Coalitions and Group Decisions

Boi Faltings

Laboratoire d'Intelligence Artificielle
boi.faltings@epfl.ch
http://moodle.epfl.ch/

Games with more than 2 players

Games with > 2 players are more complex:

- players can form *coalitions*: groups that cooperate to optimize their utility.
- players need to agree on joint decisions: social choice.

Cooperative Game

- Agents A, B and C represent servers; they can choose to not work (n) or work (w) at cost=5.
- A client is willing to pay 12 for a regression model and 20 for a regression model with causal analysis.
- One server alone cannot meet the deadline (payoff 0), two servers can produce the regression model, three servers can also produce causal analysis but extra revenue goes to agent A for license fees.

		BC			
		nn	nw	wn	ww
A	n	(0,0,0)	(0,0,-5)	(0,-5,0)	(0,1,1)
A	w	(-5,0,0)	(1,0,1)	(1,1,0)	(7,-1,-1)

Highest (combined) payoff: (w,w,w) $\Rightarrow$ 5
But not a Nash equilibrium!

Coalitions without utility transfer

Possible coalitions in this game:

- AB, BC, AC: utility $= 2$ (when third agent is excluded).
- grand coalition ABC: utility $= 5$

Coalitions AB, BC, AC are *stable*: no agent has an incentive to leave the coalition.
Coalition ABC is not stable: agents B and C can get higher payoff by leaving the coalition!

Coalitions with utility transfer

Side contract: in grand coalition, A pays 1.5 each to B and C:

		BC			
		nn	nw	wn	ww
A	n	(0,0,0)	(0,0,-5)	(0,-5,0)	(0,1,1)
	w	(-5,0,0)	(1,0,1)	(1,1,0)	(4,0.5,0.5)

$\Rightarrow$ Grand coalition is a Nash equilibrium.

Coalitional game theory:

- *coalition formation*: which group gets the highest combined revenue?
- *payoff distribution*: how are the rewards distributed?

Stability of coalitions

		BC			
		nn	nw	wn	ww
A	n	(0,0,0)	(0,0,-5)	(0,-5,0)	(0,1,1)
	w	(-5,0,0)	(1,0,1)	(1,1,0)	(4,0.5,0.5)

- B and C are better forming their own coalition: each gets 1 instead of 0.5!
- Definition: a coalition N is stable if no subset $S \subset N$ gives higher utility for all agents in S than they get in N.
- When utility can be redistributed, sufficient that S as a whole gets higher utility than S gets in N.

Stability and the core

- Question: is the grand coalition (all agents) stable?
- Rephrased: for what payoff distributions is the GC stable?
- This set of payoff distributions is called the *core* of the game.

In the example game, the core is given by:

$$\begin{aligned} \mathit{payoff}(A) &\geq 6 \\ \mathit{payoff}(B) &\geq 6 \\ \mathit{payoff}(C) &\geq 6 \end{aligned}$$

However, the core may often be empty!

Determining the core

- Let the *characteristic function* v(S) be the value that can be achieved by a coalition S; N is the coalition of all agents.
- Condition(Bondereva-Shapley): Core is nonempty iff.

$$v(N) \geq \sum_{S \subseteq N} \lambda(S) v(S)$$

for every function λ ($2^{|N|} \rightarrow [0, 1]$) that is balanced:

$$\forall i \in N, \sum_{S : i \in S} \lambda(S) = 1$$

- However, exponentially many $S \Rightarrow$ checking requires exponential time.

Games with nonempty core

- Superadditive game:

$$\forall S, T \subset N, if S \cap T = \phi, v(S \cup T) \geq v(S) + v(T)$$

- Convex game (implies superadditive):

$$\forall S, T \subset N, v(S \cup T) \geq v(S) + v(T) - v(S \cap T)$$

  Example game is convex

- Theorem: all convex games have a nonempty core!
- Stable payoff distribution is given by Shapley value.

Determining the right payoffs

- Shapley value = vector $(\phi_1, \phi_2, ..., \phi_n)$ giving the expected distribution of returns of the game.
- Shapley value should satisfy certain conditions $\Rightarrow$ axioms.
- For convex games, Shapley value should be in the core.

Conditions for a unique Shapley value

A *carrier* of a game is a minimal coalition of agents such that the result of the game is always completely decided by these agents.

1. an agent who is not member of any carrier has value $\phi_i = 0$
2. a permutation of agents gives the same permutation of Shapley values.
3. when the agents play two games I and J in parallel, the Shapley value of the combined game is the sum of the Shapley values for the individual games I and J.

$\Rightarrow$ there is a unique Shapley value!

$\Rightarrow$ for convex games, the Shapley value is in the core!

Computing the Shapley value

- Characteristic function $v(S)$ = combined payoff that coalition S can achieve together.
- Let agents $\{a_1, .., a_n\}$ be ordered and form coalitions in that order:

$$C_1 = \{a_1\}, .., C_k = \{a_1, .., a_k\}, C_n = \{a_1, .., a_n\}$$

- Given this particular ordering, the value of $U(a_{k+1})$ to the coalition C_{k+1} is $v(C_{k+1}) - v(C_k)$.
- The Shapley value of an agent is the *average* value over all possible orderings of agents.

Example (1)

Characteristic function:

AB	BC	AC	ABC
12	12	12	20

Order	U(A)	U(B)	U(C)
ABC	0	12	8
ACB	0	8	12
BAC	12	0	8
BCA	8	0	12
CAB	12	8	0
CBA	8	12	0
average	6 2/3	6 2/3	6 2/3

Example (2)

If A contributes more than the others:

AB	BC	AC	ABC
16	12	16	20

Order	U(A)	U(B)	U(C)
ABC	0	16	4
ACB	0	4	16
BAC	16	0	4
BCA	8	0	12
CAB	16	4	0
CBA	8	12	0
average	8	6	6

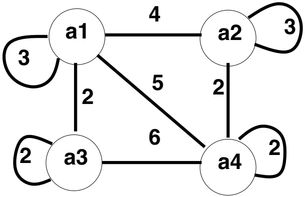
Computing the Shapley value efficiently

Explicitly computing all marginal contributions has exponential complexity. Are there classes of games where computation is efficient?

- weighted graph games: agents contribute to coalitions either individually or in pairs.
- marginal contribution nets: contribution can be in larger groups.
- weighted majority voting: Shapley value complex to compute.

Weighted graph games

Represent rewards of agents and pairs of agents as a graph:



Value of a coalition = sum of edge weights in the subgraph:

$$
\begin{array}{rl}
\{a1\} & value = 3 \\
\{a1, a2\} & value = 3 + 4 + 3 = 10 \\
\{a1, a2, a4\} & value = 3 + 4 + 3 + 2 + 5 + 2 = 19 \\
\{a1, a2, a3, a4\} & value = 29
\end{array}
$$

Shapley value of a weighted graph game

$\Rightarrow$

$$\begin{aligned} Shapleyvalue(a_i) &= w((a_i, a_i)) \\ &+0.5 \sum_{\{e_i | e_i = (a_i, a_j), j \neq i\}} w(e_i) \end{aligned}$$

Example:

$$\begin{aligned} SV(a1) &= 3 + 0.5(4 + 5 + 2) = 8.5 \\ SV(a2) &= 3 + 0.5(4 + 2) = 6 \\ SV(a3) &= 2 + 0.5(2 + 6) = 6 \\ SV(a4) &= 2 + 0.5(5 + 2 + 6) = 8.5 \end{aligned}$$

But not all games can be represented this way!

Marginal Contribution Nets

- Generalization of graphical games: also allow hyperedges.
- Computing the Shapley value: as in graphical games, but divide contributions by size of the edge (can be > 2).
- Generalize edges to conditions that could also exclude agents: can represent any game, but no easy way to compute Shapley value.

Shapley Values in Machine Learning

Payoff distribution by Shapley values is also used for credit assignment in machine learning:

- consider n datasets $\mathcal{D} = \{d_1, d_2, ..., d_n\}$, let $Q(M(D))$ be the quality of the model $M(D)$ learned from $D \subseteq \mathcal{D}$.
- model as coalitional game with joint payoff of coalition $D = Q(M(D))$.
- contribution of dataset $d_i \simeq \phi_i$ (Shapley value of dataset i).
- approximate value by sampling.
- however, stability results do not apply as data can be used in multiple coalitions.

Coalition Structures

- In some cases, agents may have a negative effect on a coalition: consume more resources than they contribute.

$\Rightarrow$ the grand coalition does not achieve the best overall payoff.

$\Rightarrow$ search for optimal division into coalitions.

- Example: separate construction workers into several crews.
- Computationally very hard problem, but good approximate solutions.

Group decision making

- Social choice: group of agents to agree on one of *n* alternative decisions $d_1, .., d_n$.
- decision should reflect joint preferences; all agents carry equal weight.
- preferences are *ordinal*: only order is expressed, no preference strength/risk attitude.
- direct revelation voting protocol: agents express their preferences, *scoring rule* determines the outcome.
- categories: 2 or ≥ 3 choices.

Properties of voting protocols

- Pareto-optimality: if every agent prefers d_i over d_j, d_j cannot be preferred over d_i in the social choice.
- Monotonicity: if an agent raises its preference for the winning alternative, it remains the winner.
- Non-imposition: for each alternative d_i, there is some set of agent preference orders so that it is chosen as the winner (with monotonicity, implies Pareto-optimality).
- Independence of losing alternatives: if the social choice function prefers d_i over d_j, then this order does not change if another alternative d_l is introduced.
- Non-dictatorship: the protocol does not always choose the alternative preferred by the same agent.

Voting with 2 alternatives

- Every agent ranks alternative $d_1 \succ d_2$ or $d_2 \succ d_1$.
- Majority voting: among 2 alternatives, agents vote for the one they prefer.
- Rank $d_1 \succ d_2$ if at least half the agents vote for d_1.
- All votes count the same.

$\Rightarrow$ best agent strategy: vote for the preferred item.

- Satisfies all desirable properties.

Majority voting with ≥ 3 alternatives

Generalize by voting for pairs of alternatives in sequence:

1. order alternatives $d_1, d_2, ..., d_n$.
2. let $x \leftarrow winner(d_1, d_2)$.
3. for $i \leftarrow 3$ to n $x \leftarrow winner(x, d_i)$
4. "surviving" x is the winner.

Vote organizer decides the order of alternatives.

Condorcet winners

- Condorcet winner:

  alternative that beats or ties all others in a pairwise majority vote.

- Depending on the preference structure, a Condorcet winner might not exist.
- Condorcet winner is Pareto-optimal, independent of loosing alternatives, satisfies monotonicity.
- Majority voting always selects the Condorcet winner.

Situation with no Condorcet winner

3 agents A_1, A_2 and A_3 choose between apples, pears and oranges:

$$\begin{array}{ll} A_1: & a \succ p \succ o \\ A_2: & p \succ o \succ a \\ A_3: & o \succ a \succ p \end{array}$$

Thus:

a is preferred over p (A_1, A_3 over A_2)
p is preferred over o (A_1, A_2 over A_3)
o is preferred over a (A_2, A_3 over A_1)

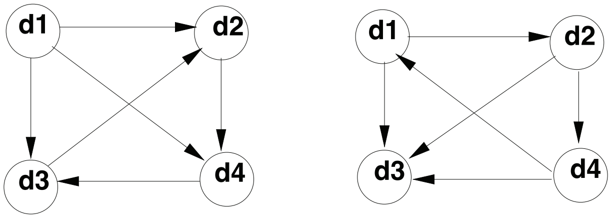
No choice is a Condorcet winner!

Manipulation in majority voting

1. order = a,p,o
   - a vs. p: a wins
   - a vs. o: **o** wins
2. order = o,p,a
   - o vs. p: p wins
   - p vs. a: **a** wins
3. order = o,a,p
   - o vs. a: o wins
   - o vs. p: **p** wins

Vote order determines outcome!

Majority graphs



- nodes = alternatives.
- directed arc from d_i to d_j: majority prefers d_i over d_j.
- Condorcet winner: node with only outgoing edges.
- left: $d1$ is a Condorcet winner (cycle does not matter).
- right: winning cycle of $d1, d2, d4$. $d3$ certainly not winner.

Manipulation of majority voting

- If there is a Condorcet winner, majority voting will select it.
- What if there is a cycle, i.e. no Condorcet winner?

$\Rightarrow$ outcome depends on sequence of votes!

- Winner is the alternative in the winning cycle that is introduced last.

$\Rightarrow$ vote organizer can always determine which of these is chosen!

Other voting protocols

Some examples of voting protocols:

- Plurality voting: every agent votes for one alternative, order alternatives by number of votes.
- Plurality with elimination: proceed in $n-1$ rounds, at each round the least preferred alternative is eliminated and those that voted for it have to vote again for a remaining alternative.
- Approval voting: vote for every acceptable alternative; the one with the most votes wins.
- Borda count: give $n-1$ votes for most preferred, $n-2$ votes for second most preferred, ..., 0 vote for least preferred alternative. Alternative with most votes wins.
- Slater ranking: best approximation to majority graph.

Complexity considerations

- Voting with many alternatives can be a considerable burden: voter has to evaluate all alternatives and rank them!
- Protocols might require many rounds (majority voting) and heavy communication.
- Simpler alternative: only vote for most preferred alternative (plurality voting).

Problems with plurality voting

3 alternatives a,b,c:

499 agents:	$a \succ b \succ c$
3 agents:	$b \succ c \succ a$
498 agents:	$c \succ b \succ a$

b is the Condorcet winner, but:

- plurality would pick a
- plurality with elimination would eliminate b, then pick c (with 501 over 499 votes).

Weighting alternatives

- Plurality voting ignores preferences beyond the best one.

$\Rightarrow$ allow further expression.

- Borda count: give
  - $n-1$ votes to most preferred alternative
  - $n-2$ to second best,
  - ...
  - 0 votes to least preferred alternative.
- Agent could not give votes for alternatives that rank very low.

Problems with Borda count (1)

3 alternatives a,b,c:

$a \succ c \succ b$	$b \succ a \succ c$	$c \succ b \succ a$
35 agents	33 agents	32 agents

Protocol	a	b	c
Borda	103	98	99
Plurality	35	33	32

without alternative c:

Protocol	a	b
Borda	35	65
Plurality	35	65

Removing c reverses choice from a to b!

Problems with Borda count (2)

4 alternatives a,b,c,d:

3 agents: $a \succ b \succ c \succ d$

2 agents: $b \succ c \succ d \succ a$

2 agents: $c \succ d \succ a \succ b$

Borda	a	b	c	d
Score	11	12	13	6

without alternative d:

Borda	a	b	c
Score	8	7	6

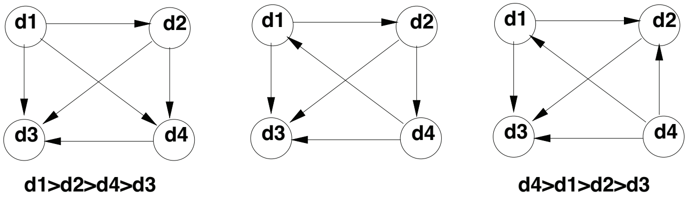
Removing d reverses order from $c \succ b \succ a$ to $a \succ b \succ c$!

Slater ranking

Combined ranking corresponds to a consistent majority graph: every alternative ranked higher beats a lower ranked one.
Slater ranking: among all possible rankings, choose the one that is closest to the agents' majority graph.
Algorithm:

- make agents vote between every pair of alternatives (or ask their preference order and simulate this vote).
- for each possible ordering, evaluate how many edges differ from the majority graph (possibly weighted by the strength of the majority).
- $\Rightarrow$ choose the one with the smallest discrepancy.
- combinatorial optimization problem: hard to solve!

Example: Slater ranking



2 of 24 possible orderings:

- left: edge $d_1 \rightarrow d_4$ is reversed.
- right: edge $d_2 \rightarrow d_4$ is reversed.

Kemeny Scores

- Ask agents to submit total orders of choices.
- For a candidate joint order, for each relation between subsequent choices d_i and d_{i+1}, count how many voters rank the two choices in the *opposite* way.
- Kemeny score of the joint order = sum of these counts.
- Winner = order with lowest Kemeny score.
- Search for joint order using branch-and-bound search.

Voting with Computers

- Computerized Voting Protocols allow more accurate decision making.
- Verification is complex: how to prove that chosen order is optimal?
- However, even simple voting protocols are hard to verify when votes are secret.

Manipulation

Voting may have anomalies, but can agents exploit them to their advantage?
Two forms:

- Manipulation of vote order by vote organizer (as in majority voting).
- Non-truthful voting: agent submits vote that does not correspond to its true preferences.

Manipulation by vote organizer

3 agents A_1, A_2 and A_3 choose between 3 alternatives a,b,c:

$$
\begin{aligned}
A_1 : &\quad a \succ b \succ c \\
A_2 : &\quad b \succ c \succ a \\
A_3 : &\quad c \succ a \succ b
\end{aligned}
$$

- order a,b,c: **c** (a wins over b, c wins over a).
- order c,b,a: **a** (b wins over c, a wins over b).
- order c,a,b: **b** (c wins over a, b wins over c).

Options introduced later in the process have a higher chance!

The Gibbard-Satterthwaite Theorem

Every (deterministic) voting protocol for ≥ 3 alternatives must have one of these three properties:

- the protocol is dictatorial, i.e. one agent decides the outcome.
- there is some candidate who cannot win under any preference profile.
- there are situations where an agent has an interest to not vote according to its true preference, i.e. to manipulate the outcome by a non-truthful vote.

Example of non-truthful vote

3 alternatives a,b,c; plurality votes of other agents:

	a	b	c
Score	3	7	8

Agent X prefers $a \succ b \succ c$:

- votes for a (truthful): c wins
- votes for b (non-truthful): b might win

Non-truthful voting $\Rightarrow$ not clear what the outcome means!

Manipulability of voting

- For many voting protocols, determining if and how the outcome can be manipulated is NP-hard, but...
- This is only the worst case: the average case is likely to be easy.
- Example heuristics:
  - Plurality: vote for most preferred alternative that is within some ϵ of winning.
  - Sensitive rules (where all alternatives are ranked): rank desired outcome first, order all others in *opposite order of other agents' preference*.
- These heuristics will find almost all manipulations.

Randomized Voting

What if outcome could be chosen by a randomized process:

- Majority voting: *probability* of choosing outcome x = fraction of agents who voted for x.
- Voting for y instead of x increases $p(y)$ by $1/n$ and decreases $p(x)$ by the same amount: expected outcome less preferred!

$\Rightarrow$ no incentive to lie about preferences.

- However, random choice could be manipulated.

Better social choice protocols

Problems with voting:

- no consideration of strength of preference $\Rightarrow$ inconsistent situations.
- every voter counts the same in every decision.
- large potential for manipulation.

Better social choice protocols are based on maximizing social welfare $\Rightarrow$ mechanism design.

Summary

- Stability of coalitions: distribute payoffs in the *core* so that no group of agents has incentive to leave coalition.
- Shapley value often falls in the core.
- Voting as social choice protocols.
- Majority voting finds Condorcet winners; but can be manipulated by choice of vote order.
- Anomalies of other voting protocols; incentives for non-truthful voting.
- Optimization-based protocols (Slater ranking, Kemeny scores) allow more rational decisions.
- Randomized choices would solve most problems.