Ranking and Rankability

Nathan McJames, David Malone and Ollie Mason
Maynooth University.

2022-09-02

Ranking

Things you might rank:

- Search results online.
- Products to advertise to people.
- Universities.
- Which sports team is best.
- Subjects available for study.
- ...

Example Ranking Techniques

Page Rank Suppose you have a random web surfer, who clicks on links on a web page randomly. Rank the pages they wind up on often highly.
(Let a_{ij} be fraction of links from page j that go to page i, corresponds to finding right eigenvalue for $\lambda = 1$.)

Linear Ordering Problem You are given some information about an ordering, e.g. a collection of n teams played m pairwise games and when team i met team j team i scored c_{ij} points. Then then you try to find a zero-one matrix x_{ij} to minimise

$$\sum_{i,j} c_{ij} x_{ij}$$

so $x_{ij} + x_{ji} = 1$ and $x_{ij} + x_{jk} + x_{ki} \leq 2$.
An order is given by $i \succ j$ iff $x_{ij} = 1$.

Rankability

Your scientists were so preoccupied with whether or not they could, they didn't stop to think if they should.

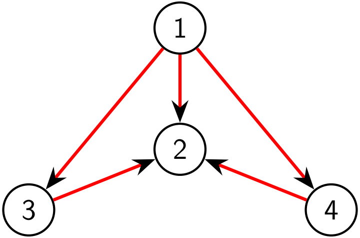
In 2019, Anderson et al[1] asked *should* we rank:

1. What is dataset's degree of rankability?
2. How should that be defined?
3. Can you measure it quickly?
4. Are subsets rankable? What does rankable look like? ...

[1] Anderson, Paul, Timothy Chartier, and Amy Langville. *The rankability of data.* SIAM Journal on Mathematics of Data Science 1.1 (2019): 121-143.

First Proposed Measure

Represent the data as a directed graph.



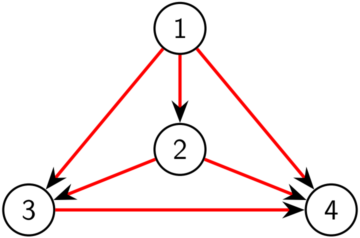
Or, as an adjacency matrix:

$$A = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

Want to compare this to something very rankable.

First Proposed Measure

Compare to *complete dominance graph*.



Or, as an adjacency matrix:

$$A = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

First Proposed Measure

$$R_{\text{edge}} = 1 - \frac{kp}{k_{\max} p_{\max}}.$$

Here k is minimum number of edges that have to be added or removed to reach total dominance graph. p is the number of distinct total dominance graphs at that distance.

These are normalised by

$$k_{\max} = \frac{n(n-1)}{2}, \qquad p_{\max} = n!$$

For our two examples these are 0.986 (as $k = 1$ and $p = 2$) and 1 ($k = 0$, $p = 1$).

First Proposed Measure

- Easy to understand.
- Not very smooth.
- Criminally slow in naive implementation.
- Normalisation is an issue.
- Some insight if you can find nearest graphs.

Second Proposed Measure

Another approach uses the spectrum of the Laplacian of the graph.[2] Let w_{ij} be how much i beats j by (or zero if j beat i). Let A be the matrix with these entries. Let

$$d_i^+ = \sum_j w_{ij} \qquad D = \begin{pmatrix} d_1^+ & 0 & 0 & 0 \\ 0 & d_2^+ & 0 & 0 \\ 0 & 0 & d_3^+ & 0 \\ 0 & 0 & 0 & \ddots \end{pmatrix}$$

Then $L = D - A$ is the graph Laplacian. $\sigma(L)$ is useful.

[2] Cameron, Thomas R., Amy N. Langville, and Heather C. Smith. *On the graph Laplacian and the rankability of data.* Linear Algebra and its Applications 588 (2020): 81-100.

Second Proposed Measure

Theorem

If L is the graph Laplacian of a tournament graph, then

$$\sigma(L) = \{d_1^+, d_2^+, \ldots d_n^+\}$$

iff the graph is acyclic.

Theorem

If L is the graph Laplacian of a tournament graph where $0 \leq w_{ij} \leq 1$ *then*

$$\sigma(L) = \{d_1^+, d_2^+, \ldots d_n^+\} = \{n-1, n-2 \ldots 1, 0\}.$$

iff the graph was a complete dominance graph.

Note, two conditions!

Second Proposed Measure

$$R_{\text{spec}} = 1 - \frac{d_H(\sigma(D), \sigma(S)) + d_H(\sigma(L), \sigma(S))}{2(n-1)}.$$

Here $d_H(X, Y) = \max(\sup_{x \in X} d(x, Y), \sup_{y \in Y} d(X, y))$ and

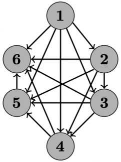
$$S = \begin{pmatrix} n-1 & 0 & 0 & 0 \\ 0 & n-2 & 0 & 0 \\ 0 & 0 & n-3 & 0 \\ 0 & 0 & 0 & \ddots \end{pmatrix}.$$

Looks a lot better

- Smooth.
- Easy to calculate.
- Works with weighted graphs.
- Doesn't have $n!$ in normalisation

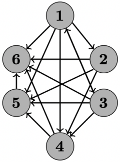
Second Proposed Measure

Complete dominance



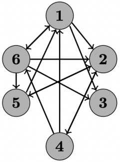
specR = 1.0000
edgeR = 1.0000

Perturbed Dominance



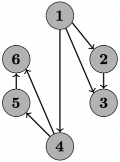
specR = 0.9382
edgeR = 0.9994

Perturbed Random C



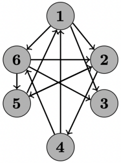
specR = 0.8202
edgeR = 0.9987

Nearly Disconnected



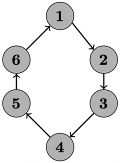
specR = 0.6000
edgeR = 0.9926

Random



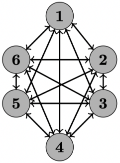
specR = 0.5891
edgeR = 0.9900

Cycle



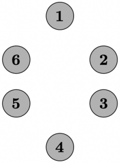
specR = 0.3000
edgeR = 0.9939

Completely Connected



specR = 0.2000
edgeR = 0.0000

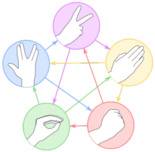
Empty



specR = 0.0000
edgeR = 0.0000.

(Examples from [2])

Examples

Data	R_{edge}	R_{spec}
Rock, Scissors, Paper	0.666667	0.5
	0.975000	0.54775
Random ($n = 6$)	0.985000	0.82470
Scottish Premiership ($n = 12$)	1.000000	0.77279
Random ($n = 12$)	1.000000	0.67809
	???	0.28890
Random ($n = 18$)	1.000000	0.67399

Examples

You can do this for some regular families.

Theorem

If the graph is a cycle of length n then

$$R_{edge} = 1 - \frac{2 + (n-1)(n-2)}{n!(n-1)},$$

and

$$R_{spec} = \begin{cases} 1 - \frac{2n-5}{2n-2} & n \text{ even} \\ 1 - \frac{n-2+|n-2-e^{-i\pi/n}|}{2n-2} & n \text{ odd} \end{cases}.$$

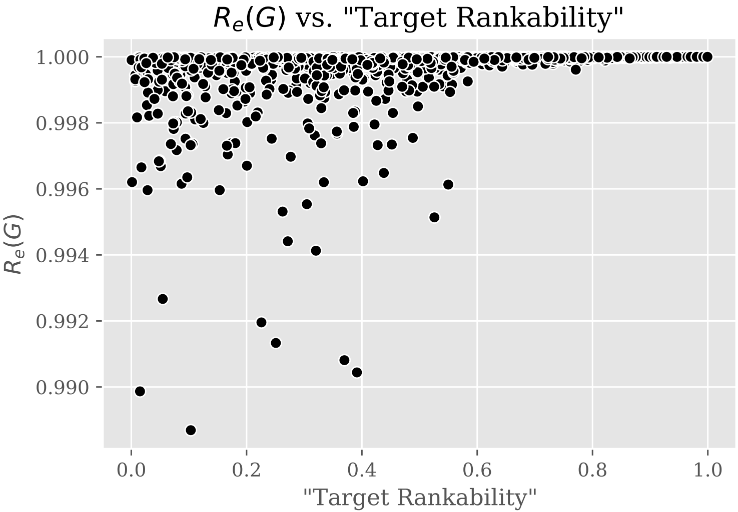
Our Suggestion

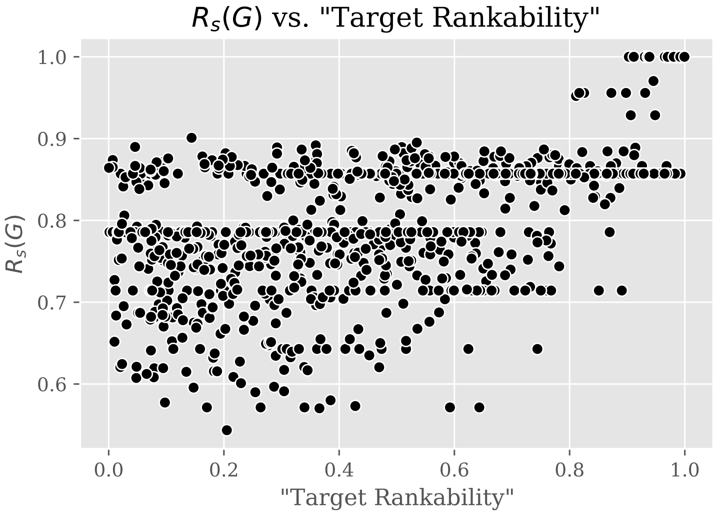
Maybe there are families of graphs that you know the rankability of, for a particular setting.[3]

E.g. Start with complete dominance graph, and flip edges with probability $p < 0.5$. We decide such graphs to have a rankability about $1 - 2p$.

Can we use these to define a rankability?

[3] McJames, Nathan, David Malone, and Oliver Mason. *A supervised learning approach to rankability.* arXiv preprint arXiv:2203.07364 (2022).


$R_e(G)$ vs. "Target Rankability"
1.000
0.998
0.996
0.994
0.992
0.990
$R_e(G)$
0.0
0.2
0.4
0.6
0.8
1.0
"Target Rankability"


$R_s(G)$ vs. "Target Rankability"
1.0
0.9
0.8
0.7
0.6
$R_s(G)$
0.0
0.2
0.4
0.6
0.8
1.0
"Target Rankability"

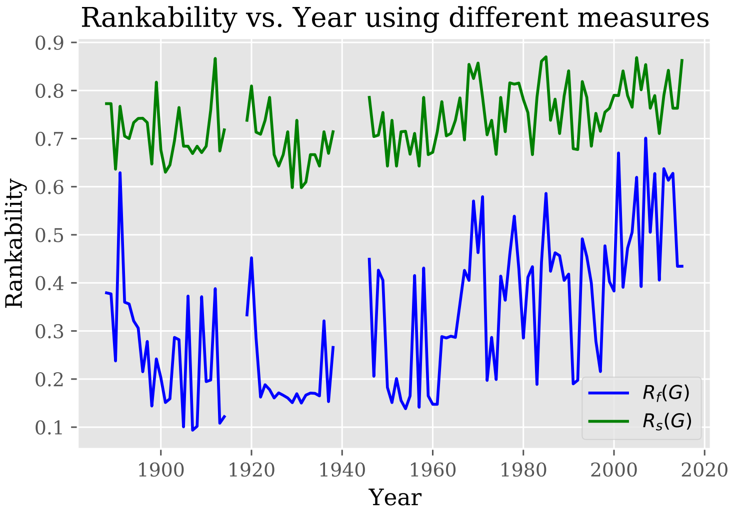
Our Suggestion

Since this is a data problem, and we can generate examples, maybe we can train something to assign rankability.

Plan:

1. Generate graphs with target rankability.
2. Calculate easy graph metrics (number of triangles, 2-cycles, standard deviation of out degrees, directed algebraic connectivity, . . .).
3. Train your favorite model to predict target rankability based on graph metrics.

Tested with random forest. Seems to work reasonably well, fast and easy to adapt, but doesn't always give rankability of 1 to complete dominance graph.



Rankability vs. Year using different measures
Rankability
Year
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
1900
1920
1940
1960
1980
2000
2020
$R_f(G)$
$R_s(G)$

To Wrap Up

- Rankability seems like good question.
- Edge, spectral and build-your-own measures.
- . . . but not sure if we have right definition.
- Are there other families that make good examples?
- Are there good graph properties to build on?
- There's a library if you want to play.[4]

---

[4] Anderson, Paul E., et al. *Developing a Ranking Problem Library (RPLIB) from a data-oriented perspective.* arXiv preprint arXiv:2206.11258 (2022).