Byzantine Spectral Ranking

Arnhav Datar, Arun Rajkumar, and John Augustine

Indian Institute of Technology, Madras

Objective: Handling malicious pairwise votes while ranking objects

- Rank aggregation from pairwise comparisons is a fundamental task in a wide spectrum of learning and social contexts such as social choice, web search, and recommendation systems.

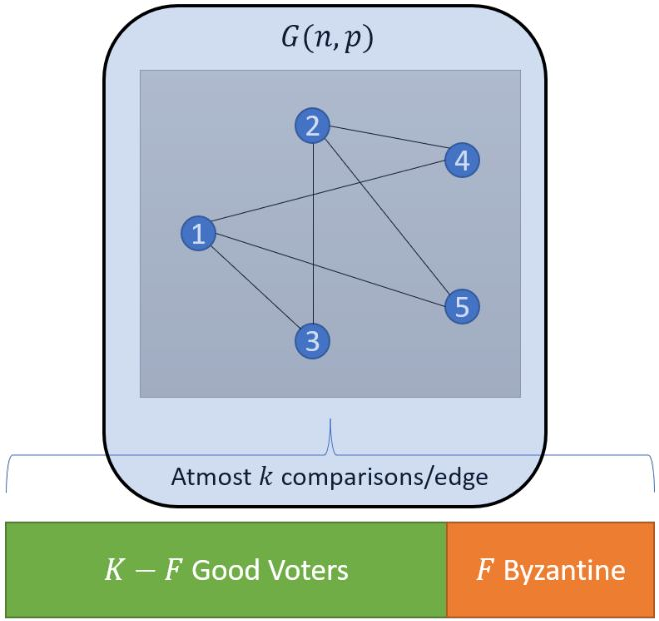
Objective: Handling malicious pairwise votes while ranking objects

- Rank aggregation from pairwise comparisons is a fundamental task in a wide spectrum of learning and social contexts such as social choice, web search, and recommendation systems.
- There has been a lot of work on ranking with the **BTL Model**.
  - n objects that are to be compared and each has a positive weight ($\tilde{\pi}_i$).
  - When a voter is asked a query for a pair, the voter claims i is better than j with probability $\tilde{\pi}_i/(\tilde{\pi}_i + \tilde{\pi}_j)$ independently at random.

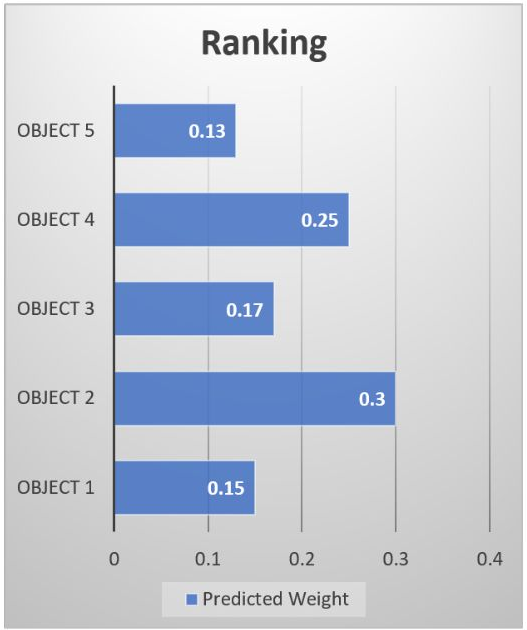
Objective: Handling malicious pairwise votes while ranking objects

- Rank aggregation from pairwise comparisons is a fundamental task in a wide spectrum of learning and social contexts such as social choice, web search, and recommendation systems.
- There has been a lot of work on ranking with the **BTL Model**.
  - n objects that are to be compared and each has a positive weight ($\tilde{\pi}_i$).
  - When a voter is asked a query for a pair, the voter claims i is better than j with probability $\tilde{\pi}_i/(\tilde{\pi}_i + \tilde{\pi}_j)$ independently at random.
- Generally, in crowd-sourced settings, voters differ significantly from each other. Some of the voters may be spammers or even direct competitors of the entity ranking the objects, leading to data poisoning.

Byzantine Voters in the BTL Model

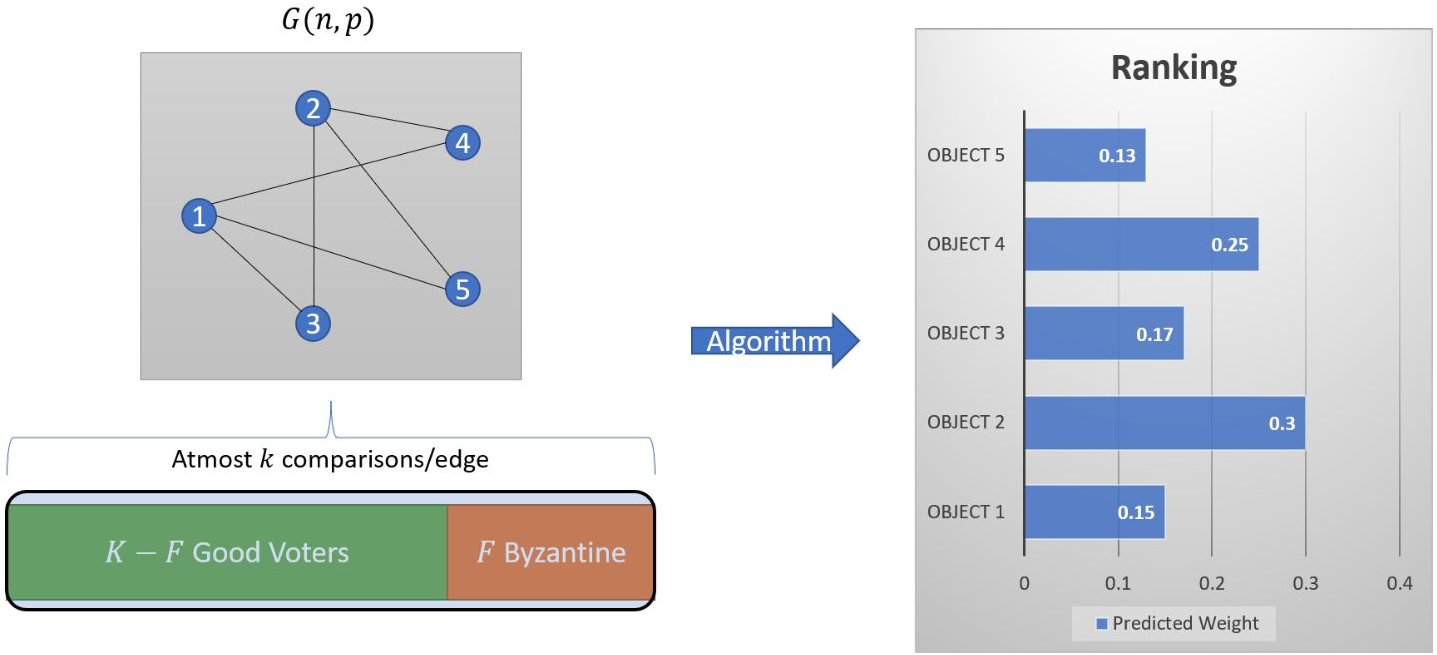


Algorithm



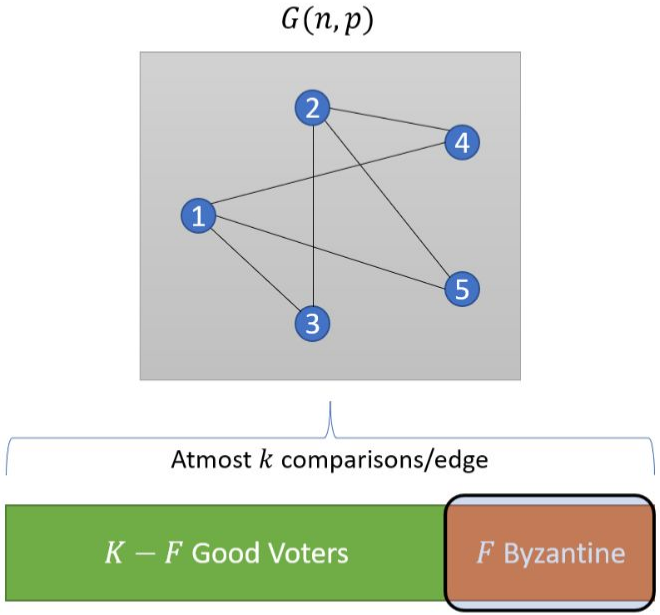
We assume that the pairs to be compared are determined by an Erdős-Rényi comparison graph $G(n,p)=(V,E)$ with atmost k comparisons per pair.

Byzantine Voters in the BTL Model

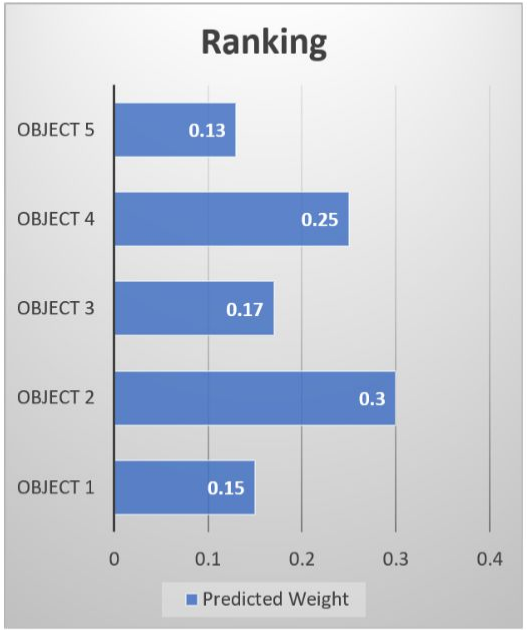


We consider a split in the voter population into $K - F$ good and F Byzantine voters. The algorithm is needed to fix the *pair to voter* mapping before the votes are collected.

Byzantine Voters in the BTL Model



Algorithm



The Byzantine voters can vote however they wish. We assume that the central adversary knows the BTL weights, the good voter's votes, the algorithm, and $G(n,p)$.

Negative Result: The Failure of Rank-Centrality

We show that Rank-Centrality will get a constant relative L_2 error with high probability.

We initially try to motivate the need for a robust ranking algorithm by showing that even a simple strategy from a Byzantine adversary will lead to an unsuitable ranking.

Theorem (Informal)

There exists a strategy that the Byzantine voters can use such that the Rank-Centrality algorithm outputs a distribution π such that with high probability

$$\frac{\|\pi - \tilde{\pi}\|}{\|\tilde{\pi}\|} \in \Omega\left(\frac{F}{K}\right)$$

Remark. A $\Theta(1)$-error algorithm is asymptotically equivalent to an algorithm that just outputs the same ranking regardless or the query responses.

Positive Result: FBSR Algorithm converges in a Byzantine Minority

The FBSR Algorithm achieves a fairly good ranking when $F < K/2$ in $\mathcal{O}(n^2)$ time.

- The FBSR algorithm relies on asking multiple voters multiple queries and based on the collective response decides whether a voter is acting suspiciously or whether it is plausible that the voter might be a good voter.
- Based on the votes, we eliminate voters who are likely to be Byzantine. The FBSR algorithm ensures that a constant fraction of good voters remain and that if any Byzantine voters remain then these voters have votes similar to good voters.

FBSR Convergence when good voters have a majority:

$$\frac{\|\pi - \tilde{\pi}\|}{\|\tilde{\pi}\|} \in \mathcal{O}\left(\max\left(\frac{\log n}{k}, \sqrt{\frac{\log\log n}{\log n}}\right)\right)$$

Negative Result: Impossibility when Byzantine Majority

There is no algorithm that can generate a satisfactory ranking when $F \geq K/2$

Theorem

If $F \geq K/2$, then no algorithm can for all weights $(\tilde{\pi})$, output weights (π^*) such that

$$\frac{\|\pi^* - \tilde{\pi}\|}{\|\tilde{\pi}\|} \leq f(n)$$

with probability $> 1/2$, where $f(n)$ is a function that converges to 0 as n goes to ∞.

Remark. This impossibility consequently shows us that the **FBSR algorithm is optimal** in terms of tolerance towards the Byzantine fraction (F/K).

Thank You