Fast algorithms for HodgeRank with application to COVID-19 symptom and NBA team scores

Contributors: Shelby Ferrier, Guangpeng Ren
Mentor: Junyuan Lin

Mathematics Department
Seaver College of Science and Engineering
Loyola Marymount University (LMU)

April 19, 2022

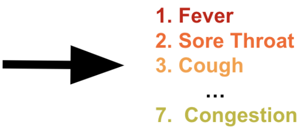
Overview

1. Background and Motivation
   - What are universal ranking problems?
   - Why are they difficult?
2. HodgeRank
   - Building the Model
3. Results
   - COVID-19 symptom scores
4. Grouping Method
   - Algorithm
   - Results on NBA data
5. Conclusion
   - Further Work

Universal ranking problems

Problem: Find a universal ranking from a group of individual rankings

Example: We'd like to rank the symptoms of COVID-19 based on severity



Example: COVID-19

"Rate your symptoms on a scale from 1-10"

Biased: different people have different pain tolerances
Incomplete: missing or corrupted data

Example: COVID-19

	Fever	Sore Throat	Cough	Nausea
Patient 1	3	2	2	5
Patient 2	7	8	9	X
Patient 3	2	2	1	3

Example: COVID-19

	Fever	Sore Throat	Cough	Nausea
Patient 1	3	2	2	5
Patient 2	7	8	9	X
Patient 3	2	2	1	3

Try taking the average of each column?

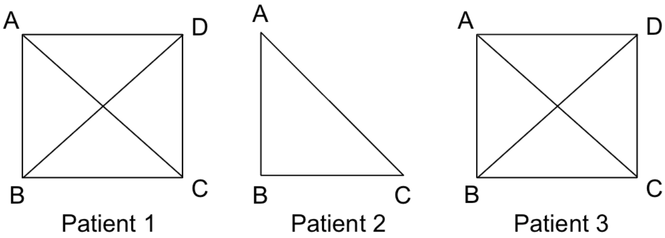
$\rightarrow$ Fever: 4, Sore Throat: 4, Cough: 4, Nausea: 4

Need method that takes into account:

- Bias
- Incompleteness

Individuals Graphs

	Fever (A)	Sore Throat (B)	Cough (C)	Nausea (D)
Patient 1	3	2	2	5
Patient 2	7	8	9	X
Patient 3	2	2	1	3

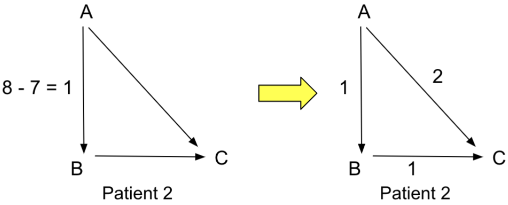


Edge Flow

	Fever (A)	Sore Throat (B)	Cough (C)	Nausea (D)
Patient 1	3	2	2	5
Patient 2	7	8	9	X
Patient 3	2	2	1	3

For $R : P \times S \to \mathbb{R}^+$, with $R(p, i)$ defined as patient p's rating of symptom i, we define the **edge flows** as follows:

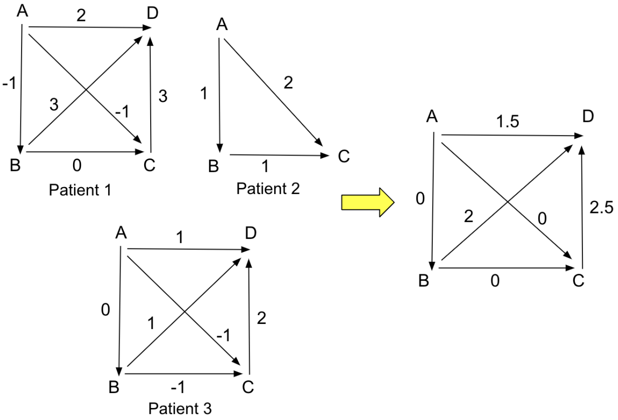
$$e_{i,j} = R(p, j) - R(p, i)$$



Colley, Lin, Hu, Aeron (2017), Jiang, Lim, Yao, Ye (2011)

Aggregating Graphs

Define total edge flow $f : E \to \mathbb{R}$ as $f(i,j) = \frac{1}{|P_{ij}|} \sum_{p \in P_{ij}} e_{i,j}$



Colley, Lin, Hu, Aeron (2017), Jiang, Lim, Yao, Ye (2011)

Solving for $\vec{r}$

	Fever (A)	Sore Throat (B)	Cough (C)	Nausea (D)
Patient 1	3	2	2	5
Patient 2	7	8	9	X
Patient 3	2	2	1	3
Univ. Rank ($\vec{r}$)	r_A	r_B	r_C	r_D

Define w_{ij} as the number of patients who have data for both symptoms i and j.

Goal: Find $\vec{r}$ that minimizes $\sum_{(i,j)\in E} w_{ij}(f(i,j) - (r_j - r_i))^2$

$\rightarrow$ Minimize the difference between every patient's ranking between two symptoms and the universal ranking between two symptoms

Colley, Lin, Hu, Aeron (2017), Jiang, Lim, Yao, Ye (2011)

Solving for $\vec{r}$

	Fever (A)	Sore Throat (B)	Cough (C)	Nausea (D)
Patient 1	3	2	2	5
Patient 2	7	8	9	X
Patient 3	2	2	1	3
Univ. Rank ($\vec{r}$)	r_A	r_B	r_C	r_D

Goal: Find $\vec{r}$ that minimizes $\sum_{(i,j)\in E} w_{ij}(f(i,j) - (r_j - r_i))^2$

$$\vec{f} = \begin{array}{c} A \to B \\ A \to C \\ A \to D \\ B \to C \\ B \to D \\ C \to D \end{array} \begin{pmatrix} 0 \\ 0 \\ 1.5 \\ 0 \\ 2 \\ 2.5 \end{pmatrix} \qquad B^T\vec{r} = \begin{bmatrix} -1 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ -1 & 0 & 0 & 1 \\ 0 & -1 & 1 & 0 \\ 0 & -1 & 0 & 1 \\ 0 & 0 & -1 & 1 \end{bmatrix} \begin{pmatrix} r_A \\ r_B \\ r_C \\ r_D \end{pmatrix}$$

Colley, Lin, Hu, Aeron (2017), Jiang, Lim, Yao, Ye (2011)

Solving for $\vec{r}$

Goal: Find $\vec{r}$ that minimizes $\sum_{(i,j)\in E} w_{ij}(f(i,j) - (r_j - r_i))^2$

$$\vec{f} = \begin{array}{c} A \to B \\ A \to C \\ A \to D \\ B \to C \\ B \to D \\ C \to D \end{array} \begin{pmatrix} 0 \\ 0 \\ 1.5 \\ 0 \\ 2 \\ 2.5 \end{pmatrix} \qquad B^T\vec{r} = \begin{bmatrix} -1 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ -1 & 0 & 0 & 1 \\ 0 & -1 & 1 & 0 \\ 0 & -1 & 0 & 1 \\ 0 & 0 & -1 & 1 \end{bmatrix} \begin{pmatrix} r_A \\ r_B \\ r_C \\ r_D \end{pmatrix}$$

$$\implies min_{\vec{r}\in\mathbb{R}^{|V|}}(||\vec{f} - B^T\vec{r}||^2_W)$$

W is defined as the diagonal matrix whose values are the weights (w_{ij}) of each edge.

Colley, Lin, Hu, Aeron (2017), Jiang, Lim, Yao, Ye (2011)

Solving for $\vec{r}$

Using calculus, we can show that

$$min_{\vec{r}\in\mathbb{R}^{|V|}} \sum_{(i,j)\in E} w_{ij}(f(i,j)-(r_j-r_i))^2 = min_{\vec{r}\in\mathbb{R}^{|V|}}(||\vec{f}-B^T\vec{r}||^2_W)$$

reduces to:

$$BWB^T\vec{r} = BW\vec{f}$$

$\vec{r}$ is the only unknown!

$$\vec{r} = (BWB^T)^{\dagger}BW\vec{f}$$

$\dagger$ denotes a pseudo-inverse.

Colley, Lin, Hu, Aeron (2017), Jiang, Lim, Yao, Ye (2011)

Example Results

	Fever (A)	Sore Throat (B)	Cough (C)	Nausea (D)
Patient 1	3	2	2	5
Patient 2	7	8	9	X
Patient 3	2	2	1	3
Univ. Rank ($\vec{r}$)	-0.375	-0.5	-0.625	1.5

Note that $\sum_{(i,j)\in E} w_{ij}(f(i,j) - (r_j - r_i))^2$ provides us with a metric for the correctness of the ranking! Error($\vec{r}$) ≈ 0.84

University of Oxford COVID-19 data set

Features:

- 14 symptoms
- 2 tiers of severity (non-severe, severe)
- 5700 respondents (1376 severe, 4324 non-severe)
- non-individualized
- Collected March 2020

	Fever	Cough	Fatigue	Dyspnea	Sputum	shortness	Myalgia	Chill	\
0	1216	978	830	608	517	491	358	358	
1	3520	2841	1911	246	1211	553	566	471	

	Dizziness	Headache	sore	Nausea	Diarhea	Congestion
0	222	155	107	81	78	39
1	523	584	419	246	251	221

Nunan, Brassey, Boyce, Gardner, Patrick-Smith *Covid-19 symptoms tracker* (2020)

Preparing data

Goal: Create an array who's columns represent symptoms and who's rows represent each of the 5700 respondents

Process:

- generated individual patient data based on Oxford data by distributing n non-zero values across each column (symptom)
- accounted for severity; (non-severe $=3$, severe $=8$)
- distributed non-zero values around a normal distribution; (sd for non-severe $=1$, sd for severe $=2$)
- bounded non-zero values to be between 1 and 10

Analysis with Hodgerank

Results:

Rank	Symptom	$\vec{r}$	Rank	Symptom	$\vec{r}$
1.	Fever	1.53	8.	Myalgia	0.125
2.	Cough	1.28	9.	Dizziness	0.0313
3.	Fatigue	0.875	10.	Headache	0
4.	Sputum Production	0.5	11.	Nausea	-0.031
5.	Dyspnea	0.25	12.	Sore Throat	-0.094
6.	Chills	0.219	13.	Diarrhea	-0.156
7.	Shortness of Breath	0.156	14.	Congestion	-0.156

Error($\vec{r}$) $\approx 3.5 * 10^{-24}$

Analysis Using Naive algorithm

Naive ranking: Average rating each person gave to a symptom

Rank	Symptom	Score	Rank	Symptom	Score
1.	Fever	3.52	8.	Chills	0.744
2.	Cough	2.84	9.	Dizziness	0.577
3.	Fatigue	2.18	10.	Headache	0.522
4.	Sputum Production	1.36	11.	Sore Throat	0.372
5.	Dyspnea	0.967	12.	Nausea	0.250
6.	Shortness of Breath	0.959	13.	Diarrhea	0.241
7.	Myalgia	0.780	14.	Congestion	0.170

Statistical Comparison

Symptom	HodgeRank rating ($\vec{r}$)	Naive rating
Fever	1.53	3.52
Cough	1.28	2.84
Fatigue	0.875	2.18
Sputum Production	0.5	1.36
Dyspnea	0.25	0.967
Chills	0.219	0.744
Shortness of Breath	0.156	0.959
Myalgia	0.125	0.780
Dizziness	0.0313	0.577
Headache	0	0.522
Nausea	-0.031	0.250
Sore Throat	-0.094	0.372
Diarrhea	-0.156	0.241
Congestion	-0.156	0.170

Runtime Limitations of HR

Time complexity of pseudo inverse: $O(n^3)$

n	10^6	10^7	10^8	10^9
t	5s	1.39 hrs	58 d	158 yrs

Methods to reduce complexity:

- Algebraic Multigrid (AMG) Method: successive subspace correction method which recursively partitions the solution space to approximate the best solution
- Dimensional reduction: reduce the size of the matrix to be inverted

Colley, Lin, Hu, Aeron (2017), Lin (2019)

Grouping Method

Key: Run the algorithm on whole group but in different groups

Steps:

1. Obtain naive ranking of elements
2. Split elements into k subgroups by naive rank
3. Run the algorithm on each of the subgroups
4. Stack subgroups rankings on top of each other

Grouping Method

Steps:

1. Obtain naive ranking of elements
2. Split elements into k subgroups by naive rank
3. Run the algorithm on each of the subgroups
4. Stack subgroups rankings on top of each other

Naive ranking (average difference with other nodes):
$r_0(i) = \sum_{j \in V} \frac{1}{|P_{ij}|} \sum_{p \in P_{ij}} [R(i) - R(j)]$

$\implies V = [v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9]$

Grouping Method

Steps:

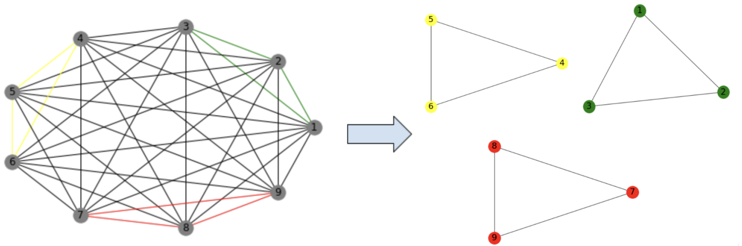
1. Obtain naive ranking of elements
2. Split elements into k subgroups by naive rank
3. Run the algorithm on each of the subgroups
4. Stack subgroups rankings on top of each other

$$V = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9\} \Longrightarrow \begin{array}{l} V_1 = \{v_1, v_2, v_3\} \\ V_2 = \{v_4, v_5, v_6\} \\ V_3 = \{v_7, v_8, v_9\} \end{array}$$

Grouping Method

Steps:

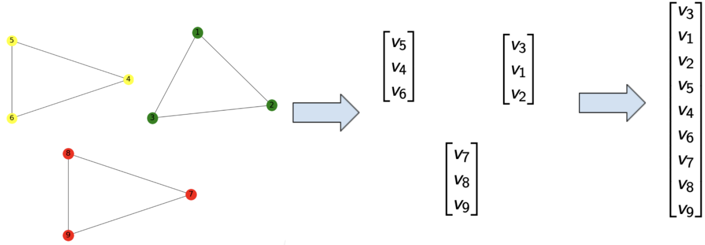
1. Obtain naive ranking of elements
2. Split elements into k subgroups by naive rank
3. Run the algorithm on each of the subgroups
4. Stack subgroups rankings on top of each other



Grouping Method

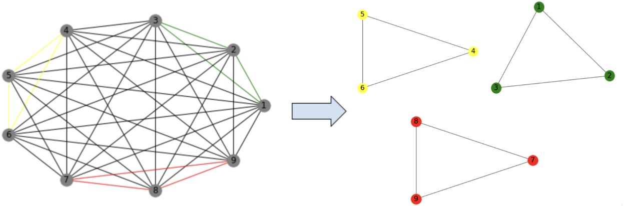
Steps:

1. Obtain naive ranking of elements
2. Split elements into k subgroups by naive rank
3. Run the algorithm on each of the subgroups
4. Stack subgroups rankings on top of each other

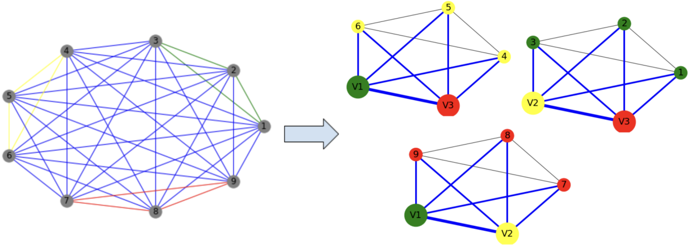


Grouping Method

Problem: Omitting much data



Solution: Include other subgroups as pseudo-nodes



Grouping Method

Steps:

1. Obtain naive ranking of elements
2. Split elements into k subgroups by naive rank
3. Run the algorithm on each of the subgroups, including the other subgroups as nodes
4. Stack subgroups rankings on top of each other, omitting the rankings of subgroups

$V = \{v_1, v_2, v_3, \ldots v_9\}$
$V_1 = \{v_1, v_2, v_3, g_2, g_3\}, g_2 = \{v_4, v_5, v_6\}, g_3 = \{v_7, v_8, v_9\}$

$$f(i,j) = \begin{cases} \frac{1}{|P_{ij}|} \sum_{p \in P_{ij}} e_{i,j} & \text{if } i \text{and } j \text{ represent single nodes} \\ \sum_{v \in i} \left[\frac{1}{|P_{vj}|} \sum_{p \in P_{vj}} e_{v,j} \right] & \text{if } i \text{ represents a subgroup and not } j \\ \sum_{v \in i, u \in j} \left[\frac{1}{|P_{vu}|} \sum_{p \in P_{vu}} e_{v,u} \right] & \text{if } i \text{ and } j \text{ represent subgroups} \end{cases} \tag{1}$$

Grouping Method

Steps:

1. Obtain naive ranking of elements
2. Split elements into k subgroups by naive rank
3. Run the algorithm on each of the subgroups, including the other subgroups as nodes
4. Stack subgroups rankings on top of each other, omitting the rankings of subgroups

$V = \{v_1, v_2, v_3, ...v_9\}$
$V_1 = \{v_1, v_2, v_3, g_2, g_3\}, g_2 = \{v_4, v_5, v_6\}, g_3 = \{v_7, v_8, v_9\}$

$$w(i,j) = \begin{cases} |P_{ij}| & \text{if } i \text{ and } j \text{ represent single nodes} \\ \sum_{v \in i} |P_{vj}| & \text{if } i \text{ represents a subgroup and not } j \\ \sum_{v \in i, u \in j} |P_{vu}| & \text{if } i \text{ and } j \text{ represent subgroups} \end{cases} \quad (2)$$

Time Complexity of Grouping Method

Time complexity of HodgeRank: $O(n^3)$ where $n =$ number of nodes
$k =$ number of groups
Time complexity of HodgeRank with grouping:
$O(k(\frac{n}{k} + k - 1)^3) = O(\frac{n^3}{k^2} + k^2) \ll O(n^3)$

Grouping method: NBA data

- 2021 season NBA game data
  - 1076 games, 30 teams
  - from stats.nba.com
- Voters: teams
- Elements to be ranked: teams
- Scores: point difference between other team and this team

GAME_ID	SEASON	HOME_TEAM_ID	VISITOR_TEAM_ID	PTS_home	PTS_away
22101005	2021	Heat	Timberwolves	104.0	113.0
22101006	2021	Bulls	Cavaliers	101.0	91.0
22101007	2021	Spurs	Pacers	108.0	119.0
22101008	2021	Warriors	Bucks	122.0	109.0
22101009	2021	Nuggets	Raptors	115.0	127.0

Grouping method: NBA data

Hodge Rank

	team	r
1	Suns	7.556319
2	Warriors	6.492398
3	Jazz	5.609594
4	Grizzlies	4.829419
5	Celtics	4.436565
6	Heat	4.335416
7	Mavericks	4.225644
8	Bucks	3.164290
9	Timberwolves	2.894025
10	Cavaliers	2.364359
11	Bulls	2.351655
12	Nuggets	2.102093
13	76ers	1.767214
14	Raptors	1.702416
15	Hawks	1.282467
16	Nets	0.101050
17	Spurs	0.045777
18	Knicks	-0.306826
19	Clippers	-1.228548
20	Hornets	-1.299388
21	Pelicans	-2.183597
22	Pacers	-2.251863
23	Lakers	-3.192266
24	Wizards	-3.263490
25	Kings	-4.236429
26	Trail Blazers	-6.199932
27	Magic	-6.842637
28	Thunder	-7.306982
29	Pistons	-8.199562
30	Rockets	-8.749182

Grouping without Pseudo-nodes

team	r
Suns	3.633397
Warriors	2.932553
Jazz	1.863423
Heat	1.362069
Mavericks	1.341824
Celtics	0.977510
Timberwolves	0.826573
Bucks	-2.960224
Grizzlies	-4.180203
Bulls	-5.796922
Cavaliers	2.689654
76ers	2.506727
Raptors	2.068132
Nets	1.782542
Nuggets	1.619619
Hornets	1.173192
Knicks	-0.754015
Hawks	-2.177948
Clippers	-2.330151
Spurs	-6.577751
Pelicans	4.890045
Kings	3.450702
Lakers	3.178861
Trail Blazers	0.362617
Pacers	-0.279938
Rockets	-0.898655
Thunder	-1.047710
Wizards	-2.308111
Magic	-2.381868
Pistons	-4.965944

Grouping with Pseudo-nodes

team	r
Suns	4.161130
Warriors	3.083983
Jazz	2.043965
Grizzlies	1.469666
Celtics	1.050483
Heat	0.865651
Mavericks	0.844736
Bucks	-0.230932
Timberwolves	-0.570963
Bulls	-1.189181
Cavaliers	1.799867
Nuggets	1.664072
76ers	1.476747
Raptors	1.210336
Hawks	0.866823
Nets	-0.369443
Spurs	-0.490471
Knicks	-0.823813
Hornets	-1.753454
Clippers	-1.880912
Pacers	1.823121
Pelicans	1.715692
Lakers	0.751654
Wizards	0.678636
Kings	-0.407239
Trail Blazers	-2.455151
Magic	-2.718680
Thunder	-3.434886
Pistons	-4.196151
Rockets	-4.882323

Statistical Comparison

Kendall's Tau:

- Statistical comparison of rankings
- Returns a number between 0 (no correlation) and 1 (perfect correlation)
- Takes in two orderings of a set

Kendall's Tau(Hodge Rank, Hodge Rank without Grouping) $\approx 0.393 \Rightarrow$ Somewhat correlated
Kendall's Tau(Hodge Rank, Hodge Rank with Grouping) ≈ 0.959 $\Rightarrow$ Highly correlated

Glen (2017)

Analysis on k

The effect of number of groups on ranking accuracy

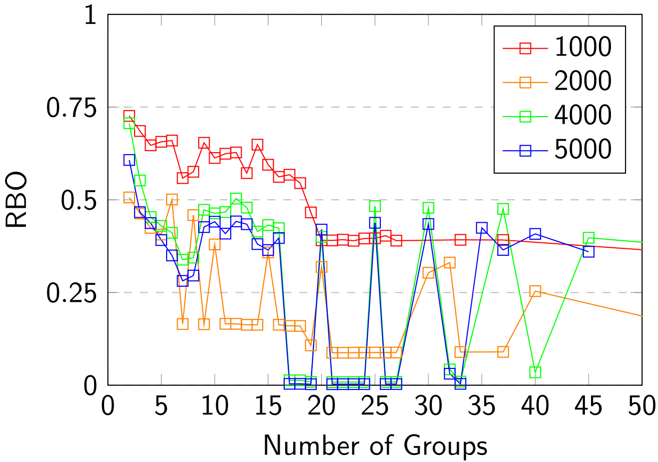
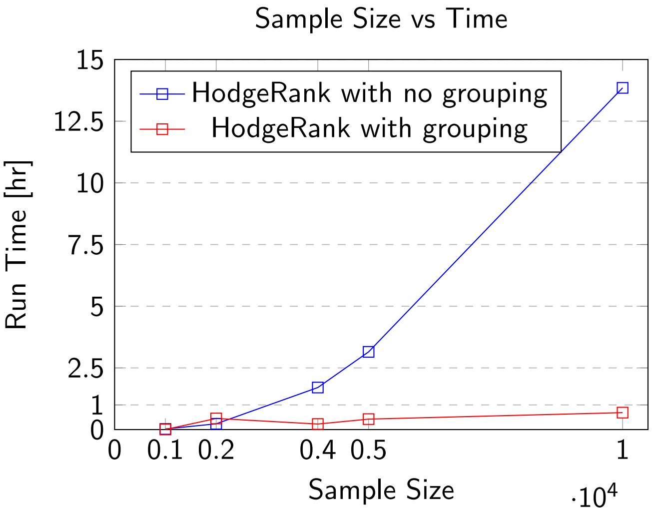


Figure: Note each line corresponds to a different sample size.

Analysis on k



Benefits of the Method

Suitable for data sets that may be:

- biased
- incomplete
- imbalanced

$\Rightarrow$ Useful on sets where every "rating agent" won't rate every item in the set.

Algorithm provides:

- ranking
- a metric for correctness

Further work for grouping method

- Embedded groupings for sets with many elements to be ranked
- Deal with "edge cases"
  - Split up subgroups in a more sophisticated way
- Test method on data sets with more nodes

References I

C. Colley, J. Lin, X. Hu, and S. Aeron. Algebraic multigrid for least squares problems on graphs with applications to hodgerank. In *2017 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, pages 627–636, 2017. doi: 10.1109/IPDPSW.2017.163.

S. Glen. Kendall's tau (kendall rank correlation coefficient), Nov 2017. URL https://www.statisticshowto.com/kendalls-tau/#:~:text=Kendall's%20Tau%20is%20a%20non,1%20is%20a%20perfect%20relationship.

X. Jiang, L.-H. Lim, Y. Yao, and Y. Ye. Statistical ranking and combinatorial hodge theory. *Mathematical Programming*, 127 (1):203–244, 2011. doi: 10.1007/s10107-010-0419-x. URL https://doi.org/10.1007/s10107-010-0419-x.

References II

Lin, Junyuan. Graph laplacian in statistical ranking.

D. Nunan, J. Brassey, K. Boyce, A. Gardner, and M. Patrick-Smith. Covid-19 symptoms tracker, Jun 2020. URL https://www.cebm.net/covid-19/covid-19-signs-and-symptoms-tracker/.

Thank you!

sferrie2@lion.lmu.edu