We consider the problem of ranking and learning the qualities \( w_1, \ldots, w_n \) of a collection of items by performing noisy comparisons among them. We assume that there is a fixed “comparison graph”, and every neighboring pair of items is compared \( k \) times.

We focus more specifically on the popular Bradley-Terry-Luce model, where comparisons are i.i.d. events, and the probability for item \( i \) to win the comparison against \( j \) is \( \frac{w_i}{w_i + w_j} \).

We propose a near-linear time algorithm allowing us to recover the weights with an accuracy that outperforms all existing algorithms, and show that this accuracy is actually within a constant factor of information-theoretic lower bounds, that we also develop. This accuracy is related to the average resistance of the comparison graph.

Our algorithm is based on a weighted least square, with weights determined from empirical outcomes of the comparisons.

We further discuss the extension to other models of comparisons, and comparisons involving multiple items.
Ranking from pairwise comparisons: a near-linear time minimax optimal algorithm for learning BTL weights

Julien Hendrickx – Lille – 10 March 2023
What if Ligue 1 has to stop now?

Who is champion?  What is the ranking?
→ who goes to L2, to European league etc.

**Possible solution:** use current standing

<table>
<thead>
<tr>
<th></th>
<th>pts</th>
<th>J</th>
<th>G</th>
<th>N</th>
<th>P</th>
<th>D</th>
<th>+/-</th>
<th>G.</th>
<th>N.</th>
<th>P.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>63</td>
<td>26</td>
<td>20</td>
<td>3</td>
<td>3</td>
<td>66</td>
<td>+61</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>55</td>
<td>26</td>
<td>17</td>
<td>4</td>
<td>5</td>
<td>49</td>
<td>+24</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>51</td>
<td>26</td>
<td>15</td>
<td>6</td>
<td>9</td>
<td>55</td>
<td>+19</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>51</td>
<td>26</td>
<td>14</td>
<td>9</td>
<td>3</td>
<td>40</td>
<td>+19</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>46</td>
<td>26</td>
<td>14</td>
<td>4</td>
<td>8</td>
<td>45</td>
<td>+16</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>45</td>
<td>26</td>
<td>13</td>
<td>6</td>
<td>7</td>
<td>46</td>
<td>+13</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>42</td>
<td>26</td>
<td>11</td>
<td>9</td>
<td>6</td>
<td>34</td>
<td>+12</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>40</td>
<td>26</td>
<td>9</td>
<td>13</td>
<td>4</td>
<td>34</td>
<td>+8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>40</td>
<td>26</td>
<td>11</td>
<td>7</td>
<td>8</td>
<td>38</td>
<td>+2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>39</td>
<td>26</td>
<td>11</td>
<td>6</td>
<td>9</td>
<td>39</td>
<td>+11</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>34</td>
<td>26</td>
<td>9</td>
<td>7</td>
<td>10</td>
<td>26</td>
<td>-8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>32</td>
<td>26</td>
<td>9</td>
<td>5</td>
<td>12</td>
<td>41</td>
<td>-5</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
What if Ligue 1 has to stop now?

Who is champion?  What is the ranking?
→ who goes to L2, to European league etc.

Possible solution: use current standing

Nice and Reims similar
But 2 weeks ago

- Nice should get more recognition
- “Current standing” option unfair for teams who only played stronger teams
What if Ligue 1 has to stop now?

Who is champion? What is the ranking? → who goes to L2, to European league etc.

- Nice should get more recognition
- “Current standing” option unfair for teams who only played stronger teams

Inherent problem when *games are not all-to-all*

- Tennis ranking
- Chess
- (...)

→ How to *build ranking / # points from results of “arbitrary” comparisons*
How to evaluate pain-killer efficiency

Asking patients number between 1 and 10?

- Good but not very objective + patient dependent
- Can’t test all on all patient
- Preference for giving “good ones”

Practical data collection: try 2 and ask which is best + learn quality
Online review

<table>
<thead>
<tr>
<th>UNDERSTANDING ONLINE STAR RATINGS:</th>
</tr>
</thead>
<tbody>
<tr>
<td>⭐⭐⭐⭐⭐ [HAS ONLY ONE REVIEW]</td>
</tr>
<tr>
<td>⭐⭐⭐⭐⭐ EXCELLENT</td>
</tr>
<tr>
<td>⭐⭐⭐⭐⭐ OK</td>
</tr>
<tr>
<td>⭐⭐⭐⭐⭐ CRAP</td>
</tr>
<tr>
<td>⭐⭐⭐⭐⭐</td>
</tr>
<tr>
<td>⭐⭐⭐⭐⭐</td>
</tr>
<tr>
<td>⭐⭐⭐⭐⭐</td>
</tr>
</tbody>
</table>

less than 5* often an insult

⇒ Not very informative

*Alternative:* did you prefer this place or this place
Comparison can be all you have

Preference expressed by action

Multiple items, not everyone compares all

How to rank/recover value based on (non-exhaustive) comparisons?
Bradley-Terry-Luce model

- Items have intrinsic quality (weight): $w_i$
- When comparing $i - j$, $i$ wins with probability $p_{ij} = \frac{w_i}{w_i + w_j}$

Example

pick coffee with 80% probability, tea with 20%

XXX football team: 3  YYY football team: 2

$\Rightarrow$ XXX should win with probability 60%

Idea: recover weights $w_i$ from the comparison results
Ranking from pairwise comparisons

- Motivation and Problem
- Weighted Least-Square Estimator
- Algorithm and Complexity
- Error Analysis
  - Error Bound
  - Lower Bound – Minimax Optimality
  - Other criteria
- Experimental Results
- A Surprising Observation
- Generalizations
- Conclusions
Ranking from pairwise comparisons

• Motivation and Problem
• **Weighted Least-Square Estimator**
• Algorithm and Complexity
• Error Analysis
  • Error Bound
  • Lower Bound – Minimax Optimality
  • Other criteria
• Experimental Results
• A Surprising Observation
• Generalizations
• Conclusions
Weight recovery

Items 1, ..., n with quality (weights) $w_1, ..., w_n \in [1, b]$

Comparison graph

$k$ i.i.d. comparisons for each edge

$i$ wins comparison against $j$ with probability

$$p_{ij} = \frac{w_i}{w_i + w_j}$$

Problem: Recover vectors of weights $w = (w_1, ..., w_n)'$ from results, up to constant multiplicative factor. Range $b$ exists but is not known

Sufficient statistics: $k$ and ratio of wins

$$R_{ij} = \frac{\# \text{ wins } i}{\# \text{ wins } j}$$
Data has network structure

*Sufficient statistics:* $k$ and ratio of wins

$$R_{ij} = \frac{\# \text{ wins } i}{\# \text{ wins } j}$$

Goal = recover values at nodes
Previous solutions

• Maximum Likelihood
  • Convex optimization problem after reformulation
  • Asymptotically optimal, but only asymptotic guarantees

• Rank centrality [Negahban, Oh, Shah 2016]
  • Based on convergence of Markov Chain built from data

\[
\frac{\|w - \hat{W}\|_2^2}{\|w\|_1} \leq O\left(\frac{1}{k}\right) \frac{b^5 \log n}{(1 - \rho)^2} \frac{d_{\max}}{d_{\min}^2},
\]

1 $-$ $\rho$ spectral gap of random walk
$d_{\max}, d_{\min}$ largest, smallest degree
$b$ maximal weight

Could scale as $n^7 b^5 / k$

Several improvements
Algorithm idea: Least-Square

Probability i wins over j: \( \frac{w_i}{w_i + w_j} \)

For large number \( k \) of comparisons i - j:

\[
\begin{align*}
\text{# win } i & \approx k p_{ij} = k \frac{w_i}{w_i + w_j} \\
\text{# win } j & \approx k p_{ji} = k \frac{w_j}{w_i + w_j}
\end{align*}
\]

\[ R_{ij} = \frac{\text{# win } i}{\text{# win } j} \approx \frac{w_i}{w_j} \]

\[ \log w_i - \log w_j \approx \log R_{ij} \]

(Naïve) Idea 1: Least-square solution of

\[ \log \hat{w}_i - \log \hat{w}_j = \log R_{ij} \quad \forall (i, j) \in E \]
Issue 1: zero wins

Lease square solution of

$$\log \hat{\omega}_i - \log \hat{\omega}_j = \log R_{ij} \quad \forall (i, j) \in E$$

$R_{ij} = \frac{\# \text{ wins } i}{\# \text{ wins } j}$

What if $i$ wins no comparison? (or all)

$R_{ij} = 0 \Rightarrow \log R_{ij} = -\infty$

→ Complete Failure, with positive probability

Solution: Replace 0 victory by $\frac{1}{2}$ victory

- Simple
- provides boundedness properties
- But creates technical complications
## Issue 2: Non-uniform Variance

Lease square solution of

\[
\log \hat{w}_i - \log \hat{w}_j = \log R_{ij} \quad \forall (i, j) \in E
\]

<table>
<thead>
<tr>
<th>Variance # win i</th>
<th>5 vs 5</th>
<th>9 vs 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>( k )</td>
<td>( k )</td>
<td>( k )</td>
</tr>
<tr>
<td>( \frac{k}{v_{ij}} )</td>
<td>( \frac{k}{4} )</td>
<td>( \frac{k}{11.11} )</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>“Variance” log R_{ij}</th>
<th>5 vs 5</th>
<th>9 vs 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \approx \frac{v_{ij}}{k} )</td>
<td>( \approx \frac{4}{k} )</td>
<td>( \frac{11.11}{k} )</td>
</tr>
</tbody>
</table>

With \( v_{ij} := \frac{w_i}{w_j} + 2 + \frac{w_j}{w_i} \)

Error in equation (9,1) expected to be larger than for (5,5)

\[ \Rightarrow \text{Corresponding equations should be treated differently.} \]
Solution: Weighted least square

Least square solution of

$$\frac{\log \hat{\omega}_i - \log \hat{\omega}_j}{\sqrt{v_{ij}}} = \frac{\log R_{ij}}{\sqrt{v_{ij}}}$$

Idea: each equation should have "the same variance"

(inspired by Best Linear Unbiased Estimator idea)

$$v_{ij} := \frac{w_i}{w_j} + 2 + \frac{w_j}{w_i}$$

$\rightarrow$ Ideal Estimator

$$\log \hat{\omega} = \arg \min_z \sum_{(i,j) \in E} \frac{(z_i - z_j - \log R_{ij})^2}{v_{ij}}$$
Weighted least square

**Ideal Estimator**

\[
\log \hat{\omega} = \arg \min_{\omega} \sum_{(i,j) \in E} \frac{(z_i - z_j - \log R_{ij})^2}{v_{ij}}
\]

**Issue 3:** \( v_{ij} := \frac{w_i}{w_j} + 2 + \frac{w_i}{w_j} \) Depends on the values we want to recover

**Iterative solution:**
- Initiate \( \hat{v}_{ij} = 4 \) for all edges
- Repeat
  - Compute estimate \( \hat{\omega} \) with \( \hat{v}_{ij} \)
  - Update \( \hat{v}_{ij} \) based on \( \hat{\omega} \)

**Empirical solution:**

\[
R_{ij} \approx \frac{w_i}{w_j} \quad \Rightarrow \quad v_{ij} := \frac{w_i}{w_j} + 2 + \frac{w_j}{w_i} \approx R_{ij} + 2 + R_{ij}^{-1}
\]
Weighted least square

**Ideal Estimator**

\[
\log \hat{w} = \arg \min_z \sum_{(i,j) \in E} \frac{(z_i - z_j - \log R_{ij})^2}{v_{ij}}
\]

**Issue 3:**

\[
v_{ij} := \frac{w_i}{w_j} + 2 + \frac{w_i}{w_j}
\]

Depends on the values we want to recover

**Empirical solution:**

\[
R_{ij} \approx \frac{w_i}{w_j} \Rightarrow v_{ij} := \frac{w_i}{w_j} + 2 + \frac{w_j}{w_i} \approx R_{ij} + 2 + R_{ij}^{-1}
\]

- Computationally cheaper
- Simpler to analyze
- More accurate (surprisingly)
Final Estimator

\[
\log \hat{w} = \arg \min_z \sum_{(i,j) \in E} \frac{(z_i - z_j - \log R_{ij})^2}{\hat{\nu}_{ij}}
\]

With \( \hat{\nu}_{ij} := R_{ij} + 2 + R_{ij}^{-1} \) \hspace{1cm} \text{Empirical “variance”}

\[
R_{ij} = \frac{\# \text{ wins } i}{\# \text{ wins } j}
\]

- \( \hat{w} \) computed by solving linear least-square problem
- But nonlinear dependence on data and \( R_{ij} \)
- No hyper parameter, tuning etc. (can be introduced)
- Can be computed in near linear time

Accuracy \( \epsilon \) in \( O \left( |E| \log^c n \log \frac{1}{\epsilon} \right) \)
Ranking from pairwise comparisons

• Motivation and Problem
• Weighted Least-Square Estimator
• Algorithm and Complexity
• Error Analysis
  • Error Bound
  • Lower Bound – Minimax Optimality
  • Other criteria
• Experimental Results
• A Surprising Observation
• Generalizations
• Conclusions
Reminder Incidence matrix $B$

Relates **nodes to edges**

**Column:** edge  
**Row:** nodes  

If edge $e$ from $i$ to $j$  

$$B_{ie} = -1$$  
$$B_{je} = 1$$  

*Orientation arbitrary*
Compact reformulation with B

Relates **nodes to edges**

- Column: edge
- Row: nodes

If edge e from i to j

\[
\begin{align*}
B_{ie} &= -1 \\
B_{je} &= 1
\end{align*}
\]

Orientation arbitrary

\[z_i - z_j = \log R_{ij}\] for all \((i, j) \in E\]

\[B^T z = \log R\]

- One equation / edge
- One variable / node

With \(R \in \mathbb{R}^{|E|}\) vector of \(R_{ij}\)
Compact reformulation with $B$

Relates **nodes to edges**

<table>
<thead>
<tr>
<th>Column:</th>
<th>edge</th>
<th>Row:</th>
<th>nodes</th>
</tr>
</thead>
</table>

If edge $e$ from $i$ to $j$

$$B_{ie} = -1$$

$$B_{je} = 1$$

Orientation arbitrary

$\rightarrow$ System

$$\frac{z_i - z_j}{\sqrt{v_{ij}}} = \frac{\log R_{ij}}{\sqrt{v_{ij}}}$$

for all $(i, j) \in E$

Can be rewritten compactly

$$V^{-1/2}B^T z = V^{-1/2} \log R$$

With $R \in \mathbb{R}^{|E|}$ vector of $R_{ij}$

$$V = \text{diag} \left( \ldots, v_{ij}, \ldots \right)$$

$v_{ij}$ approximated from data
Least-Square

*Estimator:* $\log \widehat{w}$ least square solution of

$$V^{-1/2} B^T z = V^{-1/2} \log R$$

Normal equations $\Rightarrow$ solution of

$$(V^{-\frac{1}{2}} B^T)^T V^{-1/2} B^T z = (V^{-\frac{1}{2}} B^T)^T V^{-1/2} \log R$$
Least-Square

**Estimator:** \( \log \hat{\omega} \) least square solution of

\[
V^{-1/2} B^T z = V^{-1/2} \log R
\]

Normal equations \( \rightarrow \) solution of

\[
(V^{-\frac{1}{2}} B^T)^T V^{-1/2} B^T z = (V^{-\frac{1}{2}} B^T)^T V^{-1/2} \log R
\]

\[
BV^{-1}B^T z = BV^{-1} \log R
\]

*(weighted) Laplacian matrix*
Reminder: Laplacian Matrix

Represents
- relations between nodes
- degrees

$L_{ij} = -1$ if edge $(i, j)$
$L_{ii} = \text{degree}(i)$

![Graph diagram]

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>-1</td>
<td>3</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>3</td>
<td>-1</td>
<td>-1</td>
<td>3</td>
<td>-1</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>-1</td>
<td>-1</td>
<td>2</td>
<td>-1</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>
Reminder: Laplacian Matrix

Represents
- relations between nodes
- degrees

$$L_{ij} = -1 \text{ if edge } (i, j)$$
$$L_{ii} = \text{degree}(i)$$

**Interesting properties**

- $L = BB^T$
- $L1 = 0$ (sum line = 0)
- Positive semi-definite
- $\lambda_2 > 0$ if graph connected
  + “algebraic connectivity”
Reminder: **Weighted Laplacian Matrix**

Weights $A_{ij} = A_{ji}$ on edges

- Represents Weights of relations between nodes
- Degrees/strengths of nodes

**Interesting properties**

- $L = B \text{diag} \left( A_{ij} \right) B^T$
- $L1 = 0$ (sum line = 0)
- Positive semi-definite
- $\lambda_2 > 0$ if graph connected
  + “algebraic connectivity”

$L_{ij} = -A_{ij}$ if edge $(i,j)$

$L_{ii} = \text{strength}(i) = \sum_{j \neq i} A_{ij}$

$\text{diag} \left( A_{ij} \right) \in \mathbb{R}^{\left| E \right| \times \left| E \right|}$
Final algorithm: Laplacian System

\[ BV^{-1} B^T z = BV^{-1} \log R \]

\[ =: L_V \quad \text{(weighted) Laplacian matrix} \]

\[ \log \hat{w} = \text{solutions of} \quad L_V z = BV^{-1} \log R \]

\[ R \in \mathbb{R}^{|E|} \text{ vector of } R_{ij} \]

\[ \frac{\# \text{ wins } i}{\# \text{ wins } j} \]

\[ V = \text{diag} (\ldots, v_{ij}, \ldots) \]

"variance" empirically estimated

Laplacian \( L_V \) is \textit{symmetric} and \textit{diagonally} dominant (\( L_V,ii = -\sum_{j \neq i} L_V,ij \))

[Spielman, Teng 2014], system solved up to accuracy \( \epsilon \) in \( O \left( |E| \log^c n \log \frac{1}{\epsilon} \right) \)

\( \rightarrow \text{Near linear time in size } |E| \text{ of data.} \)

For reasonable size systems, easier to use classical solver
Ranking from pairwise comparisons

• Motivation and Problem
• Weighted Least-Square Estimator
• Algorithm and Complexity
• Error Analysis
  • Error Bound
    • Lower Bound – Minimax Optimality
    • Other criteria
• Experimental Results
• A Surprising Observation
• Generalizations
• Conclusions
Error analysis

**Disclaimer:** Intuitive heuristic analysis

Formal proofs
- Exist
- Were guided by this analysis
- Involve many technical difficulties
- Probably not for a presentation.

In particular we assume
- \( E \log R_{ij} = \log \rho_{ij} \quad \rho_{ij} := \frac{w_i}{w_j} \)
- Variance \( \log R_{ij} = \frac{v_{ij}}{k} \)
- Exact \( v_{ij} \) used in the algorithm

(all this is “asymptotically” true)
Error analysis

\[ \log \hat{w} = \text{solutions of} \quad L_V z = BV^{-1} \log R \]

**How accurate is this estimate?**  \( \rightarrow \) characterize \( \Delta \log w = \log \hat{w} - \log w \)

**Scale Problem:**
- \( w, \hat{w} \) only defined \textit{up to multiplicative constant} \( p_{ij} = \frac{w_i}{w_i + w_j} \)
- \( \log w, \log \hat{w} \) defined up to \textit{additive constant}

\( \rightarrow \) Arbitrary choice: \( \log w, \log \hat{w} \) sum to 0, i.e. orthogonal to 1

\( \rightarrow \)

\[ \log \hat{w} = L_V^\dagger BV^{-1} \log R \quad \text{With } L_A^\dagger \text{ Monroe Penrose Pseudo-inverse (kernel and image orthogonal to 1) } \]

\[ \log w = L_V^\dagger BV^{-1} \log \rho \quad \rho_{ij} = \frac{w_i}{w_j} \quad \text{true ratio} \]
\[ \log \hat{w} = L_v^\dagger B V^{-1} \log R \]
\[ \log w = L_v^\dagger B V^{-1} \log \rho \]
\[ \Delta \log w = L_v^\dagger B V^{-1} \Delta \log R \]

\[
E \Delta \log w \Delta \log w^T = E \left( L_v^\dagger B V^{-1} \Delta \log R \right) \left( L_v^\dagger B V^{-1} \Delta \log R \right)^T
\]
\[
= EL_v^\dagger B V^{-1} \Delta \log R \Delta \log R^T V^{-1} B^T L_v^\dagger
\]
\[
= L_v^\dagger B V^{-1} \left( E \Delta \log R \Delta \log R^T \right) V^{-1} B^T L_v^\dagger
\]
\[
\begin{align*}
\log \hat{w} &= L_v^\dagger B V^{-1} \log R \\
\log w &= L_v^\dagger B V^{-1} \log \rho \\
\log R &\rightarrow \Delta \log w = L_v^\dagger B V^{-1} \Delta \log R
\end{align*}
\]

\[
E \Delta \log w \Delta \log w^T = E \left( L_v^\dagger B V^{-1} \Delta \log R \right) \left( L_v^\dagger B V^{-1} \Delta \log R \right)^T
\]
\[
= E L_v^\dagger B V^{-1} \Delta \log R \Delta \log R^T V^{-1} B^T L_v^\dagger
\]
\[
= L_v^\dagger B V^{-1} (E \Delta \log R \Delta \log R^T) V^{-1} B^T L_v^\dagger
\]

Square “co-variance” matrix, |E|×|E|
- Diagonal because edges independent and we assume \( E \Delta \log R_{ij} = 0 \)
- for edge \((i, j)\) value \( v_{ij} / k \)

\[
\rightarrow E \Delta \log R \Delta \log R^T = \frac{1}{k} V
\]
\[
\begin{align*}
\log \hat{w} &= L^\dagger_B V^{-1} \log R \\
\log w &= L^\dagger_B V^{-1} \log \rho \\
\rightarrow \quad \Delta \log w &= L^\dagger_B V^{-1} \Delta \log R \\
E \Delta \log w \Delta \log w^T &= E \left( L^\dagger_B V^{-1} \Delta \log R \right) \left( L^\dagger_B V^{-1} \Delta \log R \right)^T \\
&= EL^\dagger_B V^{-1} \Delta \log R \Delta \log R^T V^{-1} B^T L^\dagger_B \\
&= L^\dagger_B V^{-1} \left( E \Delta \log R \Delta \log R^T \right) V^{-1} B^T L^\dagger_B \\
&= \frac{1}{k} L^\dagger_B V^{-1} V V^{-1} B^T L^\dagger_B \\
&= \frac{1}{k} L^\dagger_B V^{-1} B^T L^\dagger_B \\
&= \frac{1}{k} L^\dagger_B L_V L^\dagger_V = \frac{1}{k} L^\dagger_V \\
\text{by property of Monroe-Penrose inverse}
\end{align*}
\]
Summary: For a given graph and vector of weight, for large enough k (non-asymptotic)

\[ E \Delta \log w \Delta \log w^T \approx \frac{1}{k} L_Y^+ \]

Pseudo-inverse of weighted Laplacian, Weights = inverse variance \( v_{ij}^{-1} \)

Square Error \( E \| \log \hat{w} - \log w \|^2 \approx \frac{1}{k} Tr(L_Y^+) \)
Reminder: Graph resistance

Weights $A_{ij} = A_{ji}$ represent conductance of wires

Effective Resistance $\Omega_{ij} = V / \text{current}$ if $V$ volts between $i$ and $j$

Average resistance: Average over all pairs

$$\Omega_{av} = \frac{1}{n} Tr (L_A^+) = \frac{1}{n} \sum_{i>1} \frac{1}{\sigma_i(L_A)}$$

With $L_A^+$ Monroe Penrose Pseudo-inverse

Alternative measure of connectivity – less centered on “worst-case”
Summary: For a given graph and vector of weight, for large enough k (non-asymptotic)

\[ E \Delta \log w \Delta \log w^T \approx \frac{1}{k} L_V^\dagger \]

Pseudo-inverse of weighted Laplacian, Weights = inverse variance $v_{ij}^{-1}$

Square Error \[ E \| \log \hat{w} - \log w \|^2 \approx \frac{1}{k} Tr(L_V^\dagger) = \frac{n}{k} \Omega_{V,av} \]

(\( \rightarrow \) Mean square error \( \frac{1}{k} \Omega_{V, av} \))
Summary: For a given graph and vector of weight, for large enough k (non-asymptotic)

\[ E \Delta \log w \Delta \log w^T \approx \frac{1}{k} L_V^+ \]

Pseudo-inverse of weighted Laplacian,
Weights = inverse variance \( v_{ij}^{-1} \)

Square Error \( E \| \log \hat{w} - \log w \|^2 \approx \frac{1}{k} Tr(L_V^+) = \frac{n}{k} \Omega_{V,av} \)

\[ = O \left( \frac{bn^2}{k} \right) = O \left( \frac{bn\Omega_{av}}{k} \right) \]

- \( \Omega_{av} \) resistance unweighted graph
- b maximal ratio of weights.

- Accuracy determined by average resistance
- \( O \left( \frac{bn^2}{k} \right) \) vs \( O \left( \frac{bn^5 n^7}{k} \right) \) (But criteria not strictly comparable)
## Bound comparison

<table>
<thead>
<tr>
<th>Graph</th>
<th>Negahban 16</th>
<th>Our result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Line</td>
<td>$b^{5/2}n^2$</td>
<td>$b\sqrt{n}$</td>
</tr>
<tr>
<td>Circle</td>
<td>$b^{5/2}n^2$</td>
<td>$b\sqrt{n}$</td>
</tr>
<tr>
<td>2D grid</td>
<td>$b^{5/2}n$</td>
<td>$b$</td>
</tr>
<tr>
<td>3D grid</td>
<td>$b^{5/2}n^{2/3}$</td>
<td>$b$</td>
</tr>
<tr>
<td>Star graph</td>
<td>$b^{5/2}\sqrt{n}$</td>
<td>$b$</td>
</tr>
<tr>
<td>2 stars joined at centers</td>
<td>$b^{5/2}n^{1.5}$</td>
<td>$b\sqrt{n}$</td>
</tr>
<tr>
<td>Barbell graph</td>
<td>$b^{5/2}n^{3.5}$</td>
<td>$b$</td>
</tr>
<tr>
<td>Geo. random graph</td>
<td>$b^{5/2}n$</td>
<td>$b$</td>
</tr>
<tr>
<td>Erdos-Renyi</td>
<td>$b^{5/2}$</td>
<td>$b$</td>
</tr>
</tbody>
</table>

Factor 1/k omitted
Ranking from pairwise comparisons

• Motivation and Problem
• Weighted Least-Square Estimator
• Algorithm and Complexity
• Error Analysis
  • Error Bound
    • Lower Bound – Minimax Optimality
  • Other criteria
• Experimental Results
• A Surprising Observation
• Generalizations
• Conclusions
Lower bound

\( \frac{1}{k} L^\dagger_V = \text{Fisher information matrix,} \)

But, many relevant estimates biased \( \rightarrow \) **Cramer-Rao not directly applicable**

Nevertheless:

**Theorem:** For any nominal weights \( w \) and any comparison graph,

There is a way of generating \( w_z \) randomly in a ball of radius \( O_{w,G}(\frac{1}{\sqrt{k}}) \)

(with \( \sum_i (w_z)_i = \sum_i w_i \))

such that for any estimator \( \hat{w} \) using the outcome \( Y \) of \( k \) comparisons

\[
E \| \log \hat{w}(Y) - \log w_z \|^2 \geq \Omega \left( \frac{1}{k} \right) Tr(L^\dagger_V)
\]

\( \rightarrow \) For large enough # comparisons, simple least square algorithm

**is minimax optimal** (up to constant factor)
Proof technique

1) Generate $w_z$ by combining i.i.d. variations along eigenvectors of $L_V$

2) Exploit **Lemma 6.1.** Let $\mu$ be any joint probability distribution of a random pair $(w, w')$, such that the marginal distributions of both $w$ and $w'$ are equal to $\pi$. Then

$$\mathbb{E}_{\pi,Y}[d(w, \hat{w}(Y))] \geq \mathbb{E}_\mu[d(w, w')(1 - \|P_w - P_{w'}\|_{TV})]$$

where $\| \cdot \|_{TV}$ represents the total-variation distance between distributions and $Y$ the observations.

(see e.g. [Hajek & Raginsky, 2019])

3) Use Pinsker’s inequality

$$\|P_w^k - P_{w'}^k\|_{TV}^2 \leq \frac{1}{2} D_{KL}(P_w^k \| P_{w'}^k)$$

+ exploit decomposition properties of KL-divergence
Ranking from pairwise comparisons

• Motivation and Problem
• Weighted Least-Square Estimator
• Algorithm and Complexity
• Error Analysis
  • Error Bound
  • Lower Bound – Minimax Optimality
  • Other criteria
• Experimental Results
• A Surprising Observation
• Generalizations
• Conclusions
Other performance criteria?

How about $E \| A \Delta \log w \|^2$

Ex: $\Delta \log w_i - \Delta \log w_j = \text{error on } (\log w_i - \log w_j)$

$\sim \text{relative error on } \frac{w_i}{w_j}$

**Direct (naïve) approach:**

$$E \Delta \log w \Delta \log w^T \approx \frac{1}{k} L_v^\dagger$$

$$E \| A \Delta \log w \|^2 = \text{Tr}(A E \Delta \log w \Delta \log w^T A^T) \approx \frac{1}{k} \text{Tr}(AL_v^\dagger A^T)$$

**Problem:** assumption $\sum_i \log w_i = 0$ not necessarily “fair”/relevant
Invariance under addition of constant
⇒ need to analyze distance between equivalence classes
Invariance under addition of constant
$\rightarrow$ need to analyze distance between equivalence classes

$z_1 + z_2 = 0$

Elements used in our analysis
Invariance under addition of constant
$\rightarrow$ need to analyze distance between equivalence classes

$z_1 + z_2 = 0$

Elements used in our analysis

*Not necessarily best* to compute distance $\|Az\|^2$
Other performance Criteria: Summary

- **Quadratic** \( E \| A \Delta \log w \|^2 \)
  - Result and minimax optimality extend
  - Direct approach \( \frac{1}{k} Tr(AL^+_V A^T) \) valid if \( A1 = 0 \)
  - Also simple expression for full rank \( A \).

In particular error on \( (\log w_i - \log w_j) \)

\[
E \| \Delta \log w_i - \Delta \log w_j \|^2 = \frac{1}{k} Tr \left( (e_i - e_j)^T L_V^+ (e_i - e_j) \right) = \Omega_{V,ij}
\]

**Resistance** between \( i \) and \( j \)

- **Nonlinear criteria**: ex: \( \sin(w, \hat{w}) \)
  - Also extends under assumptions
  - Based on \( \| \nabla V \Delta \log w \|^2 \)
Ranking from pairwise comparisons

• Motivation and Problem
• Weighted Least-Square Estimator
• Algorithm and Complexity
• Error Analysis
  • Error Bound
  • Lower Bound – Minimax Optimality
  • Other criteria
• Experimental Results
• A Surprising Observation
• Generalizations
• Conclusions
3D grid
125 nodes
$w_i$ i.i.d. geometric distribution in $[1, 20]$

$\sin(\hat{w}, w)$

$3D$ Grid

<table>
<thead>
<tr>
<th></th>
<th>LS</th>
<th>artif weight</th>
<th>iter weight</th>
<th>emp weight</th>
<th>eigenv</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.35</td>
<td>0.3</td>
<td>0.25</td>
<td>0.2</td>
<td>0.15</td>
</tr>
<tr>
<td>1</td>
<td>0.3</td>
<td>0.25</td>
<td>0.2</td>
<td>0.15</td>
<td>0.1</td>
</tr>
<tr>
<td>2</td>
<td>0.25</td>
<td>0.2</td>
<td>0.15</td>
<td>0.1</td>
<td>0.05</td>
</tr>
</tbody>
</table>

$w$ is i.i.d. geometric distribution in $[1, 20]$. $k$ is the number of samples per edge.
Erdos-Renyi

100 nodes, avg degree 10

$w_i$ i.i.d. geometric distribution in $[1, 20]$

Minimax Rate for Learning From Pairwise Comparisons in the BTL Model

Figure 1. Performance on the 2D grid, 3D grid, and Erdos-Renyi graph. All three plots show $|\sin(\hat{W}, w)|$ on the y-axis vs the number of samples per edge on the x-axis. For the plots, the weights were generated randomly in the interval $[1, 20]$. The 2D and the E-R graph have 100 nodes, while the 3D grid has 125 nodes; the average degree of the E-R graph is 10. Each data point is the average of 50 simulations.

Figure 3. $\hat{W}$ for the eigenvector method in red and the WLSM in blue on the graph of Figure 2. Close to them, as our simulations do not appear to detect any significant difference in performance. Indeed, note that the 3D grid has a very strong divergence between average resistance (constant) and spectral gap ($\approx n^2/3$), and yet our simulation on the 3D grid showed no difference between the eigenvector based method (which has been upper bounded in terms of scaling with the spectral gap) and the WLSM (which we know to scale with resistance).

Moreover, a plausible conjecture is that the methods in question achieve optimal performance not just in distance between the vectors $\hat{W}, w$ but also among $\hat{W}_i w_i$ for each node $i$ (after appropriate normalization). We conjecture this is indeed the case for the WLSM. However, our simulation suggests this may not be the case for the eigenvector method, as we have constructed an example (Figures 2 and 3) where it underperforms in this metric.

4. Conclusions

Our main contribution is the determination of the asymptotic minimax rate for inference from pairwise comparisons. In contrast to previous work, our result is exact up to constant factors.

Besides the conjectures discussed in Section 3, the most natural open question raised by our work is to understand how big the number of samples per edge $k$ has to be for the minimax rate derived in this paper to kick in. We would actually conjecture that $\text{tr}(L^\dagger)/||w||^2$ is, up to constant factors, not only the minimax rate but also the sample complexity of recovering (a scaled version of) $w$.

Acknowledgements

This work was supported by the "Learning from Pairwise Comparisons" incentive grant (MIS) of the F.R.S.-FNRS, and by NSF grants ECCS-1933027, 2007350, 1527618 and 1955981.
### Erdos-Renyi

100 nodes, avg degree 10

\( w_i \) i.i.d. geometric distribution in \([1, 20]\)

#### Minimax Rate for Learning From Pairwise Comparisons in the BTL Model

- **Figure 1.** Performance on the 2D grid, 3D grid, and Erdos-Renyi graph. All three plots show \(|\sin(\hat{W}, w)|\) on the y-axis vs the number of samples per edge on the x-axis. For the plots, the weights were generated randomly in the interval \([1, 20]\). The 2D and the E-R graph have 100 nodes, while the 3D grid has 125 nodes; the average degree of the E-R graph is 10. Each data point is the average of 50 simulations.

- **Figure 3.** \( \hat{W}^3 \) and \( \hat{W}^5 \) for the eigenvector method in red and the WLSM in blue on the graph of Figure 2.

Only **Marginal improvement**

- Did we miss something?
- Is our algorithm better?
  Or just more amenable to analysis?

#### 4. Conclusions

Our main contribution is the determination of the asymptotic minimax rate for inference from pairwise comparisons. In contrast to previous work, our result is exact up to constant factors.

Besides the conjectures discussed in Section 3, the most natural open question raised by our work is to understand how big the number of samples per edge \( k \) has to be for the minimax rate derived in this paper to kick in. We would actually conjecture that

\[
\text{tr}(L^\dagger) / ||w||^2
\]

is, up to constant factors, not only the minimax rate but also the sample complexity of recovering (a scaled version of) \( w \).

#### Acknowledgements

This work was supported by the "Learning from Pairwise Comparisons" incentive grant (MIS) of the F.R.S.-FNRS, and by NSF grants ECCS-1933027, 2007350, 1527618 and 1955981.
Worst-case ≠ Typical case for a distribution

- Eigenvector method [Negahban 16] does indeed appear to perform better than its bound.
- But, \( \approx \) as weighted least-square method with weights

\[
\left( \frac{1}{w_i} + \frac{1}{w_j} \right)^2 \quad \text{Vs our} \quad \frac{1}{w_i} + 2 + \frac{w_j}{w_i}
\]

Grows with \( \sqrt{w_i w_j} \) \quad \text{Only depends on ratio } w_i/w_j

\(\rightarrow\) Neglects information combing from edges between “small weights”

But effect can be averaged out when weights i.i.d. randomly selected
On a specific graph

Weights selected so that relevant information between small values
Conclusion on simulations

• Outperforms previously existing methods
• Effect marginal on “randomized case”
• Significantly more accurate
  • For local differences
  • When information comes from edges between small $w_i$
Ranking from pairwise comparisons

• Motivation and Problem
• Weighted Least-Square Estimator
• Algorithm and Complexity
• Error Analysis
  • Error Bound
  • Lower Bound – Minimax Optimality
  • Other criteria
• Experimental Results
• A Surprising Observation
• Generalizations
• Conclusions
Impact of variance approximation

Idealized algorithm uses

$$v_{ij} := \frac{w_i}{w_j} + 2 + \frac{w_j}{w_i}$$

Not available $\rightarrow$ approximated by empirical

$$\hat{v}_{ij} := R_{ij} + 2 + R_{ij}^{-1}$$

Theoretical analysis: empirical approx. shown “not to degrade solution too much”

But Experimentally: Empirical variance outperforms real one

![Graph showing comparison of sine error for different number of comparisons]

- Algorithm using empirical approximation
- Algorithm with real variance (only available on synthetic data)
**Implicit “regularization”**

$k=10: \ w_1 = 8, w_2 = 2$

<table>
<thead>
<tr>
<th></th>
<th>Prob.</th>
<th>$\log R_{ij}$</th>
<th>$\hat{v}_{12}$</th>
<th>Weight in least square</th>
</tr>
</thead>
<tbody>
<tr>
<td>8 wins (expected)</td>
<td>30%</td>
<td>$\log \frac{8}{2} \simeq 1.38$</td>
<td>$\frac{8}{2} + 2 + \frac{2}{8} = 6.25$</td>
<td>0.16</td>
</tr>
<tr>
<td>7 wins</td>
<td>20%</td>
<td>$\log \frac{7}{3} \simeq 0.85$</td>
<td>$\frac{7}{3} + 2 + \frac{3}{7} = 4.76$</td>
<td>0.21</td>
</tr>
<tr>
<td>9 wins</td>
<td>26%</td>
<td>$\log \frac{9}{1} \simeq 2.19$</td>
<td>$\frac{9}{1} + 2 + \frac{1}{9} = 11.11$</td>
<td>0.09</td>
</tr>
</tbody>
</table>

Empirical variance appear to “smoothen outs” dangerous outlyers.
Experimental validation

3 node graphs, $W_I = 1, W_J = 3$ → 25 wins expected

Edges towards $W_K$ set artificially at expected value

Impact of # wins + probability

Contribution to error

Figure 5.8: $\epsilon(F_{IJ}) \times P(F_{IJ})$ for $F_{IJ} \in [10, 40]$
Experimental validation
3 node graphs, $W_I = 1, W_J = 3 \rightarrow 25$ wins expected
Edges towards $W_K$ set artificially at expected value

5.4 Probabilistic analysis

Different possible observation effects have been explained in the previous sections. Depending on the realisations on each edge, one method will perform better than the other. So why does the empirical method perform better overall? To answer this question, it is necessary to introduce some probabilistic computations. Both methods can be compared considering the expectation of their error under the same circumstances as in the previous analysis. The expectation of the error can be computed as:

$$E(\sin(F_{IJ})) = \frac{1}{k} \sum_{F_{IJ}} P(F_{IJ}) \sin(F_{IJ})$$

In this expression, $\sin(F_{IJ})$ denotes the error corresponding to the case where item $I$ wins $F_{IJ}$ times. $P(F_{IJ})$ is the probability to observe the situation where item $I$ wins $F_{IJ}$ times out of $k$ comparisons.

Since the outcomes of the comparisons are i.i.d. Bernoulli, this probability can be computed as:

$$P(F_{IJ}) = A(k, F_{IJ})B(p_{IJ})^{F_{IJ}}(1 \neq p_{IJ})^{k-F_{IJ}}$$

Adding the probability of each realisation to the results, figure 5.7 is obtained:

![Figure 5.7: Error comparison and realisation probabilities](image)

The probability curve is as expected. Indeed, the expected realisation is when item $I$ wins $p_{IJ}k = 25$ times. One can see that the probability that $F_{IJ} \in [10; 40]$ is close to zero. Therefore, focus will be set on $F_{IJ} \in [10; 40]$ only.

Computing the expectation of the error for both methods determines which method is expected to work the best. Here are the results:

<table>
<thead>
<tr>
<th>Method</th>
<th>Error expectation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Artificial</td>
<td>$E_{art} = 0.0322$</td>
</tr>
<tr>
<td>Empirical</td>
<td>$E_{emp} = 0.0317$</td>
</tr>
</tbody>
</table>

The empirical method yields a lower expectation, thus outperforming the artificial method. However, this is not sufficient to explain why this method works better overall. It has been shown in the previous sections that overshoot and undershoot scenarios yield different behaviour of the methods. However, the probability for those situations to occur has not yet been taken into account.

To consider this probability, one can compute $\sin(F_{IJ})P(F_{IJ})$, as shown on figure 5.8:

![Figure 5.8: $\sin(F_{IJ})P(F_{IJ})$ for $F_{IJ} \in [10, 40] $](image)

The integral of this curve over every realisation is equal to the expectation of the error. One can see a clear distinction between overshoot and undershoot cases.

### Impact of # wins + probability

Appears to confirm implicit regularization idea
But: result of “favorable” trade-off between opposite (important) effects

### Open question
- Rigorous understanding
- Further exploitation of idea or phenomenon


Figure 5.8: $\epsilon(F_{IJ})P(F_{IJ})$ for $F_{IJ} \in [10, 40]$
Ranking from pairwise comparisons

• Motivation and Problem
• Weighted Least-Square Estimator
• Algorithm and Complexity
• Error Analysis
  • Error Bound
  • Lower Bound – Minimax Optimality
  • Other criteria
• Experimental Results
• A Surprising Observation
• Generalizations
• Conclusions
Relaxing Assumptions

• Same number $k$ of comparisons on every edge
  - Can be relaxed,
  - Some technical aspects
  - Ratio min/max # comparison for some results

• i.i.d. comparisons
  - Bounded dependence between comparison (most likely) OK
  - Persistent dependence between edges $\rightarrow$ adapting variance
Extending the notion of comparison

- Pick best out of three
- Rank three
- Comparison with ties...

- Many extensions possible (only approximative analysis so far) but depends on model specifics

  Branders, M., Vekemans, A., & Hendrickx, J. Recovering weights from comparison results in extensions of BTL model

- Multi-comparisons: sometimes non-diagonal Variance Matrix (expression of least square in terms of non-independent events)

- Game: find relation of the type

\[ w_i^{q_i} w_j^{q_j} w_k^{q_k} \approx \text{some function of the outcome (for large k)} \]
Other models - criteria

Bradley-Terry-Luce

\[ p_{ij} = \frac{w_i}{w_i + w_j} \]

- Results extend to large class of ordinal models:
  \[ p_{ij} = f(\phi(\beta_i) - \phi(\beta_j)) \]

- Technical assumption needed (e.g. \( f \) log-concave)
- Not 100% clear yet which ones are actually necessary

- Extension to (asymptotically) any continuous quality criterion

BTL:
- \( \phi = \log \)
- \( f(z) = \frac{1}{1+e^z} \)
Conclusions

- Quality of items recovered from results of comparisons on network → ranking
- Near-linear time algorithm.
- Linear least-square, \textit{coefficients nonlinear} in data.
- No hyperparameters, tuning etc.
- Outperforms past methods, Minimax optimal
- Performances Driven by $L_V^{\dagger}$ and \textit{Resistance of comparison graph}
- Many possible generalizations
- Implicit regularization, not fully understood
Some further research directions

- Online version
  - Comparison arriving one by one
  - Choosing Comparison based on past data
  - Explore and Exploit

- Regime of small # comparisons (large n)
- Prior Incorporation?
- Exploitation of implicit regularization
Thank you for your attention

Alex Olshevsky (BU), Venkatesh Saligrama (BU)

Maxime Winand

Marine Branders

Astrid Vekemans

Balint Daroczy

+ Open position to be filled ASAP

julien.hendrickx@uclouvain.be
References

• Hendrickx, J., Olshevsky, A., & Saligrama, V. *Minimax rate for learning from pairwise comparisons in the BTL model*. ICML 2020

• Hendrickx, J., Olshevsky, A., & Saligrama, V. *Graph resistance and learning from pairwise comparisons*. ICML 2019

• Daroczy B., Hendrickx, J., Olshevsky, A., & Saligrama, V. *Minimax rate for learning ordinal models from pairwise comparisons*, coming soon

• Branders, M., Vekemans, A., & Hendrickx, J. *Recovering weights from comparison results in extensions of BTL model*, Ms Thesis EPL UCLouvain 2022

• Winand, M., & Hendrickx, J. *Learning from pairwise comparisons: an empirical analysis*. MS Thesis EPL UCLouvain 2021