Pizza Democracy: A Set-Based Voting Mechanism

Matthew Chartier and Olga Zinoveva

December 2011

Introduction

We consider the problem for a group of agents of selecting toppings on a shared set of pizzas, or an 'order'. This is a relatively common problem which is also surprisingly complex. The number of possible options for an order of pizzas grows exponentially both in the number of pizzas we order, the number of toppings which are available to choose from, and in the maximum number of toppings we can get on a pizza. Then selecting an order by considering all possible combinations of pizzas seems impractical.

We can simplify the process of selecting an order by having agents provide a ranking over the toppings and using them to implicitly determine how each agent would rank any possible pizza or order of pizzas. We could potentially do this by using some scoring function to determine the score each agent assigns to each pizza in an order and then defining the score each agent assigns to an order as the sum of the scores he assigns to each pizza in the order.

The question remains as to how to use these scores to determine which pizza to select. The simplest approach might be to sum the scores each agent assigns to each order to determine the score for that order and choose the order with the highest score. This is a utilitarian approach which maximizes the aggregate value which the agents see in the toppings on the pizza. This approach isn't entirely satisfying because it isn't necessarily fair; a simple majority of agents could dictate the toppings on every pizza in an order. Since each agent is partaking of the pizza (and perhaps helping pay for it), it seems reasonable that we would want the order to appeal to each of the agents. We can take an egalitarian approach by scoring an order based on how much the least satisfied agent likes it, and defining how much an agent likes an order as how much that agent likes their favorite pizza in that order.

We also note that there are toppings with which agents may be exceedingly dissatisfied, to the point that an agent places no value on a pizza which has that topping. For example, an agent who absolutely hates pineapple or anchovies is unlikely to care that a pizza also has bacon on it, even if bacon is his favorite topping. Then in addition to ranking the preferences which they like, we'd also like agents to be able to indicate some number of toppings that they hate, so that we do not incorrectly assign utility to pizzas for which agents have no use.

Noting also that a large number of toppings may be available and that requiring a full ranking imposes a cognitive burden on agents which is unlikely to be justified by the value of the additional information, we support partial preference ordering in order to decrease the amount of information required of each agent.

Related Work

Some components of our mechanism, in particular the scoring function, are based on existing ideas. In [1], Lu and Boutilier describe the concept of 'regret,' which measures the difference between the societal utility of a selected outcome and the optimal outcome. In determining the 'max regret' of an alternative, the authors offer a method by which they calculate the score for an alternative based on the worst-case completion of a partial ranking of alternatives. We employ a similar worst-case scoring of an alternative to calculate the minimum utility for an alternative for every agent. Furthermore, using the authors' idea of finding the minimum max regret for a set of alternatives, we also maximize the lowest possible utility across all agents to establish a lower bound on the worst-case happiness of the least-happy agent.

We also make use of the claims put forth in [2] by Kalech et al. to motivate the use of partial profiles. In particular, the authors claim that, although in the worst case full preference profiles are necessary in order to determine an optimal alternative, fewer than $n - 1$ votes are needed to find this optimal alternative in most cases. This justifies our decision to elicit partial preferences in order to ease the cognitive burden on agents since we can potentially do as well without eliciting full preference rankings.

We considered the discussion of the potential manipulability of the veto rule in Walsh [3]. While the author empirically demonstrated that under certain conditions the veto rule is manipulable, we believe that its use in conjunction with ranked preferences in our mechanism is necessary for expressing the notion of unacceptable alternatives.

Mechanism

Our voting mechanism may be thought of as a hybrid of the Borda and Veto position scoring mechanisms with support for partial preference profiles. A point worth noting in considering our mechanism is that we do not ask agents for a ranking on the actual alternatives between which we are choosing (the pizza orders), but rather for a ranking on the toppings which we then use to calculate rankings for each agent on the set of possible pizzas and by extension the possible orders.

We have a set of agents A and a set of possible toppings T. We also have constant constraints t, the maximum number of number of toppings which we want to get on any pizza, and n, the number of pizzas which we would like to order. We also have P, the set of all possible pizzas with at most t distinct toppings from T, and O, the set of all possible orders containing n pizzas in P. We also have constants k and v which define the information we elicit from each agent. We request that each agent rank their top-k toppings and also provide up to v toppings which they would like to veto.

Given a set of partial votes V over the set of toppings T, we describe how to assign scores for each agent to each pizza in P, and finally how to assign scores for each agent to each order in O.

To score a pizza for an agent, we consider the set of toppings on the pizza. If the pizza contains a topping which the agent has vetoed, then the score for the pizza is 0. Otherwise, the score is the total of the Borda scores for each topping contained on the pizza. That is, if a topping has rank r (where the top-ranked topping has rank 1), then the topping contributes $|T| - r$ to the pizza's score. Given toppings on a pizza which an agent has not ranked, we calculate the score that the agent assigns a pizza assuming a worst-case completion of the agent's vote.

To score an order for an agent, we consider the set of pizzas in the order, and assign as the score for the order the maximum score the agent assigns to any pizza in the order. So the agent is assumed to be as satisfied with the order as he is with his favorite pizza in the order.

Metropolis Sampling and the Mallows Model

In order to establish some empirical results about the quality of our protocol, we needed to generate a large number of preference profiles for testing. While we could draw voter preferences from a uniform distribution, we would have to accept the impartial culture assumption, which is the idea that all preference orderings are equally likely. A more realistic way to model preferences might be the *Mallows ϕ-model* [1]. For any preference ordering r, we define the probability mass function of the Mallows model as:

$$P(r; \phi, \sigma) = \frac{1}{Z} \phi^{-d(r,\sigma)}$$

Here, σ is the reference ranking – i.e. an assumed "true" ordering where all other orderings are seen as deviations from it.

$\phi \in (0,1]$ is a dispersion parameter that controls how diverse the modeled pool of preferences is, where as $\phi \to 0$, preference orderings that are closer to σ become more likely, and the distribution becomes denser around σ.

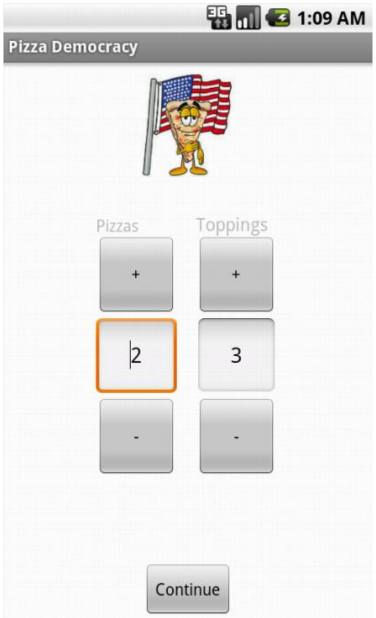
d is a function of distance, which can be defined differently for different variants of the Mallows model, and is in our case the *Kendal-*τ distance. This distance is calculated by counting the number of adjacent "swaps" required to get from one preference ordering to another. For example, $d(abcd, acdb) = 2$ because two swaps (b with c and b with d) are necessary to make the orderings identical.

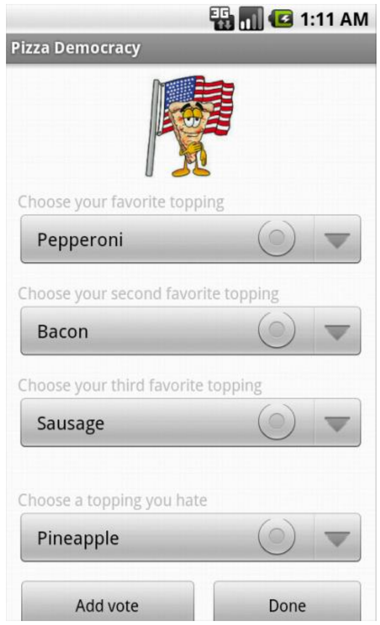
Finally, Z is the normalizing constant. Because it is intractable to calculate Z for the Mallows model, we cannot sample from the distribution directly. Instead, we use the Metropolis algorithm to estimate the probability distribution[4][5]. On each step, we generate a proposal sample r' by swapping two random elements of the current ordering r. We then calculate $d(r, \sigma)$ and $d(r', \sigma)$. If $d(r', \sigma) \leq d(r, \sigma)$, we move to r' with probability 1. Otherwise, we move with r' with probability $\phi^{d(r',\sigma)-d(r,\sigma)}$ and stay at r with probability $1 - \phi^{d(r',\sigma)-d(r,\sigma)}$. We allow 500 steps for the chain to converge to the Mallows distribution, and keep all samples generated thereafter.

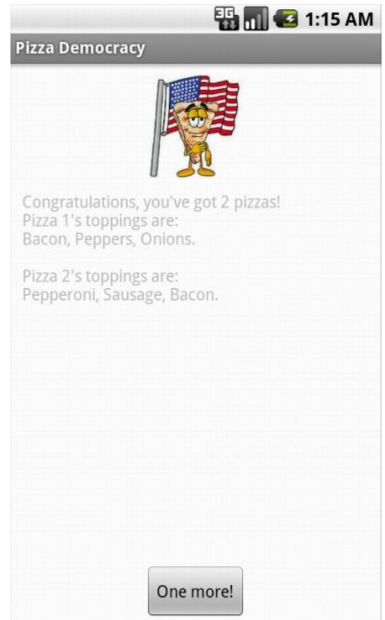
Android Implementation

We implemented our mechanism in an application for the Android mobile phone platform. In "Pizza Democracy", users are prompted to specify some number of pizzas and some maximum number of toppings. The app also elicits from each voter a ranking of his or her top 3 favorite toppings, as well as a topping which he or she hates and would like to veto.

The interface is designed in such a way that agents are able to input their own preferences on one screen containing dropdown elements allowing them to specify their top 3 ranked toppings and a topping to veto. An agent can then touch a button to submit their vote and pass the device to another agent to submit his or her preferences until all preferences have been recorded, at which point an approximation of an optimal order is calculated.







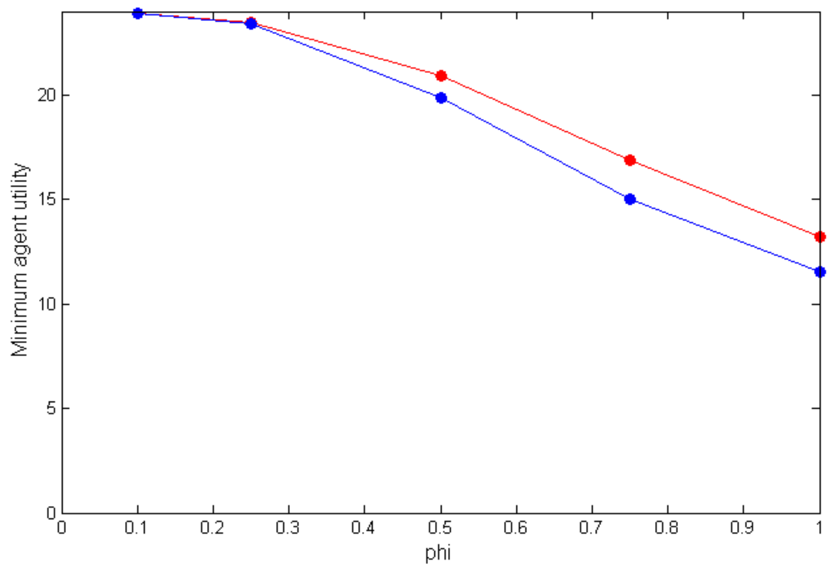
The app's simple interface demonstrates how easily agents can supply partial preferences. It's very simple, both cognitively and mechanically, to come up with the few toppings that sound best and also indicate which topping seems worst. Compared to a full ranking over the entire set of toppings, this is much easier for the user. In our experimental results, we'll show that this ease of use comes at a relatively small cost in terms of approximation quality.

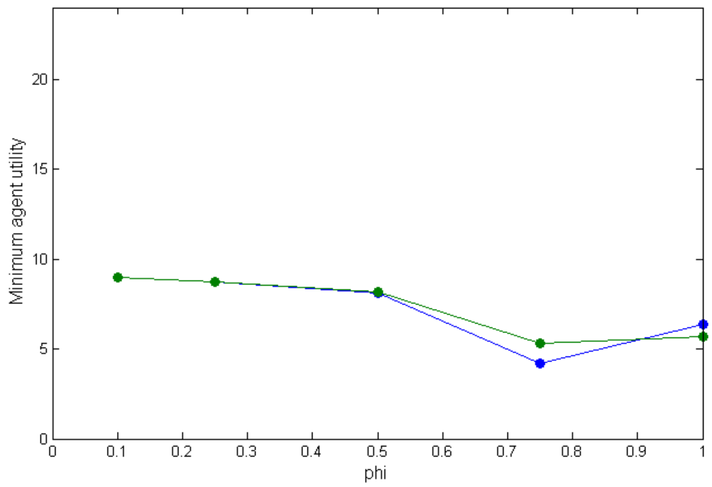
Experimental Results

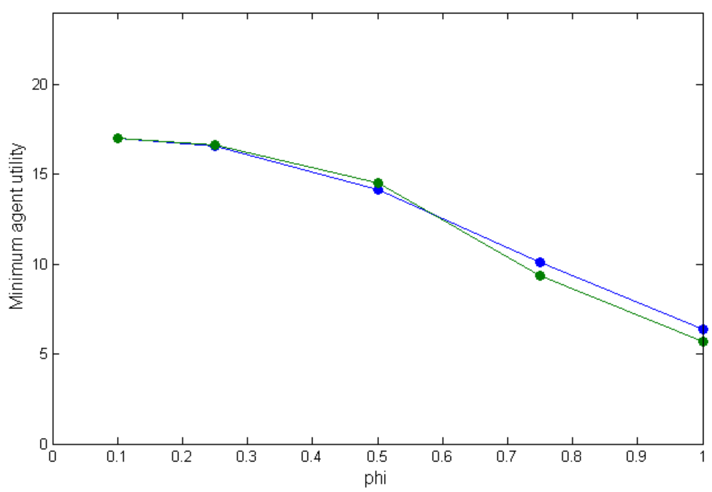
By generating sample votes using the Mallows model, we were able to create random profiles and analyze the relative performance of our mechanism incorporating both ranking and veto with partial profiles and the more traditional Borda mechanism. In all comparisons, the two mechanisms are given the same amount of partial information, though our mechanism is provided with the three top-ranked toppings and the one bottom-ranked topping and the Borda mechanism is provided with the four top-ranked toppings.

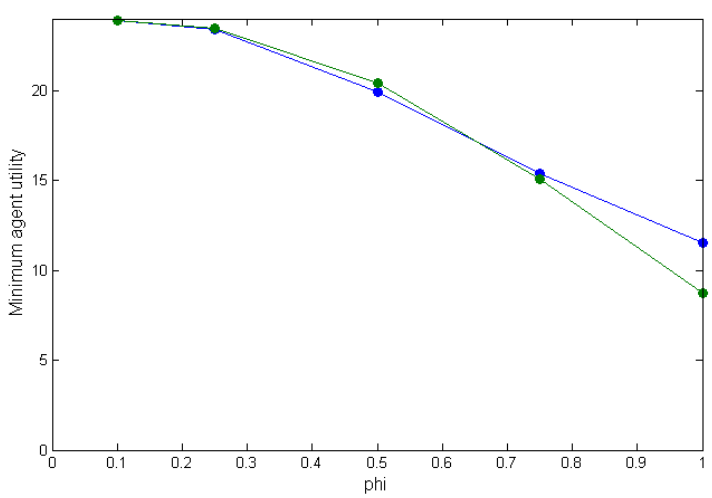
Each data point in each graph is calculated based on 100 unique trials involving 100 randomly generated votes.

In the following graph, we see a comparison between the minimum agent utility guaranteed by our mechanism using both partial (blue) and full (red) profiles. As must be expected, we perform strictly better given more information about the actual preferences of the agents, but the difference is relatively slight even as the variation in the vote distribution becomes large. This is reassuring because it suggests that the loss in approximation quality incurred by dealing with partial preferences is not overly severe, and arguably worthwhile.









In each of the three graphs on the left, we plot the relationship between the average minimum agent utility as phi increases of the order selected by both the partial profile with top-3 preferences and 1 veto (blue) and the partial profile with top-4 preferences (green).

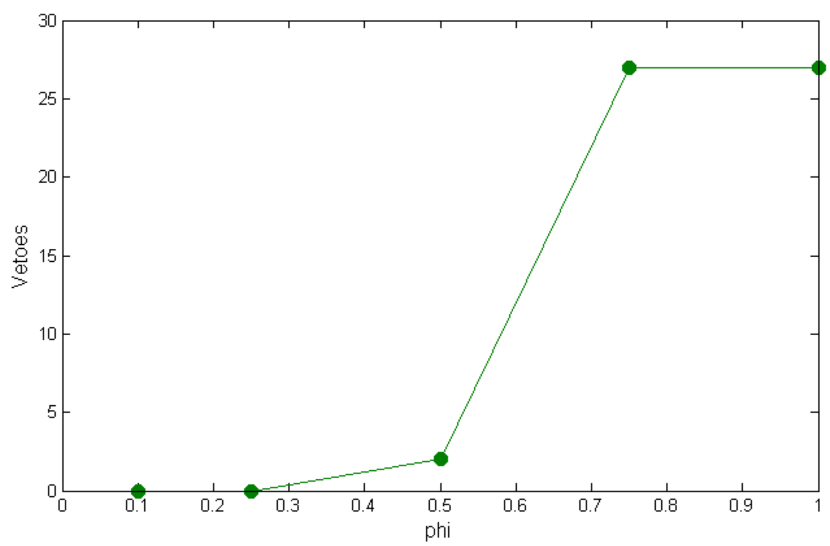
Each of the graphs represents data collected for scenarios where agents were selecting 2 pizzas. The graphs represent, from top to bottom, the 1-topping case, the 2-topping case, and the 3-topping case.

We observe that the two tend to perform quite similarly with low variation in the vote distribution, and that the profile with veto tends to pull ahead as variation increases, and that the magnitude of this difference seems to increase with the number of toppings.

Furthermore, very naturally,

the performance for both mechanisms increases with the number of toppings, which makes sense because agents rank multiple toppings, so they derive utility from more than one topping, and therefore can be a lot better off when more toppings are selected.

If we accept the idea that agents derive no utility from pizzas which contain toppings which they hate, then the following graph provides justification as to why the pure Borda scoring rule without veto is insufficient to ensure that all agents are satisfied to some degree. We show the percentage of orders selected using the Borda scoring rule on partial profiles which are vetoed by at least one agent for different degrees of variation in the vote distribution. Since the pure Borda scoring rule does not account for vetoed toppings which can ruin a pizza for an agent, it is unable to consistently avoid selecting an order containing no pizzas of any value to some agent, particularly at ϕ values above 0.5.



Manipulability of the Proposed Mechanism

An important consideration in the design of any voting mechanism is its susceptibility to manipulation. In particular, since we concern ourselves primarily with ensuring fairness among agents, we would like to ensure that the fairness of the outcome is not compromised by one or more malicious agents.

For ranking among preferred toppings, similar incentives for manipulation in our mechanism exist as do in Borda. If an agent feels that his true favorite topping is likely already very popular among other agents and will likely be included in a pizza that he will like, then an agent might choose to rank his second or third-favorite topping higher in an attempt to boost the likelihood that this topping will appear on a pizza.

This type of manipulation, however, is complicated by the nature of how orders are selected given voter preferences. Since an order is selected so as to maximize minimum satisfaction across all agents, and satisfaction of an agent is defined by how satisfied they are with their least favorite pizza, the effect of artificially inflating the perceived utility of a topping is more difficult to predict.

A more straightforward manipulation strategy would be for an agent to veto toppings which are not necessarily his or her least favorite but which he believes other agents are likely to vote for and which he or she would rather see replace by toppings he or she prefers more. A more devious agent might veto a topping which he really likes but also believe other agents like, thus affording him greater influence on the toppings selected for another pizza, while still remaining confident that the topping he liked but vetoed will still be included on some other pizzas in the order. The danger here is that an agent is then unable to veto any toppings which he legitimately does not care for, which could result in a bad outcome for the agent.

Conclusions and Future Work

The empirical results for our mechanism are positive. Though when comparing average scores of orders selected based on partial profiles with top-3 preferences and a veto with those selected based on top-4 preferences without veto we did not universally improve the average order score, this approach substantially decreased the likelihood of selecting an order which provided no value to some agent. The inclusion of additional information about top preference seems to slightly improve the quality of orders in the case where it happens that no agent vetoes the entire order, this slight tradeoff can easily be justified in the name of fairness as it results in a much lower frequency of agents being completely dissatisfied with the outcome. Our mechanism seems likely to improve expected results when agents are sufficiently dissatisfied with particular toppings such as to derive no utility from pizzas which include them.

In continuing to improve our mechanism, we will explore ways to optimize the number of preferences and vetoes we elicit from agents to optimally balance approximation quality with the cognitive burden imposed on agents, conditioning on the number of voters, the number of pizzas being ordered, the maximum number of toppings on each pizza, and the expected variation in the vote distribution. We will also further explore incentives for vote manipulation in the context of our mechanism and possible methods to resist such manipulation.

References

[1] Lu, Tyler; Boutilier, Craig. “Vote Elicitation with Probabilistic Preference Models: Empirical Estimation and Cost Tradeoffs.” ADT 2011

[2] Kalech, Meir; Kraus, Sarit; Kaminka, Gal A.; Goldman, C.V. “Practical Voting Rules with Partial Information.” Autonomous Multi-Agent Systems (2010)

[3] Walsh, Toby. “Where are the really hard manipulation problems? The phase transition in manipulating the veto rule.” IJCAI 2009

[4] Diaconis, P.; Saloff-Coste, L. “What Do We Know about the Metropolis Algorithm?” Journal of Computer and System Sciences (1998)

[5] Steyvers, Mark. “Markov Chain Monte Carlo: Metropolis-Hastings Sampling.” Computational Statistics with Matlab (book) (2011)