

Online Rank Aggregation

Shota Yasutake

Panasonic System Networks Co., Ltd

YASUTAKE.SHOUTA@JP.PANASONIC.COM

Kohei Hatano

HATANO@INF.KYUSHU-U.AC.JP

Eiji Takimoto

EIJI@INF.KYUSHU-U.AC.JP

Masayuki Takeda

TAKEDA@INF.KYUSHU-U.AC.JP

Kyushu University

Editor: Steven C.H. Hoi and Wray Buntine

Abstract

We consider an online learning framework where the task is to predict a permutation which represents a ranking of n fixed objects. At each trial, the learner incurs a loss defined as Kendall tau distance between the predicted permutation and the true permutation given by the adversary. This setting is quite natural in many situations such as information retrieval and recommendation tasks. We prove a lower bound of the cumulative loss and hardness results. Then, we propose an algorithm for this problem and prove its relative loss bound which shows our algorithm is close to optimal.

Keywords: online learning, ranking, rank aggregation, permutation

1. Introduction

The *rank aggregation* problem have gained much attention due to developments of information retrieval on the Internet, online shopping stores, recommendation systems and so on. The problem is, given m permutations of n fixed elements, to find a permutation that minimizes the sum of “distances” between itself and each given permutation. Here, each permutation represents a ranking over n elements. So, in other words, the ranking aggregation problem is to find an “average” ranking which reflects the characteristics of given rankings. In particular, the optimal ranking is called Kemeny optimal (Kemeny, 1959; Kemeny and Snell, 1962) when the distance is defined as Kendall tau distance (which we will define later). From now on, we only consider Kendall tau distance as our distance measure.

The ranking aggregation problem is a classical problem in social choice literature which deals with voting and so on (Borda, 1781; Condorcet, 1785). These days, the rank aggregation problem also arises in information retrieval tasks such as combining several search results given by different search engines. The rank aggregation problem is being studied extensively in theoretical computer science (Dwork et al., 2001; Fagin et al., 2003; Andoni et al., 2008). It is known that the rank aggregation problem is NP-hard (Bartholdi et al., 1989), even when $m \geq 4$ (Dwork et al., 2001).

Some approximation algorithms are known as well. For example, Ailon et al. proposed a $11/7$ -approximation algorithm (Ailon et al., 2008). Further, Kenyon-Mathieu and Schudy proposed a PTAS (polynomial time approximation scheme) which runs in doubly exponen-

tial in the precision parameter $\varepsilon > 0$ (Kenyon-Mathieu and Schudy, 2007). Ailon also gives algorithms for aggregation of partial rankings (Ailon, 2008).

In this paper, we consider an online version of the ranking aggregation problem, which we call “online rank aggregation”. This problem is about online prediction of permutations. Let S_n be the set of all permutations of n fixed elements. Then the online rank aggregation problem consists of the following protocol for each trial t :

1. The learner predicts a permutation $\hat{\sigma}_t \in S_n$.
2. The adversary gives the learner the true permutation $\sigma_t \in S_n$.
3. The learner receives the loss $d(\sigma_t, \hat{\sigma}_t)$, Kendall tau distance between σ_t and $\hat{\sigma}_t$.

The goal of the learner is to minimize the *cumulative loss bound* $\sum_{t=1}^T d(\sigma_t, \hat{\sigma}_t)$. This online protocol captures problems of predicting rankings naturally. For example, one might want to predict a ranking of the next week (i.e., permutation) over items and so on given rankings of the past weeks. Also, this protocol is motivated by an online recommendation task where the user’s past preferences over items are given as permutations and the recommend items using predicted rankings.

First of all, we derive a lower bound of the cumulative loss of any learning algorithm for online rank aggregation. More precisely, we show that there exists a probabilistic adversary such that for any learning algorithm for online rank aggregation, the cumulative loss is at least

$$\min_{\sigma \in S_n} \sum_{t=1}^T d(\sigma_t, \sigma) + \Omega(n^2 \sqrt{T}).$$

Then we prove hardness results. In particular, we prove that no randomized polynomial time algorithm whose cumulative loss bound matches the lower bound, under the common assumption that $\text{NP} \not\subseteq \text{BPP}$. Further, we show that, under the same assumption, there exists no fully randomized polynomial approximation scheme (FPRAS) with cumulative loss bound $(1 + \varepsilon) \min_{\sigma \in S_n} \sum_{t=1}^T d(\sigma_t, \sigma) + O(n^2 \sqrt{T})$, where FPRAS is a polynomial time algorithm runs in $1/\varepsilon$. Therefore, the cumulative loss bound of our algorithm is close to the best one achieved by polynomial time algorithms.

On the other hand, by using Kakade et al’s offline-online converter Kakade et al. (2007) and the PTAS algorithm for rank aggregation Kenyon-Mathieu and Schudy (2007), it can be shown that there exists an algorithm, for any $\varepsilon > 0$, its cumulative loss bound is $(1 + \varepsilon) \min_{\sigma \in S_n} \sum_{t=1}^T d(\sigma_t, \sigma) + O(n^2 \sqrt{T})$, with its running time is $\text{poly}(T)n^{\tilde{O}(1/\varepsilon)^6}$.

Finally, we propose an efficient algorithm for online rank aggregation. For the algorithm which we call PermRank, we prove its expected cumulative loss bound is at most

$$\frac{3}{2} \min_{\sigma \in S_n} \sum_{t=1}^T d(\sigma_t, \sigma) + O(n^2 \sqrt{T}).$$

The running time is that for solving a convex optimization problem with $O(n^2)$ variables and $O(n^3)$ linear constraints, which does not depends on T . In addition, a version of our algorithm runs in time $O(n^2)$ with a weaker loss bound that has factor 4 instead of 3/2 (omitted). We summarize the cumulative loss bounds in Table 1.

Table 1: The cumulative loss bounds $a \min_{\sigma \in S_n} \sum_{t=1}^T d(\sigma_t, \sigma) + O(n^2 \sqrt{T})$.

factor a	time complexity per iteration	
1 (optimal) ($1 + \varepsilon$)	poly time implies $NP = BPP$ $poly(n, T)$	our result combination of Kakade et al. (2007) and Kenyon-Mathieu and Schudy (2007)
3/2	$poly(n)$	our result

There are other related researches. As there have been extensive researches on online learning with experts (e.g., Weighted Majority ([Littlestone and Warmuth, 1994](#)) and Aggregating Algorithm ([Vovk, 1990](#))), it is natural for us to apply existing algorithms for the online rank aggregation problem.

First of all, a naive method would be to apply Hedge Algorithm ([Freund and Schapire, 1997](#)) with $n!$ possible permutations as experts. In this case, we can prove that the cumulative loss bound is at most

$$\min_{\sigma \in S_n} \sum_{t=1}^T d(\sigma_t, \sigma) + O\left(n^2 \sqrt{T n \ln n}\right).$$

The disadvantage of this approach is that the running time at each trial is $O(n!)$.

Next, let us consider PermELearn ([Helmbold and Warmuth, 2009](#)). Although this algorithm is not designed to deal with Kendall tau distance, it can use Spearman’s footrule, another distance measure for permutations. It is well known that the following relationship holds for Kendall tau distance d and Spearman’s footrule d_F ([Diaconis and Graham, 1977](#)): $d(\sigma, \sigma') \leq d_F(\sigma, \sigma') \leq 2d(\sigma, \sigma')$. So, by using this relationship, we can prove that the expected cumulative loss bound of PermELearn is at most

$$2 \min_{\sigma \in S_n} \sum_{t=1}^T d(\sigma_t, \sigma) + O\left(n^2 \sqrt{T \ln n}\right).$$

Its running time per trial is $\tilde{O}(n^6)$ ¹.

Finally, we show some experimental results on synthetic data sets. In our experiments, our algorithm PermRank performs much better than Hedge algorithm with permutations as experts and PermELearn.

2. Preliminaries

Let n be a fixed integer s.t. $n \geq 1$, and we denote $[n] = \{1, \dots, n\}$. Let S_n be the set of permutations on $[n]$. The *Kendall tau distance* $d(\sigma_1, \sigma_2)$ between permutations $\sigma_1, \sigma_2 \in S_n$ is defined as $d(\sigma_1, \sigma_2) = \sum_{i,j=1}^n I(\sigma_1(i) > \sigma_1(j) \wedge \sigma_2(i) < \sigma_2(j))$, where $I(\cdot)$ is the indicator function, i.e., $I(\text{true}) = 1$ and $I(\text{false}) = 0$. That is, Kendall tau distance between

1. Main computation in PermELearn is normalization of probability matrices called Sinkhorn balancing. For this procedure, there is an approximation algorithm running in time $O(n^6 \ln(n/\varepsilon))$, where $\varepsilon > 0$ is a precision parameter ([Balakrishnan et al., 2004](#)).

two permutations is the total number of pairs of elements for which the orders in two permutations disagree. By definition, it holds that $0 \leq d(\sigma_1, \sigma_2) \leq n(n-1)/2$, and it is known that Kendall tau distance satisfies the conditions of metric. The *Spearman's footrule* between two permutations $\sigma_1, \sigma_2 \in S_n$ is defined as $d_F(\sigma_1, \sigma_2) = \sum_{i=1}^n |\sigma_1(i) - \sigma_2(i)|$. It is shown that the following relationship holds (Diaconis and Graham, 1977): $d(\sigma_1, \sigma_2) \leq d_F(\sigma_1, \sigma_2) \leq 2d(\sigma_1, \sigma_2)$.

Let $N = n(n-1)/2$. A *comparison vector* \mathbf{q} is a vector in $\{0, 1\}^N$. We define the following mapping $\phi : S_n \rightarrow [0, 1]^N$ which maps a permutation to a comparison vector: For $i, j \in [n]$ s.t. $i \neq j$, $\phi(\sigma)_{ij} = 1$ if $\sigma(i) < \sigma(j)$, and $\phi(\sigma)_{ij} = 0$, otherwise. Then note that the Kendall tau distance between two permutations is represented as 1-norm distance between corresponding comparison vectors, i.e., $d(\sigma_1, \sigma_2) = \|\phi(\sigma_1) - \phi(\sigma_2)\|_1$, where 1-norm $\|\mathbf{x}\|_1 = \sum_{i=1}^N |x_i|$. For example, for a permutation $\sigma = (1, 3, 2)$, the corresponding comparison vector is given as $\phi(\sigma) = (1, 1, 0)$. Note that there are 2^N possible comparison vectors whereas there are only $n!$ possible permutations. So, in general, for some comparison vectors, there is no corresponding permutation. For example, the comparison vector $(1, 0, 1)$ represents that $\sigma(1) > \sigma(2), \sigma(2) > \sigma(3), \sigma(3) > \sigma(1)$, for which no permutation σ exists. In particular, if a comparison vector $\mathbf{q} \in \{0, 1\}^N$ has a corresponding permutation, we say that \mathbf{q} is consistent. We denote $\phi(S_n)$ as the set of consistent comparison vectors in $\{0, 1\}^N$.

For $p, q \in [0, 1]$, the *binary relative entropy* $\Delta_2(p, q)$ between p and q is defined as $\Delta_2(p, q) = p \ln \frac{p}{q} + (1-p) \ln \frac{1-p}{1-q}$. Further, we extend the definition of the binary relative entropy for vectors in $[0, 1]^N$. That is, for any $\mathbf{p}, \mathbf{q} \in [0, 1]^N$, the binary relative entropy is given as $\Delta_2(\mathbf{p}, \mathbf{q}) = \sum_{i=1}^N \Delta_2(p_i, q_i)$.

3. Lower bound

In this section, we derive a $\Omega(n^2\sqrt{T})$ lower bound of the cumulative loss for online rank aggregation. In particular, our lower bound is obtained when the adversary is probabilistic.

Theorem 1 *For any online prediction algorithms of permutations and any integer $T \geq 1$, there exists a sequence $\sigma_1, \dots, \sigma_T$ such that*

$$\sum_{t=1}^T d(\sigma_t, \hat{\sigma}_t) \geq \min_{\sigma \in S_n} \sum_{t=1}^T d(\sigma_t, \sigma) + \Omega(n^2\sqrt{T}). \tag{1}$$

Proof The proof partly follows a well known technique in (Cesa-Bianchi et al., 1997; Cesa-Bianchi and Lugosi, 2006). We consider the following strategy of the adversary: At each trial t , give the learning algorithm either the permutation $\sigma_t = \sigma^1 = (1, \dots, n)$ or $\sigma_t = \sigma^0 = (n, n-1, \dots, 1)$ randomly with probability half. Note that the corresponding comparison vectors are $\phi(\sigma^0) = (0, \dots, 0)$ and $\phi(\sigma^1) = (1, \dots, 1)$, respectively.

Then, for any $t \geq 1$ and any permutation $\hat{\sigma}_t$, $\mathbf{E}[d(\sigma_t - \hat{\sigma}_t)] = \binom{n}{2}/2$. This implies that the expected cumulative loss of any learning algorithm is exactly $\frac{\binom{n}{2}}{2}T$, because of the linearity of the expectation.

Next, we consider the expected cumulative loss of the best of σ^0 and σ^1 , that is, $\mathbf{E} \left[\min_{i=0,1} \sum_{t=1}^T d(\sigma_t, \sigma^i) \right]$. By our construction of the adversary, this expectation is re-

duced to

$$\mathbf{E} \left[\min_{p=0,1} \sum_{t=1}^T d(\sigma_t, \sigma^p) \right] = \binom{n}{2} \mathbf{E}_{y_1, \dots, y_T} \left[\min_{p=0,1} \sum_{t=1}^T |p - y_t| \right],$$

where y_1, \dots, y_T are independent random $\{0, 1\}$ -variables. The above expectation can be further written as

$$\begin{aligned} \binom{n}{2} \mathbf{E}_{y_1, \dots, y_T} \left[\min_{p=0,1} \sum_{t=1}^T |p - y_t| \right] &= \frac{\binom{n}{2} T}{2} - \frac{\binom{n}{2}}{2} \mathbf{E}_{y_1, \dots, y_T} [|(\# \text{ of 0s}) - (\# \text{ of 1s})|] \\ &= \frac{\binom{n}{2} T}{2} - \frac{\binom{n}{2}}{2} \mathbf{E}_{\delta_1, \dots, \delta_T} \left[\left| \sum_{t=1}^T \delta_t \right| \right], \end{aligned}$$

where $\delta_1, \dots, \delta_T$ are ± 1 -valued random independent variables such that $\Pr\{\delta_t = 1\} = \Pr\{\delta_t = -1\} = 1/2$. Note that, by the central limit theorem, the distribution of $\sum_{t=1}^T \delta_t$ converges to the normal distribution $N(0, \sqrt{T})$. So, for sufficiently large T , $\Pr\{|\sum_{t=1}^T \delta_t| \geq \sqrt{T}\}$ is larger than some constant. Therefore, the second term in the last equality is bounded as $-\binom{n}{2} \Omega(\sqrt{T})$.

Thus, we have $\mathbf{E} \left[\sum_{t=1}^T d(\sigma_t, \hat{\sigma}_t) - \min_{p=0,1} \sum_{t=1}^T d(\sigma_t, \sigma^p) \right] \geq \Omega(n^2 \sqrt{T})$. So, there exists a sequence $\sigma_1, \dots, \sigma_T$ such that

$$\sum_{t=1}^T d(\sigma_t, \hat{\sigma}_t) \geq \min_{p=0,1} \sum_{t=1}^T d(\sigma_t, \sigma^p) + \Omega(n^2 \sqrt{T}) \geq \min_{\sigma \in S_n} \sum_{t=1}^T d(\sigma_t, \sigma) + \Omega(n^2 \sqrt{T}).$$

■

Note that, by Corollary 5, the cumulative loss of the second version of PermRank is close to our lower bound.

4. Hardness

In this section, we discuss the hardness of online prediction with the optimal cumulative loss bound which matches the lower bound (1). We will show that existence of a randomized polynomial time prediction algorithm with the optimal cumulative loss bound implies a randomized polynomial algorithm for the rank aggregation, which is NP-hard (Bartholdi et al., 1989; Dwork et al., 2001). A formal statement is given as follows:

Theorem 2 *Under the assumption that $NP \not\subseteq BPP$, there is no randomized polynomial time algorithm whose cumulative loss bound matching the lower bound (1) for the online aggregation problem.*

Proof We assume a randomized polynomial time online algorithm A with the optimal cumulative loss bound. Given m fixed permutations, we choose a permutation uniformly randomly among them and we run A on the chosen permutation. We repeat this procedure for $T = cm^2 n^4$ times repeatedly, where c is a constant such that the average expected loss

of A w.r.t. m permutation is at most that of the best permutation plus $1/(4m)$. Then, we pick up a permutation randomly among predicted permutations $\hat{\sigma}_1, \dots, \hat{\sigma}_T$. We call this permutation as the representative permutation. Note that expected average loss of the representative permutation is at most that of the best permutation plus $1/(4m)$.

Now, we repeat this procedure for k times and get $k = O(n^4 m^2)$ representative permutations. By using Hoeffding's bound, with probability at least, say, $2/3$, the best among k representative ones has the best average loss plus $1/(2m)$. Note that, since Kendall tau distance takes integers in $[0, n(n-1)/2]$, the average loss of the best representative takes values in $\{0, 1/m, 2/m, \dots, n(n-1)T/2\}$. So, the average loss of the best representative is the same with that of the best permutation. Therefore, we can find the best permutation in polynomial time in n and m with probability at least $2/3$. Since rank aggregation is NP-hard, this implies that $NP \subseteq BPP$. ■

Now we consider the possibility of fully polynomial time randomized approximation schemes (FPRAS) with cumulative loss bound $(1 + \varepsilon) \min_{\sigma \in S_n} \sum_{t=1}^T d(\sigma, \sigma_t) + O(n^2 \sqrt{T})$, whose running time is polynomial in n, T and $1/\varepsilon$. We say that such a FPRAS has $(1 + \varepsilon)$ -approximate optimal cumulative loss bound. Then, note that if we set $\varepsilon = 1/\sqrt{T}$, then its cumulative loss bound becomes indeed optimal. This implies the following corollary.

Corollary 3 *Under the assumption that $NP \neq BPP$, there is no FPRAS with $(1 + \varepsilon)$ -approximate optimal cumulative loss bound for the online aggregation problem.*

Therefore, it is hard to improve the factor $1 + \varepsilon$ for arbitrary given $\varepsilon > 0$.

5. Our algorithm

In this section we propose our algorithm PermRank. Our idea behind PermRank consists of two parts. The first idea is that we regard a permutation as a $N(= n(n-1)/2)$ dimensional comparison vector and deal with a problem of predicting comparison vectors. More precisely, we consider a Bernoulli trial model for each component ij of a comparison vector. In other words, for each component ij , we assume a biased coin for which head appears with probability p_{ij} , and we estimate each parameter p_{ij} in an online fashion.

The second idea is how we generate a permutation from the estimated comparison vector. As we mentioned earlier, for a given comparison vector, there might not exist a corresponding permutation. To overcome this situation, we use a variant of KWIKSORT algorithm proposed by Ailon et al. (Ailon et al., 2008), LPKWIKSORT_{*h*} (Ailon, 2008). Originally, KWIKSORT is used to solve the rank aggregation problem. The basic idea of KWIKSORT is to sort elements in a brute-force way by looking at local pairwise order only. We will show later that by using LPKWIKSORT_{*h*} we can obtain a permutation whose corresponding comparison vector is close enough to the estimated comparison vector.

The algorithm uses LPKWIKSORT_{*h*} and projection techniques which are now standards in online learning researches (see, e.g., Herbster and Warmuth (2001); Helmbold and Warmuth (2009)). More precisely, after the update (and before applying LPKWIKSORT_{*h*}), PermRank projects the updated vector onto the set of probability vectors satisfying triangle inequalities: $p_{ij} \leq p_{ik} + p_{kj}$ for any $i, j, k \in [n]$, where $p_{ij} = 1 - p_{ji}$. Note that any consistent

Algorithm 1 PermRank

1. Let $\mathbf{p}_1 = (\frac{1}{2}, \dots, \frac{1}{2}) \in [0, 1]^N$.
2. For $t = 1, \dots, T$
 - (a) Predict a permutation $\hat{\sigma}_t = \mathbf{LPKWIKSORT}_h(\mathbf{p}_t)$.
 - (b) Get a true permutation σ_t and let $\mathbf{y}_t = \phi(\sigma_t)$.
 - (c) Update $\mathbf{p}_{t+\frac{1}{2}}$ as

$$p_{t+\frac{1}{2},ij} = \frac{p_{t,ij}e^{-\eta(1-y_{t,ij})}}{(1-p_{t,ij})e^{-\eta y_{t,ij}} + p_{t,ij}e^{-\eta(1-y_{t,ij})}}.$$

- (d) Let \mathbf{p}_{t+1} be the projection of $\mathbf{p}_{t+\frac{1}{2}}$ onto the set of points satisfying triangle inequalities. That is,

$$\begin{aligned} \mathbf{p}_{t+1} &= \arg \inf_{\mathbf{p}} \Delta_2(\mathbf{p}, \mathbf{p}_{t+\frac{1}{2}}) \\ \text{sub. to:} & \\ p_{ik} &\leq p_{ij} + p_{jk}, \text{ for } i, j, k \in [n] \\ p_{ij} &\geq 0, \text{ for } i, j \in [n]. \end{aligned}$$

comparison vector satisfies these triangle inequalities. The detail PermRank is shown in Algorithm 1. In particular, $\mathbf{LPKWIKSORT}_h$ uses the following function h :

$$h(x) = \begin{cases} 0 & 0 \leq x \leq \frac{1}{6} \\ \frac{3}{2}x - \frac{1}{4} & \frac{1}{6} < x \leq \frac{5}{6} \\ 1 & \frac{5}{6} < x \leq 1. \end{cases}$$

Note that h is symmetric in the sense that $h(1-x) = 1-h(x)$.

5.1. Derivation of the update

In this subsection, we derive the update rule in PermRank. The update is motivated by the following optimization problem:

$$\min_{\mathbf{p}} \eta \|\mathbf{y} - \mathbf{p}\|_1 + \Delta_2(\mathbf{p}, \mathbf{p}').$$

To solve this, we use the following relationship: For any $y_{ij} \in \{0, 1\}$ and $p_i \in [0, 1]$, $|y_{ij} - p_{ij}| = p_{ij}(1 - y_{ij}) + (1 - p_{ij})y_{ij}$. Then we define the Lagrangian as

$$L(\mathbf{p}) = \eta \sum_{ij} |y_{ij} - p_{ij}| + \sum_{ij} \Delta_2(p_{ij}, p'_{ij}),$$

where \mathbf{p}' is the probability vector before the update. Here we note that we remove the constraint that $\mathbf{p} \in [0, 1]^n$, since the binary relative entropy Δ_2 implicitly enforces the

constraint. By taking the partial derivative of L and enforcing the derivative to be zero, we get the update:

$$p_{ij} = \frac{\left(\frac{p'_{ij}}{1-p'_{ij}}\right) e^{-\eta(1-2y_{ij})}}{1 + \left(\frac{p'_{ij}}{1-p'_{ij}}\right) e^{-\eta(1-2y_{ij})}} = \frac{p'_{ij} e^{-\eta(1-y_{ij})}}{(1-p'_{ij})e^{-\eta y_{ij}} + p'_{ij} e^{-\eta(1-y_{ij})}}.$$

5.2. Our Analysis

In this subsection we show our relative loss bound of PermRank.

Lemma 1 *For each $t = 1, \dots, T$ and any comparison vector \mathbf{q} ,*

$$\Delta_2(\mathbf{q}, \mathbf{p}_t) - \Delta_2(\mathbf{q}, \mathbf{p}_{t+\frac{1}{2}}) \geq -\eta \|\mathbf{y}_t - \mathbf{q}\|_1 + (1 - e^{-\eta}) \|\mathbf{y}_t - \mathbf{p}_t\|_1.$$

We show the proof in the supplementary material, which is based on a standard technique in online learning (see, e.g., (Cesa-Bianchi and Lugosi, 2006)).

Algorithm 2 LPKWIKSORT_h (Ailon (2008))

Input: a N -dimensional vector $\mathbf{p} \in [0, 1]^N$

Output: a permutation

1. Let S_L and S_R be empty sets, respectively.
 2. Pick an integer i from $\{1, \dots, n\}$ randomly.
 3. For each $j \in \{1, \dots, n\}$ such that $j \neq i$
 - (a) With probability $h(p_{ij})$, put j in S_L .
 - (b) Otherwise, put j in S_R .
 4. Let $\mathbf{p}_L, \mathbf{p}_R$ be the comparison vector induced by S_L and S_R , respectively.
 5. Output (LPKWIKSORT_h(\mathbf{p}_L), i , LPKWIKSORT_h(\mathbf{p}_R)).
-

In order to prove the cumulative loss bound of PermRank, we will use the Generalized Pythagorean Theorem for Bregman divergences (Bregman, 1967) (For details of the definition of Bregman divergences, see, e.g., (Cesa-Bianchi and Lugosi, 2006)). Since the binary relative entropy is a Bregman divergence, so does our generalized version Δ_2 . In the following, we show a version of the Generalized Pythagorean Theorem adapted for the binary relative entropy.

Lemma 2 (Generalized Pythagorean Theorem, Bregman (1967)) *Let S be a convex set in $[0, 1]^N$ and \mathbf{p} be a point in $[0, 1]^N$ with strictly positive entries. Let $\mathbf{p}' \in S$ be the projection of \mathbf{p} onto S in terms of Δ_2 , i.e., $\mathbf{p}' = \arg \min_{\mathbf{q} \in S} \Delta_2(\mathbf{q}, \mathbf{p})$. Then, for any $\mathbf{q} \in [0, 1]^N$,*

$$\Delta_2(\mathbf{q}, \mathbf{p}) \geq \Delta_2(\mathbf{q}, \mathbf{p}') + \Delta_2(\mathbf{p}', \mathbf{p}).$$

In particular, if S is affine, the inequality holds with equality.

Using Lemma 2, we prove the next lemma.

Lemma 3 For each $t = 1, \dots, T$ and any comparison vector \mathbf{q} ,

$$\Delta_2(\mathbf{q}, \mathbf{p}_t) - \Delta_2(\mathbf{q}, \mathbf{p}_{t+1}) \geq -\eta \|\mathbf{y}_t - \mathbf{q}\|_1 + (1 - e^{-\eta}) \|\mathbf{y}_t - \mathbf{p}_t\|_1.$$

Proof By applying Lemma 2, we obtain

$$\begin{aligned} \Delta_2(\mathbf{q}, \mathbf{p}_t) - \Delta_2(\mathbf{q}, \mathbf{p}_{t+1}) &\geq \Delta_2(\mathbf{q}, \mathbf{p}_t) - \Delta_2(\mathbf{q}, \mathbf{p}_{t+\frac{1}{2}}) + \Delta_2(\mathbf{p}_{t+1}, \mathbf{p}_{t+\frac{1}{2}}) \\ &\geq \Delta_2(\mathbf{q}, \mathbf{p}_t) - \Delta_2(\mathbf{q}, \mathbf{p}_{t+\frac{1}{2}}). \end{aligned}$$

Further, by Lemma 1,

$$\Delta_2(\mathbf{q}, \mathbf{p}_t) - \Delta_2(\mathbf{q}, \mathbf{p}_{t+\frac{1}{2}}) \geq -\eta \|\mathbf{y}_t - \mathbf{q}\|_1 + (1 - e^{-\eta}) \|\mathbf{y}_t - \mathbf{p}_t\|_1,$$

which completes the proof. ■

For LPKWIKSORT_h , the following property is proved ².

Lemma 4 (Ailon (2008)) For each trial t ,

$$\mathbf{E}[d(\sigma_t, \hat{\sigma}_t)] \leq \frac{3}{2} \mathbf{p}_t \cdot \mathbf{y}_t,$$

where the expectation is about the randomization in KWIKSORT_h .

By summing up the inequality in Lemma 3 and by using Lemma 4, we obtain the cumulative loss bound of PermRank as follows:

Theorem 4 For any comparison vector $\mathbf{q} \in \{0, 1\}^N$,

$$\sum_{t=1}^T \|\mathbf{y}_t - \mathbf{p}_t\|_1 \leq \frac{\eta \sum_{t=1}^T \|\mathbf{y}_t - \mathbf{q}\|_1 + \frac{n(n-1)}{2} \ln 2}{1 - e^{-\eta}}.$$

Proof By summing up the inequality in Lemma 1 for $t = 1, \dots, T$, we get that

$$\frac{\eta \sum_{t=1}^T \|\mathbf{y}_t - \mathbf{q}\|_1 - \Delta_2(\mathbf{q}, \mathbf{p}_{T+1}) + \Delta_2(\mathbf{q}, \mathbf{p}_1)}{1 - e^{-\eta}}.$$

Since $\Delta_2(\mathbf{q}, \mathbf{p}_{T+1}) \geq 0$ and $\Delta_2(\mathbf{q}, \mathbf{p}_1) \leq \ln 2$, we complete the proof. ■

In particular, when we set $\eta = 2 \ln(1 + 1/\sqrt{T})$, by the fact that $\eta \leq e^{\frac{\eta}{2}} - e^{-\frac{\eta}{2}}$, we get

$$\frac{\eta}{1 - e^{-\eta}} \leq e^{\frac{\eta}{2}} = 1 + 1/\sqrt{T}, \text{ and } \frac{1}{1 - e^{-\eta}} = \frac{(1 + 1/\sqrt{T})^2}{1/T + 2/\sqrt{T}} \leq 1 + \sqrt{T}/2,$$

respectively. Also, recall that Kendall tau distance is at most $n(n-1)/2$. Thus we have the following corollary.

2. Originally, Lemma 4 is proved for the case where the solution of a LP relaxation of the (partial) rank aggregation problem is given as input. But, in fact, the lemma holds for any probability vectors satisfying triangle inequality.

Corollary 5 For $\eta = \ln(1 + \sqrt{1/T})$, the expected cumulative loss of Permrank is at most

$$\mathbf{E} \left[\sum_{t=1}^T d(\sigma_t, \hat{\sigma}_t) \right] \leq \frac{3}{2} \min_{\sigma \in S_n} \sum_{t=1}^T d(\sigma_t, \sigma) + O(n^2 \sqrt{T}).$$

6. Experiments

We show preliminary experimental results for artificial and real data sets. The algorithms we examine are Hedge Algorithm, PermELearn and PermRank.

For our artificial data, we specify the following way of generating permutations. First we fix a base permutation in S^n . Then, at each trial, we pick a pair over n elements randomly and reverse the order of the pair in the base permutation. After repeating this procedure s times, we give each learning algorithm the resulting permutation. In our experiments, we fix $n = 7$, $s = 1$, and $T = 600$, respectively.

Our real data set is the SUSHI Preference data set (Kamishima, 2003). In particular, we use the “sushi3a.5000.10.order” data, which contains 5000 permutations (preferences) over 10 fixed sushi items. To reduce computational costs further, we consider only 8 sushi items (we remove sea urchin and salmon roe from the original data). So, our data set has $T = 5000$ permutation over $n = 8$ items. We run each algorithm over a fixed sequence of T permutations in the data.

We run Hedge algorithm with $n!$ permutations as experts, PermELearn, and PermRank. As the parameter η , we consider $\eta \in \{0.01, 0.02, 0.05, 0.1, 0.2\}$ for these algorithms. We also compare them with the best permutation at the end. We compute the best permutation by a brute-force search over $n!$ permutations.

Note that each learning algorithm is probabilistic. So for each setting of η , we run algorithms twice and choose the one attaining the lowest average cumulative losses as the best parameter for each algorithm.

We plot the results in Figure 1. More precisely, for each algorithm, we plot its regret, i.e.,

$$(\text{regret}) = (\text{cumulative loss of the algorithm}) - (\text{that of the best fixed permutation}).$$

As can be seen in Figure 1, the regret of PermRank is smaller than that of PermELearn. On the other hand, Hedge algorithm gradually improves its regret and it gives slightly better result than PermRank at the end. Perhaps, this is because PermRank has the approximation factor $3/2$ while Hedge algorithm has the approximation factor 1 in their cumulative loss bounds, respectively. So, when the best fixed permutation has large cumulative loss, for sufficiently large T w.r.t. n , the regret of PermRank grows linearly in T because of its larger approximation factor. Note that, however, when n is large, it is not feasible to run Hedge algorithm with $n!$ experts. Also, as we show in the previous section, it is unlikely that there exists a polynomial time algorithm with the approximation factor 1. Therefore, PermRank is a practical choice for online rank aggregation.

7. Conclusions and Future Work

In this paper, we consider online rank aggregation, the online version of the rank aggregation problem. We proposed the online learning algorithm PermRank. for online rank

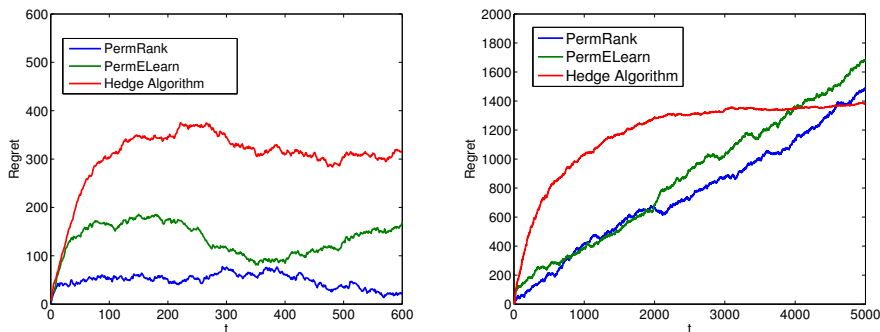


Figure 1: The regret of Hedge, PermELearn, and PermRank, for artificial data (left) and the subset of SUSHI Preference data (right).

aggregation and prove its cumulative loss bound. Then we prove a lower bound for on-line rank aggregation which is close to the upper bound of PermRank. We also prove the hardness of obtaining the optimal cumulative loss bound which matches the lower bound. Finally, our experimental results show that PermRank performs much better than the naive implementation of Hedge algorithm with $n!$ permutations as experts.

There are some open questions: (i) What if the input is not a permutation but some partial order information, e.g., which element in a pair $(i, j) \in [n]^2$ precedes (Abernethy (2010))? (ii) Can we generalize our results for partial rankings such as top k lists? The first question is posed by Abernethy as an open problem (Abernethy, 2010). Our case, the online aggregation problem, is an special case where information of all possible pairs are given. Also, the second problem is important in practice (see, e.g., Ailon (2008); Fagin et al. (2006) for researches on partial ranking).

Acknowledgement

We thank anonymous reviewers for their helpful comments. This work is supported in part by JSPS Grand-in-Aid for Young Scientists (B) 23700178 and JSPS Grand-in-Aid for Scientific Research (B) 23300003.

References

Jacob Abernethy. Can we learn to gamble efficiently? In *Proceedings of the 23rd Annual Conference on Learning Theory (COLT '10)*, pages 318–319, 2010.

Nir Ailon. Aggregation of partial rankings, p -ratings and top- m lists. *Algorithmica*, 57(2): 284–300, 2008.

Nir Ailon, Moses Charikar, and Alantha Newman. Aggregating inconsistent information: Ranking and clustering. *Journal of the ACM*, 55(5), 2008.

- Alexandr Andoni, Ronald Fagin, Ravi Kumar, Mihai Patrascu, and D. Sivakumar. Corrigendum to "efficient similarity search and classification via rank aggregation" by ronald fagin, ravi kumar and d. sivakumar (proc. sigmod'03). In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 1375–1376, 2008.
- Hamsa Balakrishnan, Inseok Hwang, and Claire J. Tomlin. Polynomial approximation algorithms for belief matrix maintenance in identity management. In *43rd IEEE Conference on Decision and Control*, pages 4874–4879, 2004.
- J. Bartholdi, C. A. Tovey, and M. A. Trick. Voting schemes for which it can be difficult to tell who won the election. *Social Choice and Welfare*, 6:157–165, 1989.
- J. C. Borda. Mémoire sur les élections au scrutin. *Histoire de l'Académie Royale des Sciences*, 1781.
- L. M. Bregman. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR Computational Mathematics and Physics*, 7:200–217, 1967.
- N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006.
- Nicolò Cesa-Bianchi, Yoav Freund, David Haussler, David P. Helmbold, Robert E. Schapire, and Manfred K. Warmuth. How to use expert advice. *Journal of the Association for Computing Machinery*, 44(3):427–485, 1997.
- M. J. Condorcet. Éssai sur l'application de l'analyse à la probabilité des décisions rendues à la pluralité des voix, 1785.
- Persi Diaconis and R. L. Graham. Spearman's footrule as a measure of disarray. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(2):262–268, 1977.
- Cynthia Dwork, Ravi Kumar, Moni Naor, and D. Sivakumar. Rank aggregation methods for the web. In *Proceedings of the Tenth International World Wide Web Conference (WWW'01)*, pages 613–622, 2001.
- Ronald Fagin, Ravi Kumar, and D. Sivakumar. Efficient similarity search and classification via rank aggregation. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 301–312, 2003.
- Ronald Fagin, Ravi Kumar, Mohammad Mahdian, D. Sivakumar, and Erik Vee. Comparing partial rankings. *SIAM Journal on Discrete Mathematics*, 20(3):628–648, 2006.
- Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.
- David P. Helmbold and Manfred K. Warmuth. Learning permutations with exponential weights. *Journal of Machine Learning Research*, 10:1705–1736, 2009.

- M. Herbster and M. Warmuth. Tracking the best linear predictor. *Journal of Machine Learning Research*, 1:281–309, 2001.
- Sham Kakade, Adam Tauman Kalai, and Látina Ligett. Playing games with approximation algorithms. In *Proceedings of the 39th annual ACM symposium on Theory of Computing (STOC'07)*, pages 546–555, 2007.
- Toshihiro Kamishima. Nantonac collaborative filtering: Recommendation based on order responses. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining (KDD'03)*, pages 583–588. ACM, 2003.
- J. G. Kemeny. Mathematics without numbers. *Daedalus*, 88:571–591, 1959.
- J. G. Kemeny and J. Snell. *Mathematical Models in the Social Sciences*. Blaisdell, 1962. (Reprinted by MIT Press, Cambridge, 1972.).
- Claire Kenyon-Mathieu and Warren Schudy. How to rank with few errors. In *Proceedings of the 39th Annual ACM Symposium on Theory of Computing (STOC'07)*, pages 95–103, 2007. Draft journal version available at http://www.cs.brown.edu/~ws/papers/fast_journal.pdf.
- Nick Littlestone and Manfred K. Warmuth. The weighted majority algorithm. *Information and Computation*, 108(2):212–261, 1994.
- V. Vovk. Aggregating strategies. In *Proceedings of the 3rd Annual Workshop on Computational Learning Theory*, pages 371–386, 1990.

Appendix A. The cumulative loss bound of Hedge algorithm

In the appendix, we analyze cumulative loss bounds for two previously known algorithms, Hedge algorithm and PermELearn, for online rank aggregation. Also, we show a proof of Lemma 1 for completeness.

First, we derive the cumulative loss bound of Hedge algorithm with $n!$ permutations as experts. Hedge algorithm is designed to deal with loss $\ell_t \in [0, 1]$. More precisely, at each trial t , each expert σ^i ($1 \leq i \leq n!$) receive loss $\ell_{t,i} \in [0, 1]$ and the loss of Hedge algorithm is expected loss $\sum_{i=1}^{n!} w_{t,i} \ell_{t,i}$ of experts, where loss is averaged with a weight vector $\mathbf{w}_t \in [0, 1]^{n!}$ such that, $\sum_i w_i = 1$. For online rank aggregation, we define the loss of expert σ^i at trial t as $\ell_{t,i} = \frac{2}{n(n-1)} d(\sigma_t, \sigma^i)$, so that the range of loss is $[0, 1]$.

Theorem 6 (Freund and Schapire (Freund and Schapire, 1997)) *For any T , the cumulative loss bounds of Hedge algorithm with N experts is bounded as follows:*

$$\sum_{t=1}^T \mathbf{w}_t \cdot \ell_t \leq \frac{\eta \min_{i=1}^N \sum_{t=1}^T \ell_{t,i} + \ln N}{1 - e^{-\eta}}$$

By using the fact that $N = n! \leq n^n$, setting $\eta = 2 \ln(1 + \sqrt{n \ln n} / \sqrt{T})$ and multiplying $n(n-1)/1$ on both sides of the inequality, we obtain the following bound of Hedge algorithm for online rank aggregation.

Corollary 7

$$\mathbf{E} \left[\sum_{t=1}^T d(\sigma_t, \hat{\sigma}_t) \right] \leq \min_{\sigma \in S_n} \sum_{t=1}^T d(\sigma_t, \sigma) + O\left(n^2 \sqrt{Tn \ln n}\right).$$

A.1. The cumulative loss bound of PermELearn

We show how to apply PermELearn (Helmbold and Warmuth, 2009) for online rank aggregation. PermELearn uses the matrix representation of permutations. More precisely, each permutation σ in S_n is represented as a $n \times n$ $\{0, 1\}$ -matrix Π such that, for each $i = 1, \dots, n$, $\sigma(i) = j$ if and only if $\Pi_{ij} = 1$. For example, for a permutation $\sigma = (2, 4, 3, 1)$, the corresponding matrix Π is

$$\Pi = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}.$$

The loss of each permutation is specified by a $n \times n$ $[0, 1]$ -matrix L , which we call *loss matrix*. Given a loss matrix L , the loss of a permutation σ whose associated matrix is Π is defined as $L \bullet \Pi = \sum_{i,j} L_{ij} \Pi_{ij}$.

Let \mathcal{P}_n be the set of the matrices representing permutations in S_n . The online learning protocol is as follows. At each trial t : (i) The learner predicts a matrix $\Pi_t \in \mathcal{P}_n$ associated with a permutation $\hat{\sigma}_t$. (ii) The adversary chooses the loss matrix L_t associated with the true permutation σ_t . (iii) The learner incurs the loss $L_t \bullet \Pi_t$. Then the following results holds.

Theorem 8 (Helmbold and Warmuth (Helmbold and Warmuth, 2009)) *For any $T \geq 1$ and $\eta > 0$, the expected cumulative loss of PermELearn is*

$$\mathbf{E} \left[\sum_{t=1}^T L_t \bullet \Pi_t \right] \leq \frac{\eta \min_{\Pi \in \mathcal{P}_n} \sum_{t=1}^T L_t \bullet \Pi + n \ln n}{1 - e^{-\eta}}.$$

Since Kendall tau distance does not seem to have a loss matrix representation, we consider an alternative distance for permutations, that is, Spearman’s footrule $d_F(\sigma, \sigma') = \sum_{i=1}^n |\sigma(i) - \sigma'(i)|$. As mentioned earlier, Spearman’s footrule d_F approximates Kendall tau distance d (Diaconis and Graham, 1977):

$$d(\sigma_1, \sigma_2) \leq d_F(\sigma_1, \sigma_2) \leq 2d(\sigma_1, \sigma_2). \tag{2}$$

Fortunately, Spearman’s footrule d_F can be written as a loss matrix. For a permutation σ and σ' , let the loss matrix L be such that $L_{ij} = \frac{|\sigma(i) - \sigma'(j)|}{n-1}$ for $i, j = 1, \dots, n$ and let Π' be the matrix form of σ' . Then, it holds that

$$L \bullet \Pi' = d_F(\sigma, \sigma') / (n - 1). \tag{3}$$

Then by using (2), (3) and setting $\eta = 2 \ln(1 + \sqrt{\ln n} / \sqrt{T})$, we obtain the cumulative loss of PermELearn for online rank aggregation.

Corollary 9 For any $T \geq 1$, the expected cumulative loss of *PermuELearn* is bounded as

$$\mathbf{E} \left[\sum_{t=1}^T d(\sigma_t, \hat{\sigma}_t) \right] \leq 2 \min_{\sigma \in S_n} \sum_{t=1}^T d(\sigma_t, \sigma) + O\left(n^2 \sqrt{T \ln n}\right).$$

A.2. Proof of Lemma 1

Proof

$$\begin{aligned} & \Delta_2(q_{ij}, p_{t,ij}) - \Delta_2(q_{ij}, p_{t+\frac{1}{2},ij}) \\ &= q_{ij} \ln \frac{p_{t+\frac{1}{2},ij}}{p_{t,ij}} + (1 - q_{ij}) \ln \frac{1 - p_{t+\frac{1}{2},ij}}{1 - p_{t,ij}} \\ &= -q_{ij} \eta (1 - y_{t,ij}) - (1 - q_{ij}) \eta y_{t,ij} - \ln \left((1 - p_{t+\frac{1}{2},ij}) e^{-\eta y_{t,ij}} + p_{t+\frac{1}{2},ij} e^{-\eta(1-y_{t,ij})} \right) \\ &= -\eta |y_{t,ij} - q_{ij}| - \ln \left((1 - p_{t+\frac{1}{2},ij}) e^{-\eta y_{t,ij}} + p_{t+\frac{1}{2},ij} e^{-\eta(1-y_{t,ij})} \right). \end{aligned}$$

Since $e^{-\eta y_{ij}} = 1 - (1 - e^{-\eta}) y_{ij}$ for $y_{ij} \in \{0, 1\}$, the terms above becomes

$$\begin{aligned} & \Delta_2(q_{ij}, p_{t,ij}) - \Delta_2(q_{ij}, p_{t+1,ij}) \\ &= -\eta |y_{t,ij} - q_{ij}| - \ln \left(1 - (1 - e^{-\eta}) ((1 - p_{t,ij}) y_{t,ij} + p_{t,ij} (1 - y_{t,ij})) \right) \\ &\geq -\eta |y_{t,ij} - q_{ij}| + (1 - e^{-\eta}) |y_{t,ij} - p_{t,ij}|. \end{aligned}$$

Finally, summing up the inequality for all $i, j \in [n]$, we complete the proof. ■