Voting

COMP6203: Intelligent Agents
December 2020

Dr Baharak Rastegari
b.rastegari@soton.ac.uk
Electronics and Computer Science
University of Southampton

Making Group Decisions

- We are now going to study a class of protocols for making *group decisions*.
- This is the domain of social choice theory, or, more informally, *voting theory*
- Suppose that there is agreement between agents that some choice needs to be made
- A single outcome needs to be selected which affects all agents (i.e. it is a *social* outcome)

Making Group Decisions

- Examples:
  - When and where to meet?
  - Which service provider should we select, given we will share the output?
  - Where to build a bridge/library/etc
  - Who should execute a task?
  - etc.
- Agents have differing preferences over outcomes
- We want means to *combine preferences* to derive a *social outcome*
- The main mechanism to achieve this is through *voting*

Components of a Social Choice Model

- Assume a set, $Ag = \{1, \ldots, n\}$, of *agents* (or *voters*)
- These are the entities that will be expressing preferences
- Voters make group decisions with respect to a set

$$\Omega = \{\omega_1, \omega_2, \ldots\}$$

  of *outcomes* (or *candidates*)
  - If we have $|\Omega| = 2$, then we have a pairwise election
  - If we have $|\Omega| > 2$, then we have a general voting scenario
- Each voter has preferences over Ω

Components of a Social Choice Model

- $\Pi(\Omega)$ is the set of all preference orderings over Ω
- Let $\succ_i \in \Pi(\Omega)$ be a preference ordering for agent i
- $\omega \succ_i \omega'$ represents that outcome ω is ranked above outcome ω' in agent i's preference order $\succ_i$
- Given a set of agents Ag, we denote by $[\succ]$ any preference ordering profile, i.e,

$$[\succ] \in \underbrace{\Pi(\Omega) \times \cdots \times \Pi(\Omega)}_{n}$$

The Social Choice Problem

The Social Choice Problem

Agents may have differing preference orderings. Given this, how do we combine these preference orderings in a principled manner to derive a group decision?

- A *social welfare function*, f takes n voters' preferences and produces a *social preference order*

$$f : \underbrace{\Pi(\Omega) \times \cdots \times \Pi(\Omega)}_{n} \to \Pi(\Omega)$$

- A simpler problem is to obtain just one of the possible outcomes — this is a *social choice function*

$$f : \underbrace{\Pi(\Omega) \times \cdots \times \Pi(\Omega)}_{n} \to \Omega$$

The Social Choice Problem

- We use $\succ^\star$ to refer to the outcome of a social welfare function
- Given a preference ordering profile $[\succ]$ and social welfare function f, $f([\succ]) = \succ^\star$ is called the aggregated ranking for $[\succ]$.
- That is, $\omega \succ^\star \omega'$ means that ω is ranked above ω' in the social outcome
- Now, let's look at *voting procedures*
    - These are mechanisms for agents to use
    - They were designed for various purposes
    - They have certain guarantees and properties
- In discussing these mechanisms, let's use an example...

The Academic Party

- Fifteen academics are trying to decide whether to buy Beer, Wine or Milk for a party
- They decide to vote, but how should the vote be organised?
- The table represents their ranking (i.e. a preference ordering profile), which translates to the preference order:
  - for academic 1, Beer$\succ_1$Wine$\succ_1$Milk
  - for academic 13, Milk$\succ_{13}$Beer$\succ_{13}$Wine

Voter	**Beer**	**Wine**	**Milk**
1	1	2	3
2	1	2	3
3	1	2	3
4	1	2	3
5	1	2	3
6	1	2	3
7	3	2	1
8	3	2	1
9	3	2	1
10	3	2	1
11	3	2	1
12	3	2	1
13	2	3	1
14	2	1	3
15	2	1	3

Plurality Vote

- Each candidate gets one point for every preference order that ranks them first.
- Plurality looks to rank candidates based on number of times they are the preferred one.
- Winner is the one with largest number of points, so aggregated ranking is

$$\text{Milk} \succ^{\star} \text{Beer} \succ^{\star} \text{Wine}$$

  i.e.
  - Beer gets 6 votes
  - Wine gets 2 votes
  - Milk gets 7 votes
- ...but that's the least preferred outcome for a majority!

Voter	Beer	Wine	Milk
1	1	2	3
2	1	2	3
3	1	2	3
4	1	2	3
5	1	2	3
6	1	2	3
7	3	2	1
8	3	2	1
9	3	2	1
10	3	2	1
11	3	2	1
12	3	2	1
13	2	3	1
14	2	1	3
15	2	1	3

The Condorcet Paradox

- There are some fundamental problems with voting procedures.
- Consider these preferences for three voters:

$$\begin{array}{rl} \text{Alice:} & \omega_1 \succ_A \omega_2 \succ_A \omega_3 \\ \text{Bob:} & \omega_2 \succ_B \omega_3 \succ_B \omega_1 \\ \text{Clare:} & \omega_3 \succ_C \omega_1 \succ_C \omega_2 \end{array}$$

- With plurality voting we have *no winner*.
- Whatever option is chosen, $\frac{2}{3}$ (a majority) of the electorate will prefer another option!
- Condorcet's paradox tells us that there are scenarios in which no matter which outcome we choose, a majority of voters will be unhappy.

The Condorcet Criterion

- A Condorcet winner is the candidate who always wins in pair-wise elections using plurality.
- A Condorcet winner does not always exist.
- A voting system satisfies the Condorcet criterion, if it always chooses a Condorcet winner when one exists.
- Rules satisfying this property are called Condorcet methods and are said to be Condorcet consistent.

Example

- Is there a Condorcet winner?
- Beer vs Wine: Beer=7, Wine=8
- Wine vs Milk: Wine=8, Milk=7
- Beer vs Milk: Beer=8, Milk=7
- Wine always wins in pairwise contests, so is a Condorcet winner.

Voter	**Beer**	**Wine**	**Milk**
1	1	2	3
2	1	2	3
3	1	2	3
4	1	2	3
5	1	2	3
6	1	2	3
7	3	2	1
8	3	2	1
9	3	2	1
10	3	2	1
11	3	2	1
12	3	2	1
13	2	3	1
14	2	1	3
15	2	1	3

Copeland method

- Each candidate is scored based on its pairwise victories minus its pairwise losses.
- Candidates are ranked according to the score (highest score gets top position).
- Easy to see that method is Condorcet consistent.
- Wine wins twice, Beer has 1 win and 1 loss, Milk always looses.
- Aggregated raking.

$$\text{Wine} \succ^\star \text{Beer} \succ^\star \text{Milk}$$

Voter	Beer	Wine	Milk
1	1	2	3
2	1	2	3
3	1	2	3
4	1	2	3
5	1	2	3
6	1	2	3
7	3	2	1
8	3	2	1
9	3	2	1
10	3	2	1
11	3	2	1
12	3	2	1
13	2	3	1
14	2	1	3
15	2	1	3

Copeland method

- The Copeland method works also when there is no Condorcet winner.
- Election with 5 candidates ranked by 100 people as follows:

$$31 : A > E > C > D > B \quad | \quad 30 : B > A > E$$

$$29 : C > D > B \quad | \quad 10 : D > A > E$$

- Pairwise comparisons:

Comparison	Result	Winner	Comparison	Result	Winner
A vs B	41/59	B	B vs D	30/70	D
A vs C	71/29	A	B vs E	59/41	B
A vs D	61/39	A	C vs D	60/10	C
A vs E	71/0	A	C vs E	29/71	E
B vs C	30/60	C	D vs E	39/61	E

- There is no Condorcet winner, but A is a Copeland winner.

Borda Count

- Some voting procedures only consider top-ranked candidates.
- The Borda count takes into account all the information from a preference order.
- This method proceeds as follows:
  - With x candidates, each voter awards x points to their first choice, $x - 1$ to their second, and so on.
  - The candidate with the most points wins.
- Beer gets 30 points, Wine gets 31 points, Milk gets 29 points.
- Aggregated ranking

$$\text{Wine} \succ^{\star} \text{Beer} \succ^{\star} \text{Milk}$$

Voter	Beer	Wine	Milk
1	1	2	3
2	1	2	3
3	1	2	3
4	1	2	3
5	1	2	3
6	1	2	3
7	3	2	1
8	3	2	1
9	3	2	1
10	3	2	1
11	3	2	1
12	3	2	1
13	2	3	1
14	2	1	3
15	2	1	3

Borda Count

- The Borda count is not a Condorcet method.
- Consider the following example:

Voter	**A**	**B**	**C**
1	1	2	3
2	1	2	3
3	1	2	3
4	3	1	2
5	3	1	2

- **A** is the Condorcet winner...
- ...but, under Borda count...
  - **A** gets 11 points
  - **B** gets 12 points,
  - **C** gets 7 points.
- So **B** is the Borda winner

Desirable Properties of Voting Procedures

- We have focused on specific examples of voting procedures.
- Some of them have less than ideal properties.
- Are there any *good* voting procedures?
- In order to answer this question, we need to specify what we mean by "good".
- This means to define the properties that a good social welfare/choice function would satisfy.

Social Welfare Functions: Desirable Properties

Pareto Efficiency

A social welfare function is Pareto efficient, if, whenever every agent i prefers ω over ω', then then $\omega \succ^* \omega'$.

Independence of Irrelevant Alternatives (IIA)

A social welfare function is independent of irrelevant alternatives, if whether ω is ranked above ω' in the social outcome depends only on the relative orderings of ω and ω' in agents' preferences.

Nondictatorship

A social welfare function is nondictatorial whenever there is no voter i such that for all ω, ω', if $\omega \succ_i \omega'$ then $\omega \succ^\star \omega'$.

IIA Example

For example if, if an agent i has preferences:

$$A \succ_i B \succ_i C \succ_i D$$

and the aggregated ranking is $C \succ^\star A$, this aggregate ranking should not change if the agent's preferences changes to e.g.:

$$A \succ_i \mathbf{B} \succ_i \mathbf{D} \succ_i C$$

In other words, the order of B and D should not affect the order between A and C.

Arrow's Theorem (Social Welfare Functions)

- We say that a social welfare function is dictatorial if it does not satisfy nondictatorship.

Arrow's Theorem

For elections with more than 2 candidates, any social welfare function satisfying Pareto efficiency and IIA is *dictatorial*.

- A *negative* result: there are fundamental limits to democratic decision making.
- Arrow's theorem tells us that we cannot hope to find a voting scheme that satisfies all of the notions of fairness that we find desirable.

Social Choice Functions: Desirable Properties

Social welfare functions require us to find an ordering: do we gain anything by using social choice functions in order to find a single outcome?

Weak Pareto Efficiency

A social choice function is weakly Pareto efficient, if, when every agent i prefers ω over ω', then ω' cannot be the outcome of the social choice function.

Monotonicity

A social choice function f is monotonic if, for every preference profile $[\succ]$, such that $f([\succ]) = \omega$, if $[\succ']$ is another profile such that $\omega \succ'_i \omega'$ whenever $\omega \succ_i \omega'$ for every agent and every alternative ω', then $f([\succ']) = \omega$ as well.

Nondictatorship

A social choice function f is nondictatorial if there does not exist an agent i such that f always selects the top choice in i's preference ordering.

Muller-Satterthwaite's Theorem

- We say that a social choise function is dictatorial if it does not satisfy nondictatorship.

Muller-Satterthwaite's Theorem

For elections with more than 2 candidates, any social choice function satisfying weak Pareto efficiency and monotonicity is *dictatorial*.

Strategic Voters (Formally)

- What happens if voters behave strategically, for instance, when they vote tactically?
- What happens when voters misrepresent their true preferences?
- A social choice function, f, is manipulable if, for some preference ordering profile

$$\succ_1, \ldots, \succ_i, \ldots, \succ_n$$

and voter i, there exists some $\succ_i'$ such that

$$f(\succ_1, \ldots, \succ_i', \ldots, \succ_n) \succ_i f(\succ_1, \ldots, \succ_i, \ldots, \succ_n)$$

  - A voter can obtain a better outcome for themselves by unilaterally changing their preference profiles.
  - That is, by misreporting their preferences to the voting procedure.

Strategic Voting

Can we engineer voting procedures that are not manipulable?

Gibbard-Satterthwaitte's Theorem

Any social choice function with at least three outcomes that satisfies *citizen sovereignty* and is truthful (i.e non-manipulable)is dictatorial.

Citizen sovereignty: for every outcome ω, there exists a preference profile $[\succ]$ such that the social choice function returns ω.

Although voting procedures are manipulable, their manipulation is computationally complex:

- It is not easy to manipulate some voting procedures intelligently.
- Various unknowns (other voters' preferences, if they also are trying to manipulate the voting, etc.).
- Computing a "lie" can be costly (depending on the voting protocol used).

Democracy still has hope...

Further Reading

Shoham, Yoav, and Kevin Leyton-Brown. *Multiagent systems: Algorithmic, game-theoretic, and logical foundations*. Cambridge University Press, 2008, Chapter 9.
http://www.masfoundations.org/mas.pdf

Wooldridge, Michael. *An Introduction to MultiAgent Systems*. Wiley, 2009, Chapter 12.

Acknowledgement

The slides are by Enrico Marchioni, who largely benefited from the slides by Enrico Gerding, who in turn used the majority of the slides from those created by Tim Norman.