is achievable using an efficient algorithmic strategy that only spends \( O(m \log m) \) time per round. The same algorithmic strategy achieves \( O(T^{2/3}) \) regret for Precision@k. Surprisingly, we show that for normalized versions of these ranking measures, namely AUC, NDCG and MAP, no online ranking algorithm can have sub-linear 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 preference 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 [12] or MAP [3]), 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 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” [15, 17, 1]. In “ranking with diversity” work in literature, the focus has been on learning an optimal ranking of a fixed set of objects where the loss in a round is simply 0 – 1, depending on whether the user finds any relevant object in top-k (of \( m \)) ranked items.

In addition to considering more practical losses associated with ranking like DCG and MAP, 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. We highlight two practical scenarios motivating the feedback model.

Economic Reason: 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 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 a 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 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 the relevance of only the top placed product.

**Privacy Reason:** A medical company wants to advertise multiple new drugs for a particular medical condition. Not all drugs are suitable for everyone since the effects of the drugs vary depending on the user attributes like age, gender etc. To satisfy most users, the company wants to produce a useful ordering of the drugs, but users are unlikely to give feedback on the usefulness (relevance) of every drug due to privacy concerns. So the company can ask for feedback about just the top ranked drug while, as above, every drug has an implicit relevance score for each user. However, the system will only get to see the relevance of the top ranked drug.

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 [7]. A very recent paper shows another practical application of the idea where feedback is neither full information nor bandit [14]. 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 [4, 10]. 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 [9], DCG and Precision@k [15], global observability holds. This immediately shows that the upper bound on regret scales as $O(T^{2/3})$ (ignoring log $T$ factors). 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 $O(T^{2/3}(\log T)^{1/3})$ 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 time $O(m \log m)$ per step and achieves a slightly improved (over generic algorithm) $O(T^{2/3})$ regret bound for PairwiseLoss, DCG and Precision@k.

For several measures, their normalized versions are considered. For example, the normalized versions of PairwiseLoss, DCG and Precision@k are called AUC [8], NDCG [11] 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 $\sigma$ gives a mapping from objects to their ranks and its inverse $\sigma^{-1}$ 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 (relevance vectors, when relevance levels are restricted to binary).

The learner’s (resp. adversary’s) actions are denoted by $\{\sigma_i : i \in [m!]\}$ (resp. $\{r_i : i \in [2^n]\}$). We sometimes interchangeably denote action $\sigma_i$ as $i$ (resp. $r_i$ as $i$). With this convention, $\sigma(i)$ is a number but $\sigma_i$ is a permutation. Also, a vector can be row or column vector depending on context. At round $t$, the learner outputs a permutation (ranking) $\sigma_t$ of the objects (possibly using some internal randomization, based on feedback history so far). The user looks at the list and generates a relevance vector $r_t$ which indicates relevance level of each object. The quality of $\sigma_t$ 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 (when $RL$ is a gain, not loss, we need to negate the quantity below):

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

We consider binary relevance but many of our techniques should easily extend to multi-graded relevance provided the performance measure has the right form.

**Precisely define minimax regret here:** What is the precise definition? What is the precise technical formula?

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 $f^s(\sigma(i)) = f(\sigma(j))$, whenever $\sigma(i) > \sigma(j), \forall i, j$. 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: $PL(\sigma, r) = \sum_{i=1}^{m} \sum_{j=1}^{m} 1(\sigma(i) < \sigma(j))1(r(i) < r(j))$. 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 in [2] that SumLoss differs from PairwiseLoss only by an $r$-dependent constant and hence the regret under the two measures are equal:

$$\sum_{t=1}^{T} PL(\sigma_t, r_t) - \sum_{t=1}^{T} PL(\sigma, r_t) = \sum_{t=1}^{T} SumLoss(\sigma_t, r_t) - \sum_{t=1}^{T} SumLoss(\sigma, r_t). \quad (2)$$
Our main results on regret bounds build on some of the theory for abstract partial monitoring games. We develop all results in context of SumLoss. We then extend the results to other ranking measures.

5 Relevant Definitions from Partial Monitoring

However, efficient algorithm already exists in stochastic case [5].

Assumption. Had the relevance vectors been stochastic in nature, the results would have held; since for \( k = 1 \), top-1 feedback is actually the same as full information feedback, for which efficient algorithms exist.

Normalised measures are not of the form \( f(\sigma) \cdot r \): PairwiseLoss, DCG and Precision@k are unnormalised versions of popular ranking measures, namely, Area Under Curve (AUC), Normalised Discounted Cumulative Gain (NDCG) and Mean Average Precision (MAP) respectively. None of the latter can be expressed in the form \( f(\sigma) \cdot r. \)

**DCG**: 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.

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

4 Summary of Results

With notations fixed, we now 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 normalised versions – NDCG, MAP and AUC – is \( \Theta(T) \). Thus, there is no algorithm that achieves sublinear regret for the normalised measures.

**Result 4**: The minimax regret, 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.

We re-emphasise that the relevance vectors are generated adversarially, without any stochastic assumption. Had the relevance vectors been stochastic in nature, the results would have held; however, efficient algorithm already exists in stochastic case [3].

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.
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     |

devolved in [4, 10]. 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., $\sigma_i$) 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 entry in cell $(i,j)$ being the relevance of top ranked object, i.e., $H_{i,j} = r_j(\sigma_i(1))$. When the learner plays action $\sigma_i$ 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, $L_{3,4} = \sigma_3 \cdot r_4 = \sum_{i=1}^{3} \sigma(i) r(i) = 2 \cdot 0 + 1 \cdot 1 + 3 \cdot 1 = 4$; $H_{3,4} = r_4(\sigma_3(1)) = r_4(2) = 1$. Other entries are computed similarly.

Let $\ell_i \in \mathbb{R}^{2^m}$ denote row $i$ of $L$. Let $\Delta$ 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 in [4], has been refined to fit our problem context.
Definition 1: Pareto-optimality 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 $C_i = \{ p \in \Delta : \text{action } i \text{ is optimal under } p \}$. If a non-empty cell $C_i$ is $2^m - 1$ dimensional, 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: Signal-matrices Signal matrix $S_i$, associated with learner’s action $\sigma_i$, is a matrix with 2 rows and $2m$ columns, with each entry 0 or 1, i.e., $S_i \in \{0,1\}^{2 \times 2m}$. The entries of $\ell$ th column of row 1 and 2 of $S_i$ are respectively: $(S_i)_{1,\ell} = I(H_{\ell} = 0)$ and $(S_i)_{2,\ell} = I(H_{\ell} = 1)$, where $I$ is the indicator function and $H$ is the feedback matrix.

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

6 Minimax Regret for SumLoss

The minimax regret of SumLoss will be established by showing that: a) SumLoss satisfies \textit{global observability}, and b) it does not satisfy \textit{local observability}.

6.1 Global Observability

Definition 3: The condition of \textit{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 $\ell_i - \ell_j \in \bigoplus_{k \in [m!]} \text{Col}(S_k^\top)$ (where Col refers to column space).

The global observability condition basically 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 \textit{SumLoss} (Proof in appendix).

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 some $\sigma_a$ (learner’s action) and $r_b$ (adversary’s action), we have

$$L_{a,b} = \sigma_a \cdot r_b = \sum_{i=1}^{m} \sigma_a(i)r_b(i) = \sum_{j=1}^{m} j r_b(\sigma_a^{-1}(j))^2 \sum_{j=1}^{m} j r_b(\tilde{\sigma}_j^{-1}(1)) \sum_{j=1}^{m} j (S_j^\top)_{r_b,2}.$$
Thus, we have

\[ \ell_a = \sum_{j=1}^{m} j (S_{\tilde{\sigma}_{j(a)}}^T)_{r_1,2}, \sum_{j=1}^{m} j (S_{\tilde{\sigma}_{j(a)}}^T)_{r_2,2}, \ldots, \sum_{j=1}^{m} j (S_{\tilde{\sigma}_{j(a)}}^T)_{r_{2m},2} \]

Equality 4 shows that \( \ell_a \) 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 \( \sigma_a \), it is true that \( \ell_a \in \oplus_{k \in [m!]} \text{Col}(S_k^T) \), \( l_a - l_b \in \oplus_{k \in [m!]} \text{Col}(S_k^T) \), \( \forall \) actions \( \sigma_a, \sigma_b \).

1. Equality 1 holds by the relation \( \sigma_a(i) = j \Rightarrow i = \sigma_a^{-1}(j) \).

2. Equality 2 holds because the relation \( \sigma_a(i) = j \Rightarrow i = \sigma_a^{-1}(j) \).

3. In equality 3, \( (S_{\tilde{\sigma}_{j(a)}}^T)_{r_1,2} \) indicates the \( r_1 \)th 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 \( H \) and from Definition 2.

4. Equality 4 holds from the observation that for a particular \( j \), \( (S_{\tilde{\sigma}_{j(a)}}^T)_{r_1,2}, (S_{\tilde{\sigma}_{j(a)}}^T)_{r_2,2}, \ldots, (S_{\tilde{\sigma}_{j(a)}}^T)_{r_{2m},2} \) forms the 2nd column of \( (S_{\tilde{\sigma}_{j(a)}}^T) \), i.e, \( (S_{\tilde{\sigma}_{j(a)}}^T)_{r,2} \).

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 \( \sigma_i \)). 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 \( \ell_i - \ell_j \in \oplus_{k \in N_{i,j}^+} \text{Col}(S_k^T) \). 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 propose the following 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 pair, if there is exactly one pair of objects, numbered \( \{ a, b \} \), whose positions differ in \( \sigma_i \) and \( \sigma_j \). Moreover, \( a \) needs to be placed just before \( b \) in \( \sigma_i \) and \( b \) needs to be placed just before \( a \) in \( \sigma_j \).

**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 in [4].

**Corollary 5.** There exists an online game for SumLoss with top-1 feedback, such that for every online algorithm, there is an adversary strategy generating relevance vectors, that guarantees the following

\[
E \left[ \sum_{t=1}^{T} \text{SumLoss}(\sigma_t, r_t) \right] - \min_{\sigma} \sum_{t=1}^{T} \text{SumLoss}(\sigma, r_t) = \Omega\left(\frac{T^2}{3}\right)
\]

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

An immediate corollary of Theorem 1 and Theorem 3.1 in [7] gives an in-efficient algorithm (inspired by the algorithm originally given in [16]) obtaining \(O\left(\frac{T^2}{3}(\log T)^{1/3}\right)\) regret.

**Corollary 6.** The algorithm in Figure 1 of [7] achieves \(O\left(\frac{T^2}{3}(\log T)^{1/3}\right)\) regret bound for SumLoss.

The results above establish that the minimax regret for SumLoss, under top-1 feedback model, is \(\tilde{O}(T^{2/3})\) where \(\tilde{O}\) hides \(\log(T)\) factors. However, the algorithm in [7] 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 Sum-Loss

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 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). Using the average vector as a full information vector for next phase, rest of the rounds in next phase follow the actions according to the distribution suggested by Follow the Perturbed Leader (FTPL) [13] (this is exploitation of previous experience). One of the key reasons for exploiting FTPL in our algorithm, instead of exponential weighing schemes, is that the structure of our problem allows distribution to be maintained implicitly over \(m\) objects, instead of explicitly maintaining distribution over \(m!\) arms, a prohibitively expensive step.

Our algorithm is motivated by the reduction from bandit-feedback to full feedback given in [6]. However, the algorithm in [6] 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 spends \(N\) rounds per phase to try out each of the \(N\) available actions – this is infeasible in our setting since \(N = m!\).

Our main theorem on regret of Algorithm [1] is as follows.
Algorithm 1 Ranking with Top-1 Feedback (RTop-1F)

\( T = \text{Time horizon}, \) \( K = \text{No. of (equal sized) blocks}, \)

Time horizon divided into blocks \( \{B_1, \ldots, B_K\} \),

randomization parameter \( \epsilon \).

Here, \( B_i = \{(i-1)(T/K) + 1, (i-1)(T/K) + 2, \ldots, i(T/K)\} \).

Initialize \( \hat{s}_0 = 0, \hat{r}_0 = 0 \).

For \( i = 1, \ldots, K \)

Update \( \hat{s}_i = \hat{s}_{i-1} + \hat{r}_{i-1} \).

Select \( m \) time points \( \{i_1, \ldots, i_m\} \) from block \( B_i \),

uniformly at random, without replacement.

For \( t \in B_i \)

If \( t = i_j \in \{i_1, \ldots, i_m\} \)

Output any permutation \( \sigma_t \) which places \( j \)th object on top.

Receive feedback on the \( j \)th object as \( r_{i_j}(j) \).

Else

Sample \( p_t \in [0, 1/\epsilon]^m \). (Product of uniform distribution in each dimension).

Output permutation \( \sigma_t = M(\hat{s}_i + p_t) \) where

\( M(y) = \text{argmin}_\sigma \sigma \cdot y \).

end for

Set \( \hat{r}_i = [r_{i_1}(1), \ldots, r_{i_m}(m)] \).

end for

Theorem 7. The expected regret of SumLoss, obtained by applying Algorithm 1 with \( K = m^{-1/3}T^{2/3} \)

and \( \epsilon = \sqrt{1/mK} \), and the expectation being taken over random learner’s actions \( \sigma_t \), is

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

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 \( r_{1:t}^{\text{avg}} \), that is, \( r_{1:t}^{\text{avg}} = \frac{1}{t} \sum_{k=1}^{t} r_k \). 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 \( r_{1:t}^{\text{avg}} \).

8 Regret Bounds for PairwiseLoss, DCG and Prec@k

As shown 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 minimax regret of DCG, when the
adversary can take any discrete valued, non-negative relevance vector is \( \Theta\left(\frac{T^2}{3}\right) \), which can be achieved by 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 \( g(r) = [g^*(r(1)), g^*(r(2)), \ldots, g^*(r(m))] \), \( g^*(i) = 2^i - 1 \) 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 multi-graded 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 always less than 5; hence \( \exp(n) \) is also small.

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

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

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 7, with \( K = m^{-1/3}T^{2/3} \) and \( \epsilon = \sqrt{\frac{T}{mK}} \), and the expectation being taken over random learner’s actions \( \sigma_t \), is

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

However, Prec@k is independent of the order of objects in top \( k \) positions of ranked list. This changes the neighboring action claims. Therefore, the minimax regret of Prec@k remains an open question, since the local observability results do not directly apply to 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 [4], 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

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

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 Empirical Results

We conducted simulation studies to compare regret rates of SumLoss and DCG, when feedback is received only for top ranked object (by applying Algorithm 1) and full relevance vector is revealed at end of each round (by applying FPL of [13]). Relevance vectors were restricted to take binary values.

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 variances among the relevance vectors, simulating the fact that, in real world, majority of users will 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 code for the algorithm was run 10 times, with the same set of relevance vectors, for averaging under the algorithm’s randomization. Fig 1 shows (averaged over time) regret with top-1 feedback for SumLoss. Averaged over time 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 3 shows the same graph for DCG. It corroborates the fact that Algorithm 1 can be applied to DCG to get same order of regret.

Fig 4 compares (averaged over time) regret, with top-1 and full information feedback respectively, for SumLoss. 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 (\( \Theta(T^{-1/3}) \)). Fig 4 shows the same comparison for DCG.

11
11 Conclusion

We introduced a novel, interesting feedback model for online ranking of a fixed set of objects 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 for $k = 1$.

The most interesting future extension of this work is to move beyond ranking 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.

References

[1] 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.

[2] Nir Ailon. Improved bounds for online learning over the permutahedron and other ranking polytopes. In AISTATS, 2014.

[3] R. Baeza-Yates, B. Ribeiro-Neto, et al. Modern information retrieval, volume 463. ACM press New York, 1999.

[4] Gábor Bartók, Dean Foster, Dávid Pál, Alexander Rakhlin, and Csaba Szepesvári. Partial monitoring–classification, regret bounds, and algorithms, 2013.
Figure 2: Comparison of average regret over time, for SumLoss, between top-1 feedback and full relevance vector feedback. Best viewed in color. Code available on request.

Figure 3: Average regret for DCG with feedback on top ranked object. Best viewed in color. Code available on request.

[5] Gábor Bartók, Dávid Pál, and Csaba Szepesvári. Minimax regret of finite partial-monitoring games in stochastic environments. Journal of Machine Learning Research-Proceedings Track, pages 133–154, 2011.

[6] Avrim Blum and Yishay Mansour. From external to internal regret. Journal of Machine Learning Research, 2007.

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

[8] Corinna Cortes and Mehryar Mohri. AUC optimization vs. error rate minimization. In NIPS, page 10, 2003.
Figure 4: Comparison of average regret over time, for SumLoss, between top-1 feedback and full relevance vector feedback. *Best viewed in color. Code available on request.*

[9] 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 (ICML-10)*, pages 327–334, 2010.

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

[11] K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems (TOIS)*, pages 422–446, 2002.

[12] 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.

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

[14] Tian Lin, Bruno Abrahao, Robert Kleinberg, and John Lui. Combinatorial partial monitoring game with linear feedback and its applications. In *Proceedings of ICML 2014 Cycle 2*, 2014.

[15] 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.

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

[17] Filip Radlinski, Paul N Bennett, Ben Carterette, and Thorsten Joachims. Redundancy, diversity and interdependent document relevance. In *ACM SIGIR Forum*, pages 46–52. ACM, 2009.
[18] 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.
We provide missing proofs of theorems and extensions that were excluded from the main body of the paper due to space constraints.

12 Regret for SumLoss

12.1 Proof of Lemma 2

Proof. For any \(p \in \Delta\), we have \(\ell_i \cdot p = \sum_{j=1}^{2^m} p_j (\sigma_i \cdot r_j) = \sigma_i \cdot (\sum_{j=1}^{2^m} p_j r_j) = \sigma_i \cdot E_r[r]\), where the expectation is taken w.r.t \(p\) \((p_j \text{ is the } j \text{ th component of } p)\). By dot product rule between 2 vectors, \(\ell_i \cdot p\) is minimized when ranking of objects according to \(\sigma_i\) and expected relevance of objects are in opposite order. That is, the object with highest expected relevance is ranked 1 and so on. Formally, \(\ell_i \cdot p\) is minimized when \(E_r[r(\sigma_i^{-1}(1))] \geq E_r[r(\sigma_i^{-1}(2))] \geq \ldots \geq E_r[r(\sigma_i^{-1}(m))]\).

Thus, for action \(i\), probability cell is defined as \(C_i = \{p \in \Delta : \sum_{j=1}^{2^m} p_j = 1, E_r[r(\sigma_i^{-1}(1))] \geq E_r[r(\sigma_i^{-1}(2))] \geq \ldots \geq E_r[r(\sigma_i^{-1}(m))]\}\).

Let \(\sigma_i^{-1}(k) = a, \sigma_i^{-1}(k+1) = b\). Let it also be true that \(\sigma_j^{-1}(k) = b, \sigma_j^{-1}(k+1) = a\) and \(\sigma_i^{-1}(n) = \sigma_j^{-1}(n), \forall n \neq \{k, k+1\}\). This indicates the sufficient condition stated in Lemma 3 for \(\{\sigma_i, \sigma_j\}\) to be neighbors.

Then, \(C_i \cap C_j = \{p \in \Delta : \sum_{j=1}^{2^m} p_j = 1, E_r[r(\sigma_i^{-1}(1))] \geq \ldots \geq E_r[r(\sigma_i^{-1}(k))] = E_r[r(\sigma_i^{-1}(k+1))] \geq \ldots \geq E_r[r(\sigma_i^{-1}(m))]\}\). Hence, there are two equalities in the non-empty set \(C_i \cap C_j\) and it is an \((2^m - 2)\) dimensional polytope. Hence condition of Definition 4 holds true and \(\{\sigma_i, \sigma_j\}\) are neighbors.

12.2 Proof of Lemma 3

Proof. From Lemma 2, we know that every one of learner’s actions are pareto-optimal and \(C_i\), associated with action \(\sigma_i\), has structure \(C_i = \{p \in \Delta : \sum_{j=1}^{2^m} p_j = 1, E_r[r(\sigma_i^{-1}(1))] \geq E_r[r(\sigma_i^{-1}(2))] \geq \ldots \geq E_r[r(\sigma_i^{-1}(m))]\}\).

Let \(\sigma_i^{-1}(k) = a, \sigma_i^{-1}(k+1) = b\). Let it also be true that \(\sigma_j^{-1}(k) = b, \sigma_j^{-1}(k+1) = a\) and \(\sigma_i^{-1}(n) = \sigma_j^{-1}(n), \forall n \neq \{k, k+1\}\). This indicates the sufficient condition stated in Lemma 3 for \(\{\sigma_i, \sigma_j\}\) to be neighbors.

Then, \(C_i \cap C_j = \{p \in \Delta : \sum_{j=1}^{2^m} p_j = 1, E_r[r(\sigma_i^{-1}(1))] \geq \ldots \geq E_r[r(\sigma_i^{-1}(k))] = E_r[r(\sigma_i^{-1}(k+1))] \geq \ldots \geq E_r[r(\sigma_i^{-1}(m))]\}\). Hence, there are two equalities in the non-empty set \(C_i \cap C_j\) and it is an \((2^m - 2)\) dimensional polytope. Hence condition of Definition 4 holds true and \(\{\sigma_i, \sigma_j\}\) are neighbors.

12.3 Proof of Theorem 4

Proof. We will explicitly show that local observability condition fails by considering the case when number of objects is \(m = 3\). Specifically, action pair \(\{\sigma_1, \sigma_2\}\), in Table 1 and 2 are neighboring actions, using Lemma 3. Moreover, from Lemma 2, since every action \(\sigma_k\) is Pareto-optimal, every \(C_k\) is \(2^m - 1\) dimensional. Since \(C_1 \cap C_2\) is \(2^m - 2\) dimensional (from Def. 4), the neighborhood action set of actions \(\{\sigma_1, \sigma_2\}\) is precisely \(\sigma_1\) and \(\sigma_2\) and contains no other actions. By definition of signal matrices \(S_{\sigma_1}, S_{\sigma_2}\) and entries \(\ell_1, \ell_2\) in Table 1 and 2, we have

\[
\begin{align*}
S_{\sigma_1} &= S_{\sigma_2} = \begin{bmatrix}
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1
\end{bmatrix} \\
\ell_1 - \ell_2 &= \begin{bmatrix}
0 & 1 & -1 & 0 & 0 & 1 & -1 & 0
\end{bmatrix}
\end{align*}
\]

(8)

It is clear that \(\ell_1 - \ell_2 \notin Col(S_{\sigma_1}^\top)\). Hence, Definition 5 fails to hold.
13 Efficient Algorithm for Obtaining Regret

13.1 Proof of Lemma

Proof. We can write \( \hat{r}_t = \sum_{j=1}^{m} r_{ij}(j)e_j \), where \( e_j \) is the standard basis vector along coordinate \( j \).

Then \( E_{i_1, \ldots, i_m}(\hat{r}_t) = \sum_{j=1}^{m} E_{ij}(r_{ij}(j)e_j) = \sum_{j=1}^{m} \sum_{k=1}^{t} \frac{r_k(j)e_j}{t} = r_{1:t}^{\text{avg}} \).

13.2 Proof of Theorem

Proof. Instead of top-1 feedback, assume that at each round, after learner reveals his action, the full relevance vector is revealed to the learner. Then an \( O(\sqrt{T}) \) expected regret for \( \text{SumLoss} \) can be obtained by applying FTPL (Follow the perturbed leader), in the following manner:

At end of every round \( t \), the full relevance vector generated by the adversary is revealed. The relevance vectors are accumulated as \( r_{1:t} = r_{1:t-1} + r_t \), where \( r_{1:s} = \sum_{i=1}^{s} r_i \). A learner’s action (permutation) for round \( t+1 \) is generated by solving \( M(r_{1:t} + p_t) \), where \( p_t \in [0, \eta]^m \) and \( \eta \) is algorithmic (randomization) parameter. It should be noted that \( M(y) = \text{argmin}_{\sigma} \sigma \cdot y \) is simply sorting of \( y \) since \( f(\sigma) = \sigma \) is a monotone function as defined in Sec.3.

The key idea is that FTPL implicitly maintains a distribution over \( m! \) actions (permutations) at beginning of each round, by randomly perturbing the scores of only \( m \) objects: score of each object is sum of (deterministic) accumulated relevance so far and (random) uniform value from \([0, \eta]\). Thus, it bypasses having to maintain explicit weight on each of \( m! \) arms, which is computationally prohibitive. This key property which introduces efficiency in our algorithm is in contrast to the general algorithms based on exponential weights, which have to maintain explicit weights, based on accumulated history, on each action and randomly select an action based on weights.

Now let us look at a variant of the full information problem. The (known) time horizon \( T \) is divided into \( K \) blocks, i.e., \( \{B_1, \ldots, B_K\} \), of equal size \( T/K \). Here, \( B_i = \{(i-1)(T/K) + 1, (i-1)(T/K) + 2, (i-1)(T/K) + 3, \ldots, i(T/K)\} \). While operating in a block, the relevance vectors within the block are accumulated, but not used to generate learner’s actions like in the full information version. Assume at the start of block \( B_i \), there was some relevance vector \( r^i \). Then at each time point in the block, a fresh \( p \in [0, \eta]^m \) is sampled and \( M(r^i + p) \) is solved to generate permutation for next time point. At the end of a block, the average of the accumulated relevance vectors \( (r^{\text{avg}}) \) for the block is used for updation, as \( r^i + r^{\text{avg}} \), to get \( r^{i+1} \) for the next block. The process is repeated for each block. At the beginning of the first block, \( r^1 = \{0\}^m \).

Formally, let the FTPL have an implicit distribution \( \rho_i \) over the permutations at the beginning of block \( B_i \). That is \( \rho_i \in \Delta \), where \( \Delta \) is the probability simplex over \( m! \) actions. Sampling a permutation using \( \rho_i \) at each time point of the block \( B_i \) means sampling a fresh \( p \in [0, \eta]^m \) at every time point \( t \) and solving \( M(s_{1:i-1} + p) \), where \( s_{1:i-1} = \sum_{j=1}^{i-1} s_j \) and \( s_j \) is the average of relevance vectors of block \( B_j \).

Since action \( \sigma \) is generated according to distribution \( \rho \), and in block \( k \), distribution \( \rho_k \) is fixed, we have \( E_{\rho_k} \sum_{t \in [B_k]} \text{SumLoss}(\sigma_t, r_t) = \sum_{t \in [B_k]} \rho_k \cdot \text{SumLoss}(\sigma_t, r_t) \). Thus,
the total expected loss of this variant of the full information problem is:

\[
E_{\rho_1, \ldots, \rho_K} \sum_{t=1}^{T} \text{SumLoss}(\sigma_t, r_t) = \\
E_{\rho_1, \ldots, \rho_K} \sum_{k=1}^{K} \sum_{t \in [B_k]} \text{SumLoss}(\sigma_t, r_t)
\]

(9)

\[
= \sum_{k=1}^{K} \sum_{t \in [B_k]} \rho_k \cdot [\text{SumLoss}(\sigma_1, r_t), \ldots, \text{SumLoss}(\sigma_m, r_t)]
\]

\[
= \sum_{k=1}^{K} \sum_{t \in [B_k]} \rho_k \cdot [\sigma_1 \cdot r_t, \ldots, \sigma_m \cdot r_t]
\]

\[
= \frac{T}{K} \sum_{k=1}^{K} \rho_k \cdot [\sigma_1 \cdot s_k, \ldots, \sigma_m \cdot s_k]
\]

(10)

\[
= \frac{T}{K} E_{\rho_1, \ldots, \rho_k} \sum_{k=1}^{K} \text{SumLoss}(\sigma_k, s_k)
\]

where \( s_k = \sum_{t \in [B_k]} \frac{r_t}{T/K} \). This clearly implies that the variant of the full information problem over \( T \) rounds reduces to full information problem in \( K \) rounds, where at end of every round \( k \in [K] \), \( \rho_k \) is being updated to \( \rho_k + 1 \) by updating \( s_{1:k-1} \) to \( s_{1:k} \).

By the regret bound of FTPL, for \( K \) rounds of full information problem, with \( \epsilon = \sqrt{D/RAK} \), we have:

\[
E_{\rho_1, \ldots, \rho_k} \sum_{k=1}^{K} \text{SumLoss}(\sigma_k, s_k)
\]

\[
\leq \min_{\sigma} \sum_{k=1}^{K} \text{SumLoss}(\sigma, s_k) + 2\sqrt{DRAK}
\]

(11)

\[
= \min_{\sigma} \sum_{t=1}^{T} \sigma \cdot s_k + 2\sqrt{DRAK}
\]

\[
= \min_{\sigma} \sum_{t=1}^{T} \sigma \cdot \frac{r_t}{T/K} + 2\sqrt{DRAK}
\]

where \( D, R, A \) are parameters implicit to the loss under consideration (To be detailed later).

Now, since \( \min_{\sigma} \sum_{t=1}^{T} \sigma \cdot \frac{r_t}{T/K} = \min_{\sigma} \frac{1}{T/K} \sum_{t=1}^{T} \text{SumLoss}(\sigma, r_t) \), combining Eq. 10 and Eq.
we get:

\[ E_{\rho_1, \ldots, \rho_K} \sum_{t=1}^{T} \text{SumLoss}(\sigma_t, r_t) \]

\[ \leq \min_{\sigma} \sum_{t=1}^{T} \text{SumLoss}(\sigma, r_t) + 2 \frac{T}{K} \sqrt{DRAK}. \tag{12} \]

However, in our top-1 feedback model, the learner does not get to see the full relevance vector at each round. Thus, we form the unbiased estimator \( \hat{s}_i \) of \( s_i \), using Lemma. \[ \text{\[8\]} \]. That is, at start of each block, we choose \( m \) time points uniformly at random, and at those time points, we output a random permutation which places each object, in turn, at top. At the end of the block, we form the relevance vector \( \hat{s} \) which is the unbiased estimator of \( s \). Note that using \( \hat{s}_i \) instead of true \( s_i \) makes the distributions \( \rho_i \) themselves random. But significantly, \( \rho_k \) is dependent only on information received up to the beginning of block \( k \) and is independent of the information collected in the block. Thus, for block \( k \), we have:

\[ E_{\rho_k/\hat{s}_1, \hat{s}_2, \ldots, \hat{s}_{k-1}} \sum_{t \in [B_k]} \text{SumLoss}(\sigma_t, r_t) \]

\[ = \frac{T}{K} E_{\rho_k/\hat{s}_1, \hat{s}_2, \ldots, \hat{s}_{k-1}} \text{SumLoss}(\sigma_k, s_k) \quad \text{(From Eq.10)} \tag{13} \]

\[ = \frac{T}{K} E_{\rho_k/\hat{s}_1, \hat{s}_2, \ldots, \hat{s}_{k-1}} E_{\hat{s}_k} \text{SumLoss}(\sigma_k, \hat{s}_k) \quad \text{(Linearity property)} \]

Crucially, since random distribution \( \rho_k \) is independent of \( \hat{s}_k \), the expectations are exchangeable and hence we have the following equation

\[ E_{\hat{s}} E_{\rho/\hat{s}} \left[ \sum_{t=1}^{T} \text{SumLoss}(\sigma_t, r_t) \right] \]

\[ = \frac{T}{K} E_{\hat{s}} E_{\rho/\hat{s}} \left[ \sum_{k=1}^{K} \text{SumLoss}(\sigma_k, \hat{s}_k) \right] \quad \text{(From Eq.10)} \]

\[ \leq \frac{T}{K} \left\{ E_{\hat{s}} \left[ \min_{\sigma} \sum_{k=1}^{K} \sigma \cdot \hat{s}_k \right] + 2 \sqrt{DRAK} \right\} \quad \text{(From Eq.11)} \]

\[ \leq \frac{T}{K} \left\{ \min_{\sigma} \sum_{k=1}^{K} \sigma \cdot s_k + 2 \sqrt{DRAK} \right\} \quad \text{(Jensen’s Inequality)} \]

\[ \leq \min_{\sigma} \sum_{t=1}^{T} \sigma_i \cdot r_t + 2 \frac{T}{K} \sqrt{DRAK} \]

\[ = \min_{\sigma} \sum_{t=1}^{T} \text{SumLoss}(\sigma, r_t) + 2 \frac{T}{K} \sqrt{DRAK} \tag{14} \]

However, since in each block, \( m \) rounds are reserved for exploration, which do not follow from distribution \( \rho \), the total expected loss is the sum of expected loss from exploitation and exploration.
Exploration leads to an extra regret of $RmK$, where $R$, as has been stated before, is an implicit parameter depending on the loss under consideration. The extra regret is because loss in each of the exploration rounds is $\leq R$ (explained later), total of $m$ rounds in each block and total of $K$ blocks. Thus, overall regret is:

$$E[\sum_{t=1}^{T} \text{SumLoss}(\sigma_t, r_t)]$$

$$\leq E_{\rho/i}[\sum_{t=1}^{T} \text{SumLoss}(\sigma_t, r_t)] + RmK \quad \text{(15)}$$

$$\Rightarrow E[\sum_{t=1}^{T} \text{SumLoss}(\sigma_t, r_t)] - \min_{\sigma} \sum_{t=1}^{T} \text{SumLoss}(\sigma, r_t)$$

$$\leq RmK + 2\frac{T}{K} \sqrt{DRAK}$$

Now, optimizing over $K$, we get $K = (DA/R)^{1/3}(T/m)^{2/3}$, and

$$E[\sum_{t=1}^{T} \text{SumLoss}(\sigma_t, r_t)] \leq \min_{\sigma} \sum_{t=1}^{T} \text{SumLoss}(\sigma, r_t)$$

$$+ O(m^{1/3}R^{2/3}(DA)^{1/3}T^{2/3}) \quad \text{(16)}$$

Now, by definition of $D, R$ and $A$ from [13], $D$ is upper bound on $\ell_1$ norm on vectors in learner’s action space, $R$ is upper bound on dot product of vectors in learner’s and adversary’s action space and $A$ is upper bound on $\ell_1$ norm on vectors in adversary’s action space. Thus, for $\text{SumLoss}$, we have

$$D = \sum_{i=1}^{m} \sigma(i) = O(m^2), \quad R = \sum_{i=1}^{m} \sigma(i)r(i) = O(m^2), \quad A = \sum_{i=1}^{m} r(i) = O(m).$$

Plugging in these values gives us result of Theorem. 7.

14 Regret Bounds for DCG and Prec@k

14.1 Extension of Results of SumLoss to DCG

We give pointers in the direction of proving the following results: a) Local observability condition fails to hold for DCG which, in combination with Theorem 4 in [4], proves that the lower bound on regret of DCG is $\Omega(T^{2/3})$. b) The efficient algorithm of Sec.7 applies to DCG, with regret of $O(T^{2/3})$. Thus, the minimax regret of DCG is $\Theta(T^{2/3})$. All results are applicable to non-binary relevance vectors. This allows us to skip the proof of global observability, which is complicated for non-binary relevance vectors.

Let adversary be able to choose $r \in \{0, 1, \ldots, n\}^m$. Then, from definition of $DCG$ in Sec.3, it is clear $DCG = f(\sigma) \cdot g(r)$. $f(\sigma)$ and $g(r)$ has already been defined for DCG. Both are composed of $m$ copies of univariate, monotonic, scalar valued function, where for $f(\cdot)$, it is monotonically decreasing whereas for $g(\cdot)$, it is increasing.
With slight abuse of notations, the Loss matrix \( L \) implicitly means Gain matrix, where entry in cell \( \{i, j\} \) of \( L \) is \( f(\sigma_i) \cdot g(r_j) \). The feedback matrix \( H \) remains the same. In Definition 1., learner action \( i \) is optimal if \( \ell_i \cdot p \geq \ell_j \cdot p, \forall j \neq i \).

In Definition 2., the maximum number of distinct elements that can be in a row of \( H \) is \( n + 1 \). The signal matrix now becomes \( S_i \in \{0, 1\}^{(n+1)\times 2^n} \), where \((S_i)_{k, \ell} = 1(H_{k, \ell} = k - 1)\).

Local Observability:

In Lemma 2, proposed for SumLoss, \( \ell_i \cdot p \) equates to \( f(\sigma) \cdot E_r[g(r)] \). From definition of DCG, and from the structure and properties of \( f(\cdot) \), \( g(\cdot) \), it is clear that \( \ell_i \cdot p \) is maximized under the same condition, i.e. \( E_r[r(\sigma_i^{-1}(1))] \geq E_r[r(\sigma_i^{-1}(2))] \geq \ldots \geq E_r[r(\sigma_i^{-1}(m))] \). Thus, all actions are pareto-optimal.

Careful observation of Lemma 3 shows that it is directly applicable to DCG, in light of extension of Lemma 2 to DCG.

Finally, just like in SumLoss, simple calculations with \( m = 3 \) and \( n = 1 \), in light of Lemma 2 and 3, show that local observability condition fails to hold.

**Implementation of the Efficient Algorithm:**

The only change in Algorithm 1 which allows extension to DCG with non-binary relevance is that relevance values will enter the algorithm via the transformation \( g^*(\cdot) \). That is, every component of relevance vector \( r \), i.e. \( r(i) \), will become \( 2^{r(i)} - 1 \). Every operation of Algorithm 1 will happen on the transformed relevance vectors. It is very easy to see that every step in algorithm will be valid by just considering the transformed relevance vectors to be some new relevance vectors with magnified relevance values. The fact that \( r \) was binary valued in SumLoss played no role in the analysis of the algorithm or Lemma 8. The properties which allowed the extension was that \( g(\cdot) \) is composed of univariate, monotonic, scalar valued functions and \( DCG(\sigma, r) \) is a linear function of \( f(\sigma) \) and \( g(r) \).

It is also interesting to note that \( M(y) = \arg\max_{\sigma} f(\sigma) \cdot y = \arg\min_{\sigma} \sigma \cdot y \). Thus, no changes in the algorithm is required, other than simple transformation of relevance values.

**Proof of Theorem 9**

Following the proof of Theorem 7, modified for DCG, Eq. 16 gives (for DCG):

\[
E(\sum_{t=1}^{T} DCG(\sigma_t, r_t)) \geq \max_{\sigma} \sum_{t=1}^{T} DCG(\sigma, r_t) - O(\sigma^{1/3}R^{2/3}DA^{1/3}T^{2/3})
\]

For DCG, \( D = \sum_{i=1}^{m} f^*(\sigma(i)) = O(m), R = \sum_{i=1}^{m} f^*(\sigma(i))g^*(r(i)) = O(m(2^n - 1)), A = \sum_{i=1}^{m} g^*(r(i)) = O(m(2^n - 1)) \) and hence the regret is \( O((2^n - 1)m^{5/3}T^{2/3}) \).

### 14.2 Extension of Results of SumLoss to Prec@k

Since \( \text{Prec@k} = f(\cdot) \cdot r \), with \( f(\cdot) \) having properties enlisted in Sec. 3, all results of SumLoss trivially extend to Prec@k, except results on local observability. The reason is that while \( f(\cdot) \) of SumLoss is strictly monotonic, \( f(\cdot) \) of Prec@k is monotonic but not strict. The gain function depends only on the objects in the top-k position of the ranked list, irrespective of the order. A careful analysis shows that Lemma 3 fails to extend to the case of Prec@k. Thus, we cannot define the neighboring action set of the Pareto optimal action pairs, and hence cannot prove or disprove local observability. The structure of neighbors in Prec@k remains an open question.

21
However, the non-strict monotonicity of Prec@k is required for solving \( M(y) = \arg\max_{\sigma} f(\sigma) \cdot y \) efficiently.

**Proof of Corollary 10**
Following the proof of Theorem 7, modified for Prec@k, Eq.16 gives (for Prec@k):

\[
E\left[ \sum_{t=1}^{T} \text{Prec@k}(\sigma_t, r_t) \right] \geq \max_{\sigma} \sum_{t=1}^{T} \text{Prec@k}(\sigma, r_t) - O(m^{1/3}R^{2/3}(DA)^{1/3}T^{2/3})
\]

for Prec@k, \( D = \sum_{i=1}^{k} f_s(\sigma(i)) = O(k) \), \( R = \sum_{i=1}^{m} f_s(\sigma(i))g_s(r(i)) = O(k) \), \( A = \sum_{i=1}^{m} r(i) = O(m) \) and hence the regret is \( O(km^{2/3}T^{2/3}) \).

### 15 Non-Existence of sublinear regret bounds for NDCG, MAP and AUC- Extensions

We show via simple calculations that for the case \( m = 3 \), global observability condition fails to hold for NDCG, when relevance vectors are restricted to take binary values. The intuition behind failure to satisfy global observability condition is that the \( \text{NDCG}(\sigma, r) = f(\sigma) \cdot g(r) \), where \( g(r) = r/Z(r) \) (See Sec.3). Thus, \( g(\cdot) \) cannot be by univariate, scalar valued functions. This makes it impossible to write the difference between two rows as linear combination of columns of (transposed) signal matrices.

**Proof.** The following calculations can be easily done:

The first and last row of Table 1, when calculated for NDCG, is:

\[
\ell_{\sigma_1} = [1, 1/2, \log_2 2/\log_2 3, (1 + \log_2 3/2)/(1 + \log_2 3),
\]

\[
1, 3/(2(1 + 1/\log_2 3)), 1, 1]
\]

\[
\ell_{\sigma_6} = [1, 1, \log_2 2/\log_2 3, 1, 1/2, 3/(2(1 + 1/\log_2 3)),
\]

\[
(1 + \log_2 3/2)/(1 + \log_2 3), 1]
\]

We remind once again that NDCG is a gain function, as opposed to SumLoss.

The difference between the two vectors is:

\[
\ell_{\sigma_1} - \ell_{\sigma_6} = [0, 1/2, 0, \log_2 3/(2(1 + \log_2 3)),
\]

\[
-1/2, 0, -\log_2 3/(2(1 + \log_2 3)), 0].
\]

The signal matrices are same as SumLoss:

\[
S_{\sigma_1} = S_{\sigma_2} = \begin{bmatrix}
1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 1
\end{bmatrix}
\]

\[
S_{\sigma_3} = S_{\sigma_5} = \begin{bmatrix}
1 & 1 & 0 & 0 & 1 & 1 & 0 \\
0 & 0 & 1 & 1 & 0 & 0 & 1
\end{bmatrix}
\]

\[
S_{\sigma_4} = S_{\sigma_6} = \begin{bmatrix}
1 & 0 & 1 & 0 & 1 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & 1 & 0
\end{bmatrix}
\]
It can now be easily checked that $\ell_{\sigma_1} - \ell_{\sigma_6}$ does not lie in the (combined) column span of the (transposed) signal matrices.

We show similar calculations for MAP and AUC:

**MAP:**

We once again take $m = 3$. The first and last row of Table 1, when calculated for MAP, is:

$\ell_{\sigma_1} = [1, 1/3, 1/2, 7/12, 1, 5/6, 1, 1]$  
$\ell_{\sigma_6} = [1, 1, 1/2, 1, 1/3, 5/6, 7/12, 1]$  

Like NDCG, MAP is also a gain function.
The difference between the two vectors is:

$\ell_{\sigma_1} - \ell_{\sigma_6} = [0, -2/3, 0, -5/12, 2/3, 0, 5/12, 0]$.  

The signal matrices are same SumLoss:

$S_{\sigma_1} = S_{\sigma_2} = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$  
$S_{\sigma_3} = S_{\sigma_5} = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \end{bmatrix}$  
$S_{\sigma_4} = S_{\sigma_6} = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$  

It can now be easily checked that $\ell_{\sigma_1} - \ell_{\sigma_6}$ does not lie in the (combined) column span of the (transposed) signal matrices.

**AUC:**

For AUC, we will show the calculations for $m = 4$. This is because global observability does hold with $m = 3$, as the normalizing factor for all relevance vectors with mixture of 0 and 1 is same. The normalizing factor changes from $m = 4$ onwards; hence global observability fails.

Table 1 will be extended since $m = 4$. Instead of illustrating the full table, we point out the important facts about the loss matrix table with $m = 4$ for AUC.

The 24 relevance vectors heading the columns are:

$r_1 = 0000$, $r_2 = 0001$, $r_3 = 0010$, $r_4 = 0100$, $r_5 = 1000$, $r_6 = 0011$, $r_7 = 0101$, $r_8 = 1010$, $r_9 = 0110$, $r_{10} = 1010$, $r_{11} = 1100$, $r_{12} = 0111$, $r_{13} = 1011$, $r_{14} = 1101$, $r_{15} = 1110$, $r_{16} = 1111$.  

We will calculate the losses of 1st and last (24th) action, where $\sigma_1 = 1234$ and $\sigma_{24} = 4321$.  

$\ell_{\sigma_1} = [0, 1, 2/3, 1/3, 0, 1, 3/4, 1/2, 1/2, 1/4, 0, 1, 2/3, 1/3, 0, 0]$  
$\ell_{\sigma_{24}} = [0, 0, 1/3, 2/3, 1, 0, 1/4, 1/2, 1/2, 3/4, 1, 0, 1/3, 2/3, 1, 0]$  

AUC, like SumLoss, is a loss function.
The difference between the two vectors is:

$\ell_{\sigma_1} - \ell_{\sigma_{24}} = [0, 1, 1/3, -1/3, -1, 1, 1/2, 0, 0, -1/2, -1, 1, 1/3, -1/3, -1, 0]$.  

23
The signal matrices for AUC with \( m = 4 \) will be slightly different. This is because there are 24 signal matrices, corresponding to 24 actions. However, every 6 action will have same signal matrix. That is, all 6 permutations which places object 1 first will have same signal matrix, 6 permutations which places object 2 first will have same signal matrix and so on. For simplicity, we denote the signal matrices as \( \{ S_1, S_2, S_3, S_4 \} \), where \( S_i \) corresponds to signal matrix where object \( i \) is placed at top:

\[
S_1 = \begin{bmatrix}
1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1
\end{bmatrix}
\]

\[
S_2 = \begin{bmatrix}
1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1
\end{bmatrix}
\]

\[
S_3 = \begin{bmatrix}
1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1
\end{bmatrix}
\]

\[
S_4 = \begin{bmatrix}
1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1
\end{bmatrix}
\]

It can now be easily checked that \( \ell_{\sigma_1} - \ell_{\sigma_{24}} \) does not lie in the (combined) column span of \( S_1, S_2, S_3, S_4 \).