Voting Systems: An Investigation into Strategic Voting and the Commutative Monoidal Structure of Elections

Jeremy Wayland

December 2019

Submitted in partial fulfillment of the requirements for an Honors Degree in Mathematics at the University of California, Berkeley, advised jointly by Professor Wesley Holliday in the Department of Philosophy and Professor Dmitry Vaintrob in the Department of Mathematics.

Acknowledgments

I would first like to thank my primary advisor Professor Holliday for his oversight and inspiration to delve into Social Choice Theory. Also, thanks to Professor Simon for standing in as a proxy advisor allowing me to sidestep some of the inconvenient bureaucracy of the Berkeley Mathematics department. Furthermore, I would like to express my gratitude to Professor Vaintrob for advising my investigations in Category Theory; his willingness to help and extensive knowledge were integral to the second part of my project. Finally, I would like to thank my family and friends for their love and support throughout my degree. I would not have made it through without them.

Preface

This thesis is the culmination of Math 196, a course satisfying one of the requirements for obtaining an honors degree in the Berkeley Mathematics Department. Most students opt for taking two graduate courses and getting sufficiently high marks to obtain honors. After choosing to write a thesis, I quickly realized that not only do a handful of students pursue this opportunity at Cal, but there is little to no structure put in place to support those that do. It is well known that contributing any meaningful research as an undergrad in mathematics is exceptionally difficult. Consequently, (in most cases) it falls on the advisor to design a project of appropriate length and difficulty within their field of research. This is often expository for undergraduates. This was not my experience or goal for this project. I wanted to produce something new and contribute to the field of mathematics. After I selected Professor Holliday as my advisor, he suggested that I look into strategic voting as he believed it bore some "low-hanging fruit". I do not think I would have been inclined to look into this topic had it not been for him, but it has proved to be stimulating and topical. I also had interests in Category Theory that stems from my natural inclination to the more abstract fields of mathematics. During my introduction to category theory during the summer of 2019, I immediately saw parallels to the structure of the election scenarios I was analyzing when considering strategic voting. However, as you will notice from the two entirely seperate papers that make up this project: these two topics have vastly different scopes. The first paper on strategic voting under uncertainty contains a hearty synopsis on prior research in the field; much of the formalism is an extension of past work (including that of Professor Holliday). It does, however, conclude with two very small, low-hanging results that are relevant to real world application. I would like to note that the research required to fully understand the cusp of this field took the better part of 4 months. Needless to say, I realized how subjective low-hanging is, but was proud of how much I learned nonetheless. I feel that this paper should most aptly be seen as a platform for motivating future work. Whilst working on strategic voting under uncertainty with Professor Holliday, I still wanted to pursue an investigation of the categorical structure underlying an election. This is the second paper included in this thesis. Although they both deal with voting, I felt it necessary to present them as two independent papers. The second, titled *The Commutative Monoidal Structure of Elections*, is an original, albeit seemingly unorthodox, application of category theory to voting systems. Although this paper has only one main result, there is promising potential to develop these ideas further, which I hope to do in the future.

Strategic Voting: Safely Manipulating under Uncertainty

Introduction

Our voting system is imperfect. This is widely appreciated in today's day and age, but it is not anthropic corruption alone that skews the purity of electing a leader. Voting methods (the ways we choose our leaders) are inherently susceptible to manipulation, provided they are constrained by a few reasonable assumptions. The Gibbard-Satterthwaite theorem [2] tells us that *any* reasonable voting method is susceptible to *strategic voting*. Reasonable meaning that the method is: 1) resolute (i.e., it selects a single winner), 2) unanimous (i.e., if all voters choose Bernie Sanders over Trump, then Trump cannot win) and finally 3) nondictorial (i.e., there is no single voter that solely dictates the outcome of the vote). Needles to say, these are restrictions that any fair election should adhere to. To vote strategically is to manipulate an election by submitting insincere preferences. This means that voters are not guaranteed their best result by being honest. What a novel thought: situations that decide power can, and often do, favor dishonesty. Even in a formal context, you cannot escape the true nature of politics.

Literature Review

Existence of manipulation is not the whole story; there is a large body of work that addresses the computational complexity of finding manipulations. In most cases, the manipulator is omniscient (i.e., is given full knowledge of everyone's preferences and the method for selecting a winner). The complexity of these situations, as well as algorithms for determining specific manipulations are well understood in many situations ([5],[4],[11],[10],[1]). However, an omniscient voter is far from realistic. It is natural to assume that a manipulator is restricted by a level of uncertainty. In other words, the manipulator's knowledge is consistent with multiple possible systems. For formal analysis, it is customary to assume that the manipulator is sure of his own preferences. Thus, the uncertainty must stem from a lack of knowledge about either the voting rule or the preferences of the other voters. The former has recently been analyzed by Holliday and Pacuit in [8]. They analyze three different manipulation criterion: sure, safe, and expected manipulations. These criteria capture different levels of confidence for the strategic voter. A *sure* manipulation is one where you are certain that submitting an insincere preference will produce an improved result, while a *safe* manipulation is one where you are only certain that your manipulation won't lead to a worse result. An *expected* manipulation occurs when your manipulation is more likely to be beneficial than not. Although they produced results in all three regimes, the most applicable to this work is their results on safe manipulation. Providing multiple possible voting methods is not sufficient to eliminate safe manipulation; however, it can reduce the incentive to manipulate. This does show that introducing even a minimal uncertainty (2 possible voting methods) can be used as a tool to incentivize honest voting. Although this work will focus on situations where you restrict the manipulator's knowledge about the preferences of other voters, there seems to be a definite motivation for future work that melds these two regimes.

One of the first investigations of strategic voting under uncertainty of voters' preferences was carried out by Conitzer, Walsh and Xia [3]. Here they develop a framework for interpreting the uncertainty as a set of possible profiles that extend your current state of knowledge. This leads to a natural epistemic interpretation of these situations as developed in [6]. Conitzer et al. were also interested in the conditions that would eliminate all safe manipulation. From a perspective of civility, their work motivates the search for a minimal set of information that will guarantee the elimination of safe manipulation. From the manipulator's perspective, it begs the question of how much information one needs to ensure that a potentially advantageous manipulation remains safe. They first show that in the extreme case that the manipulator has no information about other voter's preferences, most voting methods (with a sufficient ratio of voters to candidates) will be immune to a dominating manipulation.[1] Especially in the case of positional scoring methods, this result suggests that there is a critical point for each voting method at which the uncertainty set will eliminate safe manipulation. Conitzer et al. show that for the case of the Borda method, which will be defined formally in the next section, that slightly restricting the manipulator's information forces the computation for finding a safe manipulation to be an NP-hard problem. Naturally, these results motivate the study of specific uncertainty sets that are prevalent in everyday life. This investigation was developed by Reijngoud and Endriss in [9] and furthered by Endriss et al. in [7]. The latter analyzes 3 different types of uncertainty sets; the first assumes you know nothing aside from who wins the election. The second, a more informative set, considers that you know the score for each candidate. The last uncertainty set assumes the manipulator knows if the majority of the voting population prefers candidate A over B for each pairwise match up of candidates. The following sections will develop the notion of strategic voting under uncertainty, with an identical goal in mind: characterizing particular uncertainty sets. The notation and terminology used will closely resemble that of [8] and will focus in on the voting methods of Borda, Plurality and k-approval.

Preliminaries

Let V be a nonempty, finite set of n *voters* denoted by $\{1, \cdots, n\}$. Let C be the set of m *candidates* $\{c_1, \cdots, c_m\}$. A ballot then is a full set of preferences submitted by a given

[1]for n voters and m candidates you need $n \geq 6(m-2)$ to eliminate safe manipulations in positional scoring rules [3]

voter; this is nothing more than a linear order on C. Thus, consider the set $\mathcal{L}(C)$ which will denote the set of a all linear orders on C. For $i \in V$, we let $P_i \in \mathcal{L}(C)$ represent the truthful preferences of the i^{th} voter. Thus P_i allows you to compare the ranking of two candidates according to voter i; if i prefers c_j to $c_{j'}$ then $c_j P_i c_{j'}$. This total collection of binary rankings is often referred to as i's ballot. This paper will only consider a single manipulator, $1 \in V$, for simplicity. This will be the only voter submitting an insincere ballot which will be denoted by P_1^*. By considering the preferences of each candidate, you can define a *profile* for the population of voters, given by $\mathbb{P} = \{P_i : i \in V\}$. Throughout this work $\mathbb{P}$ will denote the generic *truthful profile* populated by genuine preferences. A *manipulation* will be represented as a function on a truthful profile that replaces P_1 with P_1^* and applies the identity to $i \in V - \{1\}$. The *manipulated profile* is the image of this function, denoted as $\mathbb{P}_{[P_i \to P_i^*]}$. The outcome of an election is determined by a *voting method*; this will be a function $f : \mathcal{L}(C)^n \to C$ that picks out a single winner.

The voting methods of interest in this paper fall under what are commonly referred to as positional scoring rules. These assign a numerical score to each candidate based on their rank in each P_i; the candidate with the maximum score is selected as the winner. For purposes of simplicity assume that scoring ties are broken lexicographically, so in favor of the candidate with the lowest index as done in [9]. Define a *scoring vector* to be $\langle s_1, s_2, \cdots s_m \rangle$ where for $j = 1, \cdots, m-1$, $s_j \geq s_{j+1}$. Then the the score of $x \in C$ for profile $\mathbb{P}$ is given by $\text{score}(\mathbb{P}, x) = \Sigma_{i=1}^n \text{score}(P_i, x)$, where $\text{score}(P_i, x) = s_r$ and r is the position that x occurs in the linear order P_i. These voting methods can be defined as follows: $\forall \mathbb{P} \in \mathcal{L}(C)^n, f(\mathbb{P}) = \max_{x \in C} \text{score}(\mathbb{P}, x)$. This assumes that this maximum intrinsically accounts for the lexicographic tie breaking rule. Thus you can fully generate a positional scoring rule, given their scoring vector. This yields the following definitions:

- *Plurality*: the rule given by $\langle 1, 0 \cdots, 0 \rangle$;
- *k-approval*: the rule given by $\langle 1, 1 \cdots, 0 \rangle$, where you have k 1's followed by zeros;
- *Borda*: the rule given by $\langle m-1, m-2 \cdots, 0 \rangle$.

Uncertainty Sets and Safe Manipulation

As noted prior, this paper aims to investigate profiles in which we know that a manipulation exists. This allows one to beg the question of how much information the manipulator needs to ensure that their manipulation is safe. *Safe*, in this context, means that insincere preferences will fair just as well, or better, than truthful preferences. It has be shown that omniscience is not always needed to preserve the safety of a manipulation [9]. Thus it is natural to restrict the information available. This is done by defining an uncertainty set as follows: let an *uncertainty set* be a subset $U \subseteq \mathcal{L}(C)^n$ that collects the profiles that agree with the information provided to the manipulator.[2] For example, consider the set generated by $\mathbb{P}$: $\{\mathbb{P}' : P_1 \simeq P_1'\}$. This is an uncertainty set for a manipulator who only

[2]This method of defining an uncertainty set by extending an incomplete knowledge state into a set of possible election scenarios was pioneered by Conitzer in [3]

knows their own ballot; the lack of information is captured by the collection of possible profiles $\mathbb{P}'$ that extend P_1. This will serve as the largest uncertainty set as we will always assume that the manipulator is certain/aware of their true preferences. The most interesting cases of uncertainty for positional scoring rules occur when the manipulator has at least the knowledge of their own preferences and the outcome of the election. This can be generated by an *uncertainty function*, U, such that $U(\mathbb{P}) = \{\mathbb{P}' : f(\mathbb{P}) = f(\mathbb{P}') \text{ and } \mathbb{P}'_1 \simeq \mathbb{P}_1\}$. The image of U allows the manipulator to compare the genuine winner to other candidates that might be selected. This gives a concrete way to assess whether or not an advantageous manipulation remains safe when considered in all profiles of an uncertainty set.

The goal of this paper is to analyze whether or not some realistic examples of uncertainty sets eliminate *safe manipulation*. Let $\preceq$ be the weak order given by the manipulator's true preferences P_1 and the identity. And $\prec$ be the strict order determined by P_1. Let the existence of a safe manipulation by a single manipulator be defined as a property of a voting method f and an uncertainty set $U(\mathbb{P})$. Then to say that $\langle f, U(\mathbb{P})\rangle$ is susceptible to a safe manipulation means that:

$\exists P_1^* \in \mathcal{L}(C)$ such that $\forall \mathbb{P}' \in U'(\mathbb{P})$:
$f(\mathbb{P}) \preceq f(\mathbb{P}'_{[P_1 \to P_1^*]})$ and $\exists \mathbb{P}^*$ such that $f(\mathbb{P}) \prec f(\mathbb{P}^*_{[P_1 \to P_1^*]})$.

This investigation is interested in situations where safe manipulation breaks down. A necessary prerequisite is a profile $\mathbb{P}$ that does in fact witness manipulation by $1 \in V$ via P_1^*. This is equivalent to saying that $\langle f, \mathbb{P}\rangle$ is susceptible to safe manipulation, where the manipulator has full information about the election. However, by considering the multitude of profiles in an uncertainty set, it is possible that P_1^* violates safety. Safety of P_1^* is broken for uncertainty set $U(\mathbb{P})$ when:

$$\exists \mathbb{P}' \in U(\mathbb{P}) : f(\mathbb{P}_{[P_i \to P_1^*]}) \prec f(\mathbb{P}) \tag{1}$$

This means there is a profile in your uncertainty set that returns a strictly less preferable candidate. We say that a manipulation P_1^* of $\mathbb{P}$ is *not safe* for $\langle f, U(\mathbb{P})\rangle$ if (1) holds.

Hopefully this will motivate the search for the following set:

$$U_{\min}(\mathbb{P}) = \bigcap \{U(\mathbb{P}) : \forall P_1^*, P_1^* \text{ is } \textit{not safe} \text{ for } \langle f, U(\mathbb{P})\rangle\}$$

This represents the maximal amount of knowledge a manipulator can have while still rendering each possible manipulation unsafe. Ensuring this knowledge threshold for each voter would reserve insincere ballots for gamblers.[3]

Results

Consider the following uncertainty sets, generated by a true profile $\mathbb{P}$ that witnesses at least one manipulation by voter $1 \in V$:

[3] Those who want to take a chance on a manipulation cannot be stopped; however, in this scenario honesty becomes the safe option which seems ideal.

$U_{\text{Omni}}(\mathbb{P}) = \{\mathbb{P}\}$
$U(\mathbb{P}) = \{\mathbb{P}' : f(\mathbb{P}) = f(\mathbb{P}_i) \text{ and } \mathbb{P}'_1 \simeq \mathbb{P}_1\}$
$U'(\mathbb{P}) = U \cap \{\mathbb{P}' : \forall i \in V, \max(P_i) = \max(P'_i)\}$
$U''(\mathbb{P}) = U \cap \{\mathbb{P}' : \mathbb{P}' = \mathbb{P}_{[P_a \to P'_a]}, P'_a \in \mathcal{L}(C)\}$
$U_{\text{Ig}}(\mathbb{P}) = \{\mathbb{P}' : \mathbb{P}'_1 \simeq \mathbb{P}_1\}$

Let these sets be referred to as the "omniscient", "winning", "first choice", "missing person", and "ignorant" priors, respectively. Arguably, these representations of initial information states are viable in the real world. The most extreme cases are captured by the ignorant and omniscient priors. The next two corollaries characterize safe manipulation for these cases.[4]

Corollary 1. *For $m \geq 3$, there exists a manipulation P^*_1 that is safe for $\langle f, U_{Omni}(\mathbb{P})\rangle$ where f is any of the following methods: Borda, k-approval, or Plurality.*

Proof. See Theorem 1 of [9] for details. ■

Corollary 2. *When $n \geq 2m - 2$, there are no safe manipulations for $\langle f, U_{Ig}(\mathbb{P})\rangle$ where f is any of the following methods: Borda, k-approval, or Plurality.*

Proof. See Theorem 5 of [9] for details. ■

These results indicate that for the three voting methods under investigation, there must be some critical prior that marks the change from *safe* to *unsafe* for any given manipulation. The winning prior has also been well studied, yielding the following result:

Corollary 3. *For $m \geq 3$ and $n \geq 4$ there exists a manipulation P^*_1 that is safe for $\langle f, (\mathbb{P})\rangle$ where f is any of the following methods: Borda or Plurality.*

Proof. See Theorem 3 of [9] for details. ■

Corollary 4. *If $m \geq 3$,$n \geq 4$, and $k \leq m - 2$, then there exists a manipulation P^*_1 that is safe for $\langle f, U(\mathbb{P})\rangle$ where f is k-approval.*

Proof. See Theorem 2 of [7] for details. ■

Lemma 1. *For an uncertainty function $\tilde{U}$ such that $\tilde{U}(\mathbb{P}) \subseteq U(\mathbb{P})$, if there exists a manipulation P^*_1 that is safe for $\langle f, U(\mathbb{P})\rangle$ then P^*_1 is also safe for $\langle f, \tilde{U}(\mathbb{P})\rangle$.*

Proof. This reduces to showing that 1) $\forall \mathbb{P}' \in U' : f(\mathbb{P}) \preceq f(\mathbb{P}'_{[P_1 \to P^*_1]})$ and 2) $\exists \mathbb{P}'' \in \tilde{U}(\mathbb{P}$ such that $f(\mathbb{P}) \prec f(\mathbb{P}''_{[P_1 \to P^*_1]})$. Since $\tilde{U}(\mathbb{P}) \subseteq U(\mathbb{P})$, $\mathbb{P}' \in \tilde{U}(\mathbb{P}) \implies \mathbb{P}' \in U(\mathbb{P})$. Since P^*_1 is safe for $\langle f, U(\mathbb{P})\rangle$, 1) follows immediately. Furthermore, the true profile must be consistent with any uncertainty set. Thus $\mathbb{P} \in \tilde{U}$, and by assumption the true profile $\mathbb{P}$ witnesses a manipulation by $1 \in V$. Thus $f(\mathbb{P}) \prec f(\mathbb{P}_{[P_1 \to P^*_1]})$. This satisfies 2) and completes the proof that P^*_1 must also be safe for $\langle f, \tilde{U}(\mathbb{P})\rangle$. ■

Theorem 1. *When $m \geq 3$ and $n \geq 4$, there exists a manipulation P^*_1 that is safe for $\langle f, U(\mathbb{P})\rangle$ and $\langle f, U''(\mathbb{P})\rangle$, where f is Borda or Plurality.*

[4]These are corollaries derived from the results of Reijgnoud and Endriss.

Proof. Given that $m \geq 3$ and $n \geq 4$, Corollary 3 implies that $\exists P_1^* \in \mathcal{L}(C)$ that is safe for $\langle f, U(\mathbb{P})\rangle$. Given that $U'(\mathbb{P}) \subseteq U(\mathbb{P})$ and $U''(\mathbb{P}) \subseteq U(\mathbb{P})$, Lemma 1 implies that P_1^* is safe for both $\langle f, U(\mathbb{P})\rangle$ and $\langle f, U''(\mathbb{P})\rangle$. ■

Theorem 2. *If $m \geq 3$,$n \geq 4$, and $k \leq m-2$, then there exists a manipulation P_1^* that is safe for $\langle f, U(\mathbb{P})\rangle$ and $\langle f, U''(\mathbb{P})\rangle$ where f is k-approval.*

Proof. Fix f as the k-approval voting method. Given the restrictions on (n, m, k), Corollary 4 implies that $\exists P_1^* \in \mathcal{L}(C)$ that is safe for $\langle f, U(\mathbb{P})\rangle$. Given that $U'(\mathbb{P}) \subseteq U(\mathbb{P})$ and $U''(\mathbb{P}) \subseteq U(\mathbb{P})$, Lemma 1 implies that P_1^* is safe for both $\langle f, U(\mathbb{P})\rangle$ and $\langle f, U''(\mathbb{P})\rangle$. ■

Conclusion

These results show that both the "first choice" and the "missing person" uncertainty sets are in fact susceptible to safe manipulation. This process can be applied to a number of other intuitive uncertainty sets that extend the "winning" prior. This could be very valuable information for both manipulators and committees looking to eliminate safe manipulation. As shown in [3], actually finding this safe manipulation is an NP-hard problem. Most manipulator's will not be able to solve this sufficiently fast to make the right decision. However, is there another way to gather useful information from this system from the standpoint of the manipulator? In larger elections, with many more than 3 candidates and 4 voters, it is possible to come up with a number of different manipulations. Thus it seems natural to inquire about which of these manipulations are safe under particular uncertainty sets. Furthermore, it has been shown that the elimination of safe manipulation is possible for any voting method if you restrict the manipulator's knowledge enough. Thus, it stands to reason by some adaptation of the Intermediate Value Theorem that there is a minimum uncertainty set that will eliminate safe manipulation. As seen from this work in conjunction with [8], there is a definite motivation for combining uncertainty about voters preferences and the voting method to find this set for any election scenario.

References

[1] J. J. Bartholdi, C. A. Tovey, and M. A. Trick. The computational difficulty of manipulating an election. *Social Choice and Welfare*, 6(3):227–241, 1989.

[2] Jean-pierre Beno. The Gibbard–Satterthwaite Theorem: a Simple Proof. *Economics Letters*, 69:319–322, 2000.

[3] Vincent Conitzer, Toby Walsh, and Lirong Xia. Dominating manipulations in voting with partial information. *Proceedings of the National Conference on Artificial Intelligence*, 1:638–643, 2011.

[4] Palash Dey. Resolving the Complexity of Some Fundamental Problems in Computational Social Choice. 012(August), 2016.

[5] Palash Dey, Neeldhara Misra, and Y. Narahari. Complexity of manipulation with partial information in voting. *Theoretical Computer Science*, 726:78–99, 2018.

[6] Hans Van Ditmarsch. Strategic voting and the logic of knowledge. (Extended Abstract). *Aamas*, pages 196–205, 2012.

[7] Ulle Endriss, Svetlana Obraztsova, Maria Polukarov, and Jeffrey S. Rosenschein. Strategic voting with incomplete information. *IJCAI International Joint Conference on Artificial Intelligence*, 2016-Janua:236–242, 2016.

[8] Wesley H. Holliday and Eric Pacuit. Strategic voting under uncertainty about the voting method. *Electronic Proceedings in Theoretical Computer Science, EPTCS*, 297:252–272, 2019.

[9] Annemieke Reijngoud and Ulle Endriss. Voter response to iterated poll information. *11th International Conference on Autonomous Agents and Multiagent Systems 2012, AAMAS 2012: Innovative Applications Track*, 1(Illc):376–383, 2012.

[10] Reyhaneh Reyhani. Strategic Manipulation in Voting Systems. (January 2013), 2013.

[11] Lirong (Duke University) Xia. Computational Voting Theory : Computational Voting Theory: Game-Theoretic and Combinatorial Aspects. pages 1–316, 2011.

The Commutative Monoidal Structure of Elections:

Introduction

This section was motivated by John Baez and his cohort of students at UCR; they have been using Category Theory to study a plethora of interesting structures that permeate mathematics, computer science, biology, chemistry and many other prolific fields. Their team has built a categorical formalism to describe all sorts of networks. Generally, a network is considered as a set with some extra structure. This structure captures the process of transforming from prescribed inputs to prescribed outputs. While exploring the field of strategic voting and attempting to settle on a specific topic, I attended a conference on Quantum Physics and Logic at Chapman University (QPL June 10-14 2019). It was here I was introduced to the idea of structured cospans, which Baez advertised as a categorical framework that could be applied to garner a deeper understanding of any type of network [2]. This seemed abundantly applicable within Social Choice Theory. What is a voting system other than a network with very specific types of inputs and outputs? Voters come in with their preferences on a given issue, and after going through a particular transition phase, they settle on a winner. Although Baez and his colleagues have analyzed many different conceptions of open networks, including linear networks, Markovian processes, Reaction Networks, Petri Nets, and others ([6],[4],[7],[5],[3]), these concepts have yet to be applied to voting. Conveniently, the abstraction of a voting method almost perfectly mirrors that of a Petri Net in [5]. Thus it is my intention to follow the work of Baez as a guideline, to define a the category of Voting Webs that characterize voting systems.[1]

Preliminaries

This project was my first experience with Category Theory, and thus I think it is appropriate for me to start at the very beginning by establishing some preliminary definitions and adopting a consistent notation for the remainder of this paper.

Definition 1. *A category C consists of the following components (1,2) and adhere to the following properties (3-5):*

1. *A class of objects, $Ob(C)$. As done in [Baez Category Theory Notes], if $x \in Ob(C)$ we simply write $x \in C$.*

[1]For a more formal definition on the structure of a voting system, see Section 1 of *Strategic Voting: Safe Manipulations Under Uncertainty* (Wayland 2019).

2. *A set of morphisms between $x, y \in C$ called $Hom_C(x, y)$. As much of Category theory is expressed through diagrams, the morphisms represent the arrows between objects in C. Analogous to group theory operations, there are certain requirements for the morphisms of a Category. As for notation, I'll adopt the convention used in [Baez Category Theory Notes] and write $f \in Hom_C(x, y)$ as $f : x \to y$.*

3. *C must be closed under composition of morphisms; given $f : x \to y$ and $g : y \to z$ then you must also have $g \circ f : x \to z$. Furthermore, this composition must be associative. I.e., $(h \circ g) \circ f = h \circ (g \circ f)$.*

4. *For any $x \in C$, you must have an identity morphism $1_x : x \to x$.*

5. *Finally, the unity laws must hold:(Left Unity) $1_x \circ f = f$ for any $f : y \to x$, (Right Unity) $g \circ 1_x = g$ for any $g : x \to y$.*

Definition 2. *A functor is a mapping between categories $F : C \to C'$ that associates an object in C' to each object in C and a morphism $F(f) : F(a) \to F(b)$ in C' for each morphism $f : a \to b$ in C such that:*

- $F(1_x) = 1_{F(x)}$;
- $F(g \circ f) = F(g) \circ F(f)$ *for all morphisms f, g in C.*

The categories that will be relevant to this paper are as follows:

a) **Set**, the category with sets as objects and functions between sets as morphisms and the subcategory of finite sets **FinSet**.

b) **CommMon**, the category of commutative monoids and monoid homomorphims and **CMC**, the category with commutative monoidal categories as objects and strict monoidal functors as morphisms.

c) **POSet**, the category of partial orderings on sets; the objects are posets and the morphisms are order preserving functions.

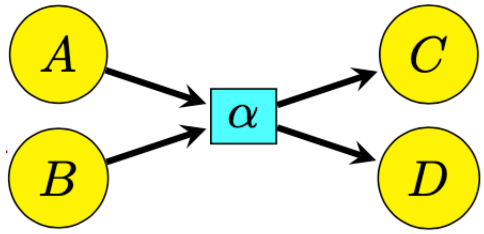
d) **Cat**, the category of "small" categories whose morphisms are functors.[2]

Naturally, I cannot possibly define all the categorical machinery working behind the scenes that allow for the construction of Petri Nets and Voting webs. However, I will attempt to be as comprehensive as possible with the qualities that are directly relevant to capturing the behavior of a voting system.

Petri Nets

What is a Petri net? Here is a simple example as shown in [5].

[2]Here "small" means you the categories must have a set of objects



Effectively they are simple computational modeling tools that appear ubiquitously in the areas of computer science, biology, and chemistry. It was first suggested by Meseguer and Montanari in 1990 that Petri nets have a commutative monoidal structure[3]. They were interested in capturing the behaviour of computation in concurrent and compositional systems, which lead them to associate Petri Nets with strict symmetric monoidal categories (as defined in [8]). They also added an additional *distributivity property* that captured their notion of concurrent computations [9]. Although credit for this initial conception must be given to Meseguer and Montanari, Baez and Master take a more streamlined approach in [5] that is more appropriate for the scale of this paper: they asscociate Petri nets with *commutative monoidal categories* that inherently captures this distributivity property. I will adopt their methodology:

Definition 3. *A commutative monoidal category is a strict monoidal category* $(C, \otimes, I)$ *such that for all* $x, y \in C$ *and morphisms f,g:*

$$a \otimes b = b \otimes a \text{ and } f \otimes g = g \otimes f$$

.

This takes a strict monoidal category and forces the symmetry isomorphisms $\sigma_{a,b}$: $a \otimes b \tilde{\rightarrow} b \otimes a$ to be identity morphisms [5]. This allows for the definition of **CMC**, the category whose objects are commutative monoidal categories and whose morphisms are strict monoidal functors.[4]

Section 2 of Baez and Master develops a pipeline for taking an interpretation of a Petri net P to the free commutative monoidal category on P and element of **CMC**. The collection of these Petri Categories which they call **PetriCat**, is a subcategory of **CMC**. They take a very similar approach as Meseguer and Montanari but with a more restrictive interpretation of the morphisms allowed in a Petri Net.

Voting Webs

This section will implement further restrictions to Petri nets that capture a) the implementation of a social welfare function to an aggregation of voters preferences and b) the

[3]The distinguishing feature is that the commutative monoid of objects is free

[4]Every monoidal functor between commutative monoidal categories is automatically a strict symmetric monoidal functor, which makes this a well defined subcategory.

execution of a computation that decides a set of winners according to a given voting rule. First I will define a categorical representation of a voting system (**VotingWeb**), from which you can (freely) generate a commutative monoidal category (**VotingWebCat**). Then I show an equivalence of **VotingWeb** and **VotingWebCat**, proving that this conception of voting systems has a commutative monoidal structure.

There are many different possible subtleties (or lack thereof) one could take to design a more specific (or more general) category that captures this computation. However, the key realization is that regardless of fine details, they should come to represent a commutative monoidal category. What is the motivation for this? It is nearly identical to the initial interest of Meseguer and Montanari. We should be able to use the monoidal structure to run two voting systems in parallel (via disjoint union) and combine populations of voters to produce an aggregated preference on a common issue. Furthermore, we should compose results of simple voting systems to create complex multistage elections that can span all combinations of population subsets and voting rules (ponder the presidential election in the United States for a minute if you don't think this is necessary).

Taking from the notation in my previous paper, consider a voting system with $|C| = n$ and $|V| = m$ under voting rule f. We can envision the system as a collection of voters coming together with packaged information (a total order on candidates) that then undergoes a transition phase (votes are counted) and returns a winning candidate. In the case of positional scoring rules, you are really transforming m strict linear orders on C, into a single (potentially non-strict) linear order on C that represents the rankings of the candidates. The similarity to a common Petri net is undeniable.

First, we need to get a handle on the our domain and codomain of a voting web computation and frame this categorically. As stated above, we are in fact transitioning between a subset of total orders on C to a singleton partial order. Thus the domain, called the set of transitions, needs to span all election possibilities. Call this domain T_{vote}. Each element of T_{vote} can be generated by the following triple: $\langle C, m, f \rangle$ for $C \in \mathbf{FinSet}, m \in \mathbb{N}$, f is a voting method. Thus $\tau \in T_{\text{vote}}$ should package together a collection of preorders that represent the preferences of m voters on the set of candidates C and a voting method. These preorders can be generated by the following forgetful functor $G : \mathbf{POset} \to \mathbf{Set}$. Consider the preimage $G^{-1}(C)$; this is a discrete subcategory of POset where the objects are all possible preorderings on the set C with only identity morphisms.

Definition 4. *Let $\mathbb{O} = G \circ G^{-1}$. This is a functor*

$$\mathbb{O} : \boldsymbol{Set} \to \boldsymbol{Set}$$

and $\mathbb{O}[C]$ represents the set of all preorders on a set C.

Baez and Master use a monad defined by the composition $\mathbb{N} = K \circ J : \mathbf{Set} \to \mathbf{Set}$, where K,J give an adjunction between **CommMon** and **Set**. Their domain of interest is $\mathbb{N}[C]$ which captures finite linear combinations of C with natural number coefficients; clearly $\mathbb{O}[C]$ can be coded as a (strict) subset of $\mathbb{N}[C]$. Thus, a Voting Web can be defined as two functions going from T_{vote} to $\mathbb{O}[C]$, in analogous fashion to a Petri Net in [5].

Definition 5. *A Voting web is a specialized Petri Net that is defined by a source and a target function that have the following form:*

$$T_{\text{vote}} \underset{t}{\overset{s}{\rightrightarrows}} \mathbb{O}[C]$$

These functions interact with a *fixed* set of transitions, T_{vote}. The functions s, t are also subject to the following restrictions:

- $|\text{Image}(s)| = n$ for $n \in \mathbb{N}$. So for a system with n voters the source function picks out precisely n preorders.
- $|\text{Image}(s)| = 1$. So the target function picks out a single preorder on C that represents the outcome of the election according to a well defined voting method f.[5]

This will be a very small subset of the possible source and target functions that appear in the context of generalized Petri Nets. However, considering the specific action of computing the outcome of an election (even with manipulation), this is to be expected. Since we are considering a fixed transition set, let $i : T_{\text{vote}} \to T_{\text{vote}}$ be the identity map on the set of voting methods. This allows for the following defintion:

Definition 6. *A **Voting Web morphism** from a petri net $V = (s, t : T_{vote} \to \mathbb{O}[C])$ to $V' = (s', t' : T_{vote} \to \mathbb{O}[C'])$ is a function $g : \mathbb{O}[C] \to \mathbb{O}[C']$ such that the diagrams below commute:*

$$\begin{tikzcd} T_{vote} \arrow[r, "s"] \arrow[d, "i"] & \mathbb{O}[C] \arrow[d, "\mathbb{O}(g)"] \\ T_{vote} \arrow[r, "s'"] & \mathbb{O}[C'] \end{tikzcd} \qquad \begin{tikzcd} T_{vote} \arrow[r, "t"] \arrow[d, "i"] & \mathbb{O}[C] \arrow[d, "\mathbb{O}(g)"] \\ T_{vote} \arrow[r, "t'"] & \mathbb{O}[C'] \end{tikzcd}$$

These last two definitions outline the behaviour of the category of voting webs, **Voting-Web**. As a restriction of Petri Nets, we know there is an underlying commutative monoidal structure. You can construct a commutative monoidal category $\hat{F}V$ from a Voting Web $V = (s, t : T_{\text{vote}} \to \mathbb{O}[C])$ as follows: let the objects of the category, $\text{Ob}(\hat{F}V)$, be defined as the free commutative monoid on C. For morphisms, consider the following recursive definition from [5]:

- for every $\tau \in T_{\text{vote}}$, include a morphism $s(\tau) \to t(\tau)$
- for any object a, include the morphism $1_a : a \to a$
- for morphisms $f : a \to b$ and $g : a' \to b'$, include a morphism to represent their tensor product $f + g : a + a' \to b + b'$
- for morphisms $f : a \to b$ and $g : b \to c$ include a morphism to represent their composition $g \circ f : a \to c$.

[5]Note that this supports single or multiple winners and that the prescription for f is contained in T_{vote}.

This recursive definition is more than exhaustive in acquiring all necessary morphisms. In order to obtain the commutative monoid $\mathrm{Mor}(\hat{F}V)$, take the set of morphisms described above and quotient out by an equivalence relation that imposes the laws of a commutative monoidal category [5].

Definition 7. *Let $\hat{F}$: **VotingWeb** $\rightarrow$ **CMC** be the functor as reflected below:*

$$\begin{tikzcd} T_{vote} \arrow[r, shift left, "s"] \arrow[r, shift right, "t"'] \arrow[d, "i"] & \mathbb{O}[C] \arrow[d, "\mathbb{O}(g)"] & \mapsto & \hat{F}V \arrow[d, "\hat{F}(g)"] \\ T_{vote} \arrow[r, shift left, "s'"] \arrow[r, shift right, "t'"'] & \mathbb{O}[C'] & & \hat{F}V' \end{tikzcd}$$

Here $\hat{F}(g) : \hat{F}V \rightarrow \hat{F}V'$ is defined on objects by $\mathbb{O}(g)$. On morphisms, $\hat{F}(g)$ is just the identity map. This functor is not necessarily a left adjoint. As done in [5], consider the corestriction of $\hat{F}$ to its image.

Definition 8. *Let **VotingWebCat** be the image of $\hat{F}$.*

This is the collection of commutative monoidal categories picked out by the functor $\hat{F}$. Each object is itself a **VotingWeb** and the morphisms are strict monoidal functors of the form $\hat{F}(g) : \hat{F}V \rightarrow \hat{F}V'$ where V, V' are related by a morphism g.

Theorem 1. *Consider the following functor:*

$$F : \textbf{\textit{VotingWeb}} \rightarrow \textbf{\textit{VotingWebCat}}$$

which is defined as the corestriction of $\hat{F}$. This is a left adjoint, showing an equivalence between the two categories.

Proof. This proof uses a condition for functoriality that is outlined in [1]: a functor is part of an equivalence of categories that is faithful, full, and essentially surjective. F is essentially surjective and full by construction [5]. Thus all that remains to be shown is that F is faithful (i.e., the restriction is injective). If $\hat{F}(g) : \hat{F}V \rightarrow \hat{F}V'$ and $F(g') : \hat{F}V \rightarrow \hat{F}V'$ are the same functor then clearly it follows that $\mathbb{O}(g) = \mathbb{O}(g')$. This implies that $g = g'$, which means that they represent the same morphism in **VotingWeb**. ∎

Conclusion:

This last theorem shows that the underlying structure of a voting system is a commutative monoid. This result barely scratches the surface on what has been shown by Baez and Master in [5] regarding Petri Nets. They develop even more structure by considering framing the Petri Nets as symmetric monoidal double categories, with the goal of designing a syntax for describing open systems of this nature with "reachability as a choice of semantics". Arguably, this is possible for Voting Webs as well, which could provide a syntax for describing the most fundamental question in voting systems: reachability. I hope to develop this formalism further in a subsequent work.

References

[1] John C. Baez. Category Theory Course. pages 1–40, 2018.

[2] John C. Baez. STRUCTURED COSPANS John C. Baez. pages 1–47, 2019.

[3] John C. Baez and Brendan Fong. A Compositional Framework for Passive Linear Networks. pages 1–54, 2015.

[4] John C. Baez, Brendan Fong, and Blake S. Pollard. A compositional framework for Markov processes. *Journal of Mathematical Physics*, 57(3):1–43, 2016.

[5] John C. Baez and Jade Master. Open Petri Nets. 2018.

[6] Brendan Fong. The Algebra of Open and Interconnected Systems. 2016.

[7] Brendan Fong and Maru Sarazola. A recipe for black box functors. pages 1–34, 2018.

[8] Saunders Mac Lane. Categories for the Working Mathematician, 1971.

[9] José Meseguer and Ugo Montanari. Petri nets are monoids. *Information and Computation*, 88(2):105–155, 1990.