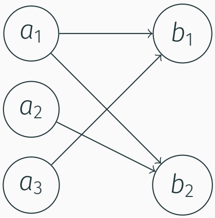
Rankings for Bipartite Tournaments via Chain Editing

Joe Singleton and Richard Booth

October 2020

Bipartite tournaments

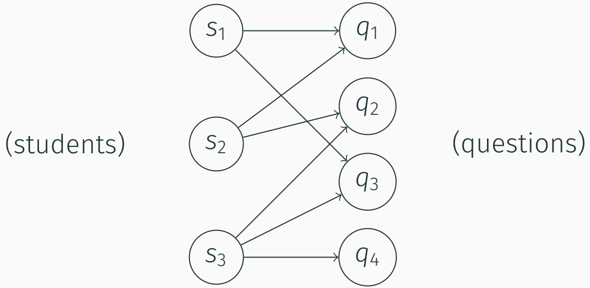
- **Tournament:** set of players together with pairwise comparisons between them
- Ranking has many applications: e.g. sports (chess, football), voting...
- We look at bipartite tournaments



- Need two rankings: one for each side
- e.g. $a_2 \simeq a_3 \prec a_1, \quad b_1 \simeq b_2$

Motivating example: an educational setting

- **Primary example:** students and exam questions



- **Ranking of the students:** who performed best on the exam?
- **Ranking of the questions:** which questions were most difficult?
- Student ranking can depend on difficulty
    - Useful if questions are crowdsourced from students themselves (e.g. PeerWise system)

Outline

Formal model

Formal model

Definition

A bipartite tournament is a triple (A, B, K) where

- $A = \{1, \ldots, m\}$ for some $m \in \mathbb{N}$
- $B = \{1, \ldots, n\}$ for some $n \in \mathbb{N}$
- K is an $m \times n$ matrix with $K_{ab} \in \{0, 1\}$ for all a, b

- $K_{ab} = 1$ if a defeats b; $K_{ab} = 0$ otherwise (no draws)
- Every $a \in A$ plays against every $b \in B$

Example

$$A = \{1, 2, 3\}, \qquad B = \{1, 2, 3, 4\}, \qquad K = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix}$$

Formal model (contd.)

Definition

A ranking operator φ assigns to each tournament K a pair of total preorders $(\preceq_K^\varphi, \preceq_K^\varphi)$ on A and B respectively

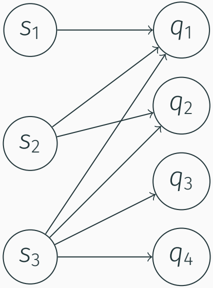
- $a_1 \preceq_K^\varphi a_2$ means a_2 is ranked *at least as strong* as a_1
- $b_1 \preceq_K^\varphi b_2$ interpreted similarly
- **Note:** ties allowed

Ranking via chain editing

Chain graphs

- Suppose there is a ‘true’ ranking of A and B
- **In an ideal world:** results are nested

$$\begin{gathered} s_1 \prec s_2 \prec s_3 \\ q_1 \prec q_2 \prec q_3 \simeq q_4 \end{gathered} \quad \Longrightarrow$$

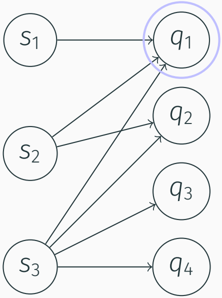


- Tournament is a chain graph: neighbourhoods form a chain w.r.t set inclusion
- Ranking can be recovered from the neighbourhoods

Chain graphs

- Suppose there is a 'true' ranking of A and B
- **In an ideal world:** results are nested

$$s_1 \prec s_2 \prec s_3$$
$$q_1 \prec q_2 \prec q_3 \simeq q_4 \implies$$

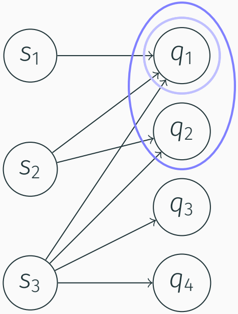


- Tournament is a chain graph: neighbourhoods form a chain w.r.t set inclusion
- Ranking can be recovered from the neighbourhoods

Chain graphs

- Suppose there is a ‘true’ ranking of A and B
- **In an ideal world:** results are nested

$$s_1 \prec s_2 \prec s_3$$
$$q_1 \prec q_2 \prec q_3 \simeq q_4 \implies$$

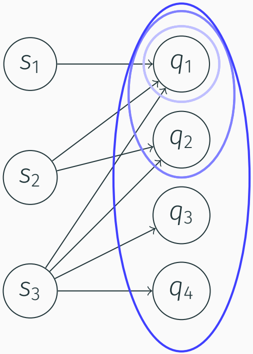


- Tournament is a chain graph: neighbourhoods form a chain w.r.t set inclusion
- Ranking can be recovered from the neighbourhoods

Chain graphs

- Suppose there is a 'true' ranking of A and B
- **In an ideal world:** results are nested

$$s_1 \prec s_2 \prec s_3$$
$$q_1 \prec q_2 \prec q_3 \simeq q_4 \implies$$



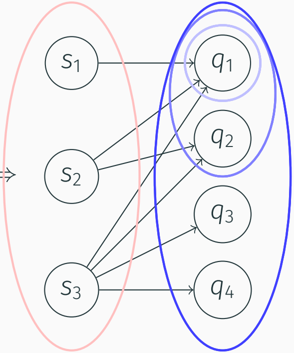
- Tournament is a chain graph: neighbourhoods form a chain w.r.t set inclusion
- Ranking can be recovered from the neighbourhoods

Chain graphs

- Suppose there is a 'true' ranking of A and B
- **In an ideal world:** results are nested

$$s_1 \prec s_2 \prec s_3$$
$$q_1 \prec q_2 \prec q_3 \simeq q_4$$

$\Longrightarrow$



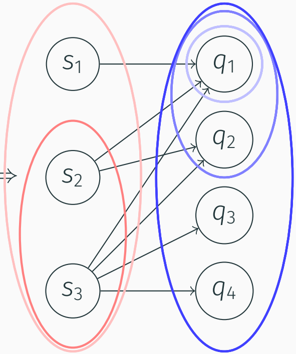
- Tournament is a chain graph: neighbourhoods form a chain w.r.t set inclusion
- Ranking can be recovered from the neighbourhoods

Chain graphs

- Suppose there is a 'true' ranking of A and B
- **In an ideal world:** results are nested

$$s_1 \prec s_2 \prec s_3$$
$$q_1 \prec q_2 \prec q_3 \simeq q_4$$

$\Longrightarrow$



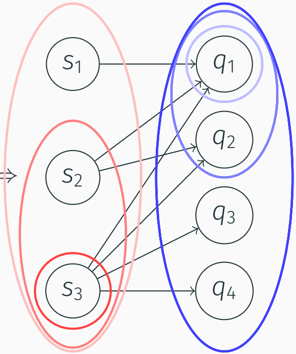
- Tournament is a chain graph: neighbourhoods form a chain w.r.t set inclusion
- Ranking can be recovered from the neighbourhoods

Chain graphs

- Suppose there is a 'true' ranking of A and B
- **In an ideal world:** results are nested

$$s_1 \prec s_2 \prec s_3$$
$$q_1 \prec q_2 \prec q_3 \simeq q_4$$

$\Longrightarrow$



- Tournament is a chain graph: neighbourhoods form a chain w.r.t set inclusion
- Ranking can be recovered from the neighbourhoods

Chain graphs (contd.)

- **In reality:** mistakes happen
- **Idea for a ranking method:** make minimal changes to fix these errors and form a chain graph

Definition (Chain Editing)

Given a bipartite graph $G = (A, B, E)$, find a chain graph G' such that $|G \triangle G'|$ is minimal

- Unfortunately, chain editing is NP-hard
- We partially address this later

Chain editing on tournaments

- In matrix terms...
- Write $K(a) = \{b \in B \mid K_{ab} = 1\}$ for the players defeated by $a \in A$ (the neighbourhood of a)

Definition (Chain tournament)

K is a chain tournament if for all $a_1, a_2 \in A$, either $K(a_1) \subseteq K(a_2)$ or $K(a_2) \subseteq K(a_1)$

- Write $\mathcal{M}(K) = \arg\min_{K' \text{ chain}} d_H(K, K')$ for the set of chain tournaments closest to K w.r.t the Hamming distance

Example

$$\mathcal{M}\left(\begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix}\right) = \left\{\begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}\right\}$$

Chain-minimal operators

- Chain editing property for φ:
- **chain-min:** for every K there is $K' \in \mathcal{M}(K)$ such that

$$a_1 \preceq_K^{\varphi} a_2 \iff K'(a_1) \subseteq K'(a_2)$$
$$b_1 \preceq_K^{\varphi} b_2 \iff (K')^{-1}(b_1) \supseteq (K')^{-1}(b_2)$$

- **Note:** there is no unique **chain-min** operator

A maximum likelihood interpretation

- **chain-min** was motivated by fixing noise in K to find the 'true' ranking
- Can be made precise by maximum likelihood estimation
  - Define possible states of the world θ
  - Given K, maximise $P(K \mid \theta)$
  - Output rankings according to θ

The probabilistic model

Definition

A state of the world θ is a pair $(\boldsymbol{x}, \boldsymbol{y})$ where

- $\boldsymbol{x} = (x_1, \ldots, x_{|A|}) \in \mathbb{R}^{|A|}$ and $\boldsymbol{y} = (y_1, \ldots, y_{|B|}) \in \mathbb{R}^{|B|}$ are **skill levels** of players
- some **explainability** conditions are satisfied...

- **Intuition:** a capable of defeating b in state θ iff $x_a \geq y_b$
- **Noise model:**
  - X_{ab} is binary random variable for the outcome between a and b
  - false positive w.p α_+ (if $x_a < y_b$)
  - false negative w.p α_- (if $x_a \geq y_b$)
  - Independent noise:

$$P(K \mid \theta) = \prod_{a,b} P(X_{ab} = K_{ab} \mid \theta)$$

Maximum likelihood operators

Definition

φ is an MLE operator if for every K there is $\theta = (x, y)$ such that

1. $P(K \mid \theta)$ is maximal
2. $a_1 \preceq_K^\varphi a_2$ iff $x_{a_1} \leq x_{a_2}$
3. $b_1 \preceq_K^\varphi b_2$ iff $y_{b_1} \leq y_{b_2}$

- i.e. for each K, find an MLE $\theta = (x, y)$, and rank according to $\boldsymbol{x}$ and $\boldsymbol{y}$

MLE and chain-min

Theorem

If $\alpha_+ = \alpha_- < \frac{1}{2}$, then

$$\varphi \textit{ MLE} \iff \varphi \textbf{\textit{ chain-min}}$$

Proof outline:

- **Lemma 1:** θ MLE for K iff $d_H(K, K_\theta)$ is minimal
- **Lemma 2:** K is a chain tournament iff $K = K_\theta$ for some θ
- It follows that $\mathcal{M}(K)$ consists of K_θ across all MLEs θ, which implies the result.

(similar results for other values of α_+, α_-)

Relaxing chain editing

Chain definability

- Chain editing has intuitive and theoretical backing, but...
- **Two problems:** NP-hardness and **Anonymity** failure
- Can be resolved by removing minimisation requirement in chain editing
- **chain-def:** for every K there is a chain tournament K' such that

$$a_1 \preceq_K^{\varphi} a_2 \iff K'(a_1) \subseteq K'(a_2)$$
$$b_1 \preceq_K^{\varphi} b_2 \iff (K')^{-1}(b_1) \supseteq (K')^{-1}(b_2)$$

Characterising chain-def

Theorem

*φ satisfies **chain-def** if and only if for every K:*

$$|\mathrm{ranks}(\preceq_K^{\varphi}) - \mathrm{ranks}(\preceq_K^{\varphi})| \leq 1$$

- **Idea for achieving chain-def:** iteratively choose ranks of A and B
- Greedy algorithm for finding a chain graph
- e.g. based on neighbourhood cardinality heuristic:

$$K = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix}$$

Characterising chain-def

Theorem

*φ satisfies **chain-def** if and only if for every K:*

$$|\text{ranks}(\preceq_K^{\varphi}) - \text{ranks}(\preceq_K^{\varphi})| \leq 1$$

- **Idea for achieving chain-def:** iteratively choose ranks of A and B
- Greedy algorithm for finding a chain graph
- e.g. based on neighbourhood cardinality heuristic:

$$K = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix}$$

$$\preceq_K^{\varphi} : 1 \qquad \preceq_K^{\varphi} : 1$$

Characterising chain-def

Theorem

*φ satisfies **chain-def** if and only if for every K:*

$$|\mathrm{ranks}(\preceq^{\varphi}_{K}) - \mathrm{ranks}(\preceq^{\varphi}_{K})| \leq 1$$

- **Idea for achieving chain-def:** iteratively choose ranks of A and B
- Greedy algorithm for finding a chain graph
- e.g. based on neighbourhood cardinality heuristic:

$$K = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix}$$

$\preceq^{\varphi}_{K} : 1 \qquad \preceq^{\varphi}_{K} : 1$

Characterising chain-def

Theorem

*φ satisfies **chain-def** if and only if for every K:*

$$|\mathrm{ranks}(\preceq_K^{\varphi}) - \mathrm{ranks}(\preceq_K^{\varphi})| \leq 1$$

- **Idea for achieving chain-def:** iteratively choose ranks of A and B
- Greedy algorithm for finding a chain graph
- e.g. based on neighbourhood cardinality heuristic:

$$K = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix}$$

$\preceq_K^{\varphi} : 3 \prec 1 \qquad \preceq_K^{\varphi} : 3 \simeq 4 \prec 1$

Characterising chain-def

Theorem

*φ satisfies **chain-def** if and only if for every K:*

$$|\mathrm{ranks}(\preceq_K^\varphi) - \mathrm{ranks}(\preceq_K^\varphi)| \leq 1$$

- **Idea for achieving chain-def:** iteratively choose ranks of A and B
- Greedy algorithm for finding a chain graph
- e.g. based on neighbourhood cardinality heuristic:

$$K = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix}$$

$\preceq_K^\varphi : 3 \prec 1$ $\quad$ $\preceq_K^\varphi : 3 \simeq 4 \prec 1$

Characterising chain-def

Theorem

*φ satisfies **chain-def** if and only if for every K:*

$$|\mathrm{ranks}(\preceq_K^\varphi) - \mathrm{ranks}(\preceq_K^\varphi)| \leq 1$$

- **Idea for achieving chain-def:** iteratively choose ranks of *A* and *B*
- Greedy algorithm for finding a chain graph
- e.g. based on neighbourhood cardinality heuristic:

$$K = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix}$$

$\preceq_K^\varphi : 2 \prec 3 \prec 1 \qquad \preceq_K^\varphi : 5 \prec 3 \simeq 4 \prec 1$

Characterising chain-def

Theorem

*φ satisfies **chain-def** if and only if for every K:*

$$|\mathrm{ranks}(\preceq_K^{\varphi}) - \mathrm{ranks}(\preceq_K^{\varphi})| \leq 1$$

- **Idea for achieving chain-def:** iteratively choose ranks of A and B
- Greedy algorithm for finding a chain graph
- e.g. based on neighbourhood cardinality heuristic:

$$K = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix}$$

$\preceq_K^{\varphi} : 2 \prec 3 \prec 1$ $\qquad$ $\preceq_K^{\varphi} : 5 \prec 3 \simeq 4 \prec 1$

Characterising chain-def

Theorem

*φ satisfies **chain-def** if and only if for every K:*

$$|\mathrm{ranks}(\preceq_K^{\varphi}) - \mathrm{ranks}(\preceq_K^{\varphi})| \leq 1$$

- **Idea for achieving chain-def:** iteratively choose ranks of A and B
- Greedy algorithm for finding a chain graph
- e.g. based on neighbourhood cardinality heuristic:

$$K = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix}$$

$\preceq_K^{\varphi} : 4 \prec 2 \prec 3 \prec 1 \qquad \preceq_K^{\varphi} : 2 \prec 5 \prec 3 \simeq 4 \prec 1$

Characterising chain-def

Theorem

*φ satisfies **chain-def** if and only if for every K:*

$$|\mathrm{ranks}(\preceq_K^{\varphi}) - \mathrm{ranks}(\preceq_K^{\varphi})| \leq 1$$

- **Idea for achieving chain-def:** iteratively choose ranks of A and B
- Greedy algorithm for finding a chain graph
- e.g. based on neighbourhood cardinality heuristic:

$$K = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix}$$

$\preceq_K^{\varphi} : 4 \prec 2 \prec 3 \prec 1$ $\qquad$ $\preceq_K^{\varphi} : 2 \prec 5 \prec 3 \simeq 4 \prec 1$

Characterising chain-def (contd.)

- Iteratively choosing ranks can be generalised: we call such operators interleaving operators

Theorem

*φ satisfies **chain-def** if and only if φ is an interleaving operator*

- **Cardinality-based example:**
  - Polynomial time: ✓
  - Anonymity: ✓

Conclusion and future work

Conclusion and future work

So far...

- We studied chain editing for ranking bipartite tournaments
- Obtained a maximum likelihood interpretation
- Resolved computational difficulties by relaxing to **chain-def**

In the future...

- Allow draws and abstentions
- Approximation algorithms for chain editing
- Experimental analysis