IOS Press

Approximate Sorting

Joachim Giesen[*]
Max Plank Institute for Computer Science
Saarbrücken, Germany

Eva Schuberth[*]
Institute for Theoretical Computer Science
ETH Zurich, Switzerland

Miloš Stojaković[†]
Department of Mathematics and Informatics
University of Novi Sad, Serbia

Abstract. We show that any comparison based, randomized algorithm to approximate any given ranking of n items within expected Spearman's footrule distance $n^2/\nu(n)$ needs at least

$$n\left(\min\{\log \nu(n), \log n\} - 6\right)$$

comparisons in the worst case. This bound is tight up to a constant factor since there exists a deterministic algorithm that shows that $6n \log \nu(n)$ comparisons are always sufficient.

Keywords: algorithms, sorting, ranking, Spearman's footrule metric, Kendall's tau metric

1. Introduction

Our motivation to study approximate sorting comes from the following market research application. We want to find out how a respondent ranks a set of products. In order to simulate real buying situations the respondent is presented pairs of products out of which he has to choose the one that he prefers, i.e., he has to perform paired comparisons. The respondent's ranking is then reconstructed from the sequence of his choices. That is, a procedure that presents a sequence of product pairs to the respondent in order to obtain the product ranking is nothing else than a comparison based sorting algorithm. We can measure the efficiency of such an algorithm in terms of the number of (pairwise) comparisons needed in order to

[*] Partly supported by the Swiss National Science Foundation under the grant "Robust Algorithms for Conjoint Analysis".

[†] Corresponding author. Partly supported by the Ministry of Science and Environmental Protection, Republic of Serbia, and Provincial Secretariat for Science, Province of Vojvodina.

obtain the ranking. The information theoretic lower bound on sorting [6] states that there is no procedure that can determine a ranking by posing less than $n \log \frac{n}{e}$ paired comparison questions to the respondent, i.e., in general $\Omega(n \log n)$ comparisons are needed. Even for only moderately large n that easily is too much since respondents often get worn out after a certain number of questions (independent of n) and do not answer further questions faithfully anymore. On the other hand, it might be enough to know the respondent's ranking approximately. In this paper we pursue the question of how many comparisons are necessary and sufficient in order to approximately rank n products.

In order to give sense to the term "approximately" we need some metric to compare rankings. Assume that we are dealing with n products. Since a ranking is a permutation of the products, this means that we need a metric on the permutation group S_n. Not all of the metrics, e.g., the Hamming distance that counts how many products are ranked differently, are meaningful for our application. For example, if in the respondent's ranking one exchanges every second product with its predecessor, then the resulting ranking has maximal Hamming distance to the original one. Nevertheless, this ranking still tells a lot about the respondent's preferences. In marketing applications Kendall's tau metric [3] is frequently used since it seems to capture the intuitive notion of closeness of two rankings and also arises naturally in the statistics of certain random rankings [7].

Our results. Instead of working with Kendall's metric we use Spearman's footrule metric [3] which essentially is equivalent to Kendall's metric, since the two metrics are within a constant factor of each other [3]. The maximal distance between any two rankings of n products in Spearman's footrule metric is less than n^2. We show that in order to obtain a ranking at distance $n^2/\nu(n)$ to the actual ranking, with any strategy, a respondent has in general to perform at least $n\,(\min\{\log \nu(n), \log n\} - 6)$ comparisons in the worst case, i.e., there is an instance for which any comparison based algorithm performs at least $n\,(\min\{\log \nu(n), \log n\} - 6)$ comparisons. Moreover, if we allow the strategy to be randomized such that the obtained ranking is at expected distance $n^2/\nu(n)$ to the respondent's ranking, we can show that the same bound on the minimum number of comparisons holds.

On the other hand, there is a deterministic strategy (algorithm) that shows that $6n \log \nu(n)$ comparisons are always sufficient.

Related work. At first glance our work seems related to work done on pre-sorting. In pre-sorting the goal is to pre-process the data such that fewer comparisons are needed afterwards to sort them. For example in [4] it is shown that with $O(1)$ pre-processing one can save $\Theta(n)$ comparisons for Quicksort on average. Pre-processing can be seen as computing a partial order on the data that helps for a given sorting algorithm to reduce the number of necessary comparisons. The structural quantity that determines how many comparisons are needed in general to find the ranking given a partial order is the number of linear extensions of the partial order, i.e., the number of rankings consistent with the partial order. Actually, the logarithm of this number is a lower bound on the number of comparisons needed in general [5]. Here we study another structural measure, namely, the maximum diameter in the Spearman's metric of the set of rankings consistent with a partial order. Our results show that with $o(n \log n)$ comparisons one can make this diameter asymptotically smaller than the diameter of the set of all rankings. That is not the case for the number of linear extensions which stays in $\Theta(2^{n \log n})$.

Notation. The logarithm $\log$ in this paper is assumed to be binary, and by id we denote the identity (increasing) permutation of $[n]$.

2. Lower Bound

Here, we show that in order to obtain a ranking reasonably close to the actual ranking, a respondent has to perform a substantial number of comparisons in the worst case. More precisely, for any (possibly randomized) comparison based algorithm that outputs a ranking at distance $n^2/\nu(n)$ to the actual ranking, there is an instance for which it performs (in expectation) at least $n\,(\min\{\log\nu(n), \log n\} - 6)$ comparisons.

The distance of an approximate ranking from the actual ranking will be measured in Spearman's footrule metric,

$$D(\pi, \mathrm{id}) = D(\pi) = \sum_{i=1}^{n} |i - \pi(i)|,$$

where $\pi(i)$ is the rank of the element of rank i in the approximate ranking, i.e., $|i - \pi(i)|$ measures deviation of the approximated rank from the actual rank. Note that for any ranking the distance in the Spearman's footrule metric to id is at most $\frac{n^2}{2}$.

For $r > 0$, by $B_D(\mathrm{id}, r)$ we denote the ball centered at id of radius r with respect to the Spearman's footrule metric, so

$$B_D(\mathrm{id}, r) := \{\pi \in S_n : D(\pi, \mathrm{id}) \le r\}.$$

Next we estimate the number of permutations in a ball of radius r.

Lemma 2.1.

$$|B_D(id, r)| \le \left(\frac{2e(r+n)}{n}\right)^n.$$

Proof:
Every permutation $\pi \in S_n$ is uniquely determined by the sequence $\{\pi(i) - i\}_i$. Hence, for any sequence of non-negative integers d_i, $i = 1, \ldots, n$, there are at most 2^n permutations $\pi \in S_n$ satisfying $|\pi(i) - i| = d_i$.

If $D(\pi, \mathrm{id}) \le r$, then $\sum_i |\pi(i) - i| \le r$. Since the number of sequences of n non-negative integers whose sum is at most r is $\binom{r+n}{n}$, we have

$$|B_D(\mathrm{id}, r)| \le \binom{r+n}{n} 2^n \le \left(\frac{2e(r+n)}{n}\right)^n.$$

□

Using the previous lemma and Yao's Principle [8], we give a lower bound for the worst case running time of any (randomized) comparison based approximate sorting algorithm.

Theorem 2.1. Let $\mathcal{A}$ be a randomized approximate sorting algorithm based on comparisons, let $\nu = \nu(n)$ be a function, and let $r = r(n) = \frac{n^2}{\nu(n)}$.

If for every input permutation $\pi \in S_n$ the expected Spearman's footrule distance of the output to id is at most r, then the algorithm performs at least $n\,(\min\{\log\nu, \log n\} - 6)$ comparisons in expectation in the worst case.

Proof:
Let k be the smallest integer such that $\mathcal{A}$ performs at most k comparisons for every input. For a contradiction, let us assume that

$$k < n\left(\min\{\log \nu, \log n\} - 6\right).$$

First, we are going to prove

$$\frac{1}{2}n! > 2^k\left(\frac{2e(2r+n)}{n}\right)^n. \tag{1}$$

Since $\log \nu - 6 > k/n$, we have $\frac{\nu}{2^6} > 2^{k/n}$ and since $\nu = \frac{n^2}{r}$ we get

$$\frac{n}{2e} > 2^{k/n}\frac{2e \cdot 2r}{n}. \tag{2}$$

On the other hand, from $\log n - 6 > k/n$ we get $\frac{n}{2^6} > 2^{k/n}$ implying

$$\frac{n}{2e} > 2^{k/n}\frac{2e \cdot n}{n}. \tag{3}$$

Putting (2) and (3) together, we obtain

$$\frac{n}{e} > 2^{k/n}\frac{2e(2r+n)}{n}.$$

Hence

$$\frac{1}{2}n! \geq \left(\frac{n}{e}\right)^n > 2^k\left(\frac{2e(2r+n)}{n}\right)^n,$$

proving (1).

We denote by R the source of random bits for $\mathcal{A}$. One can see R as the set of all infinite 0-1 sequences, and then the algorithm is given a random element of R along with the input. For a permutation $\pi \in S_n$ and $\alpha \in R$, we denote by $\mathcal{A}(\pi, \alpha)$ the output of the algorithm with input π and random bits α.

We fix $\widetilde{\alpha} \in R$ and run the algorithm for every permutation $\pi \in S_n$. Note that with the random bits fixed the algorithm is deterministic. For every comparison made by the algorithm there are two possible outcomes. We partition the set of all permutations S_n into classes such that all permutations in a class have the same outcomes of *all the comparisons* the algorithm makes. Since there is no randomness involved, we have that for every class C there exists a $\sigma \in S_n$ such that for every $\pi \in C$ we have $\mathcal{A}(\pi, \widetilde{\alpha}) = \sigma \circ \pi$, where $\circ$ is the multiplication in the permutation group S_n. In particular, this implies that the set $\{\mathcal{A}(\pi, \widetilde{\alpha}) : \pi \in C\}$ is of size $|C|$. On the other hand, since the algorithm in this setting is deterministic and the number of comparisons of the algorithm is at most k, there can be at most 2^k classes. Hence, each permutation in S_n is the output for at most 2^k different input permutations. From Lemma 2.1 we have $|B_D(\mathrm{id}, 2r)| \leq \left(\frac{2e(2r+n)}{n}\right)^n$, and this together with (1) implies that at least

$$n! - 2^k\left(\frac{2e(2r+n)}{n}\right)^n > \frac{1}{2}n!$$

input permutations have output at distance to id more than $2r$.

Now, if both the random bits $\alpha \in R$ and the input permutation $\pi \in S_n$ are chosen at random, the expected distance of the output $\mathcal{A}(\pi, \alpha)$ to id is more than r. Therefore, there exists a permutation π_0 such that for a randomly chosen $\alpha \in R$ the expected distance $d_D(\mathcal{A}(\pi_0, \alpha), \mathrm{id})$ is more than r. Contradiction. □

3. Algorithm

The idea of ASORT algorithm is to partition the products into a sorted sequence of equal-sized bins such that the elements in each bin have smaller rank than any element in subsequent bins. It is based on a well-studied variation of Quicksort algorithm in which the median is chosen to be the pivot element (see, e.g., [2]).

The output of the algorithm is the sequence of bins. Note that we do not specify the ordering of elements inside each bin, but consider any ranking consistent with the ordering of the bins. As it turns out, any such ranking approximates the actual ranking of the elements in terms of Spearman's footrule metric well.

The algorithm ASORT iteratively performs a number of median searches, each time placing the median into the right position in the ranking. Here the median of n elements is defined to be the element of rank $\lfloor \frac{n+1}{2} \rfloor$.

```
ASORT ($B$ : set, $m$ : int)
1  $B_{01} := B$   //  $B_{ij}$ is the $j$'th bin in the $i$'th round
2  for $i := 1$ to $m$ do
3     for $j := 1$ to $2^{i-1}$ do
4        compute the median of $B_{(i-1)j}$
5        $B_{i(2j-1)} := \{x \in B_{(i-1)j} \mid x \leq \textit{median}\}$
6        $B_{i(2j)} := \{x \in B_{(i-1)j} \mid x > \textit{median}\}$
7     end for
8  end for
9  return $B_{m1}, \ldots, B_{m(2^m)}$
```

To compute the median in line 4 and to partition the elements in line 5 and 6 we use the deterministic algorithm by Blum et al. [1] that performs at most $5.73n$ comparisons in order to compute the median of n elements and to partition them according to the median. We note that in putting the algorithm ASORT to practice one may want to use a different median algorithm, like, e.g., RANDOMIZEDSELECT [2].

In each round, the sum of the cardinalities of all the bins is n. Hence, one round takes at most $5.73n$ comparisons. As the algorithm runs for m rounds overall, the total number of comparisons is less than $6nm$.

Theorem 3.1. Let $r = \frac{n^2}{\nu(n)}$. Any ranking consistent with the ordering of the bins computed by ASORT in $\log \nu(n)$ rounds, i.e., with less than $6n \log \nu(n)$ comparisons, has a Spearman's footrule distance of at most r to the actual ranking of the elements from B.

Proof:
The distance of the actual ranking of the elements in B to any ranking consistent with the ordering of the bins computed by ASORT in m rounds can be bounded by

$$n \left(\left\lceil \frac{n}{2^m} \right\rceil - 1 \right) \leq \frac{n^2}{2^m}.$$

Plugging in $m = \log \nu(n)$, we see that the distance is at most r. As we saw earlier, the algorithm performs at most $6nm = 6n \log \nu(n)$ comparisons. □

Acknowledgments. We are indebted to Jiří Matoušek for comments and insights that made this paper possible.

References

[1] Blum, M., Floyd, R. W., Pratt, V., Rivest, R. L., Tarjan, R. E.: Linear time bounds for median computations, *STOC '72: Proceedings of the fourth annual ACM symposium on Theory of computing*, ACM Press, 1972.

[2] Cormen, T. H., Leiserson, C. E., Rivest, R. L.: *Introduction to Algorithms, 2nd ed.,* The MIT Press/McGraw-Hill, 2001.

[3] Diaconis, P., Graham, R. L.: Spearman's Footrule as a Measure of Disarray, *Journal of the Royal Statistical Society*, **39(2)**, 1977, 262–268.

[4] Hwang, H. K., Yang, B. Y., Yeh, Y. N.: Presorting algorithms: an average-case point of view, *Theoretical Computer Science*, **242(1-2)**, 2000, 29–40.

[5] Kahn, J., Kim, J. H.: Entropy and Sorting, *STOC '92: Proceedings of the twenty-fourth annual ACM symposium on Theory of computing*, ACM Press, 1992.

[6] Knuth, D. E.: *The Art of Computer Programming,* vol. 3, Addison Wesley, 1973.

[7] Mallows, C. L.: Non-null ranking models, *Biometrica*, **44**, 1957, 114–130.

[8] Yao, A. C.: Probabilistic computations: Towards a unified measure of complexity, *FOCS'77: Proceedings of 18th Annual Symposium on Foundations of Computer Science*, IEEE Computer Society Press, 1977.