Computational Social Choice: Autumn 2010

Ulle Endriss
Institute for Logic, Language and Computation
University of Amsterdam

Social Choice Theory

SCT studies collective decision making: how should we aggregate the preferences of the members of a group to obtain a "social preference"?

$$\triangle \succ_1 \bigcirc \succ_1 \square$$

$$\square \succ_2 \triangle \succ_2 \bigcirc$$

$$\bigcirc \succ_3 \square \succ_3 \triangle$$

$$?$$

SCT is traditionally studied in Economics and Political Science, but now also by "us": *Computational Social Choice*.

Introduction

The course will cover issues at the interface of *computer science* and *mathematical economics*, including in particular:

- (computational) logic
- multiagent systems
- artificial intelligence
- social choice theory
- game theory
- decision theory

There has been a recent *trend* towards research of this sort. The broad philosophy is generally the same, but people use different names to identify various flavours of this kind of work, e.g.:

- Algorithmic Game Theory
- Social Software
- *and:* Computational Social Choice

Very few specific *prerequisites* are required to follow the course. Nevertheless, we will frequently touch upon *current research* issues.

Organisational Matters

- **Lecturer:** Ulle Endriss (`u.endriss@uva.nl`), Room C3.140
- **TA:** Umberto Grandi (`u.grandi@uva.nl`), Room C3.119
- **Timetable:** Tuesdays 11–13 in Room G3.13 (+ a few tutorials)
- **Examination:** There will be several *homework* assignments on the material covered in the course. In the second block, every student will have to study a *recent paper*, write a short essay on the topic, and present their findings in a talk.
- **Website:** Lecture slides, coursework assignments, and other important information will be posted on the course website:

  `http://www.illc.uva.nl/~ulle/teaching/comsoc/2010/`
- **Seminars:** There are occasional talks at the ILLC that are directly relevant to the course and that you are welcome to attend (e.g., at the Computational Social Choice Seminar).

Topics

The main topic for 2010 will be *voting theory* (8-10 lectures), which we will investigate from all sorts of angles. Some keywords:

- axiomatic method: impossibility theorems, characterisation results
- complexity of voting (computational, communication, . . .)
- strategic manipulation
- voting in combinatorial domains and preference modelling
- maybe: proportional representation, electronic voting

The remaining lectures will be spent on other topics, such as:

- judgment aggregation
- stable matchings
- fair division

If interested, you can arrange *(individual) projects* on some of these (and related) topics with members of the COMSOC Group later on.

Prerequisites

There are no formal prerequisites. But: you should be comfortable with *formal* material and you will be asked to *prove* stuff.

There are two areas for which we will assume some background knowledge that some of you may not yet have. This material will be covered in two *tutorials* in the first few weeks:

- **Complexity Theory:** definition of complexity classes such as P and NP; completeness with respect to a complexity class; proving NP-completeness via reduction
- **Game Theory:** non-cooperative games in strategic form; Pareto optimal outcomes; dominant strategies; pure and mixed Nash equilibria; computing Nash equilibria for small games

Related Courses

- Strategic Games
  Krzysztof Apt
- Cooperative Games
  Stéphane Airiau
- Games and Complexity
  Peter van Emde Boas
- Autonomous Agents and Multiagent Systems (MSc AI)
  Shimon Whiteson

Plan for Today

This course is about *collective decision making:* How can we map individual inputs of a group of agents into a joint decision?

Today we will see some examples, problems, ideas, paradoxes, or just *issues* that illustrate the main question addressed in the course:

▶ *How does collective decision making work?*

The remainder of the course will then be devoted to developing (some of) these rather vague ideas in a rigorous manner.

Three Voting Procedures

Voting is the prototypical form of collective decision making.

Here are three *voting procedures* (there are many more):

- *Plurality:* elect the candidate ranked first most often (i.e., each voter assigns one point to a candidate of her choice, and the candidate receiving the most votes wins)
- *Borda:* each voter gives $m-1$ points to the candidate she ranks first, $m-2$ to the candidate she ranks second, etc., and the candidate with the most points wins
- *Approval:* voters can approve of as many candidates as they wish, and the candidate with the most approvals wins

Example

Suppose there are three *candidates* (A, B, C) and 11 *voters* with the following *preferences* (where boldface indicates *acceptability*, for AV):

5 voters think: **A** $\succ$ B $\succ$ C

4 voters think: **C** $\succ$ B $\succ$ A

2 voters think: **B** $\succ$ **C** $\succ$ A

Assuming the voters vote *sincerely*, who *wins* the election for

- the plurality rule?
- the Borda rule?
- approval voting?

<u>Conclusion:</u> We need to be very clear about which *properties* we are looking for in a voting procedure . . .

The Axiomatic Method: May's Theorem

Three attractive properties ("axioms") of voting procedures:

- *Anonymity:* voters should be treated symmetrically
- *Neutrality:* candidates should be treated symmetrically
- *Positive Responsiveness:* if a (sole or tied) winner receives increased support, then she should become the sole winner

One of the classical results in voting theory:

Theorem 1 (May, 1952) *A voting procedure for two candidates satisfies anonymity, neutrality and positive responsiveness if and only if it is the plurality rule.*

K.O. May. A Set of Independent Necessary and Sufficient Conditions for Simple Majority Decisions. *Econometrica*, 20(4):680–684, 1952.

Example with Three Candidates

Suppose the *plurality rule* is used to decide an election: the candidate receiving the highest number of votes wins.

Assume the preferences of the people in, say, Florida are as follows:

49%:	Bush $\succ$ Gore $\succ$ Nader
20%:	Gore $\succ$ Nader $\succ$ Bush
20%:	Gore $\succ$ Bush $\succ$ Nader
11%:	Nader $\succ$ Gore $\succ$ Bush

So even if nobody is cheating, Bush will win this election. <u>But:</u>

- In a *pairwise contest*, Gore would have defeated anyone.
- It would have been in the interest of the Nader supporters to *manipulate*, i.e., to misrepresent their preferences.

Is there a better voting procedure that avoids these problems?

The Gibbard-Satterthwaite Theorem

More properties of voting procedures:

- A voting procedure is *manipulable* if it may give a voter an incentive to misrepresent her preferences.
- A voting procedure is *dictatorial* if the winner is always the top candidate of a particular voter (the dictator).

Another classical result (not stated 100% precisely here):

Theorem 2 (Gibbard-Satterthwaite) *For more than two candidates, every voting procedure is either dictatorial or manipulable.*

A. Gibbard. Manipulation of Voting Schemes: A General Result. *Econometrica*, 41(4):587–601, 1973.

M.A. Satterthwaite. Strategy-proofness and Arrow's Conditions. *Journal of Economic Theory*, 10:187–217, 1975.

Complexity as a Barrier against Manipulation

By the Gibbard-Satterthwaite Theorem, any voting procedure for ≥ 3 candidates can be manipulated (unless it is dictatorial).

Idea: So it's always *possible* to manipulate, but maybe it's *difficult!*
Tools from *complexity theory* can be used to make this idea precise.

- For *some* procedures this does *not* work: if I know all other ballots and want X to win, it is *easy* to compute my best strategy.
- But for *others* it does work: manipulation is *NP-complete*.

Recent work in COMSOC has expanded on this idea:

- NP is a worst-case notion. What about average complexity?
- Also: complexity of winner determination, control, bribery, ...

J.J. Bartholdi III, C.A. Tovey, and M.A. Trick. The Computational Difficulty of Manipulating an Election. *Soc. Choice and Welfare*, 6(3):227–241, 1989.

P. Faliszewski and A.D. Procaccia. AI's War on Manipulation: Are We Winning? *AI Magazine*. In press (2010).

Vickrey Auctions

We have seen that *manipulation* is a serious problem in voting.
In domains other than voting we can sometimes do better.

Suppose we want to sell a single item in an auction.

- *First-price sealed-bid auction:* each bidder submits an offer in a sealed envelope; highest bidder wins and pays what they offered
- *Vickrey auction:* each bidder submits an offer in a sealed envelope; highest bidder wins but pays *second highest price*

In the Vickrey auction each bidder has an incentive to submit their *truthful valuation* of the item!

William Vickrey received the 1996 Nobel Prize in Economic Sciences for "contributions to the economic theory of incentives".

W. Vickrey. Counterspeculation, Auctions, and Competitive Sealed Tenders. *Journal of Finance* 16(1):8–37, 1961.

Yet Another Example

Suppose 13 voters are asked to each vote *yes* or *no* on three issues; and we use the plurality rule for each issue independently to select a winning combination:

- 3 voters each vote for YNN, NYN, NNY.
- 1 voter each votes for YYY, YYN, YNY, NYY.
- No voter votes for NNN.

But then NNN wins: 7 out of 13 vote *no* on each issue.

This is an instance of the *paradox of multiple elections:* the winning combination received the fewest number of (actually: *no*) votes.

S.J. Brams, D.M. Kilgour, and W.S. Zwicker. The Paradox of Multiple Elections. *Social Choice and Welfare*, 15(2):211–236, 1998.

Combinatorial Domains

Many social choice problems have a *combinatorial structure:*

- During a *referendum* (in Switzerland, California, places like that), voters may be asked to vote on n different propositions.
- Elect a *committee* of k members from amongst n candidates.
- Find a good *allocation* of n indivisible goods to agents.

Seemingly small problems generate huge numbers of alternatives:

- Number of 3-member committees from 10 candidates: $\binom{10}{3} = 120$ (i.e. $120! \approx 6.7 \times 10^{198}$ possible rankings)
- Allocating 10 goods to 5 agents: $5^{10} = 9765625$ allocations and $2^{10} = 1024$ bundles for each agent to think about

We need good *languages* for representing preferences!

Preference Representation Languages

There are many different languages for representing preferences. When choosing a language, we should consider these criteria:

- *Cognitive relevance:* How close is a given language to the way in which humans would express their preferences?
- *Elicitation:* How difficult is it to elicit the preferences of an agent so as to represent them in the chosen language?
- *Expressive power:* Can the chosen language encode all the preference structures we are interested in?
- *Succinctness:* How compact is the representation of (typical) preferences? Is one language more succinct than another?
- *Complexity:* What is the computational complexity of related decision problems, such as comparing two alternatives?

Y. Chevaleyre, U. Endriss, J. Lang, and N. Maudet. Preference Handling in Combinatorial Domains: From AI to Social Choice. *AI Magazine*, 29(4):37–46, 2008.

Judgment Aggregation

Preferences are not the only structures we may wish to aggregate. JA studies the aggregation of judgments on logically inter-connected propositions. Example:

	A	B	C
Judge 1:	yes	yes	yes
Judge 2:	no	yes	no
Judge 3:	yes	no	no
Majority:	yes	yes	no

A: witness is reliable

B: *if* witness is reliable *then* guilty

C: guilty

note that $A \wedge B \rightarrow C$

Problem: While each individual set of judgments is logically consistent, the collective judgement produced by the majority rule is not.

C. List. The Theory of Judgment Aggregation: An Introductory Review. *Synthese*. In press (2010).

The Stable Marriage Problem

Given: 100 men; 100 women; each with a linear preference ordering over the members of the opposite sex.

Problem: Find a *stable matching*. There should be no man and woman that would prefer each other over their assigned partners.

Solution: The Gale-Shapley algorithm works as follows.

- In each round, each man who is not yet engaged proposes to his favourite amongst the women he has not yet proposed to.
- In each round, each woman picks her favourite from the proposals she's receiving and the man she's currently engaged to (if any).
- Stop when everyone is engaged.

D. Gale and L.S. Shapley. College Admissions and the Stability of Marriage. *American Mathematical Monthly*, 69:9–15, 1962.

Analysis

The Gale-Shapley algorithm is correct and efficient:

- The algorithm always *terminates*.
- The algorithm always returns a *stable* matching. For if not, the unhappy man would have proposed to the unhappy woman . . .
- The algorithm has *quadratic complexity:* even in the worst case, no man will propose twice to the same woman. For instance:
  - each man has a different favourite $\rightsquigarrow$ 1 round (n proposals)
  - all men have the same preferences $\rightsquigarrow$ $\frac{n(n+1)}{2}$ proposals

What about other properties? Who is better off, men or women?

Fair Division

Fair division is the problem of dividing one or several goods amongst two or more agents in a way that satisfies a suitable fairness criterion.

This can be considered a problem of *social choice:*

- A group of agents each have individual preferences over a collective agreement (the allocation of goods to be found).
- But: in fair division preferences are often assumed to be cardinal (*utility functions*) rather than ordinal (as in voting)
- And: fair division problems come with some *internal structure* often absent from other social choice problems (e.g., I will be indifferent between allocations giving me the same set of goods)

Today we'll only look into one particular subarea: cake-cutting . . .

U. Endriss. *Lecture Notes on Fair Division.* ILLC, University of Amsterdam, 2010.

Cake-Cutting

"Cake-cutting" is the problem of fair division of a single *divisible* (and heterogeneous) good between n *agents*.

The *cake* is represented by the unit interval $[0, 1]$:

```
|----------------------|
0                      1
```

Each agent i has a *utility function* u_i mapping finite unions of subintervals of $[0, 1]$ to the reals, satisfying . . . some simple properties that don't really matter for this short exposition.

S.J. Brams and A.D. Taylor. *Fair Division: From Cake-Cutting to Dispute Resolution.* Cambridge University Press, 1996.

J. Robertson and W. Webb. *Cake-Cutting Algorithms: Be Fair if You Can.* A.K. Peters, 1998.

Cut-and-Choose

The classical approach for dividing a cake between *two agents:*

- ▶ *One agent cuts the cake in two pieces (which she considers to be of equal value), and the other one chooses one of the pieces (the piece she prefers).*

The cut-and-choose procedure is *proportional:*

- ▶ Each agent is guaranteed at least one half (general: $1/n$) according to her own valuation.

What if there are more than two agents?

The Banach-Knaster Last-Diminisher Procedure

In the first ever paper on fair division, Steinhaus (1948) reports on a solution for *arbitrary* n proposed by Banach and Knaster.

(1) Agent 1 cuts off a piece (that she considers to represent $1/n$).

(2) That piece is passed around the agents. Each agent either lets it pass (if she considers it too small) or trims it down further (to what she considers $1/n$).

(3) After the piece has made the full round, the last agent to cut something off (the "last diminisher") is obliged to take it.

(4) The rest (including the trimmings) is then divided amongst the remaining $n-1$ agents. Play cut-and-choose once $n = 2$. ✓

Each agent is guaranteed a *proportional* piece.

H. Steinhaus. The Problem of Fair Division. *Econometrica*, 16:101–104, 1948.

Computational Social Choice

Research can be broadly classified along two dimensions —
The kind of *social choice problem* studied, e.g.:

- electing a winner given individual preferences over candidates
- aggregating individual judgements into a collective verdict
- fairly dividing a cake given individual tastes

The kind of *computational technique* employed, e.g.:

- algorithm design to implement complex mechanisms
- complexity theory to understand limitations
- logical modelling to fully formalise intuitions
- knowledge representation techniques to compactly model problems
- deployment in a multiagent system

Y. Chevaleyre, U. Endriss, J. Lang, and N. Maudet. *A Short Introduction to Computational Social Choice*. Proc. SOFSEM-2007.

Homework

For the voting procedure you have been assigned

(a) find out how it works and prepare to be able to present it in 90 seconds (on the blackboard), and

(b) find something nice to say about your procedure and prepare for explaining what that is in a further 90 seconds.

Up to 5 points for each question (all or nothing).

Read the COMSOC Survey (Chevaleyre et al., 2007) and browse through the COMSOC Website (workshops, PhD theses, ...):

http://www.illc.uva.nl/COMSOC/

Y. Chevaleyre, U. Endriss, J. Lang, and N. Maudet. *A Short Introduction to Computational Social Choice*. Proc. SOFSEM-2007.

What next?

- Tutorial on complexity theory to be held later this week (only for those who think they might need it).
- No class next week (I'll be at COMSOC-2010 in Düsseldorf).
- Then we'll start with a systematic introduction to voting theory.