Interest Cardinal Social Choice System

The limitations of traditional voting systems have long intrigued me. Particularly, current voting systems often force voters into artificial dilemmas, bundling unrelated issues into single choices. For example, group decisions on abortion rights are influenced by votes cast on an economic basis (and vise versa). This led me to a question: *How can we design a system that avoids the pitfalls of party-driven polarization and issue-bundling?*

Another problem with current systems is their propensity to Mob Rule: modern elections don't take interest into account, so resolutions that are disproportionately harmful to minority groups often pass. This led me to another question: *How can we design a system that incorporates interest?*

Answering these questions became the foundation for my **Interest-Based Cardinal Social Choice System (CSCS)**—a novel framework designed to maximize average happiness by addressing false dichotomies produced by issue bundling, and naturally incorporating voter interest.

In this novel voting system, n winners are selected from a set of $2n$ alternatives (options). The system pairs alternatives into n pairwise elections (labeled $0, 1, ..., n-1$), and each voter is allowed to distribute a single unit of voting power across the elections according to their preferences.

This split encapsulates nuanced voter preferences: voters who agree on something like economics (e.g. election 0) don't also have to agree on something like international policy (e.g. election 1).

Letting the voter distribute the vote naturally incorporates interest into the system, as voters have an incentive to place more of their vote on elections they care more about. So, if someone cares more about the economy than international relations, they could put 0.7 votes on the economy and 0.3 votes on IR.

GitHub repository of code used in the project: https://github.com/ronak-u-agarwal/cardinal-social-choice-system.

1 Implementation

1.1 Vector representations and calculations

Preferences:
Assign each voter an n-dimensional preference vector with each entry being a -1 or 1. The value in the ith position represents a voter's preference in the ith election, where -1 corresponds to preferring the "alternative -1", and 1 corresponds to preferring "alternative 1."

True interest:
Assign each voter a vector on the $(n-1)$-dimensional standard simplex. Briefly, a simplex is a geometric object such that each point on the $(n-1)$-dimensional standard simplex represents an n-dimensional nonnegative vector with entries summing to 1. This vector of true interest is meant to capture how much a voter truly "cares" about each of the n elections. For example, the true interest vector (0.75, 0.25) represents a voter who cares about election 0 three times as much as election 1 (the notion of "cares three times as much" will be rigorously defined later).

Vote distribution:
Assign each voter a n-dimensional vote distribution describing how they divides their vote. Like the interest vector, the vote distribution vector lies on the $(n-1)$-dimensional standard simplex.

Calculating winners:
First, take each voter's cast vote (someone with preferences (-1, 1) and distribution (0.6, 0.4) will cast the vote (-0.6, 0.4)). Add up these cast-votes for all voters in the population to get a total tally vector. A negative entry in position i represents "alternative -1" winning election i, and a positive entry in position i represents "alternative 1" winning election i.

Calculating happiness of voter given an outcome:
If a voter assigns x interest to an election, then a win adds x to their happiness, and a loss subtracts x. Due to the above setup, we can calculate this happiness by taking the element-wise product of the preferences and the outcome vector, then taking the dot product of this vector with true interest. Because interest adds up to 1, a happiness score of -1 represents losing all elections, and a score of 1 represents winning all elections.

Furthermore, we now have a notion of what comparisons like "double interest" mean: if losing elections i and j impacts a voter's happiness the same amount as losing election k, the sum of the voter's interest in i and j would equal the voter's interest in k. By grounding these comparisons in mathematical terms, abstract ideas are translated into testable, quantitative measures, which is essential for connecting theoretical constructs to real-world voter dynamics. However, one pitfall is that our measure might not specifically represent reality: specifically, happiness might not be an additive construct.

1.2 Theoretical results

The first and main result is that if voters use their true interest vector as their vote distribution vector, this CSCS maximizes average happiness, where happiness is defined by the above calculation.

The proof of this is as follows:

Total happiness $= \sum_{elections} \sum_{voters} H(v, e, r)$, where $H(v, e, r)$ is the function that calculates a voter v's happiness with result r in election e. Because of the setup, this is equivalent to: $(P(v,e) \cdot r) \cdot (I(v,e))$, where $P(v,e)$ represents voter v's preference in election e, and $I(v,e)$ represents voter v's interest in election e. Thus, toggling the result from r to $-r$ inverts the sign of happiness:

$$H(v,e,-r) = -H(v,e,r) \Rightarrow \sum_{voters} H(v,e,-r) = -\sum_{voters} H(v,e,r)$$

In other words, if one alternative contributes total happiness H, then the other alternative contributes $-H$ total happiness. Thus, maximal happiness is achieved by selecting the alternative that contributes positive happiness (if they aren't tied).

When vote distributions align with interest distributions, the margin by which an alternative wins directly corresponds to its contribution to total happiness. This requires some care with signs: for example, if the tally vector is something like (-3.4, 1.2, -6.5), then the alternative -1 in election 0 contributes 3.4 to total happiness, while alternative 1 in election 0 would contribute -3.4 to total happiness. Thus, the alternative that is selected by the election is the alternative that contributes a positive value to average happiness, and the other alternative would contribute a negative number (given that there is not a tie).

Therefore, given that voting distribution vectors match interest vectors, this CSCS maximizes average happiness.

2 Simulating Populations

To test my system, I create a simulated population of voters. I was curious to see what types of populations my system works best with, so I created populations with a variety of traits/tendencies by adjusting the bias in my random initialization of voter interest and preference.

2.1 Interest

To initialize voter interest and vote distribution, I needed to generate n-dimensional vectors adding up to 1. In other words, I needed to sample points from the $(n-1)$-dimensional standard simplex.

As a baseline, I wanted to be able to sample from the simplex uniformly, so any nonnegative vector adding up to 1 would be equally likely. However, uniformly sampling points from the simplex proved more challenging than anticipated, as my initial methods (softmaxing random numbers, squaring coordinates on a sphere, etc.) introduced bias. I eventually came to a randomization technique (the breakpoints method) that guaranteed equal likelihood for all vectors.

With the breakpoints method, I chose $n-1$ points at random on the interval between 0 and 1, sorted them from least to greatest into a vector $(a_1, a_2, \ldots, a_{n-1})$, then took the n

lengths: $(a_1 - 0, a_2 - a_1, \ldots, 1 - a_{n-1})$. This method uniformly samples from the simplex because the probability density of a specific point (a, b, c) is invariant.

Proof: Suppose we have arbitrary (a, b, c), where $a + b + c = 1$. Then

$$Pdf((a, b, c)) = 2 \cdot Pdf(a) \cdot Pdf(a + b) = 2$$

As (a, b, c) was arbitrarily chosen on the simplex, we can see that probability density on the simplex is uniform. Also, note that this PDF is defined on a parameterized 2-D simplex with area 1/2, as opposed to the standard 2-D simplex in R^3 with area $\sqrt{3}/2$.

I also designed biased simplex sampling methods to simulate real-world scenarios, such as voters consistently prioritizing high-stakes elections like the presidency. This approach allowed me to explore how varying interest distributions influenced system outcomes and strategies that groups can take to manipulate results.

2.2 Preference

To initialize voter preference, I needed to generate n-dimensional vectors with values -1 or 1. As a baseline, I wanted to be able to sample from the preference space uniformly, so I randomly chose each entry for each voter.

I also wanted to test how different groups might compete with strategic voting to beat each other. To simulate this, I created ”parties” where voters generally had the same preferences, with a little bit of noise.

3 Simulating Strategic Behavior

Recognizing that any system is susceptible to manipulation, I explored how coalitions of voters might strategically distribute their votes to skew outcomes. While straightforward to understand on a general level, the specifics get tricky: how much information do we assume voters have about other voters? How do voters know who to collaborate with? How much are voters willing to sacrifice for the sake of collaboration? Does a game-theoretic outcome exist? Thus, I needed to transform a broad questions about abstract concepts into specific, testable sub-questions about measurable quantities.

These specific questions naturally emerged as I started my investigation by implementing Monte Carlo simulations to model randomized voting strategies. Specifically, I generated random walks across the simplex to alter voter distribution vectors, and calculated resulting happiness to identify strategic voting and Nash equilibria.

One difficulty that arose with performing random walks on each voter was that the outcome of the group vote remained relatively unchanged: the alterations to each voter's distribution canceled each other out. Thus, I needed to refine my technique to capture group coordination.

To do this, I had to make "coordination" more concrete. Specifically, I came to the game theoretic conclusion that voters will never vote against their own preferences for the sake of collaboration. However, they could set their vote-distribution equal to 0 if they disagree with their collaborator's ideal set of winners.

This led to a key finding: the **Representative Voter Theorem (RVT)**, which re-represents a coalition's behavior as a single "representative" voter. Importantly, this analytic finding rigorously defines what "coordination" means in a collaborating group.

Briefly, the RVT is that, given certain conditions, a group of m voters can be represented as one single voter with m votes, derivable preference and interest vectors, and a restricted vote distribution vector.

When to join a coalition

One large assumption in the above discussion is that voters are willing to shift their vote to match group needs. For this assumption to hold, joining a coalition must benefit each member. This creates a complicated game theory problem: voters need to decide their best strategy, but that "best strategy" depends on what others are doing. As other voters face the same problem, the result is a recursive mutual dependence that complicates the system.

To address this, I developed a computationally expensive but theoretically stable solution that evaluates all $\binom{2^n}{2}$ possible pairs of outcomes. For each pair of outcomes, voters determine their preferences, and the outcome with greater support "wins," as the group favoring it can imitate the smaller group and freely distribute their remaining votes. If a single outcome defeats all others in this manner, it becomes the game-theoretic winner.

This approach provides a systematic framework to explore strategic behavior, but its computational intensity and potential for oscillatory strategies (no winner) underscore the need for further refinement.

4 Conclusion and further research

This project introduces an interest-based Cardinal Social Choice System (CSCS) designed to address fundamental issues in traditional voting systems, such as issue-bundling and lack of representation for intensity of preferences. By allowing voters to distribute their voting power across multiple pairwise elections, the system aims to maximize average happiness while minimizing the risks of mob rule and polarization.

Linear algebra and probability theory informed the theoretical design, while Python enabled me to implement and simulate the system efficiently. Throughout the project, I engaged in a constant interplay between general and specific questions, systematically refining broad inquiries into precise, testable terms.

However, several directions for future research remain. Exploring alternative measures of happiness could broaden the system's applicability, and testing the system in real-world scenarios would provide valuable insights into its practicality and impact.

One intriguing conjecture for further investigation is whether a uniform vote distribution across all elections might yield a game-theoretically stable solution. More broadly, refining the model's computational efficiency and addressing the potential for strategic oscillations will be critical steps towards finding strategic voting Nash equilibria.