Displaying Justifications for Collective Decisions

Arthur Boixel\(^1\) and Ulle Endriss\(^1\) and Olivier 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 \(X^* \subseteq X\). When \(X^*\) is a singleton, then we may think of that single element of \(X^*\) as the “best” alternative in \(X\); otherwise, we may think of the elements of \(X^*\) 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:

\[
\begin{align*}
\#2 : & \text{ Chianti} \succ \text{ Brunello} \succ \text{ Amarone} \\
\#1 : & \text{ Brunello} \succ \text{ Amarone} \succ \text{ Chianti} \\
\#1 : & \text{ Brunello} \succ \text{ Chianti} \succ \text{ Amarone} \\
\#1 : & \text{ Amarone} \succ \text{ Chianti} \succ \text{ Brunello}
\end{align*}
\]

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? \(\triangle\)

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 \(R^*\), a target outcome \(X^*\), and a corpus \(\mathcal{A}\) of axioms we may rely on. In its most basic form, a justification for \(X^*\) is simply a reference to a set
\(\mathcal{A}^n \subseteq \mathcal{A}\), a so-called normative basis, such that every voting rule satisfying the axioms in \(\mathcal{A}^n\) will return \(X^*\) for \(R^*\).

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? △

What is still missing from our notion of justification is the explanatory component. So let us refine our definition by requiring that \(\mathcal{A}^n\) must be paired with an explanation \(\mathcal{A}^e\) made up of a set of instances of the axioms in \(\mathcal{A}^n\). Here an instance of an axiom is an application of that axiom to a specific situation (e.g., specific profiles and alternatives). \(\mathcal{A}^e\) must be such that every voting rule that satisfies it will return \(X^*\) for \(R^*\). In addition, we may require that \(\mathcal{A}^e\) 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 \(R_1\)):

\[
\begin{align*}
\#1 & : \text{Chianti} \succ \text{Brunello} \succ \text{Amarone} \\
\#1 & : \text{Brunello} \succ \text{Chianti} \succ \text{Amarone}
\end{align*}
\]

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 \(R_2\)):

\[
\begin{align*}
\#1 & : \text{Chianti} \succ \text{Brunello} \succ \text{Amarone} \\
\#1 & : \text{Brunello} \succ \text{Chianti} \succ \text{Amarone} \\
\#1 & : \text{Amarone} \succ \text{Chianti} \succ \text{Brunello}
\end{align*}
\]

\(R_2\) 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\}. △

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.
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 $\mathcal{A}^N$ is satisfiable but becomes unsatisfiable once we add a formula saying that the outcome should not be equal to $X^*$.

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 $\mathcal{A}$ [Nardi et al., 2022].

Given an unsatisfiable set of formulas encoding $\mathcal{A}^N$ together with the requirement that $X^*$ 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 $\mathcal{A}^E$. So we can relegate this task to an MUS enumeration tool [Liffton 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 Barreto Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham

[Rossi et al., 2006] 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].


