Optimal Policy Selection Under Bilateral Preferences

Daniel Bruwel Katie O'Reilly

June 20, 2025

Abstract

Many situations involve a selection of policies that affect two subgroups or agents differently. Examples range from economic policies impacting different parties to household decisions affecting members of a couple or roommates. These decisions are typically binary—for instance, an economic policy may be passed or rejected, or a household decision may favor one member's preference over another's. In this note, we analyze the problem of finding the optimal subset of policies to select. We propose a simple objective function and examine greedy algorithms, brute force, continuous approximations, and other methods. We show that the greedy algorithm is optimal for a broad class of problems, making brute force approaches unnecessary.

1 Introduction

[1]

2 Problem Formulation

You have a collection of n policies, denoted by the subscript i. Each policy can either be implemented (1) or not implemented (0). For each policy we denote $\pi = [\pi_i]$ as the vector of relisations of each policy. This is a binary vector of length n.

We have2−agents (possibly more but we only consider the two agent case, the agents could represent a collection or party). Each member has a scoring function $s : \pi \to \mathbb{R}$ that distribues lienarly over the policies. That is to say $s(\pi) = \sum_i s_i(\pi_i)$. This distributive property can be relaxed if different policies interact with each other, but we will not consider this case here.

We further modify each score with some utility function $u : \mathbb{R} \to \mathbb{R}$, which is a monotonic function that transforms the score into a utility. The utility of each agent is then given by $u(s(\pi))$.

The choice of this scoring function depends on the context of the problem. For example, for an economic policy, the score could represent the expected economic benefit, while for a household decision, it could represent the satisfaction of each member. One option for a househould decision is to use the "$50 : 50$ rule" where each person says something like "I'm leaning 70% towards passing the policy, and 30% towards rejecting it".

In many cases (including the $50 : 50$ rule and expceted economic outcome), the scoring function is linear in the policies. In this case, we can write the score of each agent as:

$$s_a(\pi) = m_a \cdot \pi + c_a$$

Where the subscript a denotes the agent, m_a is a vector of weights for each policy, and c_a is a constant term. We will consider this linear case in the rest of this note.

We will also chose to use $u(x) = \ln(x)$ as this is the unique sclae invariant constant relative risk aversion utility.

Combining this leads to the final log-loss as

$$\mathcal{L}(\mathbf{x}) = -\left(m_0 \cdot \pi + b_0\right)\left(m_1 \cdot \pi + b_1\right)$$

Our goal is to find the vector π that minimises this loss.

3 Analysis

3.1 Efficient Frontier

For each policy selection π, we can plot the point $(s_0(\pi), s_1(\pi))$. It is easy to maximise s_a by simply takign $\pi = 0$ whenever $b_a < 0$ and $\pi = 1$ whenever $b_a \geq 0$. This gives us an $a-$optimal policy selection. For any fixed c there is a policy selection that maximises s_a subject to $s_{a'} \geq c$. Plotting these create a sort of "efficiency frontier", wherein every policy will fall "within" this frontier. On this plot we can overaly the lines of constant utility. The figure below shows such a frontier on randomly simulated data (see apendix for the simulation process for this figure).

To calculate the efficiency frontier, we are essentially asking for a collection of policies to include that maximises $s_1(\pi)$ subject to a constraint on $s_0(\pi)$. This is a knapsack problem, which is NP-hard. However, we can use a greedy algorithm to approximate the solution. Once we have the efficiency frontier, we can find the point on the frontier that maximises the utility of both agents.

If we analyise the distribution for policies assuming the outcome of each policiy (i.e. 0 or 1 is randomly selected), and that each persons preference score is random, then we have that $s_i(\pi_i)$ $\rho(x)$ for $x \in [0, 1]$. Taking multiple scores for multiple policies gives $s_a(\pi)$ $\mathcal{N}(\mu)a, \sigma_a)$ by the central limit theorem. As such, the distribution for possible scores is going to form a bivariate normal distribution. The efficiency frontier is then going to be eliptical in shape.

3.2 Optimisation Algorithms

3.2.1 Brute Force

For a small number of policies we can simply exhaustively search through all possible policy selections. This involves checking every combination of policies to find the one that maximises the utility of both agents. The brute force approach is computationally expensive, especially as the number of policies increases, but it guarantees finding the optimal solution.

3.2.2 Greedy Algorithms

We have two alternate options that are both greedy algorithms. This first is to calculate the efficiency frontier as described above, and then to find the point on the frontier that maximises the utility of both agents. The second option is to start at some $a-$optimal policy, and then to iteratively add policies that increase the utility of both agents until no more policies can be added. We will only focus on the second option.

3.2.3 Simulated Annealing

Similar in spirit to the greedy algorithm, we can use simulated annealing to find a good policy selection. The idea is to start with an initial policy selection and then iteratively make small changes to the selection, accepting changes that improve the utility of both agents. The temperature

parameter controls how likely we are to accept changes that decrease utility, allowing us to explore the solution space more thoroughly.

3.2.4 Continuous Approximation

We can extend π so that rather than being binary, it is a continuous variable in $[0,1]^n$. If we look at our objective function, we have

$$\begin{aligned}\mathcal{L}(\mathbf{x}) &= -\left(m_0 \cdot \pi + b_0\right)\left(m_1 \cdot \pi + b_1\right) \\ &= -\left(m_0^T\pi + b_0\right)\left(m_1^T\pi + b_1\right) \\ &= -(m_0^T\pi)(m_1^T\pi) - b_0 m_1^T\pi - b_1 m_0^T\pi - b_0 b_1 \\ &= -\pi^T M\pi - c^T\pi - b_0 b_1\end{aligned}$$

Where here we defined $M = m_0 m_1^T$ and $c = b_0 m_1 + b_1 m_0$. We can ignore the constant term $-b_0 b_1$ as it does not depend on π. We can also do the standard substitution to replace $\pi^T M\pi$ with $\pi^T Q\pi$ where $Q = \frac{1}{2}\left(M + M^T\right)$ to make the objective function a standard quadratic form. We now have a quadratic programming problem with box constraints on π. This can be solved efficiently using standard quadratic programming techniques. Doing so will yield a continuous approximation of the optimal policy selection, and we can make the final selection by rounding the continuous solution to the nearest binary vector.

4 Computational Experiments and Results

References

[1] V. Chvátal. A greedy heuristic for the set-covering problem. *Mathematics of operations research*, 4(3):233–235, 1979.