Voting: A Social Ranking Application

(Based on Shoham and Leyton-Brown (2008). *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*, Cambridge.)

Leen-Kiat Soh

Introduction

- Consider a setting in which the set of agents is *the same* as the set of outcomes: **agents are asked to vote to express their opinions about each other, with the goal of determining a social ranking**
  - A special social choice problem that has a computational flavor
- Such settings have great practical importance. E.g.,
  - **Search engines rank Web pages by considering hyperlinks** from one page to another to be votes about the importance of the destination pages
  - **Online auction sites** employ ***reputation*** *systems* to provide assessments of agents' trustworthiness based on ratings from past transactions

Introduction | The Ranking Systems Setting

- $N = O$: the set of agents is the same as the set of outcomes
- Agents' preferences are such that each agent divides the other agents into a set that it likes *equally*, and a set that it dislikes *equally* (or, equivalently, has no opinion about)
- Formally, for each $i \in N$ the outcome set O (equivalent to N) is partitioned into two sets $O_{i,1}$ and $O_{i,2}$, with $\forall o_1 \in O_{i,1}, \forall o_2 \in O_{i,2}, o_1 \succ_i o_2$, and with $\forall o, o' \in O_{i,k}$ for $k \in \{1, 2\}, o \sim_i o'$
- We call this the ***ranking systems setting***, and call **a social welfare function in this setting a *ranking rule***

Properties

If we receive the same number of votes, but one of my voters receives more votes than your voters (if we can somehow arrange both sets of voters in some order), then I rank higher than you

- **Definition 9.5.2 Strong Transitivity.** *Consider a preference profile in which outcome o_2 receives at least as many votes as o_1, and it is possible to pair up all the voters for o_1 with voters for o_2 so that each voter for o_2 is weakly preferred by the ranking rule to the corresponding voter for o_1. Further assume that o_2 receives more votes than o_1 and/or that there is at least one pair of voters where the ranking rule strictly prefers the voter for o_2 to the voter for o_1. Then* ***the ranking rule satisfies strong transitivity if it always strictly prefers o_2 to o_1.***

Outcome2 receives at least as many votes as Outcome1

Each voter for Outcome2 is weakly preferred to each voter for Outcome1

Voter2 receives more vote than Voter1

At least for one pair of voters: ranking rule strictly prefers voter for outcome2 to the voter for outcome1

Strong transitivity if Outcome2 is strictly preferred to Outcome1

Why is it Strong transitivity?

It does not take into account the *number* of votes that a voting agent places

Consider an example in which Alibaba votes for almost all the candidates, whereas Sesame votes only for one. If Alibaba and Sesame are ranked the same by the ranking rule, **strong transitivity requires that their votes must count equally**. However, we might feel that Sesame has been *more* decisive, and therefore feel that Sesame’s vote should be counted *more* strongly than Alibaba’s.

Properties 2

- **Definition 9.5.3 Weak Transitivity.** *Consider a preference profile in which outcome o_2 receives at least as many votes as o_1, and it is possible to pair up all the voters for o_1 with voters for o_2* ***who have both voted for exactly the same number of outcomes*** *so that each voter for o_2 is weakly preferred by the ranking rule to the corresponding voter for o_1. Further assume that o_2 receives more votes than o_1 and/or that there is at least one pair of voters where the ranking rule strictly prefers the voter for o_2 to the voter for o_1. Then the ranking rule satisfies* weak transitivity *if it always strictly prefers o_2 to o_1.*
    - Same as Strong Transitivity except it is restricted to apply ***only*** to settings in which the voters vouch for exactly the ***same*** number of candidates

Properties 3

If we receive the same number of votes, but **everyone** of my voters receives more votes than **everyone** of your voters (if we can somehow arrange both sets of voters in some order), then I rank higher than you

- **Definition 9.5.7 Strong Quasi-transitivity.** *Consider a preference profile in which outcome o_2 receives at least as many votes as o_1, and it is possible to pair up all the voters for o_1 with voters from o_2 so that each voter for o_2 is weakly preferred by the ranking rule to the corresponding voter for o_1. Then the ranking rule satisfies* strong quasi-transitivity *if it weakly prefers o_2 to o_1, and strictly* strong *prefers o_2 to o_1 if either o_1 received no votes or* ***each*** *paired voter for o_2 is strictly preferred by the ranking rule to the corresponding voter for o_1.*
  - A different weakening of strong transitivity which does ***not*** care about the number of outcomes that agents vote for
  - but instead requires strict preference only when the ranking rule strictly prefers ***every*** paired voter for o_2 over the corresponding voter for o_1

Properties 4

- **Definition 9.5.4 RIIA, informal.** *A ranking rule satisfies* ***ranked independence of irrelevant alternatives*** *(RIIA) if the relative rank between pairs of outcomes is always determined according to the same rule, and this rule depends* ***only*** *on (1)* ***the number of votes each outcome received****; and (2)* ***the relative ranks of these voters****.*

Is the above fair?

A Ranking Algorithm

forall $i \in N$ **do** $rank(i) \leftarrow 0$
repeat
  forall $i \in N$ **do**
    if $|voters\_for(i)| > 0$ **then**
      $rank(i) \leftarrow \frac{1}{n+1}[|voters\_for(i)| + \max_{j \in voters\_for(i)} rank(j)]$
    else
      $rank(i) \leftarrow 0$
until *rank converges*

1. The number of votes that i receives

2. The ranking of the highest-ranked voter for i

A practical, iterative algorithm to compute ranking in the Ranking Systems setting that satisfies RIIA and Strong Quasi-Transitivity; but, is it fair?

Can the 2nd component be modified? How?

Connection to MAS?

The Ranking Algorithm can be used by a MAS to compute the social choice (and social welfare)

Multisensor data fusion: each sensor (as an agent) has its own area to monitor; and the areas may overlap. How to resolve conflicts in sensory data? How to decide on detection? How to track an object moving through the areas? How to identify the type of detected objects?

Recommendation system: how to aggregate ratings from customers?

Silly Question: In a ranking system for chess players, a player A absorbs a percentage of a player B's ranking points if A beats B, while B loses a percentage of its points if B has more points than A before the match. What are some potential issues with this ranking system?