Finding the median of three permutations under the Kendall-tau distance

Guillaume Blin, Maxime Crochemore, Sylvie Hamel, Stéphane Vialette

Finding the median of three permutations under the Kendall-τ distance - Extended Abstract

Guillaume Blin[1], Maxime Crochemore[1], Sylvie Hamel[⋆2] and Stéphane Vialette[1]

[1] Université Paris-Est, IGM-LabInfo - UMR CNRS 8049, France
{gblin, Maxime.Crochemore, vialette}@univ-mlv.fr
[2] DIRO - Université de Montréal - QC, Canada
sylvie.hamel@umontreal.ca

Abstract. Given m permutations π^1, $\pi^2 \dots \pi^m$ of $\{1, 2, \dots, n\}$ and a distance function d, the **median** problem is to find a permutation π^* that is the "closest" of the m given permutations. Here, we study the problem under the Kendall-τ distance that counts the number of pairwise disagreements between permutations. This problem is also known, in the context of rank aggregation, as the Kemeny Score Problem and has been proved to be NP-hard when $m \geq 4$. In this article, we investigate the case $m = 3$.

1 Indroduction

The problem of finding the median of a set of m permutations of $[n]$ under the Kendall-τ distance is best known in the literature as the Kemedy Score Problem. In this problem we have m voters that have to order n candidates from their best-liked candidate to their least-liked one. The problem then consist in finding a "Kemedy consensus", *i.e*, an order of the candidates that agree the most with the order of the m voters, *i.e.*, that minimizes the sum of the disagreements. This problem has been proved to be NP-complete when $m \geq 4$ [5] (the complexity is unknown for $m = 3$ and polynomial-time solvable for $m = 2$) and some approximation algorithms have been derived. First, a randomized algorithm with approximation factor 11/7 [1] and then a deterministic one with approximation factor 8/5 [10]. In 2007, a PTAS result has been obtained [8] and a year later, some fixed-paramater algorithms have been described [2]. Here, we focus on $m = 3$.

This article is organized as follow. In Section 2, we gives some basic definitions for the problem. In Section 3, we show how we can reduce the search space for the brute force algorithm by deriving some combinatorial properties of the median. Finally we present our heuristic and what still need to be done in section 4 and 5. This work is a work in progress. Since it is an extended abstract, all the proofs has been omitted but are available on request.

2 Definitions and Notations

A **permutation** π is a bijection of $[n] = \{1, 2 \dots, n\}$ onto itself. The set of all permutations of $[n]$ is denoted $\mathcal{S}_n$. As usual we denote a permutation π of $[n]$ as

⋆ supported by NSERC through an Individual Discovery Grant

$\pi = \pi_1\pi_2\dots\pi_n$. The **identity permutation** correspond to the identity bijection of $[n]$ and is denoted $\imath = 12\dots n$. A pair (π_i, π_j) of elements of the permutation π is called an **inversion** if $\pi_i > \pi_j$ and $i < j$. The **number of inversion** of a permutation π is denoted $\mathrm{inv}(\pi)$.[1]
The **Kendall-τ distance**, denoted d_{KT}, counts the number of pairwise disagreements between two permutations and can be defined formally as follows: for permutations π and σ of $[n]$, we have that

$$d_{KT}(\pi,\sigma) = |(i,j) : i < j \textit{ and } [(\pi[i] < \pi[j] \textit{ and } \sigma[i] > \sigma[j]) \\ \textit{or } (\pi[i] > \pi[j] \textit{ and } \sigma[i] < \sigma[j])|,$$

where $\pi[i]$ denote the position of integer i in permutation π. Note that we can easily computed $\mathrm{inv}(\pi)$ as $\mathrm{inv}(\pi) = d_{KT}(\pi, \imath)$.
The problem consider in this article will be called **median of three problem under the Kendall-τ distance** and can be stated as follow:

Given π^1, π^2 and π^3, we want to find π^* such that
$d_{KT}(\pi^*,\pi^1)+d_{KT}(\pi^*,\pi^2)+d_{KT}(\pi^*,\pi^3) \leq d_{KT}(\pi,\pi^1)+d_{KT}(\pi,\pi^2)+d_{KT}(\pi,\pi^3)$,
for all $\pi \in \mathcal{S}_n$,

In order to represent the disagreements between pairs of element in the median with respect to π^1, π^2 and π^3, we introduce here the notion of disagreements graph.

Definition 1 *We call the* **disagreements graph** *of the median $\pi^* = \pi^*_1\pi^*_2\dots\pi^*_n$ with respect to $\pi^1 = \pi^1_1\dots\pi^1_n$, $\pi^2 = \pi^2_1\dots\pi^2_n$ and $\pi^3 = \pi^3_1\dots\pi^3_n$, denoted $\mathcal{G}(\pi^*)$, the graph obtained from π^* by drawing weighted edges between each pairs (π^*_i, π^*_j), with $i < j$. The* **weight** *of an edge (π^*_i, π^*_j), denoted $w(\pi^*_i, \pi^*_j)$, represent the number of disagreements of this pair in π^* with the same pair of elements in π^1, π^2 and π^3,* i.e., *the distance contribution of this pair in the total Kendall-τ distance.*

Example 1 *Given $\pi^1 = 2134$, $\pi^2 = 4123$ and $\pi^3 = 4231$ we can compute (since here n is small) the median π^* by choosing, in all permutation of 4 elements, the one that minimize the Kendall-τ distance. Doing that, we know here that the median is $\pi^* = 4213$. The disagreements graph for this π^* is given Figure 1.*

3 Reducing the search space

When dealing with permutations, searching the whole set of permutations quickly becomes impossible since there are $n!$ permutations of $[n]$. To be able to compare our heuristic with the brute force algorithm for permutations of $[n]$ where $n > 12$, we need to reduce the search space so that the computation will take place in a reasonable time. Here, given three permutation π^1, π^2 and π^3, we derived some combinatorial properties of their median π^* which will considerably reduce the search space.

[1] Since the inversions are generators of $\mathcal{S}_n$, we can view $\mathcal{S}_n$ with these generators as a Coxeter group. In this context, the number of inversions of a permutation π is called the **length** of π and is denoted by $\ell(\pi)$. See Chapter 5 of [7] for more details.

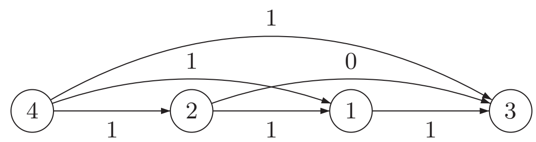

Fig. 1. Disagreements graph of $\pi^* = 4213$ with $\pi^1 = 2134$, $\pi^2 = 4123$ and $\pi^3 = 4231$.

Combinatorial properties of the median

Theorem 1 *Let $\pi^* = \pi^*_1 \dots \pi^*_n$ be the median of π^1, π^2 and π^3, three permutations of $[n]$, with respect to the Kendall-τ distance. Then, for all pairs (i, j) such that $i < j$ and $\pi^k[i] < \pi^k[j]$ for all $1 \leq k \leq 3$, (respectively $\pi^k[i] > \pi^k[j]$ for all $1 \leq k \leq 3$), we have $\pi^*[i] < \pi^*[j]$ (respectively $\pi^*[i] > \pi^*[j]$).*

This first theorem states that all the pairs of elements that appears in the same order in π^1, π^2 and π^3 should also appears in that order in the median π^*. Note that this theorem has already been stated and proved in the area of applied finance and uses what they called an Extended Condorcet Criterion [9]. To the best of our knowledge, this is the first time that this result is proved in the context of permutations.

Theorem 2 *Let $\pi^* = \pi^*_1 \dots \pi^*_n$ be the median of π^1, π^2 and π^3, three permutations of $[n]$, with respect to the Kendall-τ distance. Without loss of generality, suppose that π^1 is the permutation that is the closest of the two others,* i.e, *$d_{KT}(\pi^1, \pi^2) + d_{KT}(\pi^1, \pi^3) \leq d_{KT}(\pi^2, \pi^1) + d_{KT}(\pi^2, \pi^3)$ and $d_{KT}(\pi^1, \pi^2) + d_{KT}(\pi^1, \pi^3) \leq d_{KT}(\pi^3, \pi^1) + d_{KT}(\pi^3, \pi^2)$. Then*

$$\mathrm{inv}(\pi^*) \leq \frac{\mathrm{inv}(\pi^1) + \mathrm{inv}(\pi^2) + \mathrm{inv}(\pi^3) + d_{KT}(\pi^1, \pi^2) + d_{KT}(\pi^1, \pi^3)}{3}$$

and

$$\mathrm{inv}(\pi^*) \geq \frac{\mathrm{inv}(\pi^1) + \mathrm{inv}(\pi^2) + \mathrm{inv}(\pi^3) - d_{KT}(\pi^1, \pi^2) - d_{KT}(\pi^1, \pi^3)}{3}.$$

Theorem 2 gives upper and lower bounds on the number of inversion in the median π^*. This is really interesting since there exist a CAT-algorithm that computes all permutation of $[n]$ having exactly k inversions [6]. Table 1 compares the computation time needed to find the median of 3 permutations of $[n]$, for $4 \leq n \leq 11$, using 1) the brute force algorithm and 2) the brute force algorithm optimize by the results of Theorem 1 and 2.

4 Our heuristic

The idea of our algorithm is to apply a series of "good" cyclic movements on the starting permutations to make them closer to the median. Formally we have the following definitions and algorithm.

n	4	5	6	7	8	9	10	11
time BruteForce	0	0.0002	0.0005	0.00415	0.03955	0.425	5.03	63.33
time BruteForce + opt.	0	0	0.0002	0.0012	0.0064	0.0238	0.1496	1.0052

Table 1. Running time, in seconds, of the brute force algorithm with and without the optimizations

Definition 2 *Given $\pi = \pi_1 \dots \pi_n$, we call* **cyclic movement** *of a segment $\pi[i..j]$ of π, denoted $c[i,j](\pi)$, the cycling shifting of one position to the right ($c_r[i,j]$) or to the left ($c_\ell[i,j]$) of the segment inside the permutation π:*

$$c_r[i,j](\pi) = \pi_1 \dots \pi_{i-1}\pi_{\mathbf{i+1}} \dots \pi_{\mathbf{j}}\pi_{\mathbf{i}}\pi_{j+1} \dots \pi_n,$$

$$c_\ell[i,j](\pi) = \pi_1 \dots \pi_{i-1}\pi_{\mathbf{j}}\pi_{\mathbf{i}} \dots \pi_{\mathbf{j-1}}\pi_{j+1} \dots \pi_n$$

When $j = i+1$, a cyclic movement correspond to a **transposition**.

Definition 3 *Given three permutations π^1,π^2 and π^3, we will say that a cyclic movement is a* **k-move** *if*

$$\sum_{m=1}^{3} d_{KT}(c[i,j](\pi), \pi^m) = \sum_{m=1}^{3} d_{KT}(\pi, \pi^m) + k.$$

Definition 4 *A* **good** *cyclic movement $c[i,j]$ is a k-move, where $k < 0$.*

This means that if we apply a good cyclic movement to π we obtain a permutation that is closer to the median than π, *i.e.*, we have $\sum_{m=1}^{3} d_{KT}(c[i,j](\pi), \pi^m) < \sum_{m=1}^{3} d_{KT}(\pi, \pi^m)$. Theorem 3 gives us a way to easily find these good moves (in fact any k-move) on a starting permutation π by summing the weights of the edges, in the disagreements graph $\mathcal{G}(\pi)$ that are change by these moves.

Theorem 3 *Let π^1,π^2 and π^3 be three permutations. Let π be a starting permutation from which we want to derive π^*, the median of π^1,π^2 and π^3 with respect to the Kendall-τ distance. We have that $c_r[i,j](\pi)$ (resp. $c_\ell[i,j](\pi)$) is a k-move, $k \in \mathbb{Z}$, iff $j - i \equiv k \bmod 2$ and*

$$\sum_{t=i+1}^{j} w_{\mathcal{G}(\pi)}(\pi_i, \pi_t) \left(resp. \sum_{t=i}^{j-1} w_{\mathcal{G}(\pi)}(\pi_t, \pi_j) \right) = \frac{3(j-i)+k}{2}.$$

Now, we present our heuristic whose pseudo-code is depicted in Figure 2. The idea is to begin our search for the median in any of the starting permutation π^1, π^2 or π^3 and to apply good movements to this starting point till there is no more possible good movement. We apply three time our pseudo-code, with $\pi = \pi^m$, $1 \le m \le 3$ and our "median" is the best result we obtain from these three runs.
We tested this heuristic on all possible triplets of premutations of $[n]$ for $3 \le n \le 5$, and on 2000 random triplets, for $6 \le n \le 12$. Table 2 shows that the percentage of errors of our heuristic slowly increases from 0 to 1.6 %, as n increases from 3 to 12. Table 2 also shows that, in the case, when our heuristic does not find the real median π^*, the difference between the Kendall-τ distance of our median and π^* is always one.

```
Algorithm FindMedian (π, [π^1, π^2, π^3])
    n ← length(π)
    bool ← 0 (will be change to 1 if there is no more possible "good" movement)
    chang ← 0 (will tells us if some movements where made)
    WHILE bool <> 1 DO
        FOR i from 1 to n − 1 DO
            FOR j from i + 1 to n DO
                IF c_r[i, j](π) or c_ℓ[i, j](π) is a good movement THEN
                    π ← c_good[i, j](π)
                    chang ← chang +1
                END IF
            END FOR
        END FOR
        IF chang = 0 THEN
            bool ← 1
        END IF
    END WHILE
    RETOURNER π
```

Fig. 2. Pseudo-code of our heuristic FindMedian

n	3	4	5	6	7	8	9	10	11	12
number of computed medians	20	2024	280840	2000	2000	2000	2000	2000	2000	2000
% of errors	0	0	0	0	0.05	0.25	0.35	0.6	1.1	1.6
mean of the distances difference	0	0	0	0	1	1	1	1	1	1

Table 2. Percentage of errors of our heuristic for permutations of $[n]$, $1 \leq n \leq 12$

Considering 0-moves

When our heursitic does not find the median π^*, it means that we are stuck in a local minimum and there is no more possible good move that we can make. We decide in this case to apply a fixed number of 0-moves in hope that these moves will help us go out of the local minimum. Given a permutation π, we can easily find these 0-moves with Theorem 3. Among these 0-moves, if at least one has the property described in Theorem 4 we are guaranteed to move out of the local minimum. So, the 0-moves with this properties will be call "good".

Theorem 4 *Let π^1,π^2 and π^3 be three permutations. Let π be a starting permutation from which we want to derive π^*, the median of π^1,π^2 and π^3 with respect to the Kendall-τ distance. If $c_r[i,j](\pi)$ (resp. $c_\ell[i,j](\pi)$) is a 0-move and $w_{\mathcal{G}(\pi)}(\pi_{i-1}, \pi_{i+1}) = 2$ (resp. $w_{\mathcal{G}(\pi)}(\pi_{j-1}, \pi_{j+1}) = 2$), then there exist a -1-move in $c_r[i,j](\pi)$ (resp. $c_\ell[i,j](\pi)$).*

To try to see if we always find the median π^* by applying alternatively our heuristic and 0-moves (good or random), we tested this idea, with a permitted number of 0-moves of at most 2, on 400 random triplets of permutations of $[n]$, $7 \leq n \leq 14$. In all of those computed examples, we did found the median π^*.

5 What's left to do

Since this article is a work in progress, there is still a lot of question we need to answer. Stating only a few, we have the following ones: Starting in one permutation and applying any combinations of good and 0-moves, do we always end in the same permutation? Is our heuristic + 0-moves an exact algorithm and if so what is its complexity? Can we find combinatorial properties that will completely described the set of 0-moves that can make us move out of a local minimum?

Acknowledgements
We thanks Quentin Dejean and Anthony Estebe who programmed all the algorithms and tests as an internship project for their patience with our numerous demands and they really did great work.

References

1. N. Ailon, M. Charikar and N. Newman, *Aggregating inconsistent information: Ranking and clustering*, In Proceedings of the 37th STOC, pp.684–693, 2005.
2. N. Betzler, M.R. Fellows, J. Guo, R. Niedermeier and F.A. Rosamond, *Fixed-Parameter Algorithms for Kemeny Scores*, LNCS 5034, pages 60–71, 2008.
3. T. Biedl, F.J. Brandenburg and X. Deng, *Crossings and Permutations*, LNCS 3843, pages 1–12, 2005.
4. V. Conitzer, A. Davenport and J. Kalagnanam, *Improved Bounds for Computing Kemeny Rankings*, in Proceedings of the 21st National Conference on Artificial Intelligence (AAAI), pages 620–627, 2006.
5. C. Dwork, R. Kumar, M. naor and D. Sivakumar, *Rank Aggregation Methods for the Web*, in proceedings of the 10th WWW, pp.613-622, 2001.
6. S. Effler and F. Ruskey, *A CAT algorithm for generating permutations with a fixed number of inversions*, Information Processing Letters, Vol. 86-2, pp.107–112, 2003.
7. J.E. Humphreys, *Reflection Groups and Coxeter Groups*, Cambridge University Press, (1990), 204 pages.
8. C. Kenyon-Mathieu and W. Schudy, *How to rank with few errors*, In Proceedings of the 39th STOC, pp. 95-103, 2007.
9. M. Truchon, *An Extension of the Condorcet Criterion and Kemeny Orders*, Internal Report, cahier 98-15 du Centre de Recherche en Économie et Finance Appliquées, Université Laval, 16 pages, 1998.
10. A. vanZuylen and D.P. Williamson, *Deterministic algorithms for rank aggregation and other ranking and clustering problems*, in Proceedings of the 5th WAOA, LNCS 4927, pp.260–273, 2007.