Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence (IJCAI-22)
Demonstrations Track

Displaying Justifications for Collective Decisions

Arthur Boixel[1] and **Ulle Endriss**[1] and **Oliviero Nardi**[2]
[1]ILLC, University of Amsterdam
[2]TU Wien
{a.boixel, u.endriss}@uva.nl, oliviero.nardi@tuwien.ac.at

Abstract

We present an online demonstration tool illustrating a general approach to computing justifications for accepting a given decision when confronted with the preferences of several agents. Such a justification consists of a set of axioms providing a normative basis for the decision, together with a step-by-step explanation of how those axioms determine the decision. Our open-source implementation may also prove useful for realising other kinds of projects in computational social choice, particularly those requiring access to a SAT solver.

1 Introduction

Suppose we have to select a "best" alternative from a given set of alternatives on the basis of the preferences expressed by several agents. We of course could delegate this decision to a computer program implementing one of the many voting rules that have been proposed in the literature [Brams and Fishburn, 2002]. But sometimes we expect more than simply being presented with the output returned by such a program. Sometimes we would like to see a meaningful justification for why a given choice really is the right one. Such a justification should appeal to basic normative principles we can agree with; and it should present the reasoning steps involved in showing that the suggested outcome really is entailed by those principles—in a manner that is easy to understand.

This point has been made by a number of authors in recent years [Cailloux and Endriss, 2016; Procaccia, 2019; Boixel and Endriss, 2020], and it ties in with broader concerns regarding the explainability of algorithmic decision making powered by AI [Miller, 2019; Arrieta *et al.*, 2020].

Realising the ideal of (automatically) justifying collective decisions from first principles presents itself as a natural challenge for the field of computational social choice, given its concern with both the normative and the algorithmic aspects of collective decision making [Brandt *et al.*, 2016]. Here we present an online tool we developed to showcase one particular approach addressing this challenge [Boixel and Endriss, 2020; Boixel *et al.*, 2022; Nardi *et al.*, 2022].

Roadmap. In Section 2 we introduce the problem of computing a justification for a given target outcome when presented with a profile of preferences and a corpus of axioms encoding normative principles of interest. Then, in Section 3 we present our demonstration tool and in Section 4 we briefly describe the AI techniques used to build it, before discussing possible directions for future developments in Section 5.

2 Justifying Collective Decisions

In this section we provide an informal account of the approach to finding axiomatic justifications for collective decisions we developed in a series of recent papers [Boixel and Endriss, 2020; Boixel *et al.*, 2022; Nardi *et al.*, 2022].

We are concerned with decision-making scenarios in which several *agents* each express their individual *preferences* by providing a ranking of the *alternatives* in a finite set X. We treat all agents the same, so when talking about such a *profile* of preferences we only keep track of *how many* agents support any given ranking. Making a collective decision amounts to selecting an *outcome*, a nonempty subset $X^\star \subseteq X$. When $X^\star$ is a singleton, then we may think of that single element of $X^\star$ as the "best" alternative in X; otherwise, we may think of the elements of $X^\star$ as all being "tied for best".

Example 1. If you were to ask five sommeliers to rank three of the best known Italian wines—Amarone, Brunello, and Chianti—you might obtain the following preference profile:

$$\begin{array}{ll} \#2: & \text{Chianti} \succ \text{Brunello} \succ \text{Amarone} \\ \#1: & \text{Brunello} \succ \text{Amarone} \succ \text{Chianti} \\ \#1: & \text{Brunello} \succ \text{Chianti} \succ \text{Amarone} \\ \#1: & \text{Amarone} \succ \text{Chianti} \succ \text{Brunello} \end{array}$$

That is, the first ranking is reported by two individuals, while the other rankings have just one supporter each.

Observe that the well-known *Borda rule* would declare a tie between Brunello and Chianti (with 5 points each), while the *Copeland rule* would select Chianti (winning all pairwise majority contests). So what is the right choice, and why? △

We might justify the outcome selected by a voting rule F by appealing to the *axioms* characterising F [Zwicker, 2016]. Examples include the *Pareto Principle*, saying that a dominated alternative should never be selected, and the *Neutrality Principle*, postulating symmetric treatment of the alternatives. But this is not the route we follow here. Instead, we want to justify outcomes by appealing to axioms *directly*.

So suppose we are given a profile $R^\star$, a target outcome $X^\star$, and a corpus $\mathbb{A}$ of axioms we may rely on. In its most basic form, a justification for $X^\star$ is simply a reference to a set

$\mathcal{A}^{\mathrm{N}} \subseteq \mathbb{A}$, a so-called *normative basis*, such that every voting rule satisfying the axioms in $\mathcal{A}^{\mathrm{N}}$ will return $X^\star$ for $R^\star$.[1]

Example 2. Let's return to our oenological case study. We can justify the outcome $\{$Chianti$\}$ by reference to a normative basis consisting of just one axiom, the *Condorcet Principle*, which demands that any alternative beating all others in pairwise majority contests should be the only winner.

Can we also justify the tied outcome $\{$Brunello, Chianti$\}$? Yes, we can. As an expert in social choice theory would be able to confirm, the normative basis consisting of the aforementioned Neutrality and Pareto Principles together with the *Reinforcement Principle* does the job. The latter says: when one group of agents selects Y and another selects Y', then their union should select $Y \cap Y'$ (unless that intersection would be empty). But what if you are no such expert? $\triangle$

What is still missing from our notion of justification is the explanatory component. So let us refine our definition by requiring that $\mathcal{A}^{\mathrm{N}}$ must be paired with an *explanation* $\mathcal{A}^{\mathrm{E}}$ made up of a set of instances of the axioms in $\mathcal{A}^{\mathrm{N}}$. Here an *instance* of an axiom is an application of that axiom to a specific situation (e.g., specific profiles and alternatives). $\mathcal{A}^{\mathrm{E}}$ must be such that every voting rule that satisfies it will return $X^\star$ for $R^\star$. In addition, we may require that $\mathcal{A}^{\mathrm{E}}$ can be presented in a structured form, as a step-by-step derivation.

Example 3. We can explain how Neutrality, Pareto, and Reinforcement force the selection of $\{$Brunello, Chianti$\}$ as follows. First, consider this subprofile (let's call it R_1):

$$\begin{array}{ll} \#1: & \text{Chianti} \succ \text{Brunello} \succ \text{Amarone} \\ \#1: & \text{Brunello} \succ \text{Chianti} \succ \text{Amarone} \end{array}$$

By Pareto, Amarone cannot win in R_1. By Neutrality, the other two alternatives either must both win or both lose. So the only possible outcome for R_1 is $\{$Brunello, Chianti$\}$.

Now let us consider the rest of the group (subprofile R_2):

$$\begin{array}{ll} \#1: & \text{Chianti} \succ \text{Brunello} \succ \text{Amarone} \\ \#1: & \text{Brunello} \succ \text{Amarone} \succ \text{Chianti} \\ \#1: & \text{Amarone} \succ \text{Chianti} \succ \text{Brunello} \end{array}$$

R_2 is completely symmetric: if we rename Amarone to Brunello, Brunello to Chianti, and Chianti to Amarone we end up in the exact same profile. So the outcome must be invariant under this permutation as well, meaning that the full set $\{$Amarone, Brunello, Chianti$\}$ is the only option.

Finally, when we join the two subprofiles, Reinforcement forces the desired outcome of $\{$Brunello, Chianti$\}$. $\triangle$

While the general problem of computing justifications is highly intractable [Boixel and de Haan, 2021], for small-scale scenarios such as this, it is possible to automate the process.

3 The Online Demonstration Tool

We have developed an online demonstration tool that allows anyone to compute and explore axiomatic justifications for small-scale decision-making scenarios of their own design. In this section we describe the functionality of this tool and show how it can be applied to the example discussed earlier. The tool is available at the following address:

https://demo.illc.uva.nl/justify/

To build a profile, we first choose names for the alternatives involved in the decision-making scenario, and then define the preferences of the voters over those alternatives (see Figures 1 and 2). Next, we pick the outcome for which we want to find a justification, and finally we specify for each of the axioms available whether we would be happy for that axiom to feature in that justification (see Figure 3). Hovering over an axiom will reveal a short intuitive definition. Pressing the submit-button will launch the justification engine.

If no justification meeting our requirements exists, or if none can be found within the search depth or time limit in place, a message to this effect will be displayed. Otherwise, the justification found will be presented on the screen.

Such a justification consists in a step-by-step explanation for why the target outcome should win, similar to a mathematical proof. Indeed, a justification is internally represented as a proof tree, with each node being a step in the explanation. Most of the steps correspond to an application of an axiom instance that constrains the possible outcomes for either the profile we are interested in or some of its subprofiles. Other steps amount to simple case distinctions.

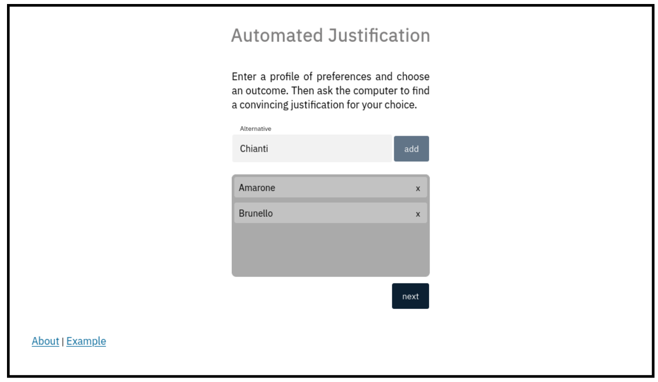

Figure 1: The landing page of the demonstration tool.

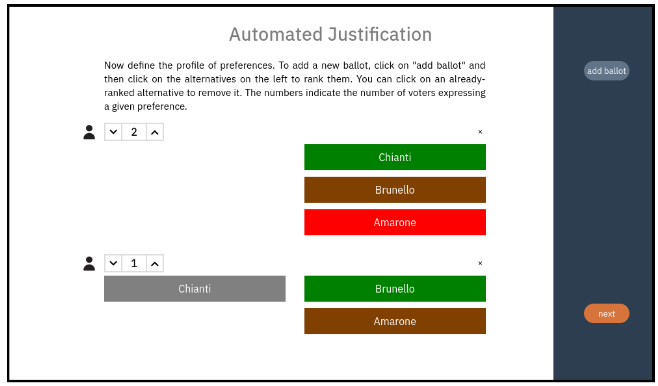

Figure 2: Constructing a preference profile.

[1]There is one further technical requirement: $\mathcal{A}^{\mathrm{N}}$ may not be *trivial* in the sense of there not existing even one voting rule that satisfies $\mathcal{A}^{\mathrm{N}}$. An example for a trivial set of axioms is the set of those involved in Arrow's famous impossibility theorem [Arrow, 1963].

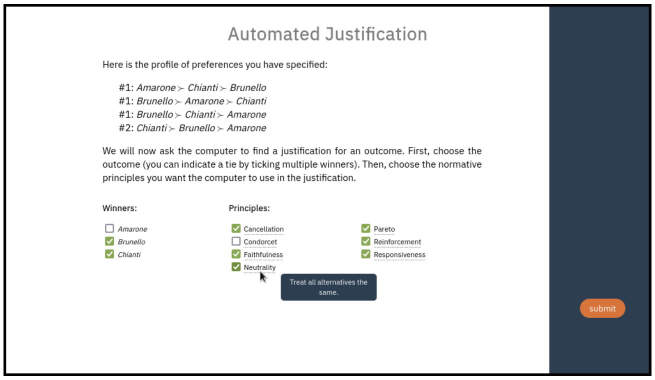

Figure 3: Choosing outcome and normative principles.

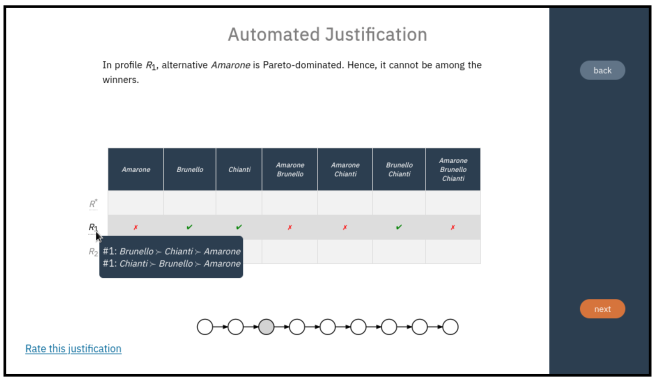

Figure 4: Displaying a step in an explanation.

In our tool, we display one step at a time, describing each step in an intuitive way. Figure 4 shows an example, namely the application of the Pareto Principle to subprofile R_1 to exclude Amarone from the set of winners (as also explained in Example 3). One can easily navigate back and forth between different steps in the explanation. On each page (corresponding to a step), we show a table that records, for every profile mentioned so far, the outcomes that are still available for it. This table is updated at every step, as more and more constraints imposed by the axiom instances over the available outcomes are made explicit. The justification is complete when the only remaining outcome for the profile specified by the user is the outcome she wanted to justify.

When a justification is displayed, the user has the option of rating both the degree to which it is easy to understand and the degree to which it is convincing. We hope the data collected in this manner will prove useful in informing future research on the topic of explainability is social choice.

4 Techniques

Building an application—such as our online demonstration tool—that can provide axiomatic justifications to its users requires the integration of a whole battery of AI techniques, ranging from computational social choice, to automated reasoning, to search. Here we provide a brief overview.

The starting point for the automation of the task of finding justifications is the fundamental insight that—for a fixed set of alternatives and an upper bound on the number of agents—we can rewrite any axiom of interest as a formula of propositional logic with variables of the form $p_{R,x}$, encoding that in profile R alternative x should be part of the outcome. This makes it possible to use SAT solvers [Biere *et al.*, 2009; Ignatiev *et al.*, 2018] to reason about axioms. This insight has been used repeatedly in computational social choice to prove impossibility theorems [Tang and Lin, 2009; Geist and Peters, 2017]. But we can also use it to check whether a given set of axioms $\mathcal{A}^{\mathrm{N}} \subseteq \mathbb{A}$ constitutes a valid normative basis for choosing $X^\star$ in profile $R^\star$. To do so, we simply need to check whether the encoding of $\mathcal{A}^{\mathrm{N}}$ is satisfiable but becomes unsatisfiable once we add a formula saying that the outcome should *not* be equal to $X^\star$.[2]

Owing to the impressive efficiency of modern SAT solvers, these checks can be performed very quickly. The main bottleneck is the generation phase, as the encoding of an axiom will typically be huge. To address this challenge, we have developed a *search algorithm* that constructs the encoding of the set of formulas to be checked in an incremental fashion by exploring a graph on the set of all possible profiles induced by the axioms in the corpus $\mathbb{A}$ [Nardi *et al.*, 2022].

Given an unsatisfiable set of formulas encoding $\mathcal{A}^{\mathrm{N}}$ together with the requirement that $X^\star$ must not be the outcome, any *minimally unsatisfiable subset* (MUS) of that set will contain all the information we need to identify a set of axiom instances involved in some explanation $\mathcal{A}^{\mathrm{E}}$. So we can relegate this task to an *MUS enumeration tool* [Liffiton *et al.*, 2016].

Finally, we have developed a method for turning such an MUS into a structured proof [Boixel *et al.*, 2022], inspired by *tableau-based calculi* from the field of *automated deduction* [D'Agostino *et al.*, 1999]. As there usually are many different such proofs, we used *answer set programming* [Gebser *et al.*, 2012] as a means of selecting one that meets certain optimality criteria, such as being as short as possible.

We implemented our demonstration tool in Python, with an eye on reusability, particularly of the packages providing fundamental reasoning abilities. The code is available here:

https://github.com/comsoc-amsterdam/comsoc/

5 Future Directions

There are a number of directions in which to take this research agenda further. Examples include improving the performance of the justification algorithm, translating our tableau-based explanations into natural language, running experiments with users to improve our understanding of what kind of explanation they experience as most helpful, and extending our approach to other types of decision-making scenarios. Regarding the latter, recent work by Loustalot Knapp [2022] suggests that the general approach also has potential in the area of *matching under preferences* [Manlove, 2013].

References

[Arrieta *et al.*, 2020] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham

[2] It is also possible to use the tools of *constraint programming* [Rossi *et al.*, 2006] to the same effect, and in our initial work on the topic we followed this alternative route [Boixel and Endriss, 2020].

Tabik, Alberto Barbado, Salvador García, Sergio Gil-López, Daniel Molina, Richard Benjamins, Raja Chatila, and Francisco Herrera. Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion*, 58:82–115, 2020.

[Arrow, 1963] Kenneth J. Arrow. *Social Choice and Individual Values*. John Wiley and Sons, 2nd edition, 1963. First edition published in 1951.

[Biere *et al.*, 2009] Armin Biere, Marijn Heule, and Hans van Maaren, editors. *Handbook of Satisfiability*. IOS Press, 2009.

[Boixel and Endriss, 2020] Arthur Boixel and Ulle Endriss. Automated justification of collective decisions via constraint solving. In *Proceedings of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS-2020)*. IFAAMAS, 2020.

[Boixel and de Haan, 2021] Arthur Boixel and Ronald de Haan. On the complexity of finding justifications for collective decisions. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI-2021)*. AAAI Press, 2021.

[Boixel *et al.*, 2022] Arthur Boixel, Ulle Endriss, and Ronald de Haan. A calculus for computing structured justifications for election outcomes. In *Proceedings of the 36th AAAI Conference on Artificial Intelligence (AAAI-2022)*. AAAI Press, 2022.

[Brams and Fishburn, 2002] Steven J. Brams and Peter C. Fishburn. Voting procedures. In Kenneth J. Arrow, Amartya K. Sen, and Kotaro Suzumura, editors, *Handbook of Social Choice and Welfare*, volume 1, chapter 4, pages 173–236. Elsevier, 2002.

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

[Cailloux and Endriss, 2016] Olivier Cailloux and Ulle Endriss. Arguing about voting rules. In *Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS-2016)*. IFAAMAS, 2016.

[D'Agostino *et al.*, 1999] Marcello D'Agostino, Dov M. Gabbay, Reiner Hähnle, and Joachim Posegga, editors. *Handbook of Tableau Methods*. Elsevier, 1999.

[Gebser *et al.*, 2012] Martin Gebser, Roland Kaminski, Benjamin Kaufmann, and Torsten Schaub. *Answer Set Solving in Practice*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2012.

[Geist and Peters, 2017] Christian Geist and Dominik Peters. Computer-aided methods for social choice theory. In Ulle Endriss, editor, *Trends in Computational Social Choice*, chapter 13, pages 249–267. AI Access, 2017.

[Ignatiev *et al.*, 2018] Alexey Ignatiev, Antonio Morgado, and João Marques-Silva. PySAT: A Python toolkit for prototyping with SAT oracles. In *Proceedings of the 21st International Conference on Theory and Applications of Satisfiability Testing (SAT-2018)*. Springer, 2018.

[Liffiton *et al.*, 2016] Mark H. Liffiton, Alessandro Previti, Ammar Malik, and João Marques-Silva. Fast, flexible MUS enumeration. *Constraints*, 21(2):223–250, 2016.

[Loustalot Knapp, 2022] Daniela Loustalot Knapp. Justification of matching outcomes. Master's thesis, ILLC, University of Amsterdam, 2022.

[Manlove, 2013] David Manlove. *Algorithmics of Matching under Preferences*. World Scientific, 2013.

[Miller, 2019] Tim Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artificial intelligence*, 267:1–38, 2019.

[Nardi *et al.*, 2022] Oliviero Nardi, Arthur Boixel, and Ulle Endriss. A graph-based algorithm for the automated justification of collective decisions. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS-2022)*. IFAAMAS, 2022.

[Procaccia, 2019] Ariel D. Procaccia. Axioms should explain solutions. In Jean-François Laslier, Hervé Moulin, M. Remzi Sanver, and William S. Zwicker, editors, *The Future of Economic Design*, pages 195–199. Springer, 2019.

[Rossi *et al.*, 2006] Francesca Rossi, Peter van Beek, and Toby Walsh, editors. *Handbook of Constraint Programming*. Elsevier, 2006.

[Tang and Lin, 2009] Pingzhong Tang and Fangzhen Lin. Computer-aided proofs of Arrow's and other impossibility theorems. *Artificial Intelligence*, 173(11):1041–1053, 2009.

[Zwicker, 2016] William S. Zwicker. Introduction to voting theory. In Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel D. Procaccia, editors, *Handbook of Computational Social Choice*, chapter 2, pages 23–56. Cambridge University Press, 2016.