

Making Group Decisions from Natural Language-Based Preferences

Farhad Mohsin, Lei Luo, Wufei Ma, Inwon Kang
Zhibing Zhao, Ao Liu, Rohit Vaish, Lirong Xia

Abstract

Traditional approaches for group decision-making—in particular, *voting*—typically assume that agents explicitly submit well-defined ordinal preferences (e.g., rankings) over the alternatives. However, in real-world settings such as online discussion forums, preferences are often expressed implicitly in *natural language*. We propose a framework for making group decisions from natural language-based preferences. Our approach combines ideas from random utility theory and social choice by first learning a general random utility model for each agent, and subsequently using a voting rule to aggregate individual preferences to make a group decision. To circumvent the computational intractability associated with the latter step, we show that fractional voting outcomes, which are often easy to compute, are identical to those under the randomized modeling with high probability. Preliminary experiments on the efficacy of the framework were conducted on a newly-collected dataset.

1 Introduction

Suppose a family is deciding which college their child should attend after receiving ten offers. The family members may have different preferences over the colleges: for example, the child likes college *A*'s atmosphere; the dad prefers college *B* because of its low cost; and the mom prefers college *C* because it is close to home. How should the decision be made?

This example illustrates the problem of group decision making, which is a central topic in the field of *social choice*. One of the most popular approaches is *voting*—agents are asked to submit their preferences over the alternative to the center, who then applies a *voting rule* to make a group decision for the agents.

The classical paradigm of voting works well in group decision making scenarios with few alternatives and low frequency, in particular presidential elections. However, it faces two major challenges in modern, large-scale, and more-frequent group decision making scenarios: the *preference bottleneck* [8] and the *computational bottleneck* [9].

The preference bottleneck arises when the agents' preferences are too complicated to be efficiently communicated. This often happens when the number of alternatives is large but can be also a common phenomenon when the preferences are uncertain or even unknown to the agents themselves. A typical approach to addressing preference bottleneck is by conducting *preference elicitation* [7, 11, 41, 24], where interactive questions are computed to efficiently elicit agents' preferential information that are sufficient for making a group decision.

The computational bottleneck arises when it is computationally hard to compute the outcomes of voting rules based on the preferential information provided by the agents. This typically happens when the number of alternatives is larger than a few under certain voting rules such as Kemeny [12], or can happen under many voting rule when agents' preferences are compactly represented [20]. Various algorithms and computational theories were developed to address the computational bottleneck, as summarized in the Handbook of Computational Social Choice [9].

In this paper, we aim at providing an immersive solution to the preference bottleneck and the computational bottleneck by leveraging the power of AI. Take the college-choice example for instance, we want to build an intelligent system that silently learns the family members’ preferences from their conversations in natural language (without explicitly querying them), and then compute a group decision based on the learned preferences. The key questions we want to answer in this paper is:

How can we learn agent preferences from natural language to make a group decision?

While the idea of aggregating sentiments expressed in natural language is not new [14, 15], successfully addressing the problem appears to be technically challenging to tackle, despite many previous work on sentiment analysis and stance detection [21, 26] and many voice-enabled systems, including the business group decision-making system developed by IBM [10]. In particular, we are not aware of a dataset on agents’ collective preferences (not simply their sentiment), and we are not aware of a previous technical work that explicitly combines preference learning from natural language and group decision making.

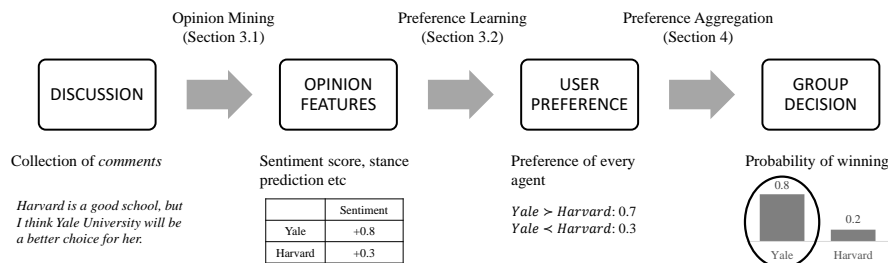


Figure 1: The proposed framework.

Our Contributions

Our primary **conceptual contribution** is a group decision-making framework illustrated in Figure 1. We focus on natural language-based preferences in this paper and leave its extensions to multi-modal inputs as future work. The framework consists of three stages. In the *opinion mining* stage, features that indicate preferences, such as sentiment scores, are extracted from natural language. In the *preference learning* stage, we use the features extracted in the first stage to train a machine learning model to predict agents’ preferences. Finally, in the *preference aggregation*, voting rules are applied to agents’ predicted preferences to make a group decision.

Our **theoretical contributions** lie in the preference aggregation stage (Section 5). We first look at the problem in a fractional preference setting and show that we can aggregate fractional preferences in polynomial time for a large number of voting rules, including positional scoring rules, weighted majority graph-based rules and multi-round voting rules to compute a single winner. Then, we consider a randomized setting, showing that the fractional profile winner also has a high conditional probability of being the winner in the randomized setting.

For **experimental analysis**, we created a new natural language dataset of discussions about preferences over colleges from the College Confidential¹ forum. The discussions involve two or three alternatives (colleges) per thread. We use Amazon Mechanical Turk to crowd-label agents’ preferences in the discussions. The dataset currently has 53 different discussions of varying lengths. We apply our framework on the dataset to predict group decisions to get high training accuracy but moderate testing accuracy (53 ~ 61% test

¹<https://talk.collegeconfidential.com/>

accuracy for group-decision prediction for three-alternative cases for multiple voting rules). This indicates that the problem of predicting group decisions from natural language can be challenging given low amount of data and using direct NLP features like sentiment values.

To evaluate our aggregation results more than just three alternatives, we experiment with the preference data from the Moral Machine (MM) experiment [3], which consists of agents’ preferences over moral dilemmas. Our results indicate that with high probability the fractional profile winner will also be the randomized profile winner, showing effectiveness of the aggregation method.

2 Related Work and Discussions

Sentiment analysis [21, 4, 2] is one of the most popular NLP tools used, and it is very relevant to the problem of learning preferences. Since we are interested about preferences between alternatives, target-specific sentiment analysis is of more concern [5, 37]. At the moment, there are online tools that complete the task of entity-specific sentiment analysis as well. However, sentiment alone can not help model preferences. Another interesting task is stance detection [26, 34, 27, 34], which identifies whether an agent is in favor of, or against a target. Work has also been done in collaborative filtering literature, where additional text features like sentiment have been considered [13, 29, 19]. While the goal is different, this aspect of the work has similar motivation as this work. Sentiment and opinion features are typically considered for single pieces of text. While simple voting based ideas have been proposed to get from sentiment analysis to preference aggregation [14, 15], our work considers more general preference models and expands on that concept.

There are several NLP datasets built for closely related tasks, such as, [25, 40] for predicting approval rating from text; [27, 34] for stance detection. [1] has a large amount of debate data but more are about abstract topics rather than specific alternatives. The UGI corpus [6] focuses on group decision-making, but the discussions express explicit rankings instead of discussing the alternatives.

Random utility models (RUM) [36] are a family of models for modeling probabilistic preferences. We show results using two special cases of general RUMs, the Plackett-Luce model (PL) [30, 22] with features, and Thurstone’s Case V model (TV) [35] with features. PL models are in particular popular for easy-to-compute probabilities for rankings and comparisons. General RUMs subsume the model with bilinear features used in [42] and is similar to use in modeling dyad ranking [17, 32, 33]. See [39] for an exposition to different random utility models and algorithms for learning them.

Aggregation over distribution over preferences, which we also consider, was studied in different extents in [31, 28, 23]. Hazon et al. [16] discusses a dynamic programming algorithm to aggregate arbitrary distribution over preferences, which can be combined with our model but ends up being computationally expensive. Zhao et al. [42] introduce randomized voting rules which require similar computation as this paper. However, instead of assigning winning probabilities proportional to score, we predict a single winner under different voting rules and provide theoretical results about this result. We cover a larger umbrella of voting rules in this paper compared to most other papers and present general results for all anonymous voting rules.

3 Preliminaries

Fractional Preference Profiles. Assume we have n agents and set of alternatives $\mathcal{A} = \{a_1, \dots, a_m\}$. Let $\mathcal{L}(\mathcal{A})$ be the set of all rankings over \mathcal{A} . For a particular ranking σ , if alternative a_j is ranked i -th in σ , $\sigma(j) = i$; e.g. for the highest ranked alternative a_j in σ ,

$\sigma(j) = 1$. Instead of a single preferred ranking, we may say each agent has some partial preference for each ranking. So, we define a *fractional preference profile* $\pi \in [0, 1]^{m!}$ such that $\pi(\sigma)$ is the *fractional preference* for a particular ranking $\sigma \in \mathcal{L}(\mathcal{A})$. If we have fractional preference profiles π_1, \dots, π_n for each agent, we may interpret $\pi_i(\sigma)$ in two different ways. In a probabilistic setting, we may assume that $\pi_i(\sigma)$ is the probability that agent i 's preferred ranking is σ . Or that agent i gives fractional vote of weight $\pi_i(\sigma)$ to the ranking σ .

Random Utility Models (RUM). A RUM assumes an associated utility distribution for each alternative. Given utilities u_1, \dots, u_m for the alternatives in \mathcal{A} , the probability of a particular ranking $\Pr[a_{j_1} \succ \dots \succ a_{j_m}] = \Pr[u_{j_1} > \dots > u_{j_m}]$.

The Plackett-Luce (PL) model [30, 22] and Thurstone's Case V (TV) model are two of the most popular random utility models, where PL uses Gumbel as the distributions of all utilities [22] while TV uses Gaussian distributions. PL and TV are formally defined below.

Definition 3.1 (Plackett-Luce model (PL)). The parameter space is $\Theta = \{\vec{\theta} = \{\theta^j | 1 \leq j \leq m\}\}$. The sample space is $\mathcal{L}(\mathcal{A})^n$. Given a parameter $\vec{\theta} \in \Theta$, the probability of any full ranking $\sigma = a_{j_1} \succ a_{j_2} \succ \dots \succ a_{j_m}$ is $\Pr_{\text{PL}}(\sigma | \vec{\theta}) = \prod_{p=1}^{m-1} \frac{\exp(\theta^{j_p})}{\sum_{q=p}^m \exp(\theta^{j_q})}$.

Definition 3.2 (Thurstone's Case V Model (TV)). The parameter space is $\mathcal{M} = \{\vec{\mu} = \{\mu^j | 1 \leq j \leq m\}\}$. The sample space is $\mathcal{L}(\mathcal{A})^n$. Given a parameter $\vec{\mu} \in \mathcal{M}$, the utility for any alternative a_j , $u_j \sim \mathcal{N}(\mu^j, \frac{1}{2})$, and the probability of any full ranking $\sigma = a_{j_1} \succ a_{j_2} \succ \dots \succ a_{j_m}$ is $\Pr_{\text{TV}}(\sigma | \vec{\mu}) = \Pr[u_{i_1} > \dots > u_{i_m}]$.

Probabilities for PL model are more tractable for computation. Next, we define k -mixture of PL models (k -PL in short), a generalization of PL.

Definition 3.3 (k -mixture of PL models (k -PL)). Given $k \in \mathbb{N}$, the k -mixture Plackett-Luce model is defined as follows. The sample space is $\mathcal{L}(\mathcal{A})^n$. The parameter space has two parts. The first part is the mixing coefficients $\vec{\alpha} = (\alpha_1, \dots, \alpha_k)$, where for all $1 \leq \kappa \leq k$, $\alpha_\kappa \geq 0$ and $\sum_{\kappa=1}^k \alpha_\kappa = 1$. The second part is $(\vec{\theta}_{(1)}, \dots, \vec{\theta}_{(k)})$, where $\vec{\theta}_{(\kappa)} \in \Theta$ is the parameter of the κ -th Plackett-Luce component. The probability of any ranking σ is $\Pr_{k\text{-PL}}(\sigma | \vec{\theta}) = \sum_{\kappa=1}^k \alpha_\kappa \Pr_{\text{PL}}(\sigma | \vec{\theta}_{(\kappa)})$.

In short, the probability of k -PL is a convex combination of k PL models. For agent a_i , we use $\vec{\theta}_{i,(\kappa)}$ (or $\alpha_{\kappa,i}$) to denote the κ -th PL component (or the κ -th mixing coefficient).

For general RUMs, let $X_{ij} \in \mathbb{R}^d$ denote the feature vector of alternative a_j for the i -th agent. Then the perceived utilities for agent i are given by the following process— $u_{ij} \sim \Pr(\cdot | X_{ij}, B)$. Here B is the parameter of the general RUM.

We define k -mixture of Plackett-Luce model (k -PL-X) with features as follows.

Definition 3.4 (k -mixture of Plackett-Luce model with features (k -PL-X)). Given any agent i , each alternative a_j is characterized by a d -dimensional feature vector X_{ij} . The parameter space has two parts. The first part is the mixing coefficients $\vec{\alpha} = (\alpha_1, \dots, \alpha_k)$, where for all $1 \leq \kappa \leq k$, $\alpha_\kappa \geq 0$ and $\sum_{\kappa=1}^k \alpha_\kappa = 1$. The second part is $(\vec{\beta}_{(1)}, \dots, \vec{\beta}_{(k)})$, where $\vec{\beta}_{(\kappa)} \in \Theta$ is the parameter of the κ -th Plackett-Luce component. The sample space is $\mathcal{L}(\mathcal{A})^n$. Given a parameter $\vec{\beta} \in \Theta$, the probability of any ranking $\sigma_i = a_{j_1} \succ a_{j_2} \succ \dots \succ a_{j_m}$ given by agent i is

$$\Pr_{k\text{-PL-X}}(\sigma_i | \vec{\alpha}, \vec{\beta}) = \sum_{\kappa=1}^k \alpha_\kappa \cdot \prod_{p=1}^{m-1} \frac{\exp(\vec{\beta}_{(\kappa)} \cdot X_{ij_p})}{\sum_{q=p}^m \exp(\vec{\beta}_{(\kappa)} \cdot X_{ij_q})}$$

k -PL-X can be treated as a convex combination of k PL models with features. Similarly k -TV mixture models and models with features can also be defined for Thurstone's Case V models. These definitions and more properties of RUMs are discussed in Appendix A.

The distribution over rankings given by PL and k -PL model enables us to compute various fractional preferences efficiently. Here, we present the distribution of k -PL model and the distribution for PL can be treated as a special case of $k = 1$. For a given k -PL parameters $\vec{\alpha} = (\alpha_1, \dots, \alpha_k)$, $\vec{\theta} = (\vec{\theta}_{(1)}, \dots, \vec{\theta}_{(k)})$ and any ranking σ , we have $\pi(\sigma) = \Pr_{k\text{-PL}}(\sigma | \vec{\alpha}, \vec{\theta})$. The probability of many partial rankings (*i.e.*, $a_1 \succ$ others) can also be computed efficiently. For example, let $\pi^{(j)}$ denote the probability of a_j being ranked highest. We have $\pi^{(j)} = \sum_{\sigma(j)=1} \pi(\sigma) = \sum_{\kappa=1}^k \alpha_\kappa \cdot \frac{\exp(\theta_{(\kappa)}^j)}{\sum_{\ell=1}^m \exp(\theta_{(\kappa)}^\ell)}$. Similarly, let $\pi^{(a_j \succ a_\ell)}$ denote the probability of $a_j \succ a_\ell$. Then we have $\pi^{(a_j \succ a_\ell)} = \sum_{\sigma(j) < \sigma(\ell)} \pi(\sigma) = \sum_{\kappa=1}^k \alpha_\kappa \cdot \frac{\exp(\theta_{(\kappa)}^j)}{\exp(\theta_{(\kappa)}^j) + \exp(\theta_{(\kappa)}^\ell)}$.

Similarly, for given k -TV parameters $\vec{\alpha} = (\alpha_1, \dots, \alpha_k)$, $\vec{\mu} = (\vec{\mu}_{(1)}, \dots, \vec{\mu}_{(k)})$, we will have $\pi^{(a_j \succ a_\ell)} = \sum_{\kappa=1}^k \alpha_\kappa \cdot \Phi\left(\mu_{(\kappa)}^j - \mu_{(\kappa)}^\ell\right)$, where Φ is the standard normal CDF. Even though the probability for full rankings are computationally expensive to compute for TV model, probabilities of pairwise preferences can still be calculated efficiently.

Voting Rules. A voting rule $r : \mathcal{L}(\mathcal{A})^n \mapsto \mathcal{A}$ is a function that maps a preference profile (n ranking over alternatives in \mathcal{A}) to a single winner. For *anonymous* voting rules, where the agent identity does not matter, the definition of voting rule may easily be extended to provide a winner from a fractional preference as $r : \mathbb{R}^{m!} \mapsto \mathcal{A}$. For example, a positional scoring rule is characterized by a score vector $\vec{s} = \{s_1, \dots, s_m\}$ such that $s_1 \geq \dots \geq s_m$ and $s_1 > s_m$. And for each ranking σ , the alternative a_j gets a score of $s_{\sigma(j)}$. For fractional preference profile π , an alternative a_j would get additive score $\pi(\sigma) \times s_{\sigma(j)}$ for each $\sigma \in \mathcal{L}(\mathcal{A})$ and the alternative with maximum total score shall be the winner. Some popular scoring rules are: Plurality, whose scoring vector is $\{1, 0, \dots, 0\}$; Borda, whose scoring vector is $\{m-1, \dots, 1, 0\}$; approval voting with top- ℓ approval, whose scoring vector is $\{1, \dots, 1, 0, \dots, 0\}$.
(ℓ 1's)

Definition 3.5 (Weighted Majority Graph (WMG)). Given any fractional profile π , WMG is a directed graph where the nodes are all the alternatives, there exists edges in both directions for each pair of alternatives and weight for edge (a_j, a_ℓ) is $\pi^{(a_j \succ a_\ell)}$.

$\pi^{(a_j \succ a_\ell)}$ is defined as before as the fractional preference for all rankings with $a_j \succ a_\ell$. Voting rules that depend on pairwise preferences only, such as Copeland and maximin, can be considered *WMG based voting rules*, because once the WMG is constructed, it is easy to compute the winner, generally in polynomial time for m .

When r is Copeland, the winner is the alternative beating most other alternatives in pairwise elections. Formally, define pairwise indicator

$$w_r(a_j, a_\ell, F) = \begin{cases} 1 & , \sum_{i=1}^n \pi_i^{(a_j \succ a_\ell)} > \sum_{i=1}^n \pi_i^{(a_\ell \succ a_j)} \\ 0 & \text{otherwise} \end{cases}$$

Then, the Copeland winner maximizes Copeland score $\text{score}_r(a_j, F) = \sum_{\ell \neq j} w_r(a_j, a_\ell, F)$. The *maximin* winner maximizes the *maximin score* $\text{score}_r(a_j, F) = \min_{\ell \neq j} \sum_{i=1}^n \pi_i^{(a_j \succ a_\ell)} - \pi_i^{(a_\ell \succ a_j)}$. It is a well-known fact that *Borda winner* can also be computed using WMG.

Multi-round Voting Rules. There are several multi-round voting rules, mostly all of which focus on eliminating alternatives repeatedly on some criteria until an alternative has definite majority. For example, Single Transferable Voting (STV) has $m-1$ rounds. In each round, the alternative with least plurality score is eliminated. And in the next round, all rankings where the eliminated alternative was preferred would be counted for the next ranked alternative.

Comment Text	Sentiment Score		Preference
	H	Y	
1. We thought that Harvard (H) was a much <i>more prestigious</i> school than Yale (Y).	+0.7	+0.1	$H \succ Y$
2. My daughter <i>chose to go</i> to Harvard (H) <i>over</i> Yale (Y) but she was happy with both offers.	0.0	0.0	$H \succ Y$

Table 1: Examples of opinion features in text

4 The Framework

In this section we formally propose the framework that learns and aggregates agents’ preferences based on natural language interactions, which consists of three parts: Opinion mining (Section 4.1), preference learning (Section 4.2), and preference aggregation (Section 5).

We view the framework mostly as a conceptual contribution, and while we implement an instance of this framework, it was not our goal to find a state-of-the-art solution to the problem, and our solution can definitely be improved upon. The main technical contributions are the theoretical guarantees in the third components, which will be presented in the next section.

4.1 Opinion mining

A discussion is an array of individual comments, where each comment is associated with an agent i . Each agent can have one or more comments associated with them. We assume that we know beforehand the set of alternatives, \mathcal{A} , that are being discussed. With the comments and the set of alternatives as input, output of the opinion mining section will be individual NLP based features for each alternative for each comment.

For example, a common NLP feature associated with preferences is sentiment. A positive sentiment expressed towards an alternative can mean preference for that alternative. So, in our implementation, using state-of-the-art tools for target-specific sentiment extractions (such as Google’s Natural Language API ² and Watson Natural Language Understanding API ³), we extract sentiment values for specific alternatives. See Table 1 for some examples of this. For each comment by agent i , the sentiment scores and magnitude expressed toward alternative j are part of the opinion features.

In addition to extracted sentiment values, direct language features like n-grams and word-embeddings can also be considered as features for the learning problem.

4.2 Preference Learning

For each comment, we have NLP features representing the comment, and labels regarding to the preferences expressed. The goal of the preference learning problem would be to learn a model that can predict preferences over the alternatives. So, given a learned model, we can get a distribution over preferences over alternatives.

For each agent i and alternative a_j , we can have a feature vector X_{ij} consisting of NLP features, agent features and alternative features. We consider all features we can extract from the Opinion Mining stage. Additionally, we may use features of the agents and alternatives themselves in the form— $\vec{u}_i \otimes \vec{v}_j$, where \otimes denotes the Kronecker product, \vec{u}_i is the feature vector of agent u_i , and \vec{v}_j is the feature vector of a_j .

²<https://cloud.google.com/natural-language/>

³<https://www.ibm.com/cloud/watson-natural-language-understanding>

In our implementation, we learn two types of models based on this data. First, considering pairwise preferences between alternatives, we can get a pair of alternative features and a preference label. For a pair of alternatives, A, B , possible labels can be $A \succ B, B \succ A, A = B$, No preference. So, we get a multiclass classification problem. We considered traditional classification algorithms like random forests, SVM, logistic regression, shallow neural nets for our implementation.

Secondly, we can also learn random utility models with features. As we shall see in Section 5, this allows us to aggregate distributions over preferences in a meaningful way. For example, Plackett-Luce model with features (PL-X) are used to model the preference profile in our datasets. See [39] for various methods to learn general RUMs. We use maximum likelihood estimation (MLE) to learn the Plackett-Luce parameters β : $\vec{\beta}^* = \operatorname{argmax}_{\vec{\beta}} \sum_i \ln \Pr_{\text{PL-X}}(\sigma_i | \vec{\beta})$

4.3 Preference aggregation

Based on the learned model, given a new set of alternatives with feature vectors $\{X_{ij}\}$, we can get specific predictions for preferences or distributions over preferences. To get a group decision, we need to aggregate these predictions/distributions, and we can apply various voting rules there.

In case we have specific prediction for pairwise preferences, any weighted-majority-graph based voting rule can be used to aggregate the preference predictions. For distributional preferences it becomes computationally intractable, which leads to our proposed solutions and theoretical results. In particular, we consider the case where distribution over preferences are compactly represented using random utility models and we want to learn the output of a specific voting rule. We present theoretical result for some voting rules under this scenario in Section 5. In our implementation, given the learned PL-X parameters, β , we get different PL parameters, $\vec{\theta}_j$ for each agent over the alternatives in the test scenarios. We then aggregate the distributions represented by these models to get a group decision.

We should mention that the preference aggregation part of the framework would also work for non-NLP problems as well. For example, if agents made pairwise comparisons for sample scenarios from a large set of alternatives, we can still learn a model to express preferences and then apply the aggregation results. An example of this would be learning preference models from pairwise data in the Moral Machine dataset, and in a new (test) scenario, predicting the group decision based on the learned models.

5 Efficient preference aggregation of distributional preferences

Based on the learned models, given a new discussion, we can predict distributions over preferences for each agent, represented compactly using random utility models. In this section, we consider the problem of aggregating these distribution over preferences to get a group decision using voting rules.

5.1 Aggregating Fractional Preferences

Assume that we have n agents and agent i has k -PL (or k -TV) mixing coefficients $\vec{\alpha} = (\alpha_1, \dots, \alpha_k)$ and parameters $(\vec{\theta}_{i(1)}, \dots, \vec{\theta}_{i(k)})$ or $(\vec{\mu}_{i(1)}, \dots, \vec{\mu}_{i(k)})$.

$\vec{\alpha}$ and $\vec{\theta}_{i(\kappa)}$'s can then be used to compute the fractional preference profile π_i for agent i . As mentioned in Section 3, getting full fractional profiles for TV model can be computation-

ally expensive, but we can make use of them, when we are considering pairwise preferences only.

Let $\Pi = \{\pi_1, \dots, \pi_m\}$ be the set of all fractional profiles. The total preference for a particular ranking σ is just the sum of fractional preference for all agents. We define this as the *sum-fractional profile* $F(\sigma) = \sum_{i=1}^n \pi_i(\sigma)$. So $F(\Pi) = \{F(\sigma)\}_{\sigma \in \mathcal{L}(\mathcal{A})} \in \mathbb{R}^{m!}$ will be a vector representing the total fractional preference among all agents for each ranking. In general, when clear from context we will just use F to mean $F(\Pi)$. All voting rules introduced in this paper can be extended to fractional profiles. For a voting rule r , we say $r(F)$ is the fractional profile winner (FP-winner) for Π . Numeric examples of voting rules with fractional profiles are given in the Appendix.

The following proposition can be considered a simple extension of a result of [42].

Proposition 5.1. *Given m alternatives and n agents, where agent i has k -PL mixing coefficients $\alpha_1, \dots, \alpha_k$ and parameters $(\vec{\theta}_{i(1)}, \dots, \vec{\theta}_{i(k)})$, FP-winner $r(F)$ can be computed in $\mathcal{O}(kmn)$ time when r is plurality.*

Positional scoring rules other than plurality can be generally hard to compute. Fortunately, there are scoring rules such as approval voting where only the top- ℓ ranked alternatives get any score. For such cases, we have the following general theorem (for which Proposition 5.1 is a special case with $\ell = 1$).

Theorem 5.2. *Given m alternatives and n agents, where agent i has k -PL mixing coefficients $\alpha_1, \dots, \alpha_k$ and parameters $(\vec{\theta}_{i(1)}, \dots, \vec{\theta}_{i(k)})$, for scoring rules that have score vectors with $\ell < m$ non-zero scores, FP-winner can be computed in $\mathcal{O}(k\ell m^\ell n)$ time.*

Now, we turn to WMG-based voting rules. Once the WMG is constructed, both Copeland and maximin winners can be computed in $\mathcal{O}(m^2)$ time. The following lemma asserts that the WMG can be computed in polynomial time from agents' RUM parameters, if the models are k -PL or k -TV.

Lemma 5.3. *Given m alternatives and n agents, where agent i has k -PL (or k -TV) mixing coefficients $\alpha_1, \dots, \alpha_k$ and parameters $(\vec{\theta}_{i(1)}, \dots, \vec{\theta}_{i(k)})$ (Or $(\vec{\mu}_{i(1)}, \dots, \vec{\mu}_{i(k)})$), the WMG of F can be computed in $\mathcal{O}(km^2n)$ time.*

In light of Lemma 5.3, the FP-winner of the WMG-based rules discussed in this paper can be computed in polynomial time.

Theorem 5.4. *Given m alternatives and n agents, where agent i has k -PL (or k -TV) mixing coefficients $\alpha_1, \dots, \alpha_k$ and parameters $(\vec{\theta}_{i(1)}, \dots, \vec{\theta}_{i(k)})$ (Or $(\vec{\mu}_{i(1)}, \dots, \vec{\mu}_{i(k)})$), FP-winner for Borda, Copeland, and maximin can be computed in polynomial time.*

In most multi-round voting rules, a simple voting rule like plurality is applied in repeatedly. The main challenge is recomputing the score in each round as score received by the eliminated alternatives is then distributed among remaining ones. The next Theorem states that for a wide range of multi-round rules this can be done efficiently.

Theorem 5.5. *For a multi-round voting rule r , if FP-winner for voting rule applied in each round can be computed in polynomial time, then the multi-round FP-winner $r(F)$ can also be computed in polynomial time.*

For example, Theorem 5.5 and Proposition 5.1 together implies that STV FP-winner can be computed in polynomial time, and we get the following corollary.

Corollary 5.6. *Given m alternatives and n agents, where agent i has k -PL mixing coefficients $\alpha_1, \dots, \alpha_k$ and parameters $(\vec{\theta}_{i(1)}, \dots, \vec{\theta}_{i(k)})$, FP-winner for STV can be computed in $\mathcal{O}(km^2n)$ time.*

5.2 Aggregating Randomized Preferences

So far, we have been treating $\pi_i(\sigma)$ as a fractional vote by agent i for ranking σ . But we can also consider the problem in a randomized setting, in which we assume that $v_i(\sigma)$ is the indicator random variable (RV) defining whether agent i 's preference is σ . We define the total number of agents with preference σ as $H(\sigma)$ and $H(\Pi) = \{H(\sigma)\}_{\sigma \in \mathcal{L}(\mathcal{A})}$. $H(\Pi) \in \mathbb{R}^{m!}$ is an $m!$ -dimensional RV, denoting the total number votes in favor of each rankings.

Lemma 5.7. *Given set of fractional preferences Π , for all $\sigma \in \mathcal{L}(\mathcal{A})$, $\mathbb{E}(H(\sigma)) = F(\sigma)$*

Just like $F, F(\Pi)$, we use H instead of $H(\Pi)$ when it is clear from context. Now, since H is an RV, so the winner under the randomized setting would also be a random variable. We call this $r(H)$, the randomized-profile winner (RP-winner), and we would get a probability for each alternative to be the winner. Let us define the winning probability for alternative a_j as $p_j = \Pr[r(H) = a_j]$.

Since we can compute $\pi_i(\sigma)$ for all σ and all $i \leq n$, it is possible use a dynamic programming algorithm in [16] to exactly compute p_j 's. However, this algorithm would not be computationally tractable. So we approximately compute probability p_j for $j \leq m$ with Algorithm 1. The sampling complexity for this problem is expressed in Theorem 5.8.

Algorithm 1: Monte Carlo voting

```

1 Input: Distribution of preferences
    $PL_i(\vec{\theta}_i)$ , voting rule  $r$ ;
2 Output:  $\tilde{p}_j$  for each  $a_j$ ;
3 Initialization:  $c_j = 0$  for each  $j \leq m$ ;
4 for  $t = 1, \dots, T$  do
5   | Sample a linear order for each
   |   agent  $j$ ;
6   | Compute the winner  $a_j$  using
   |   voting rule  $r$ ;
7   |  $c_j \leftarrow c_j + 1$ ;
8 end
9 Compute  $\tilde{p}_j = \Pr[r(H) = a_j] = \frac{c_j}{T}$ ;

```

Theorem 5.8. *Given any $\epsilon, \delta > 0$, if $p = \langle p_1, \dots, p_m \rangle$ are true winning probabilities and \tilde{p} is the estimate from Algorithm 1, we need $\mathcal{O}(\frac{m^2}{\epsilon^2} \ln(\frac{2m}{\delta}))$ samples so that $TVD(p, \tilde{p}) \leq \epsilon$ with probability $1 - \delta$ where $TVD(p, \tilde{p})$ is the total variation distance between the two distributions.*

How Often is Fractional-Profile Winner also the Randomized-Profile Winner? For a group decision-making setting, computing the probability of winning for each alternative is not always necessary. Finding the alternative with highest probability of winning can be a much higher priority. Since we have an efficient aggregation method to compute $r(F)$, we can focus on finding $\Pr[r(H) = r(F)]$ and in particular whether we can guarantee that $r(F)$ is the alternative with highest probability of winning in H .

Before we delve further in this section, we define the concept of *margin of victory* (MoV) [38]. In a regular scenario, MoV of an election, is the smallest number k such that k agents can change the winner by voting differently. It can be considered a measure for how robust the election result is. We change the definition slightly to consider fractional profiles and FP-winners.

Definition 5.1 (Margin of Victory). Given a fractional preference profile Π and a voting rule r , the MoV is defined as $\text{MoV}(F(\Pi), r) = \min_{\Pi'} \frac{1}{2} \|F(\Pi) - F(\Pi')\|_1$, where $r(F(\Pi)) \neq r(F(\Pi'))$.

Now, we present the main theorem for this subsection.

Theorem 5.9. *Given a set of m alternatives \mathcal{A} , n agents with fractional preference profiles π_1, \dots, π_n , let fractional profile be F and randomized profile be H . For any anonymous voting rule r , for any $d \geq 0$, $\Pr[r(H) = r(F) \mid \text{MoV}(F, r) \geq d] \geq 1 - 2m! \cdot \exp(-\frac{d^2}{d+2n})$.*

This theorem does not depend on any voting rule based properties and thus will hold for any anonymous voting rule r . While the bound itself may be weak for cases with small N and relatively large m , this fact actually works to our advantage. Because the true probability of $r(F)$ being the winner could be much higher. Also note that while the theorem uses the value of MoV, we actually never need to compute it, because none of our voting rule algorithms depend on it. Now, if the probability of an alternative winning under randomized preferences is ≥ 0.5 , that guarantees that it is the highest probable winner. For the FP-winner, we get the following sufficient condition for this to happen. For low m (compared to \sqrt{n}), such a margin of victory is often attainable for real preference profiles.

Theorem 5.10. *Given a set of m alternatives \mathcal{A} , n agents with fractional preference profiles π_1, \dots, π_n , let fractional profile be F and randomized profile be H . For any anonymous voting rule r , we can get $\Pr[r(H) = r(F)] \geq \frac{1}{2}$ for a margin of victory $\text{MoV}(F, r)$ that is $\Omega(m \log(m) + \sqrt{n})$.*

6 Experiments and Results

College Confidential Data.

Instances: The dataset consists of multiple “discussions”. Each discussion has fixed alternatives (set of alternatives can be different for each discussion) and a number of users (agents) discussing their preferences over the alternatives. The discussion consists of multiple serialized comments. For each comment made by the agents, for all possible pairs of alternatives, we have human annotation of preference expressed over the alternatives.

Collection Process:

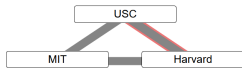
Discussion 1, Comment 1/10

Title of the original post: usc vs harvard vs mit

Please analyze the comment in the blue outline.

Username: CollegeGeek

Even though USC is a well known school, it is nowhere near Harvard or MIT. I would recommend MIT or Harvard depending on your interests, but if all choices were available to you, I would definitely say Harvard and MIT are much better choices than USC.



Preference expressed in blue comment

☐ USC preferred over Harvard

☐ Harvard preferred over USC

☐ Equal preference

☐ No preference expressed

Question: 1/20

Prev Next

School Info	Original Comment	Past Comments
USC University of Southern California	Harvard Harvard University	MIT Massachusetts Institute of Technology

Context information about each school:

	University of Southern California	Harvard University	Massachusetts Institute of Technology
Location	Los Angeles, California	Cambridge, Massachusetts	Cambridge, Massachusetts
Undergraduate Size	18740	6755	4530
Institution Type	USC Fisher Museum of Art, USC Shoah Foundation Institute for Visual History and Education	Private University	National Space Grant College and Fellowship Program, Land-grant university, National Sea Grant College Program, Private university
Faculty size	3945	1957	4016

Figure 2: Sample Task faced by agents

We scraped all discussions in the College Confidential forum made between Jan 2017 through Jun 2020 that had two or more alternatives and had at least 40 comments. The threshold of 40 comments were chosen so that there was at least some meaningful discussion of ideas instead of few isolated comments.

For the annotation, we recruited participants on Amazon Mechanical Turk. Every participant was given 20 comments to annotate, and each comment in turn was shown to up to 3 participants. For each comment, the participants choose whether the agent expressed a preference or not, and in case they did what the exact preference was. A sample task faced by participants is shown in Figure 2.

Statistics: In total, we have labeled data from 53 discussions, 40 two-alternative and 13 three-alternative discussions. Each discussion has at least 30 individual comments which compare between the alternatives. Considering the three pairwise comparisons in a comparison of three alternative, the complete dataset consists of 2974 pairwise comparisons.

Analysis and Results:

To learn from the data, we extract entity-level sentiment features and additional text features e.g. Parts of speech (PoS) tagging, n-grams etc. to create feature vector for the scenarios. We found that the text features (PoS, n-grams), while useful to identify whether there is a preference expressed in a comment, is much less helpful than the sentiment features themselves. Thus, we consider two ways to train classifiers.

1. Two stage learning - first stage: *preference vs no-preference*, second stage three class classifier with labels $A > B, A < B, A = B$

2. Single stage four class classifier with labels *no preference, $A > B, A < B, A = B$*

When learning the PL-X parameters, we also implement a two-stage method. The first stage learns a preference vs no-preference classifier, while the PL-X model is trained only on comments that have expressed preferences. To have a baseline for comparison, we consider a predictor that predicts the class based on a single value: sentiment score, predicting preference for the alternative that has higher sentiment associated with it.

We show the results for random forest (RF), shallow neural nets (NN) and PL-X models in Table 2. While some classifiers have high training accuracy, test accuracy is low and comparable to the simple baseline for all models. This indicates that predicting preference can be a challenging task based on sentiment and other simple text features. While we also tried multinomial logistic regression, SVM and decision trees among other classifiers, they had poorer training and test accuracy, hence we do not present those results.

	Training accuracy	Cross-validation accuracy
Baseline	0.56	0.56
Single stage RF	0.94	0.60
Single stage NN	0.76	0.59
Two-stage RF	0.93	0.56
Two-stage NN	0.75	0.58
RF + PL-X	0.69	0.56

Table 2: Comment level prediction accuracy for pairwise preference

Aggregation results: We then predict aggregated group decisions based on the individual comment level predictions. For regular classifiers, we use the two-stage variants and we take the predictions as strict preference for one alternative or a tie and then aggregate the predictions. For PL-X however, we may additionally use fractional profiles as indicated in Section 5 to make the group decision.

Note that since the labels are for pairwise comparisons, it makes sense to apply WMG-based voting rules. Hence, we test on Copeland, maximin and Borda rules. Here, also we present both training and testing accuracy. Again, we see that the training accuracy can be particularly high for RF and NN, we can get high training accuracy, the test set performance

	Copeland		Maximin		Borda	
	Test	Train	Test	Train	Test	Train
Baseline	6/13	29/53	6/13	32/53	6/13	31/53
RF	8/13	41/53	8/13	42/53	8/13	43/53
NN	8/13	37/53	7/13	36/53	7/13	47/53
PL-X (fractional)	8/13	33/53	8/13	32/53	7/13	31/53
PL-X (classifier)	7/13	34/53	8/13	32/53	7/13	33/53

Table 3: Group prediction accuracy for various voting rules when trained on two alternative discussions and tested on three alternative discussions

does not improve much from the baseline. For the testing scheme, we trained on all two-alternative discussions and tested on three-alternative discussions. Since all voting rules become trivial for two alternatives, we tested on three-alternative scenarios. These results (Table 3) with somewhat high training accuracy but poor test accuracy again indicate the challenge of successfully learning and aggregating group decision from natural language.

Moral Machine Data. The Moral Machine dataset [3] consists of pairwise comparisons between alternatives posing a moral dilemma. Using the pairwise comparison data and alternative features, we learn an individual PL-X models for each agent. Given a new set of alternatives, using the PL-X parameters, we can learn a distribution over the alternatives π_i for each agent i . Using these distributions we can check our theoretical results for preference aggregation (Section 5).

For a set of alternatives, \mathcal{A} , we can compute sum-fractional preference profile $F(\Pi)$. For this experiment, we consider the scoring rules, plurality and Borda for voting. From $F(\Pi)$, we can compute $r(F)$ and $\text{MoV}(F, r)$ when r is plurality and Borda. We modify Algorithm 1 for computing MoV for scoring rules presented in Xia [38] to work with fractional profiles. Additionally, for \mathcal{A} , using the Algorithm 1 defined in Section 5.2, we can estimate the probability of the fractional-profile winner also being the randomized profile winner, i.e $\Pr[r(H) = r(F)]$. We repeat this process for different randomly sampled alternative sets of size $|\mathcal{A}| = 4$ and $N = 1000$ random agents.

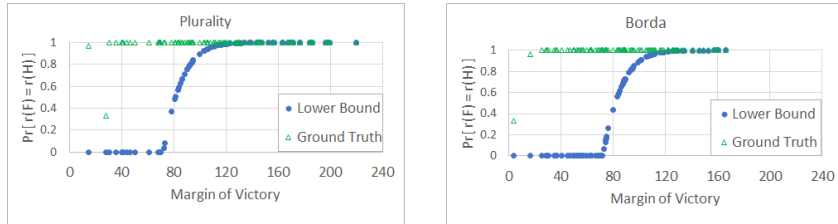


Figure 3: Margin of Victory vs Probability that FP-winner is also RP-winner

Figure 3 is a scatter plot for estimates and lower bounds for $\Pr[r(H) = r(F)]$, plotted against MoV. We notice that even for low margin of victory, when the theoretical guarantee is not that strong, the probability that the fractional winner is the randomized-profile winner as well is close to 1. This gives confidence in our aggregation framework. We also show further experimental results in Appendix D.

7 Discussion and Future Work

We propose a framework that learns preference models for agents from natural language and then efficiently aggregate preferences to provide a group decision. We also present an initial implementation of the framework in a newly created dataset for the specific problem of group decision-making from natural language. From our experiments, we notice that even sentiment extracted using state-of-the-art sentiment analysis methods are alone not enough to express sentiments. For preferences, we noticed that sentiments about alternatives are not always what matters. But rather figuring out the context of other entities about which sentiments were expressed is an important problem of preference learning and may improve the framework performance. Our dataset, in its current state, may not be large enough to train deep learning based models and it is our goal to expand the dataset with more labeled discussions. In that case, making use of modern deep learning methods in NLP to directly solve the preference learning problem might become more tractable. Also, it would also be interesting to expand on this framework towards building a multi-modal group decision-making system by learning preferences from other possible inputs besides natural language.

Acknowledgments

We thank Tania Bedrax-Weiss, Craig Boutilier, participants of the COMSOC video seminar, and anonymous reviewers for helpful discussions and comments. This work is supported by NSF #1453542, ONR #N00014-17-1-2621, and a gift fund from Google.

References

- [1] Rob Abbott, Brian Ecker, Pranav Anand, and Marilyn Walker. Internet argument corpus 2.0: An SQL schema for dialogic social media and the corpora to go with it. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, 2016.
- [2] Apoorv Agarwal, Boyi Xie, Ilia Vovsha, Owen Rambow, and Rebecca J Passonneau. Sentiment analysis of twitter data. In *Proceedings of the workshop on language in social media (LSM 2011)*, pages 30–38, 2011.
- [3] Edmond Awad, Sohan Dsouza, Richard Kim, Jonathan Schulz, Joseph Henrich, Azim Shariff, Jean-François Bonnefon, and Iyad Rahwan. The moral machine experiment. *Nature*, 563, 2018.
- [4] Rushlene Kaur Bakshi, Navneet Kaur, Ravneet Kaur, and Gurpreet Kaur. Opinion mining and sentiment analysis. In *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, pages 452–455. IEEE, 2016.
- [5] Siddharth Batra and Deepak Rao. Entity based sentiment analysis on twitter. *Science*, 9(4):1–12, 2010.
- [6] Indrani Bhattacharya, Michael Foley, Christine Ku, Ni Zhang, Tongtao Zhang, Cameron Mine, Manling Li, Heng Ji, Christoph Riedl, Brooke Foucault Welles, and et al. The unobtrusive group interaction (UGI) corpus. In *Proceedings of the 10th ACM Multimedia Systems Conference*, 2019.
- [7] Craig Boutilier. A POMDP formulation of preference elicitation problems. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 239–246, Edmonton, AB, Canada, 2002.
- [8] Craig Boutilier. Computational Decision Support: Regret-based Models for Optimization and Preference Elicitation. In P. H. Crowley and T. R. Zentall, editors, *Comparative Decision Making: Analysis and Support Across Disciplines and Applications*. Oxford University Press, 2013.
- [9] Felix Brandt, Vincent Conitzer, Ulle Endriss, Jerome Lang, and Ariel D. Procaccia, editors. *Handbook of Computational Social Choice*. Cambridge University Press, 2016.
- [10] Tathagata Chakraborti, Kshitij P Fadnis, Kartik Talamadupula, Mishal Dholakia, Bipav Srivastava, Jeffrey O Kephart, and Rachel KE Bellamy. Planning and visualization for a smart meeting room assistant. *AI Communications*, 32(1):91–99, 2019.
- [11] Vincent Conitzer and Tuomas Sandholm. Vote elicitation: Complexity and strategy-proofness. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 392–397, Edmonton, AB, Canada, 2002.
- [12] Cynthia Dwork, Ravi Kumar, Moni Naor, and D. Sivakumar. Rank aggregation methods for the web. In *Proceedings of the 10th World Wide Web Conference*, pages 613–622, 2001.

- [13] Miguel Á García-Cumbreras, Arturo Montejo-Ráez, and Manuel C Díaz-Galiano. Pessimists and optimists: Improving collaborative filtering through sentiment analysis. *Expert Systems with Applications*, 40(17):6758–6765, 2013.
- [14] Umberto Grandi, Andrea Loreggia, Francesca Rossi, and Vijay A Saraswat. From sentiment analysis to preference aggregation. In *ISAIM*, 2014.
- [15] Umberto Grandi, Andrea Loreggia, Francesca Rossi, and Vijay Saraswat. A borda count for collective sentiment analysis. *Annals of Mathematics and Artificial Intelligence*, 77(3):281–302, 2016.
- [16] Noam Hazon, Yonatan Aumann, Sarit Kraus, and Michael Wooldridge. On the evaluation of election outcomes under uncertainty. *Artificial Intelligence*, 189, 2012. ISSN 0004-3702.
- [17] Thomas Hofmann, Jan Puzicha, and Michael I Jordan. Learning from dyadic data. In *Advances in neural information processing systems*, pages 466–472, 1999.
- [18] David R. Hunter. Mm algorithms for generalized bradley-terry models. *Ann. Statist.*, 32, 2004. doi: 10.1214/aos/1079120141.
- [19] Pythagoras Karampiperis, Antonis Koukourikos, and Giannis Stoitsis. Collaborative filtering recommendation of educational content in social environments utilizing sentiment analysis techniques. In *Recommender Systems for Technology Enhanced Learning*, pages 3–23. Springer, 2014.
- [20] Jerome Lang, Jerome Mengin, and Lirong Xia. Voting on Multi-Issue Domains with Conditionally Lexicographic Preferences. *Artificial Intelligence*, 265:18–44, 2018.
- [21] Bing Liu. *Sentiment analysis and opinion mining*. Morgan and Claypool Publ., 2012.
- [22] R. Duncan. Luce. *Individual choice behavior - A Theoretical Analysis*. Wiley, 1959.
- [23] Thomas Lukasiewicz, Maria Vanina Martinez, and Gerardo I Simari. Probabilistic preference logic networks. In *Proceedings of the Twenty-first European Conference on Artificial Intelligence*, pages 561–566. IOS Press, 2014.
- [24] Debmalya Mandal, Ariel D. Procaccia, Nisarg Shah, and David P. Woodruff. Efficient and Thrifty Voting by Any Means Necessary. In *Proceedings of NeurIPS*, 2019.
- [25] Julian McAuley and Jure Leskovec. Hidden factors and hidden topics: Understanding rating dimensions with review text. In *Proceedings of the 7th ACM Conference on Recommender Systems*, 2013.
- [26] Saif Mohammad, Svetlana Kiritchenko, Parinaz Sobhani, Xiaodan Zhu, and Colin Cherry. SemEval-2016 task 6: Detecting stance in tweets. In *Proceedings of the 10th International Workshop on Semantic Evaluation*, 2016.
- [27] Saif M. Mohammad, Parinaz Sobhani, and Svetlana Kiritchenko. Stance and sentiment in tweets. *ACM Trans. Internet Technol.*, 17, 2017. ISSN 1533-5399.
- [28] Ritesh Noothigattu, Snehalkumar Gaikwad, Edmond Awad, Sohan Dsouza, Iyad Rahwan, Pradeep Ravikumar, and Ariel Procaccia. A voting-based system for ethical decision making. In *Proceedings of AAAI Conference on Artificial Intelligence*, 2018.

- [29] Nikolaos Pappas and Andrei Popescu-Belis. Sentiment analysis of user comments for one-class collaborative filtering over ted talks. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, pages 773–776, 2013.
- [30] R. L. Plackett. The analysis of permutations. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 24, 1975.
- [31] Adarsh Prasad, Harsh Pareek, and Pradeep Ravikumar. Distributional rank aggregation, and an axiomatic analysis. In *International Conference on Machine Learning*, pages 2104–2112, 2015.
- [32] Dirk Schäfer and Eyke Hüllermeier. Dyad ranking using a bilinear plackett-luce model. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 227–242. Springer, 2015.
- [33] Dirk Schäfer and Eyke Hüllermeier. Dyad ranking using plackett-luce models based on joint feature representations. *Machine Learning*, 107(5):903–941, 2018.
- [34] Parinaz Sobhani, Diana Inkpen, and Xiaodan Zhu. A dataset for multi-target stance detection. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, 2017.
- [35] Louis L Thurstone. A law of comparative judgment. *Psychological review*, 34(4):273, 1927.
- [36] Kenneth E. Train. *Discrete Choice Methods with Simulation*. Cambridge University Press, 2nd edition, 2009.
- [37] Bo Wang, Maria Liakata, Arkaitz Zubiaga, and Rob Procter. Tdparse: Multi-target-specific sentiment recognition on twitter. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 483–493, 2017.
- [38] Lirong Xia. Computing the margin of victory for various voting rules. In *Proceedings of the 13th ACM Conference on Electronic Commerce, EC*, 2012.
- [39] Lirong Xia. Learning and decision-making from rank data. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 13(1):1–159, 2019.
- [40] Yelp. Yelp dataset. <https://www.yelp.com/dataset/challenge>, 2014. [Online; accessed 17-February-2020].
- [41] Zhibing Zhao, Haoming Li, Junming Wang, Jeffrey Kephart, Nicholas Mattei, Hui Su, and Lirong Xia. A Cost-Effective Framework for Preference Elicitation and Aggregation. In *Proceedings of Uncertainty in Artificial Intelligence*, 2018.
- [42] Zhibing Zhao, Haoming Li, Junming Wang, Jeffrey O. Kephart, Nicholas Mattei, Hui Su, and Lirong Xia. A cost-effective framework for preference elicitation and aggregation. In *Proceedings of the Thirty-Fourth Conference on Uncertainty in Artificial Intelligence, UAI*, 2018.

Farhad Mohsin (Corresponding author)
Rensselaer Polytechnic Institute
Troy, NY, USA
Email: mohsif@rpi.edu

Lei Luo
Duke University
Durham, NC, USA

Wufei Ma
Purdue University
West Lafayette, IN, USA

Inwon Kang
Rensselaer Polytechnic Institute
Troy, NY, USA

Zhibing Zhao
Microsoft Research
Bellevue, WA

Ao Liu
Rensselaer Polytechnic Institute
Troy, NY, USA

Rohit Vaish
Tata Institute of Fundamental Research (TIFR)
Mumbai, India

Lirong Xia
Rensselaer Polytechnic Institute
Troy, NY, USA

A Discussion of Random Utility Models

In this , we use a more general Plackett-Luce model with features for preference learning. Let $X_{ij} \in \mathbb{R}^d$ denote the feature vector of a_j given agent i . We define the Plackett-Luce model with features as follows.

Definition A.1 (Plackett-Luce model with features (PL-X)). Given any agent i , each alternative a_j is characterized by a d -dimensional feature vector X_{ij} . The parameter space is $\Theta = \{\vec{\beta} = \{\beta_k | 1 \leq k \leq d\}\}$. The sample space is $\mathcal{L}(\mathcal{A})^n$. Given a parameter $\vec{\beta} \in \Theta$, the probability of any ranking $\sigma_i = a_{j_1} \succ a_{j_2} \succ \dots \succ a_{j_m}$ given by agent i is

$$\Pr_{\text{PL-X}}(\sigma_i | \vec{\beta}) = \prod_{p=1}^{m-1} \frac{\exp(\vec{\beta} \cdot X_{ij_p})}{\sum_{q=p}^m \exp(\vec{\beta} \cdot X_{ij_q})}.$$

Internal Consistency of RUMs: Because of the *internal consistency* that fractional preferences have when defined using PL parameters, the updated score each round may also be computed efficiently.

Proposition A.1 (Internal consistency). *Given PL parameters $\theta^1, \dots, \theta^m$, consider any partial ranking $\sigma_{\mathcal{B}} = \{a_{j_1} \succ \dots \succ a_{j_b}\}$ over alternatives in $\mathcal{B} = \{a_{j_1}, \dots, a_{j_b}\}$ where $\mathcal{B} \subset \mathcal{A}$. The fractional preference for all rankings $\sigma \in \mathcal{L}(\mathcal{A})$ which agree with $\sigma_{\mathcal{B}}$ regarding the order of alternatives in \mathcal{B} is $\sum_{\substack{\sigma \in \mathcal{L}(\mathcal{A}) \\ \sigma_{\mathcal{B}} \text{ are consistent}}} \pi(\sigma) = \pi(\sigma_{\mathcal{B}}) = \prod_{p=1}^{b-1} \frac{\exp(\theta^{j_p})}{\sum_{q=j_p}^{j_b} \exp(\theta^j)}$.*

This result, proven in [18], implies that computing the updated results in each round is equivalent to ignoring the already eliminated alternatives and just considering the PL parameters for existing alternatives. This is useful both for computing pairwise preferences and also preferences in multi-round voting rules.

B Numerical Example for Fractional Profiles

Example 1: For $\mathcal{A} = \{a_1, a_2, a_3\}$, suppose we have fractional profiles as in Table 4 for 3 agents.

σ	$\pi_1(\sigma)$	$\pi_2(\sigma)$	$\pi_3(\sigma)$
$a_1 \succ a_2 \succ a_3$	1/2	0	1/3
$a_1 \succ a_3 \succ a_2$	1/2	0	1/3
$a_2 \succ a_1 \succ a_3$	0	1	0
$a_2 \succ a_3 \succ a_1$	0	0	0
$a_3 \succ a_1 \succ a_2$	0	0	1/6
$a_3 \succ a_2 \succ a_1$	0	0	1/6

Table 4: Example of fractional profiles for $m = 3, n = 3$

Based on these fractional preferences we can calculate alternative scores for different voting rules. For example, we show plurality, Borda and maximin scores in Table 5

alternative	Plurality score	Borda score	Maximin score
a_1	5/3	9/2	11/6
a_2	1	3	7/6
a_3	1/3	3/2	1/3

Table 5: Example of fractional profiles with $m = 3, n = 3$

We see that, under all three voting rules, a_1 has the maximum score and is FP-winner for these fractional profiles.

C Various Proofs

Proof for Proposition 5.1

Proposition. *Given m alternatives and n agents, where agent i has k -PL mixing coefficients $\alpha_1, \dots, \alpha_k$ and parameters $(\vec{\theta}_{i(1)}, \dots, \vec{\theta}_{i(k)})$, FP-winner $r(F)$ can be computed in $\mathcal{O}(kmn)$ time when r is plurality.*

Proof. The score vector when $r = \text{plurality}$ is $\vec{s} = \{1, 0, \dots, 0\}$. So, to compute plurality winner, we only need $\pi_i^{(j)}$ for all $i \leq n$ and $j \leq m$. And by definition, plurality winner is

$$r(F) = \operatorname{argmax}_{a_j} \sum_{i=1}^n \pi_i^{(j)} = \operatorname{argmax}_{a_j} \sum_{i=1}^n \frac{\exp(\theta_i^j)}{\sum_{k=1}^m \exp(\theta_i^k)}$$

Clearly, the sum for each j can be computed in $\mathcal{O}(n)$ and thus the argmax can be computed in $\mathcal{O}(mn)$ time. \square

Proof for theorem 5.2

Theorem. *Given m alternatives and n agents, where agent i has k -PL mixing coefficients $\alpha_1, \dots, \alpha_k$ and parameters $(\vec{\theta}_{i(1)}, \dots, \vec{\theta}_{i(k)})$, for scoring rules that have score vectors with $\ell < m$ non-zero scores, FP-winner can be computed in $\mathcal{O}(k\ell m^\ell n)$ time.*

Proof. Only the first ℓ ranks contribute to alternative scores. Thus, if we enumerate all possible permutations over the top ℓ positions and compute the fractional preference for each permutation, we can compute score for each alternative. There are $\frac{m!}{(m-\ell)!}$ such permutations.

Now, Assume $\sigma_{|\ell} = a_{j_1} \succ \dots \succ a_{j_\ell}$ is one such permutation. for k -PL parameters as described,

$$\pi_i(\sigma_{|\ell}) = \sum_{\kappa=1}^k \alpha_\kappa \cdot \prod_{p=1}^{\ell} \frac{\exp(\theta_{i(\kappa)}^{j_p})}{\sum_{q=p}^m \exp(\theta_{i(\kappa)}^{j_q})}.$$

So computing each $\pi_i(\sigma_{|\ell})$ would take $\mathcal{O}(k\ell)$ time, there are $\mathcal{O}(m^\ell)$ such permutations, and n users, we can compute score for all alternatives in $\mathcal{O}(k\ell m^\ell n)$ time. \square

Proof for Lemma 5.3

Lemma. *Given m alternatives and n agents, where agent i has k -PL (or k -TV) mixing coefficients $\alpha_1, \dots, \alpha_k$ and parameters $(\vec{\theta}_{i(1)}, \dots, \vec{\theta}_{i(k)})$ (Or $(\vec{\mu}_{i(1)}, \dots, \vec{\mu}_{i(k)})$), the WMG of F can be computed in $\mathcal{O}(km^2n)$ time.*

Proof. The WMG for F , same as that for a single fractional profile π_i would have m nodes and thus $\mathcal{O}(m^2)$ edges. Weight for edge (a_j, a_ℓ) for k -PL model would be $\sum_{i=1}^n \pi_i^{(a_j \succ a_\ell)} = \sum_{i=1}^n \sum_{\kappa=1}^k \alpha_\kappa \frac{\exp(\theta_{i(\kappa)}^j)}{\exp(\theta_{i(\kappa)}^j) + \exp(\theta_{i(\kappa)}^\ell)}.$

For k -TV model, this becomes $\pi^{(a_j \succ a_\ell)} = \sum_{\kappa=1}^k \alpha_\kappa \cdot \Phi(\mu_{(\kappa)}^j - \mu_{(\kappa)}^\ell)$

For each case, each edge weight is obviously computable in $\mathcal{O}(kn)$ time, which leads to the desired result. \square

Lemma C.1. *Given m alternatives and n agents, where agent i has k -PL mixing coefficients $\alpha_1, \dots, \alpha_k$ and parameters $(\vec{\theta}_{i(1)}, \dots, \vec{\theta}_{i(k)})$, consider a multi-round voting rule given where voting rule r_q is used in round q . Let run-time of r_q for m voters be $T(r_q, m)$, then total run-time for the multi-round procedure shall be $\mathcal{O}(\sum_{q=1}^m T(r_q, m - q + 1))$.*

Proof. For each round until the last one, the eliminated alternative needs to be computed instead of the winner. For all rules we have discussed, finding lowest score alternative requires the same time as finding winning alternative. Now, Lemma A.1 tells us that for a single PL, computing the fractional preference for a partial ranking is as simple as ignoring the other alternatives. Now, the probability of k -PL is a convex combination of k PL models. In round q , if the set of remaining alternatives is $\mathcal{B} \subset \mathcal{A}$ with $|\mathcal{B}| = m - q + 1$, for some $\sigma_B \in \mathcal{L}(\mathcal{B})$

$$\pi_i(\sigma_B) = \sum_{\kappa=1}^k \alpha_\kappa \prod_{p=1}^{b-1} \frac{\exp(\theta^{j_p})}{\sum_{q=j_p}^{j_b} \exp(\theta^q)}$$

Thus, computing π_i for $i \leq n$ and consequentially fractional preference for any other partial ranking in round q concerns computing over only the remaining $m - q + 1$ alternatives. So, we can say that the run-time for round q is $T(r_q, m - q + 1)$, thus completing the proof. \square

Proof for Lemma 5.7

Lemma. *Given set of fractional preferences Π , for all $\sigma \in \mathcal{L}(\mathcal{A})$, $\mathbb{E}(H(\sigma)) = F(\sigma)$*

Proof. The probability that agent i 's preferred ranking is σ is $\pi_i(\sigma)$ and the related indicator RV is $v_i(\sigma)$. So, $\mathbb{E}(v_i(\sigma)) = \pi_i(\sigma)$. Now, $H(\sigma) = \sum_{i=1}^n v_i(\sigma)$. So, $\mathbb{E}(H(\sigma)) = \sum_{i=1}^n \mathbb{E}(v_i(\sigma)) = \sum_{i=1}^n \pi_i(\sigma) = F(\sigma)$. \square

Proof for theorem 5.8

Theorem. *Given any $\epsilon, \delta > 0$, if $p = \langle p_1, \dots, p_m \rangle$ are true winning probabilities and \tilde{p} is the estimate from Algorithm 1, we need $\mathcal{O}(\frac{m^2}{\epsilon^2} \ln(\frac{2m}{\delta}))$ samples so that $TVD(p, \tilde{p}) \leq \epsilon$ with probability $1 - \delta$ where $TVD(p, \tilde{p})$ is the total variation distance between the two distributions.*

Proof. For alternative a_j , define indicator random variable X_j , which takes the value 1 when a_i wins, and 0 otherwise. Then X_j is Bernoulli distributed with probability $\pi(a_i)$, and each trial is independent. Let $\epsilon > 0$ be arbitrary. Define $\epsilon_i = \frac{\epsilon}{m}$.

Then we use Chernoff bound and have

$$\Pr[|\frac{\sum_{t=1}^T X_i^{(t)}}{T} - \pi(a_i)| \geq \epsilon_i] \leq 2e^{-\frac{T\epsilon_i^2}{3\pi^*(a_i)}}$$

The total variation distance between the winner distributions computed using the Monte Carlo algorithm \tilde{p} and the underlying ground truth p can be defined as $\sum_{a_j \in \mathcal{A}} |p_j - \tilde{p}_j|$. Applying the union bound, with probability at least $1 - 2 \sum_{i=1}^m e^{-\frac{T\epsilon_i^2}{3m^2\pi^*(a_i)}}$, the total variation distance is below $\sum_{i=1}^m \epsilon_i = \epsilon$.

So, to have prob. $> \delta$, we would need at least $T = \frac{3m^2}{\epsilon^2} \ln \frac{2m}{\delta}$ samples to have $TVD(p, \tilde{p}) \leq \epsilon$ \square

Proof for theorem 5.9

Theorem. *Given a set of m alternatives \mathcal{A} , n agents with fractional preference profiles π_1, \dots, π_n , let fractional profile be F and randomized profile be H . For any anonymous voting rule r , for any $d \geq 0$, $\Pr[r(H) = r(F) \mid MoV(F, r) \geq d] \geq 1 - 2m! \cdot \exp(-\frac{d^2}{d+2N})$.*

Proof. Assuming that $F(\sigma) > 0$ for all $\sigma \in \mathcal{L}(\mathcal{A})$, we get the following Chernoff bounds for $\delta > 0$

$$\begin{aligned} \Pr[|H(\sigma) - F(\sigma)| \geq \delta F(\sigma)] &\leq 2 \exp\left(-\frac{\delta^2}{\delta+2} F(\sigma)\right) \\ \iff \Pr[|H(\sigma) - F(\sigma)| \geq d] &\leq 2 \exp\left(-\frac{d^2}{d+2F(\sigma)}\right) \\ &\leq 2 \exp\left(-\frac{d^2}{d+2N}\right) \end{aligned}$$

with $d = \delta F(\sigma)$. The last inequality comes from the fact that $F(\sigma) \leq N$.

Using the union bound with this for all σ , we get

$$\Pr\left[\bigcap_{\sigma \in \mathcal{L}(\mathcal{A})} (|H(\sigma) - F(\sigma)| \leq d)\right] \geq 1 - 2m! \exp\left(-\frac{d^2}{d+2N}\right)$$

This basically means that with high probability, total vote for all rankings in H will be within d of what it is in F . Suppose, now for some voting rule r , we compute the winner and MoV based on F to get $r(F)$ and $\text{MoV}(F, r)$. Now, if $\text{MoV}(F, r) \geq d$, and $H(\sigma)$ is within d to $F(\sigma)$ for all $\sigma \in \mathcal{L}(\mathcal{A})$ that would be sufficient for $r(F) = r(H)$, independent of the voting rule and that concludes the proof. \square

Proof for Theorem 5.10

Theorem. *Given a set of m alternatives \mathcal{A} , n agents with fractional preference profiles π_1, \dots, π_n , let fractional profile be F and randomized profile be H . For any anonymous voting rule r , we can get $\Pr[r(H) = r(F)] \geq \frac{1}{2}$ for a margin of victory $\text{MoV}(F, r)$ that is $\Omega(m \log(m) + \sqrt{n})$.*

Proof. Instead of applying the union bound where we did in the proof of Theorem 5.9, we could use it before to get the following-

$$\Pr\left[\bigcap_{\sigma \in \mathcal{L}(\mathcal{A})} (|H(\sigma) - F(\sigma)| \leq d)\right] \geq 1 - \sum_{\sigma \in \mathcal{L}(\mathcal{A})} \exp\left(-\frac{d^2}{d+2F(\sigma)}\right)$$

We know the added fact that $\sum_{\sigma \in \mathcal{L}(\mathcal{A})} F(\sigma) = N$. Because the fractional votes must add up to 1 for each agent. Thus, to get a tighter lower bound, we can solve the following optimization problem

$$\text{maximize } \sum_{p=1}^{m!} \exp\left(-\frac{d^2}{d+2x_p}\right)$$

where $\sum_{p=1}^{m!} x_p = n$

Because of symmetry, the KKT points for this $m!$ -dimensional optimization problem would be all points where some x_p are 0 and the rest are all equal. Thus, we can represent the optimum value as

$$\max \left\{ (m! - k) \exp(-d) + k \exp\left(-\frac{d^2}{d + \frac{n}{k}}\right) \right\}_{p=1}^{m!}$$

Note that this already leads to a tighter bound than Theorem 5.9, however we proceeded to present the simpler result because there is no unique k that gives a general inequality for all values of d . Thus we chose to present the simpler but weaker result there.

Although we get no fixed k for all values of d , as d grows higher, the value for $k = 1$ dominates the other values. Particularly, for $d > \sqrt{(n)}$,

$$\max \left\{ (m! - k) \exp(-d) + k \exp\left(-\frac{d^2}{d + \frac{n}{k}}\right) \right\}_{p=1}^{m!} = (m! - 1) \exp(-d) + \exp\left(-\frac{d^2}{d + n}\right)$$

Now, we can find $d = \Omega(m \log m)$ s.t. $(m! - 1) \exp(-d)$ goes to zero. Also, as d grows slightly from $\sqrt{(n)}$, $\exp(-\frac{d^2}{d+n})$ also becomes low. And thus, combined if $d = \Omega(m \log(m) + \sqrt{(n)})$, we get the desired result. \square

D Additional Simulation Plots for Moral Machine Data

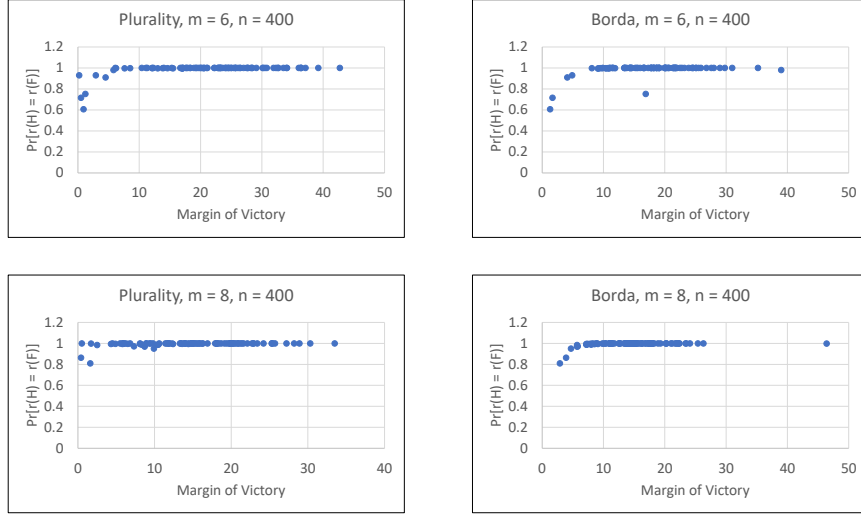


Figure 4: Margin of Victory vs Probability that FP-winner is also RP-winner

$\Pr[r(F) = r(H)]$ was predicted with less than ideal number of samples for these plots to ensure probabilistic guarantees as in Theorem 5.8. So they are slightly noisy. But the trend indicates that the probability is near 1 in all cases.