Optimizing Voting Rules for Social Welfare and Beyond

Demonstration Track

Ben Armstrong
Tulane University
New Orleans, United States
barmstrong@tulane.edu

Hrishi Kabra
Tulane University
New Orleans, United States
hkabra@tulane.edu

Nicholas Mattei
Tulane University
New Orleans, United States
nsmattei@tulane.edu

ABSTRACT

We present a website and Python package, `optimal_voting`, that provides tools for finding the positional scoring rule (PSR) that optimizes for a user-selected target function over a (set of) profiles. The web interface provides an intuitive interface for creating profiles, analyzing them, and exploring various PSRs; while the library provides extensive functionality for researchers and system implementers. By formulating finding the optimal PSR within a larger optimization system, the target of the optimization can be any of a number of widely studied functions including classical social welfare functions (utilitarian, egalitarian, Nash), distortion, or axiom violation rate. By restricting optimization to positional scoring rules we ensure that our results are intuitively explainable. Our library has already been used to generate significant results for two papers and can be directly applied to several further settings.

KEYWORDS

Social Choice; Positional Scoring Rules; Optimization

ACM Reference Format:
Ben Armstrong, Hrishi Kabra, and Nicholas Mattei. 2026. Optimizing Voting Rules for Social Welfare and Beyond: Demonstration Track. In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), Paphos, Cyprus, May 25 – 29, 2026*, IFAAMAS, 3 pages. https://doi.org/10.65109/SSLT9308

1 INTRODUCTION

Social choice studies the process of making a collective decision by taking into account the preferences of multiple individual stakeholders (agents). Applications of this classic setting range from scoring athletes across many competitions [22], aggregating scores in conference reviewing [6, 19], to aggregating rankings given by recommender systems [1], to political elections [4]. Much of the research in the field of *computational* social choice studies fundamental algorithmic possibilities or determines the computational hardness of achieving certain outcomes, while other work often aims to develop voting rules with superior properties [12].

Voting rules are evaluated in many ways: on the axioms they satisfy [12, 13]; the social welfare they provide [9]; on functions tailored for specific purposes, such as ranking papers under review [6]; or distortion from some ground truth model [3]. In recent years neural networks have been trained as novel voting rules optimized for several of these tasks, including axiom satisfaction [13, 17, 20] and utility optimization [2]. However, such rules pose risks to practical usage due to the black-box nature of neural networks.

On the other hand, *positional scoring rules* (PSRs) are a well studied, interpretable class of voting rule. Explaining voting rules is complex [5, 8, 14, 18], but in general PSRs are accepted as being interpretable and explainable for end users. PSRs are defined by a **score vector** which awards points to alternatives based on each voter's preferences. A voter ranking some alternative at the i^{th} position awards points equal to the score vector's i^{th} value. For example, the *plurality* rule, $(1, 0, ..., 0)$, awards one point to each voter's first preference. The Borda rule, $(m-1, m-2, ..., 0)$ for m alternatives, aims for greater fidelity over the ranking [12]. The alternative receiving the highest sum of points across all voters wins. In the optimization view of Boutilier et al. [9], an evaluation function considers that winner, alongside voter preferences, and gives a score for a rule based on the winners it selects across many profiles. This is often a social welfare (SW) function, e.g., the sum of all voter utilities for the winning alternative (utilitarian SW). In our framing this can be any function over profile, utilities, and results.

Contribution. We introduce `optimal_voting`, a Python package for finding *novel and interpretable* positional scoring rules maximizing a target evaluation function. We also provide an intuitive web interface for analyzing and exploring these PSRs as well as defining target optimization functions and profiles[1]. Our library has already provided results for multiple distinct projects [6, 13]. The platform has two main components: (1) a web interface which allows users to optimize a positional scoring rule given a social welfare function and (set of) preference profiles, and (2) a Python package which powers the web interface and exposes additional functionality. Our web interface is a DJANGO website which allows the user to:

- Select an optimization target from four existing social welfare functions: utilitarian, egalitarian, Nash, and malfare.
- Sample preference from well-known distributions, load real-world data from PrefLib [21], or create custom profiles.
- Compare welfare of optimized positional scoring vectors against many canonical scoring vectors, e.g., Figure 1.

The `optimal_voting` package[2] powers the functionality of our website and provides additional functionality, discussed in Section 3.

2 WEB INTERFACE FUNCTIONALITY

The web interface can be used as a stand alone application to analyze a voting profile – akin to e.g., OPRA [15], Pynx [11], Spliddit [16], or WHALE[3] – and perform basic optimization tasks, while

Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), C. Amato, L. Dennis, V. Mascardi, J. Thangarajah (eds.), May 25 – 29, 2026, Paphos, Cyprus. © 2026 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). https://doi.org/10.65109/SSLT9308

[1] A video overview of the work is at: youtu.be/v1vwvr9uC9Y
[2] https://pypi.org/project/optimal-voting/
[3] https://whale5.imag.fr/

Scoring Rule	Winner	Utilitarian SW	Nash SW	Egalitarian SW	Malfare SW
Plurality	Option A	0.98	0.91	1.00	1.00
Veto	Option B	1.00	1.00	1.00	1.00
Borda	Option D	1.00	0.94	1.00	1.00
Harmonic	Option A	0.98	0.91	1.00	1.00
Plurality + Veto	Option A	0.98	0.91	1.00	1.00
Two Approval	Option B	1.00	1.00	1.00	1.00

Figure 1: Comparison across score vectors on a single profile. Social welfare is normalized to a maximum value of 1.

optimal_voting goes beyond these, allowing for computing optimal PSRs on different domains and optimization targets.

Creating and Importing Profiles. A profile represents the preference of a group of voters; it contains, for each voter, an *ordinal ranking* over alternatives and the *cardinal utility* the voter receives should each alternative be elected. The user may create ordinal profiles manually, sample them from common preference distributions (e.g., Impartial Culture (IC), Mallows, Single-Peaked [7]), or import real-world preferences from PrefLib [21]. The user may then generate utilities from various distributions or set them manually.

Optimization. Optimization can be done over a single profile or over many profiles at once. Ultimately, optimization searches for a score vector which maximizes an optimization target across all provided profiles. On the web interface four social welfare functions are supported as optimization targets: utilitarian, egalitarian, Nash, and malfare. We use simulated annealing for optimization [4] which takes in one or more profiles (ordinal and cardinal data), and an optimization target. The optimal_voting package is integrated into the web backend and performs the optimization; several thousand steps of annealing complete in seconds for large PrefLib collections.

Comparison and Analysis. With a single profile, our interface allows comparing many classical scoring vectors, providing both the winning alternative and the (normalized) SW granted by the winner. Figure 1 shows one aspect of this analysis. In optimization mode (typically with multiple profiles), the web interface also shows the generated novel score vector optimized for the given profiles, the SW history during optimization, and the average score and SW given to each alternative, as shown in Figure 2.

3 ADDITIONAL FUNCTIONALITY IN LIBRARY

In addition to the web interface functionality, the optimal_voting library allows for more complex use cases, including:

- **Custom Optimization Targets:** Rather than using a known SW function, a user can provide any function which provides a score associated with each winning alternative and profile.
- **Optimizing for Distortion:** While social welfare is an interesting optimization target to study, theorists are often more interested in the worst-case ratio of the maximum SW to the actual SW achieved, i.e., distortion [3, 9].
- **Randomized Scoring Rules:** PRSs can be interpreted as lotteries where an alternative's score is the relative probability of winning [10]. The library supports deterministic or probabilistic modes, allowing flexibility over the website.

[4]Gradient descent and mixed-integer programs are also available in the library.

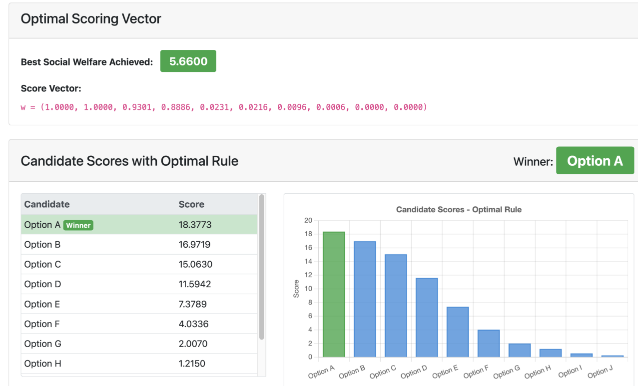

Figure 2: Web interface showing positional scoring rule and scores for each alternative after optimizing egalitarian social welfare over 100 profiles sampled from Mallow's distribution.

4 POTENTIAL AND SUCCESSFUL USE CASES

Our focus on PSRs and arbitrary optimization targets allows distinct use cases from similar applications: whalrus[5] and Pnyx [11] have a focus on simply computing the output of existing rules. Tools for optimization, such as Spliddit [16], generate solutions for single instances, often with limited interpretability, rather than an interpetable rule which can be used for many future decisions.

Web: Profile Analysis. Our web interface allows comparing the social welfare of alternatives across many welfare functions. This allows users to make decisions, or select a rule for future decisions, by deciding on a welfare function to use. For example, a group may submit preferences about a movie to watch and agree that the movie should make the least satisfied person as happy as possible. They can then identify the movie which maximizes egalitarian SW.

Web or Library: Deepen Understanding of Social Welfare Functions. Theoretical work has studied optimal scoring rules on IC preferences under utilitarian SW [9]. Our work allows for experimental analysis of settings not yet theoretically tractable, potentially guiding future theoretical and empirical inquiry for optimal voting in novel domains. Early experiments suggest the existence of a novel scoring rule family that optimizes egalitarian SW.

Library: Custom Optimization Targets. Our library allows custom optimization targets which we have already used to support a number of research findings. Caiata et al. [13] study how often axioms are violated by different voting rules and uses optimal_voting as one included voting rule; finding that while Borda scores often perform well a novel PSR reduces violations. Berker et al. [6] use optimal_voting with a custom target to find a PSR that provides *consistent* rankings over alternatives, with optimized rules significantly outperforming others. Finally, we have applied our framework to find PSRs for recommender systems that provide higher utility to end users under various fairness constraints [1].

[5]https://francois-durand.github.io/whalrus/

ACKNOWLEDGMENTS

All authors were supported in part by NSF Awards IIS-RI-2007955, IIS-III-2107505, IIS-RI-2134857, IIS-RI-2339880 and CNS-SCC-2427237 as well as the Harold L. and Heather E. Jurist Center of Excellence for Artificial Intelligence at Tulane University and the Tulane University Center of Excellence for Community-Engaged Artificial Intelligence (CEAI).

REFERENCES

[1] Amanda Aird, Paresha Farastu, Joshua Sun, Elena Stefancová, Cassidy All, Amy Voida, Nicholas Mattei, and Robin Burke. 2024. Dynamic Fairness-Aware Recommendation Through Multi-Agent Social Choice. *ACM Transactions on Recommender Systems* 3, 2 (2024), 1–35.

[2] Cem Anil and Xuchan Bao. 2021. Learning to Elect. *Advances in Neural Information Processing Systems* 34 (2021), 8006–8017.

[3] Elliot Anshelevich, Aris Filos-Ratsikas, Nisarg Shah, and Alexandros A Voudouris. 2021. Distortion in Social Choice Problems: The First 15 Years and Beyond. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence Survey Track*.

[4] Kenneth J Arrow, Amartya Sen, and Kotaro Suzumura. 2010. *Handbook of social choice and welfare*. Vol. 2. Elsevier.

[5] Antoinette Baujard, Roberto Brunetti, Isabelle Lebon, and Simone Marsilio. 2025. How people understand voting rules. *Public Choice* (2025), 1–30.

[6] Ratip Emin Berker, Ben Armstrong, Vincent Conitzer, and Nihar B Shah. 2025. Designing Rules to Pick a Rule: Aggregation by Consistency. *arXiv preprint arXiv:2508.17177* (2025).

[7] Niclas Boehmer, Piotr Faliszewski, Łukasz Janeczko, Andrzej Kaczmarczyk, Grzegorz Lisowski, Grzegorz Pierczyński, Simon Rey, Dariusz Stolicki, Stanisław Szufa, and Tomasz Wąs. 2024. Guide to numerical experiments on elections in computational social choice. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*. 7962–7970.

[8] Arthur Boixel, Ulle Endriss, and Oliviero Nardi. 2022. Displaying Justifications for Collective Decisions. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022*. ijcai.org, 5892–5895.

[9] Craig Boutilier, Ioannis Caragiannis, Simi Haber, Tyler Lu, Ariel D Procaccia, and Or Sheffet. 2012. Optimal Social Choice Functions: A Utilitarian View. In *Proceedings of the 13th ACM Conference on Electronic Commerce*. 197–214.

[10] Felix Brandt. 2017. Rolling the dice: Recent results in probabilistic social choice. *Trends in computational social choice* (2017), 3–26.

[11] Felix Brandt, Guillaume Chabin, and Christian Geist. 2015. Pnyx: A Powerful and User-friendly Tool for Preference Aggregation.. In *AAMAS*. 1915–1916.

[12] Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel D Procaccia. 2016. *Handbook of Computational Social Choice*. Cambridge University Press.

[13] Joshua Caiata, Ben Armstrong, and Kate Larson. 2026. What Voting Rules Actually Do: A Data-Driven Analysis of Multi-Winner Voting. *40th Annual AAAI Conference on Artificial Intelligence* (2026).

[14] Olivier Cailloux and Ulle Endriss. 2016. Arguing about Voting Rules. In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems, Singapore, May 9-13, 2016*. ACM, 287–295.

[15] Yiwei Chen, Jingwen Qian, Junming Wang, Lirong Xia, and Gavriel Zahavi. 2021. Opra: an open-source online preference reporting and aggregation system. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 16011–16013.

[16] Jonathan Goldman and Ariel D Procaccia. 2015. Spliddit: Unleashing Fair Division Algorithms. *ACM SIGecom Exchanges* 13, 2 (2015), 41–46.

[17] Levin Hornischer and Zoi Terzopoulou. 2025. Learning How to Vote with Principles: Axiomatic Insights into the Collective Decisions of Neural Networks. *Journal of Artificial Intelligence Research* 83 (2025).

[18] Inwon Kang, Qishen Han, and Lirong Xia. 2023. Learning to Explain Voting Rules. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*. 2883–2885.

[19] Jing Wu Lian, Nicholas Mattei, Renee Noble, and Toby Walsh. 2018. The conference paper assignment problem: Using order weighted averages to assign indivisible goods. In *proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32.

[20] Leonardo Matone, Ben Abramowitz, Ben Armstrong, Avinash Balakrishnan, and Nicholas Mattei. 2025. Deepvoting: Learning Voting Rules with Tailored Embeddings. *arXiv preprint arXiv:2408.13630* (2025).

[21] Nicholas Mattei and Toby Walsh. 2013. Preflib: A Library For Preferences http://www.preflib.org. In *International conference on algorithmic decision theory*. Springer, 259–270.

[22] Quang Nguyen, Hannah Butler, and Gregory J Matthews. 2021. An Examination of Olympic Sport Climbing Competition Format and Scoring System. *arXiv preprint arXiv:2111.05310* (2021).