MS&E 336/CS 366: Computational Social Choice. Aut 2020-21
Course URL: `http://www.stanford.edu/~ashishg/msande336/index.html`.
Instructor: Ashish Goel, Stanford University.

Lecture 1, 9/20/2021. Scribed by Bryce McLaughlin.

Related Reading: Chapter 1 and 2.1-2.7 of HCSC [1]

Introduction

Class Basics:

- Course Name: Computational Social Choice MS&E 336/CS 366
- Instructor: Ashish Goel
- Location: McCullough 122
- Webpage: `http://web.stanford.edu/~ashishg/msande336/`
- Books: *Handbook of Computational Social Choice* [1] and *Algorithmic Game Theory* [2]
- Deliverables:
  - Scribe a lecture in a group of two (10%)
  - Two homeworks which can be done in groups of 2-3 (40%)
  - A take-home midterm exam (20%)
  - Project Report (Prelim: Nov 17, Final: Dec 3) with Optional Presentation (Dec 1) (30%)
  - Read 4-6 papers.

Motivating Example: Public Referendum

Say you are in a city and need to decide on what services to spend the city budget on. You want to get the public's opinion on what should be improved. How could you ask while making sure you get a representative sample of opinions?

- Ask everyone - Computationally expensive and self-selecting for only those who can/desire to respond.
- Do a random sample of the city - Different demographics still have different response rates, so a one shot sample will still likely be biased.
- Ask community leaders - How do you define a community leader and how do you know when your list is exhaustive?

Solving this problem involves many different areas outside of Social Choice Theory:

- Statistics - How to sample and survey
- Market Design - How to price and incentivize
- Law - How to enforce with consistency

Thus, social choice theory does not take into account all aspects of decision-making and implementation, but instead provides a normative framework to set up the rules of running different votes. By developing this in isolation we can ensure our rules are set up in a way that properties of the output solution will be true without allowing our personal preferences to help inform who should win.

Single Winner Elections

A single winner election is characterized by N voters and M candidates (also known as alternatives). Each voter $i \in [N]$ submits their preference ranking $\succeq_i$ over the M candidates. The collection of all preference rankings $P = \{\succeq_1, \ldots, \succeq_N\}$ is called a *profile.*

Notation

$a \succeq_i b$ - voter i likes candidate a at least as much as candidate b. Desired properties:

- Complete: either $a \succeq_i b$ or $b \succeq_i a$.
- Reflexive: $a \succeq_i b$ and $b \succeq_i a \implies a = b$.
- Anti-symmetric: $a \succeq_i a$.
- Transitive: $a \succeq_i b$ and $b \succeq_i c \implies a \succeq_i c$.

Throughout this course we will use f to refer to both a *Social Choice Function* (SCF) and a *Social Welfare Function* (SCF).

- A SCF takes in a profile and returns a (not-necessarily unique) winner.
- A SWF takes in a profile and returns a (non-strict) ranking of the candidates.

A SWF is in a sense gives a more complete output than a SCF. Moving forward we will refer to both using f, which takes in a profile and returns a result.

Desiderata on f

Below we list some qualities we would like f to have:

- *Non-dictatorial* - A single voter's preference function does not solely determine f.

- *Monotonic* - If $a \in f(P)$ (a is a winning candidate), and P' is constructed as follows (where $\succeq$ represents a preference in P and $\succeq'$ represents a preference in P'):
  - One voter j is such that if $a \succeq_j b$ then $a \succeq'_j b$, and if $c \succeq_j b$ then $c \succeq'_j b$ for all $b, c \neq a$. In other words, only a becomes more preferred under $\succeq'_j$ compared to $\succeq_j$ with all other alternates remaining unchanged in preference.
  - $\succeq_i = \succeq'_i$ for all voters $i \neq j$,

  then $a \in f(P')$. Note that this can be applied iteratively to move a up across multiple voters while maintaining a as a winning candidate.

- *Non-empty* - f has a value for every input profile.

- *Resolute* - f has exactly one value for every input profile.

- *Pareto Optimal* - If $a \in f(P)$ and there is a candidate b such that $b \succeq_i a$ for all voters i, then $b \in f(P)$.

- Fair to candidates:
  - *Neutrality* - If a and b are exchanged in every profile, then the times they are chosen by f are exchanged as well.
  - *Non-imposition* - For each candidates a there exists a profile such that $f(p) = \{a\}$.

- Fair to voters:
  - *Anonymity* - Exchanging the preference functions of voter i and voter j should not change the outcome of f.
  - *Instrumentality* - For each voter i there exists a profile P such that i can switch their preference function and change the output of f.
  - *Condorcet Criterion* - If there exists a candidate a such that for any other candidate b,

    $$|\{i|a \succeq_i b\}| \geq \frac{N}{2},$$

    then a should be a winner. In other words; in a pairwise projection of the election, a beats everyone. If this condition meets a they are known as a Condorcet Winner.
  - *Reinforcement* - Suppose a is in $f(S)$ and $f(T)$ for two profiles S and T which contain none of the same voters, but contain votes on the same candidate. Then $a \in f(S \cap T)$

Interestingly enough, their exist no f that satisfies both reinforcement and the Condorcet criterion

References

[1] F. Brandt, V. Contizer, U. Endriss,J. Lang, and A.D. Procaccis. *Handbook of Computational Social Choice*. Cambridge University Press, 2016.

[2] N. Nisan. *Algorithmic Game Theory*. Cambridge University Press, 2007.