Task 3: Ranking Based on Pairwise Preferences

In this assignment, you will produce a ranking of movies that reflects your preferences, based on a set of pairwise comparisons using methods from the *Crowdsourcing for Language Resources and Evaluation* lecture.

Dataset

1. Select $10 \leq n \leq 15$ movies you've watched from https://www.imdb.com/chart/top/ . **Make sure you keep the order of items**; you will compare your results with them.
2. Generate $n\,log(n)$ (35-60) random pairs of movies.
3. Annotate each generated pair with your preference vote: the left object is more preferred, or the right object is more preferred.

Tasks and Points

1. Collect and annotate movie data (10)
2. Fit the Bradley–Terry model to movie data and report the results (30)
3. Calculate Spearman's ρ between the obtained ranking and IMDb ranking (20)
4. Evaluate the results: does the ranking reflect your preferences? (10)
5. Compute bootstrap confidence intervals on your scores (30)

Remark on Confidence Intervals

Perform the following procedure $k = 1000$ times:

1. Sample the entire pair dataset with replacements (sample size = dataset size).
2. Fit the Bradley–Terry model on this sample and remember each item's score.

This results in a dataset in which for every item we have k score estimates. After that, pick the 2.5% and 97.5% percentiles of each item's bootstrap score distribution, and report them. Feel free to adapt the bootstrap confidence intervals computation approach in the references.

Note: Instead of movies feel free to use another set of items you deeply care about. It might be books, food, travel locations, celebrities, video games, or anything else where the items can be compared to each other. In this case, there will be no 'external' rating to compare to, scores for Spearman's ρ will be redistributed to sum up to 100.

References

- Bradley-Terry Model: https://en.wikipedia.org/wiki/Bradley%E2%80%93Terry_model
- Crowd-Kit Implementation: crowdkit.aggregation.pairwise.bradley_terry.BradleyTerry
- Crowd-Kit Example: Readability-Pairwise.ipynb
- Spearman's ρ in SciPy: scipy.stats.spearmanr
- Bootstrap Confidence Intervals: Compute Bootstrap Confidence Intervals for Elo Scores
- Percentiles: numpy.percentile