Social Choice
(Preference Aggregation)

January 26, 2006

Social choice theory

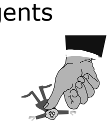
- Study of decision problems in which a group has to make the decision
- The decision affects all members of the group
  - Their opinions should count!
- Applications:
  - Political elections
  - Other elections
  - Note that outcomes can be vectors
    - Allocation of money among agents, allocation of goods, tasks, resources…

Social choice theory

- CS applications:
  - Multiagent planning [Ephrati&Rosenschein]
  - Computerized elections [Cranor&Cytron]
    - Note: this is not the same as electronic voting
  - Accepting a joint project, rating Web articles [Avery,Resnick&Zeckhauser]
  - Rating CDs…

Assumptions

1. Agents have preferences over alternatives
   - Agents can rank order the outcomes
     - a>b>c=d is read as "a is preferred to b which is preferred to c which is equivalent to d"
2. Voters are sincere
   - They truthfully tell the center their preferences
3. Outcome is enforced on all agents



Formal model

- Set of agents N={1,2,…,n}
- Set of outcomes O
- Set of strict total orders on O, L

- **Social choice function** $C:L^n \rightarrow O$

- **Social welfare function** $C:L^n \rightarrow L^-$ where L^- is the set of weak total orders on O

The problem

- Majority decision:
  - If more agents prefer a to b, then a should be chosen
- Two outcome setting is easy
  - Choose outcome with more votes!

- What happens if you have 3 or more possible alternatives?

Case 1: Agents specify their top preference

Canadian Election System

- Plurality Voting
  - One name is ticked on a ballot
  - One round of voting
  - One candidate is chosen

Is this a "good" system?

What do we mean by good?

Example: Plurality

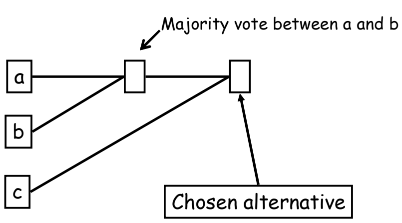
- 3 candidates
  - Lib, NDP, C
- 21 voters with the preferences
  - 10 Lib>NDP>C
  - 6 NDP>C>Lib
  - 5 C>NDP>Lib
- Result: Lib 10, NDP 6, C 5
  - But a majority of voters (11) prefer all other parties more than the Libs!

What can we do?

- Majority system
  - Works well when there are 2 alternatives
  - Not great when there are more than 2 choices
- Proposal:
  - Organize a series of votes between 2 alternatives at a time
  - How this is organized is called an agenda
    - Or a cup (often in sports)

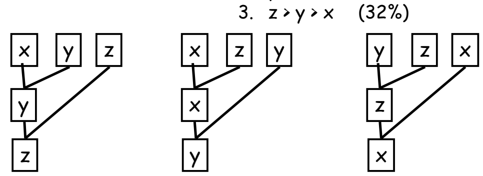
Agendas

- 3 alternatives {a,b,c}
- Agenda a,b,c



Agenda paradox

- *Binary protocol (majority rule) = cup*
- Three types of agents:
  1. x > z > y (35%)
  2. y > x > z (33%)
  3. z > y > x (32%)

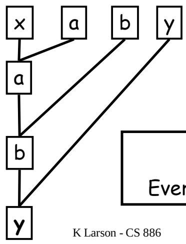


- Power of agenda setter (e.g. chairman)
- Vulnerable to irrelevant alternatives (z)

Another problem: Pareto dominated winner paradox

Agents:

1. x > y > b > a
2. a > x > y > b
3. b > a > x > y



BUT

Everyone prefers x to y!

Case 2: Agents specify their complete preferences

Maybe the problem was with the ballots!

Ballot

X>Y>Z

Now have more information

Condorcet

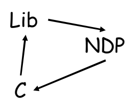
- Proposed the following
  - Compare each pair of alternatives
  - Declare “a” is socially preferred to “b” if more voters strictly prefer a to b
- Condorcet Principle: If one alternative is preferred to <u>all other</u> candidates then it should be selected

Example: Condorcet

- 3 candidates
  - Lib, NDP, C
- 21 voters with the preferences
  - 10 Lib>NDP>C
  - 6 NDP>C>Lib
  - 5 C>NDP>Lib
- Result:
  - NDP win! (11/21 prefer them to Lib, 16/21 prefer them to C)

A Problem

- 3 candidates
  - Lib, NDP, C
- 3 voters with the preferences
  - Lib>NDP>C
  - NDP>C>Lib
  - C>Lib>NDP
- Result:
  - No Condorcet Winner



Borda Count

- Each ballot is a list of ordered alternatives
- On each ballot compute the rank of each alternative
- Rank order alternatives based on decreasing sum of their ranks

A>B>C	A: 4
A>C>B	B: 8
C>A>B	C: 6

Borda Count

- Simple
- Always a Borda Winner
- BUT does not always choose Condorcet winner!
- 3 voters
  - 2: b>a>c>d
  - 1: a>c>d>b

Borda scores:

a:5, b:6, c:8, d:11

Therefore a wins

BUT b is the Condorcet winner

Inverted-order paradox

- Borda rule with 4 alternatives
  - Each agent gives 1 points to best option, 2 to second best...
- Agents:
  1. x > c > b > a
  2. a > x > c > b
  3. b > a > x > c
  4. x > c > b > a
  5. a > x > c > b
  6. b > a > x > c
  7. x > c > b > a
- x=13, a=18, b=19, c=20
- Remove x: c=13, b=14, a=15

Borda rule vulnerable to irrelevant alternatives

- Three types of agents:

  1. x > z > y (35%)
  2. y > x > z (33%)
  3. z > y > x (32%)

- Borda winner is x
- Remove z: Borda winner is y

Desirable properties for a voting protocol

- Universality
  - It should work with any set of preferences
- Transitivity
  - It should produce an ordered list of alternatives
  - That is, we work with social welfare function
- Pareto efficient
  - If all all agents prefer x to y then in the outcome x should be preferred to y
  - SWF W is pareto efficient if for any $o_1, o_2 \in O$, $\forall$ i $o_1 \succ_i o_2$ implies that $o_1 \succ_W o_2$

Desirable properties for a voting protocol

Independence of Irrelevant Alternatives (IIA)
- Comparison of two alternatives depends only on their standings among agents' preferences, not on the ranking of other alternatives
- SWF W is IIA if for any $o_1, o_2 \in O$, and two preference profiles $\succ'$, $\succ''$, $\forall$ i $o_1 \succ_i' o_2 \leftrightarrow o_1 \succ_i'' o_2$ implies that $o_1 \succ_{W(\succ')} o_2 \leftrightarrow o_1 \succ_{W(\succ'')} o_2$

- No dictators
  - SWF W has no dictator if $\neg \exists I \, \forall o_1, o_2 \, (o_1 \succ_i o_2 \Rightarrow o_1 \succ_W o_2)$

Arrow's Theorem (1951)

- If there are 3 or more alternatives and a finite number of agents then there is **no** protocol which satisfies the 5 desired properties

Is there anything that can be done?

- Can we relax the properties?
- No dictator
  - Fundamental for a voting protocol
- Paretian
  - Also seems to be pretty desirable
- Transitivity
  - Maybe you only need to know the top ranked alternative
    - Stronger form of Arrow's theorem says that you are still in trouble
- Independence
- Universality
  - Some hope here (1 dimensional preferences, spacial preferences)

Take-home Message

- Despair?
  - No ideal voting method
  - That would be boring!

- A group is more complex than an individual
- Weigh the pro's and con's of each system and understand the setting they will be used in

- Do not believe anyone who says they have the best voting system out there!

Proof of Arrow's theorem (slide 1 of 3)

- Follows [Mas-Colell, Whinston & Green, 1995]
- Assuming *G* is Paretian and independent of irrelevant alternatives, we show that *G* is dictatorial
- **Def.** Set $S \subseteq A$ is decisive for x over y whenever
  - $x \succ_i y$ for all $i \in S$
  - $x \prec_i y$ for all $i \in A-S$
  - => $x \succ y$
- **Lemma 1.** If S is decisive for x over y, then for any other candidate z, S is decisive for x over z and for z over y
- **Proof.** Let S be decisive for x over y. Consider: $x \succ_i y \succ_i z$ for all $i \in S$ and $y \succ_i z \succ_i x$ for all $i \in A-S$
  - Since S is decisive for x over y, we have $x \succ y$
  - Because $y \succ_i z$ for every agent, by the Pareto principle we have $y \succ z$
  - Then, by transitivity, $x \succ z$
  - By independence of irrelevant alternatives (y), $x \succ z$ whenever every agent in S prefers x to z and every agent not in S prefers z to x. I.e., S is decisive for x over z
- To show that S is decisive for z over y, consider: $z \succ_i x \succ_i y$ for all $i \in S$ and $y \succ_i z \succ_i x$ for all $i \in A-S$
  - Then $x \succ y$ since S is decisive for x over y
  - $z \succ x$ from the Pareto principle and $z \succ y$ from transitivity
  - Thus S is decisive for z over y ②

Proof of Arrow's theorem (slide 2 of 3)

- Given that S is decisive for x over y, we deduced that S is decisive for x over z and z over y.
- Now reapply Lemma 1 with decision z over y as the hypothesis and conclude that
  - S is decisive for z over x
  - which implies (by Lemma 1) that S is decisive for y over x
  - which implies (by Lemma 1) that S is decisive for y over z
  - Thus: **Lemma 2**. If S is decisive for x over y, then for any candidates u and v, S is decisive for u over v (i.e., S is *decisive*)
- **Lemma 3.** For every $S \subseteq A$, either S or A-S is decisive (not both)
- **Proof**. Suppose $x \succ_i y$ for all $i \in S$ and $y \succ_i x$ for all $i \in A-S$ (only such cases need to be addressed, because otherwise the left side of the implication in the definition of decisiveness between candidates does not hold). Because either $x \succ y$ or $y \succ x$, S is decisive or A-S is decisive

Proof of Arrow's theorem (slide 3 of 3)

- **Lemma 4.** If S is decisive and T is decisive, then $S \cap T$ is decisive
- **Proof.**
  - Let $S = \{i: z \succ_i y \succ_i x\} \cup \{i: x \succ_i z \succ_i y\}$
  - Let $T = \{i: y \succ_i x \succ_i z\} \cup \{i: x \succ_i z \succ_i y\}$
  - For $i \notin S \cup T$, let $y \succ_i z \succ_i x$
  - Now, since S is decisive, $z \succ y$
  - Since T is decisive, $x \succ z$
  - Then by transitivity, $x \succ y$
  - So, by independence of irrelevant alternatives (z), $S \cap T$ is decisive for x over y.
    - (Note that if $x \succ_i y$, then $i \in S \cap T$.)
  - Thus, by Lemma 2, $S \cap T$ is decisive
- **Lemma 5.** If $S = S_1 \cup S_2$ (where S_1 and S_2 are disjoint and exhaustive) is decisive, then S_1 is decisive or S_2 is decisive
- **Proof.** Suppose neither S_1 nor S_2 is decisive. Then $\sim S_1$ and $\sim S_2$ are decisive. By Lemma 4, $\sim S_1 \cap \sim S_2 = \sim S$ is decisive. But we assumed S is decisive. Contradiction ②
- **Proof of Arrow's theorem**
  - Clearly the set of all agents is decisive. By Lemma 5 we can keep splitting a decisive set into two subsets, at least one of which is decisive. Keep splitting the decisive set(s) further until only one agent remains in any decisive set. That agent is a dictator. QED

Stronger version of Arrow's theorem

- In Arrow's theorem, social choice functional G outputs a ranking of the outcomes
- The impossibility holds even if only the highest ranked outcome is sought:
- **Thrm**. Let $|O| \geq 3$. **If a** social choice *function* f: R -> outcomes is *monotonic* and *Paretian,* then f is dictatorial
  - f is *monotonic* if [x = f(R) and x *maintains its position* in R'] => f(R') = x
  - x *maintains its position* whenever $x >_i y \Rightarrow x >_i' y$