A Restricted Markov Tree Model for Inference and Generation in Social Choice with Incomplete Preferences

John A. Doucette
New College of Florida
Sarasota, FL, USA
jdoucette@ncf.edu

Robin Cohen
University of Waterloo
Waterloo, ON, Canada
rcohen@uwaterloo.ca

ABSTRACT

We introduce a probabilistic graphical model of ranked preferences for social choice based on a restricted version of a kth-order Markov tree. The system is similar to Plackett's model, and is based on the intuition that, in some domains, an agent's next most preferred alternative is highly predictable given a few of the immediately preceding alternatives. We provide a bound on the data requirements to learn the parameters of such a model and show eventual consistency between most probable sequences of the model and both the centroid of a Mallows model and the induced ranking of a RUM, theoretically and on artificial data; this is followed by a full evaluation of the model as a social choice system, including comparisons with existing approaches on 6 real world datasets. An application of the system to a simulated multiagent coordination task is also demonstrated, in which the proposed system offers pronounced advantages over existing approaches.

1. INTRODUCTION

Multiagent decisions arise when a society of autonomous agents must select from among some set of alternative actions which the society will undertake *collectively*. When this coordination problem is solved by having each agent express an individual preference regarding the various alternatives, and applying an algorithm to aggregate these individual preferences into a group preference, the problem is studied using the tools of *social choice* theory [3, 7].

Both individual and group preferences can be expressed as a *ranking* of the alternatives available to the agent society. An individual agent's ranking over a set of alternatives (candidates) C, denoted $\succ$, is a binary relation over elements of C that is transitive, reflexive and antisymmetric (i.e. a partial ordering of C). Thus, for two candidates $c_i, c_j \in C$, $c_i \succ c_j$ denotes c_i preceding c_j in the ordering of the candidates (c_i is preferred to c_j). A social choice function f aggregates rankings from the individual members of a society into a single communal ranking, implementing some aggregation algorithm. The two longstanding and central results in social choice theory [2, 10, 21] show that for the special case where all agents' rankings are total orders over C, no aggregation algorithm can produce a result that is properly representative of the preferences of the society, as specified by a set of axiomatic fairness properties. Consequently, there exist many competing social choice functions, which may be considered better or worse for a given society depending on the circumstances of the choice it faces. For example, recent work [6, 8] advances social choice functions that minimize the maximum possible regret that the society could collectively experience as a result of the function's choice of aggregate ranking. This approach is an effective method of making a decision when rankings provided by individual agents are incomplete [13], or under the assumption that agents preferences have been induced by underlying utility functions that are unknown [6]. Alternative techniques include building a model of how agents' preferences might be generated, and then using the individual rankings to infer the parameters of the model. The resulting model parameters can then be used to suggest an aggregated ranking directly [23, 19, 15, 17], or to impute any missing preference information before applying an existing social choice function [9].

In this work, we offer a particular approach for modelling how agent preferences can be generated, a probabilistic graphical model called a restricted Markov tree. The resulting model is shown to be consistent with two existing families of ranked preference distributions, but with the additional benefit of modelling the entire distribution of preferences, rather than just estimating the most common ranking. This means that a side effect of inferring the parameters of our model is to provide an excellent model for the imputation of missing preference information. We demonstrate that the new technique is highly effective when used to impute missing preferences from real world electoral data, and also show, through the use of a simulation, that the new model outperforms competitors for certain problem domains.

While learning from ranked preferences is central to analysis of data in domains like the Netflix challenge, as indicated above, the application which motivates our work and is used for discussion in this paper is that of social choice (i.e. aggregating individual ranked preferences to reach communal decisions); a companion concern is thus addressing how to predict missing preference information, of use in various real world scenarios.

2. BACKGROUND

There has been considerable recent interest in learning models of ranked preference data. Generally, such approaches begin by the specification of a model, or family of models, from which preferences might be drawn, and then proceed to show how to infer the parameters of the model efficiently from a sample of ranked preference data. This sample can take many forms, ranging from arbitrary pairwise relationships between the alternatives presented through to more constrained representations like top-t orderings (where the t most preferred candidates are ordered, and the remainder are unordered, except insofar as they follow the first t), through to total orders over the set.

Appears in: *Proceedings of the 16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2017), S. Das, E. Durfee, K. Larson, M. Winikoff (eds.), May 8–12, 2017, São Paulo, Brazil.*

The simplest such model is the *Mallows Distribution*, roughly analogous to a Gaussian distribution over rankings [15]. In a Mallows model, a *centroid ranking* over the alternatives, μ, fulfills roughly the role of the mean in a Gaussian, while a dispersion parameter ϕ is analogous to the variance. The probability of a ranking R being produced by a particular Mallows model is given by $\frac{1}{Z}e^{-\phi d(R,\mu)}$, where Z is a normalization constant, and $d(R, \mu)$ is the Kendall-Tau distance between R and μ [11] (i.e. the number of adjacent swaps required to transform R into μ). Several recent papers discuss efficient inference over mixtures of Mallows models (or classes which include them) via Gibbs sampling [12, 17].

An alternative, simple, preference model is the random utility model or RUM [14, 18]. In this model, the subjective utility (quality) assigned by a given agent or ranker to given alternative is a random variable drawn from some distribution over the reals. These subjective utilities are not observed. Instead, an agent's subjective utilities for all alternatives together induces an ordering over the alternatives, which the ranker then expresses. Like with the Mallows model, there has been considerable recent interest in inferring the parameters (typically means and variances) of the subjective utilities distributions of alternatives, given ranked data to learn from. These approaches range from being able to infer the parameters of RUMs where distributions lie in certain families (e.g. where all subjective utility distributions are normal distributions with identical variances [4]), to more general classes, including models with ranker-specific distributions [5].

Several other techniques have been developed that do not rely on a specific model from which preferences are drawn in order to infer an aggregated group ranking from individual rankings. Typically these approaches are more concerned with facilitating *decision making* (i.e. social choice, voting), than inferring a particular true ranking. The Minimax Regret approach [13] determines this order by ranking candidates with respect to the lowest possible rank they could achieve given an adversarial completion of any preferences that are incomplete (i.e. less than total rankings), under a given voting or aggregation rule. In contrast, the approach of [9] is to learn predictive models for the completion of different users' preferences, and to complete them with said predictions before making a decision using a given voting or aggregation rule.

The approach described in this paper describes a new model of preferences that is detailed enough to provide reasonable inference of a central ranking, but is still able to provide a *predictive* model of individual user's preferences. This is accomplished by learning a Markov model that describes preference rankings as **sequences**. An agent that can provide a portion of their own ranking can then have the remainder predicted by the model (allowing for applications like the imputation-based social choice approach mentioned above), while at the same time, a 'most probable sequence' can be extracted from the model as a representation of the overall decision.

3. MODEL

Among distributions over (complete) preferences, the most straightforward and nonrestrictive model is a general joint distribution over the set of possible total orderings. Agents' preferences are modelled being sampled from a set $\mathbf{R}$ of random variables $R_1, ..., R_{|C|}$, which all have domain C, the set of candidates or alternatives. The elements of $\mathbf{R}$ are constrained to take on mutually exclusive values (i.e. $R_i = c \rightarrow R_j \neq c, \forall\, 1 \leq i \neq j \leq |C|$). If, in an agent's preferences, candidate c is preceded by exactly $i - 1$ candidates, then in the corresponding assignment of values to the random variables in $\mathbf{R}$, $R_i = c$. The model is defined by a probability distribution $P(R_1, ..., R_{|C|})$ over possible ballots. Given such a model, it is also very easy to impute an incomplete preference provided by an agent. For example, if an agent states that they like c_1 the most, and c_2 second most, one could simply consult the model, find the most common completion of ballots that started with c_1 and c_2, and impute that completion. If the model was learned from actual ballots, then this idea amounts to the notion of looking for other agents who also ranked c_1 and c_2 first, and completing the original agent's ranking based on the preferences of these other agents. Although the algorithm for accomplishing this inference is straightforward, the size of the input to the algorithm can be prohibitively large, because the probability distribution may contain enormous amounts of detail. There are $|C|!$ possible ballots, and in theory, the probabilities for each can be separately encoded. If most of the ballot is incomplete, reasoning over possible completions is similarly hard.

One way of representing a probability distribution over a sequence of values for random variables is with a Markov Model, which stores a probability distribution for the next value in the sequence given some number of previous values, as well as a distribution over starting values for the sequence. This is similar in spirit to Plackett's model [18], which shows that a nested series of logistic models can be used to learn higher order sequences of patterns. A Markov Model assumes that the value of the variable at the present step is dependent only on the value of the variable in the previous step of the sequence, or perhaps in some (finite) number of additional sequence steps in the past. This assumption allows it to factor the joint distribution over a sequence of values $\mathbf{R}$ into a series of simpler distributions. For example, a first-order Markov Model computes the probability of a sequence as:

$$P(\mathbf{R}) = P(R_1) \prod_{i \in 2,...,|R|} P(R_i|R_{i-1})$$

where $P(R_1)$ is a distribution over the value of the first variable in the sequence, and $P(R_i|R_{i-1})$ is the probability of the next variable in the sequence taking on value R_i given that it had value R_{i-1} in the previous step.

The relationships between elements of the sequence in a Markov Model can be presented pictorially using the standard notation for Bayesian Networks. Figure 1 shows several examples of such figures for different values of k, the number of previous steps on which future steps depend. All of the models are over sequences of length exactly four, and the four nodes shown correspond to the value of the variables in the sequence at each of four steps. Edges show dependencies between the nodes. A probability distribution is needed at each node, showing the probability for the value of the sequence at this step, given the node's parents, $P(R_i|R_{i-j}, ..., R_{i-1})$. The grey nodes in each model all share a probability distribution, since they all have the same number of parents. The assumption that all nodes beyond the j^{th} share a probability distribution is called the Markovian assumption.

A Markov Tree is a particular way of storing, representing, and utilizing the probability distributions that are contained in a Markov Model. In a Markov Tree, the root stores the probability distribution over values of the initial variable R_1, much like the first node in a Markov Model. Both the root of a Markov Tree, and the first node of a Markov Model store the probability distribution $P(R_1)$. The root of a Markov Tree has one child for each possible value that the variable could take initially. For example, in a second-order Markov Tree, each node at the second level of the tree stores the probability distribution for R_2 given that R_1 took on a specific value in the previous step (i.e. $P(R_2|R_1 = x)$ for some x), and likewise has a child for each possible value that R_2 could take on. Collectively, *all* the nodes at the second level of the tree store the information contained in $P(R_2|R_1)$, which is the probability distribution

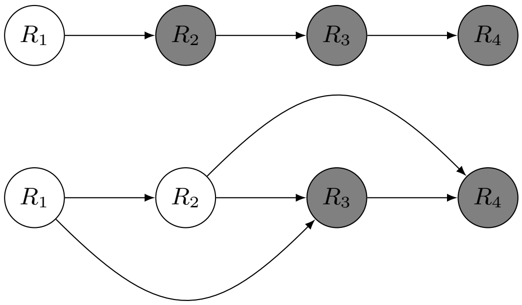

Figure 1: A graphical representation of first and second order Markov Models for a sequence of length four. Each node corresponds to the state of a random variable at the corresponding step in the sequence. An edge from R_i to R_j shows that the probability distribution for R_j depends on the value of R_i. Gray shaded nodes within the same model have identical probability distributions (by the Markov assumption).

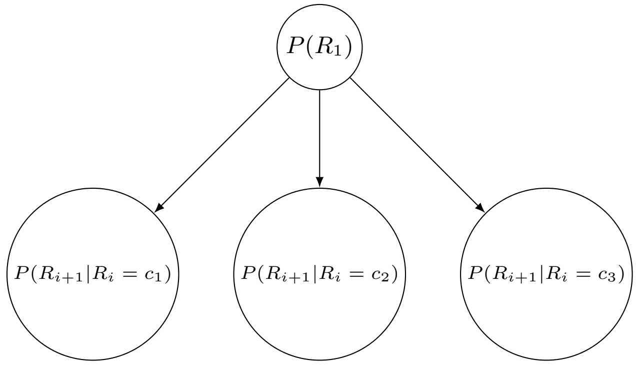

Figure 2: A graphical representation of a first order Markov Tree. The internal node of the tree (the root) stores information about the distribution of values for R_1, while the leaves store information about the distribution of R_i given the values of the preceding $k = 1$ members of R, which are identical for every $i > k$ by the Markovian Assumption.

stored at the second node in a Markov Model. Similarly, subsequent non-leaf levels of the tree will collectively store the distributions of subsequent nodes in a Markov Model. For instance, in a third-order Markov Tree, the third level of the tree collectively stores $P(R_3|R_1, R_2)$. Each node will store $P(R_3|R_1 = x, R_2 = y)$ for some specific values x and y. The leaves of the tree are special, and correspond to the entire set of grey nodes shown in Figure 1. In a Markov Tree of order k (i.e. a tree modelling a distribution where the i^{th} preference depends on the preceding k preferences), each leaf of the tree stores a distribution $P(R_i|R_{i-k}, ..., R_{i-1} = S)$ where S is a sequence of length k and $i > k$, meaning the leaves[1] collectively store the distribution $P(R_i|R_{i-k}, ..., R_{i-1})$. Figure 2 shows a Markov Tree that encodes all the information required to represent a first-order Markov Model assuming that the variables can take on only 3 distinct values at each step, denoted $\{c_1, c_2, c_3\}$. Markov Trees are used in this paper to represent Markov Models, because a special restriction must be applied to a Markov Model in order to represent distributions over *ballots*, and this restriction is more easily encoded into a Markov Tree. These restricted Markov Trees are discussed formally and at length in the next section.

In this section, we describe the proposed model for learning preference distributions using k^{th} -order Markov trees. The intuitive idea behind the model is that in some domains, ranked alternatives can be thought of as embedded in a (often small) feature space. For example, when ranking political alternatives, the candidates might be accurately representable on the left-right axis, or on the slightly less well known two-dimensional model (liberal/conservative economic policies and liberal/conservative social policies). Once the candidates are embedded in such a space, a ranking can be thought of as a **trajectory** through the space. Insofar as some trajectories are more common or popular than others, we can infer the relative quality of different rankings of the candidates.

From a generative prospective, this models the idea that ranked data are generated based on the candidates an agent has thought about in the recent past. This would be applicable to domains where humans are confronted with excessive amounts of choice, like ordering the set of all possible films, or all possible meals. In practice, such an agent probably does not construct a full ranking of the alternatives. Instead, the agent will first rank a few options that come to mind immediately. Thinking of these options will suggest certain successors, which in turn will suggest other successors. More formally, we assume an agent's *seed ranking*, of size k, is a top order drawn from a distribution over the set of top-k rankings. Following that, an agent's next most preferred alternative is drawn from a distribution that depends only on the immediately preceding k items. The process is then repeated, with some probability of stopping at each step, to yield a top-order over some (potentially large) set of candidates.

A Markov tree of order k is nearly (though not quite) sufficient to represent the process described above, and can be efficiently learned from a sample of ranked data. Since a Markov tree is fairly compact, provided that k is small, and has a readily interpretable representation, we believe it may be a useful and informative alternative to more complex existing models for learning user preferences, at least in domains with fairly homogenous preference trajectories. Interestingly, we can also show that the new model can efficiently learn the overall rankings of several existing families of voting models, including both random utility models and Mallows models.

3.1 Formal Description of the Model

Formally, the proposed model is to use a **restricted** Markov Tree, in which an agent's j^{th} most preferred alternative is modelled as a discrete random variable R_j with domain equal to the set of alternatives that are to be ranked, C. R_j is distributed according to a categorical distribution with parameters dependent on k preceding variables R_i for $j - k \leq i < j$, and conditionally independent of the rankings of any other agent given the parameters of the model itself (i.e. rankings are drawn i.i.d.). As mentioned above, a Markov tree is not quite capable of modelling the generative process that was described. This is because each candidate can be ranked only once, whereas in a true Markov tree of order k, the $k+2^{nd}$ candidate cannot depend on the first. To address this, the following **constraints** are imposed: $P(R_j = c|R_y = c \wedge y \neq j) = 0$, and also $P(R_j = c|\{R_y \mid y < j \wedge R_y \neq c\}) = P(R_j = c|\{R_y \mid j - k \leq y < j\})$ where k is said to be the *order* of the restricted tree. The resulting model is a Markov Tree of order k, but with mutual exclusivity constraints encoded. Such a model can be compactly represented

[1] Because of the Markov assumption, the distribution stored at each of the leaf nodes is identical, and so only needs to be recorded once in a generic form, and can be reused when reasoning about all values of R_i for $i > k$.

as an ordinary Markov tree of order k. When reasoning about R_j, one only needs to restrict the domain of R_j to those elements of C that did not appear previously in the ranking, renormalizing the probabilities of the remaining elements of C in accordance with Luce's axiom [14], which supposes that an agent's distribution of preferences over candidates a and b ought not to be affected by the addition or removal of some third candidate c from C.

The parameters of the model are encoded in several sets. The first set corresponds to the leaves of the tree, and are used to impute the $k + 1^{st}$ candidate given a list of k candidates that are ranked already. For a given sub-sequence S of k candidates drawn from C, $\theta_{S,c,k+1}$ denotes the value of the parameter corresponding to $P(R_j = c|\{R_y = S_{y-(j-k)+1} \mid j - k \leq y < j\})$, for any $j > k$. $P(\theta_{S,c,k+1})$ is then the prior probability distribution over values of $\theta_{S,c,k+1}$ in the model, and the complete set of parameters corresponding to multinomial distribution over values of R_j given the observation of each possible sub-sequence S is denoted with $\Theta_{C,k+1} = \{\theta_{S,c,k+1} \mid S \in \rho(C,k) \wedge c \in C\}$, where $\rho(C,k)$ is the set of all permutations of k elements of C. Additionally, the model will have $k - 1$ further sets of parameters for the initial levels of nodes in the Markov tree, corresponding collectively to the joint distribution over the first k candidates ranked, which naturally cannot be predicted by a distribution that requires k prior rankings. These parameters are denoted with $\Theta_{C,y} = \{\theta_{S,c,y} \mid S \in \rho(C,y) \wedge c \in C\}$, and are defined for every $1 \leq y \leq k$. Each of the parameters $\theta_{S,c,y}$ corresponds to $P(R_j = c|\{R_x = S_{x-(j-y)+1} \mid j - y \leq x < j\})$, predicting the y^{th} candidate on the basis of the preceding $y - 1$. Thus, for example, a first order tree has two sets of parameters, $\Theta_{C,2}$, stored in the leaves, and $\Theta_{C,1}$, stored in the root node. In a slight abuse of notation, the set of all parameters for a complete k^{th} order model is denoted with $\Theta_{C,\vec{k}}$. Throughout, there are occasional abuses of notation by treating S as though it were a set. S is always a subsequence rather than a set, but $S \setminus s$ may be used to indicate the sequence with its *last* element (s) truncated. Similarly, $|S|$ may denote the length of S.

Consider for example a first order model (i.e. $k = 1$) describing the distribution over a set of four candidates $C = \{c_1, c_2, c_3, c_4\}$. The model will have four parameters in $\Theta_{C,1}$, and 16 in $\Theta_{C,2}$ (4 in each of the 4 leaf distributions). Example values for the parameters are summarized in Table 1, which provides some intuition about the notation used in the model. The left hand table shows the distribution over initial elements of the sequence, in this case, over agents' most preferred candidates. The right hand table shows the distribution of an agent's second preference, conditioned on their first. An agent's third preference will obey an identical distribution, conditioned on their second, but re-normalized to remove their first preference from consideration. For example, suppose that an agent's first preference is c_2, and second preference is c_3. Looking in the right hand table, the probability of selecting c_1 as the agent's third preference is 0.1, given by $\theta_{c_3,c_1,2}$. Likewise, the probability of selecting c_4 as the agent's third preference is 0.8, as indicated by $\theta_{c_3,c_4,2}$. Since c_2 has been assigned as the agent's first preference, these probabilities do not sum to 1. By applying Luce's Axiom, the probabilities are renormalized, reflecting the idea that, apart from the mutual exclusivity constraint, the distribution depends only on the immediately preceding preference. Therefore, c_4 will be selected as the next preference with probability $\frac{0.8}{0.1+0.8} \approx 0.89$ and c_1 will be selected with probability $\frac{0.1}{0.1+0.8} \approx 0.11$.

Having described the model, it now remains to describe how its parameters should be learnt. Learning the parameterization of $\Theta_{C,k}$ is the most difficult. This requires a set of *contiguous* partial rankings over the subsets of k candidates in C. A contiguous ranking over a set C is defined by a successor function $\succ_{\text{contig}}$ such that $\forall\ a, b \in C,\ a \succ_{\text{contig}} b \iff \text{rank}(b) \leq \text{rank}(a) \wedge \not\exists\ c \in C\ s.t.\ \text{rank}(b) \leq \text{rank}(c) \leq \text{rank}(a)$, where $\text{rank}(x)$ is the number of candidates that are preceded by x in the ranking. That is, it is a partial ordering where certain pairs (those where $a \succ_{\text{contig}} b$ is defined) are *known* to be contiguous in the ranking, independent of the fact that certain other candidates may be unranked. Although the same symbol ($\succ$) is used, this should not be confused with the more general ideas of a partial or total order. This restriction at first appears substantial, but any total order over the candidate set can be represented with a contiguous ranking, and so can any top-k orders, and partial orders in which relatively few pairs of candidates are incomparable (e.g. $a \succ b \succ c \succ (d \sim e)$ is a contiguous ranking where a, b, and c appear contiguously in the first three positions, ahead of d and e, but the relative order of d and e is undetermined).

c	$\theta_{S,c,1}$	Value
c_1	$\theta_{\emptyset,c_1,1}$	0.1
c_2	$\theta_{\emptyset,c_2,1}$	0.4
c_3	$\theta_{\emptyset,c_3,1}$	0.4
c_4	$\theta_{\emptyset,c_4,1}$	0.1

S	c	$\theta_{S,c,2}$	Value
c_1	c_1	$\theta_{c_1,c_1,2}$	0
	c_2	$\theta_{c_1,c_2,2}$	0.4
	c_3	$\theta_{c_1,c_3,2}$	0.3
	c_4	$\theta_{c_1,c_4,2}$	0.3
c_2	c_1	$\theta_{c_2,c_1,2}$	0.8
	c_2	$\theta_{c_2,c_2,2}$	0
	c_3	$\theta_{c_2,c_3,2}$	0.2
	c_4	$\theta_{c_2,c_4,2}$	0
c_3	c_1	$\theta_{c_3,c_1,2}$	0.1
	c_2	$\theta_{c_3,c_2,2}$	0.1
	c_3	$\theta_{c_3,c_3,2}$	0
	c_4	$\theta_{c_3,c_4,2}$	0.8
c_4	c_1	$\theta_{c_4,c_1,2}$	0.2
	c_2	$\theta_{c_4,c_2,2}$	0.5
	c_3	$\theta_{c_4,c_3,2}$	0.3
	c_4	$\theta_{c_4,c_4,2}$	0

Table 1: Tables showing an example of the probability distributions used in a first-order Markov Tree. The distribution over initial states R_1 is shown on the left, while the transition probabilities (i.e. the distribution of R_i given R_{i-1}) are shown in the table on the right. The red numbers correspond to the probabilities that are re-normalized and used to determine an agent's third preference, given that their first preference was c_2, and their second was c_3.

Given a single total ordering of the candidates $\succ$, learning proceeds by splitting the ordered list corresponding to $\succ$ into a series of sub-sequences, used as observations to adjust the parameters of $\Theta_{C,\vec{k}}$. The candidate ranked first S_1 is an observation for $R_1 = S_1$, which consequently influences the posterior distribution of $\theta_{\emptyset,R_1,1}$. Similarly, the first j ranked candidates serve as an observation of $R_j = S_j|\{R_i = S_i, i < j\}$ for each $j < k$, which influences the posterior distribution of $\theta_{S_{1..j-1},S_j,j}$. After the first k candidates ranked in $\succ$, each subsequent ranked candidate defines an observation of $R_j = S_j|\{R_i = S_i, j - k \leq i < j\}$. Given a contiguous ranking that is not total, a similar process is followed. All contiguous subsequences of length $k + 1$ can be extracted and used as observations of $R_j = S_j|\{R_i = S_i, j - k \leq i < j\}$, and any top-order of length $j < k$ can also be used to provide observations for the other model parameters. Collectively, these sequences constitute the training data. To estimate the parameters from the training data, standard statistical estimation techniques are adopted. In particular, given a set of ballots that all begin with the same subsequence S and rank at least one more candidate c_i following S, the parameters in $\{\theta_{S,c_i,|S|} \mid c_i \in C \setminus S\}$ can be learned via the Maximum Aposteriori estimate for a categorical distribution:

$$\hat{\theta}_{S,c_i,|S|} = \frac{\alpha_i + N_{S,c_i,|S|}}{\sum_{c_j \in C \setminus S}(\alpha_j + N_{S,c_j,|S|})}$$

where $N_{S,c_i,|S|}$ is the number of observations which contain sequence S followed by c_i, and α_i is the number of *prior* observations of c_i following S (i.e. α collectively parameterize a Dirichlet prior for the categorical distribution defined by $\theta_{S,c_i,|S|}$ for every $c_i \in C \setminus S$; this allows the incorporation of prior information).

It is easy to see that a tree of this class, properly parameterized, can generate any sequence of alternatives. The full distribution over the first k candidates is represented fully, and with proper parameterizations, it can generate any initial sequence. The remainder of the sequence will be sampled from the $k+1^{th}$ node, which will initially assign probabilities to each previously unranked candidate. After selecting a candidate, the distribution conditional on the second through $k+1^{th}$ candidate will be renormalized by restricting all $k+1$ candidates thus far selected. This will proceed until the final candidate is selected, provided that all candidates have nonzero probabilities of appearing in all conditionings of the $k+1^{th}$ node. The distribution as represented is not proper, because the representation is compact. If the tree were expanded out fully, so that each node beyond the k^{th} corresponds to a particular restriction of the k^{th} node, then a proper probability distribution (i.e. where the sum of the probability of all events is 1) could be obtained. As it is, if the compact distribution is sampled from with appropriate restrictions, then the product of the *renormalized* parameters used at each sampling step will yield a proper probability, even if multiplying the raw parameter values would not. Note that normalization is not required during training, because we only train on the contiguous *subsequences* of length k or less. Note also that, if Laplacian smoothing is not used, re-normalization may be impossible.

Once a model has been learned, it is straightforward to impute a ballot. A top-order S can be extended by sampling a candidate from $C \setminus S$ from the appropriate learned categorical distribution.

3.2 Convergence Rates and Consistency

Suppose a set of rankings are generated from a restricted Markov Tree of order k with unknown parameter values. It is reasonable to suppose that another restricted Markov Tree of order k could be constructed, and that by observing the rankings generated from the first tree, could be trained to have parameter estimates that would converge to those of the original tree as more data was observed. A bound on the amount of data required to learn the parameters of the above model to within a given tolerance ϵ would be useful however. For simplicity, the learning process analyzed to develop a bound is maximum likelihood estimation rather than maximum aposterori estimation.

THEOREM 1. *Given a set of candidates C and a restricted Markov Tree T of order k describing a distribution over candidate sequences of C, the error in a learned estimate $\hat{\theta}_{S,c,|S|}$ of a parameter $\theta_{S,c,|S|}$ in a second restricted Markov Tree T' for a given sequence $S \in C^k$ and $c \in C$ is in $O(\theta_{S,c,|S|}\sqrt{\epsilon})$ with probability at least $(1-\alpha)^k$ after observing $N = \frac{-\ln((\alpha/|C|)^2)}{\theta_{S,c,|S|}} \prod_{0<j<k}(\theta_{S_{1,j-1},S_j,j}(1-4\sqrt{\epsilon})^{-1})$ sequences drawn from T.*

The gist of the proof is that for parameters in the top level of the tree (e.g. $\theta_{\emptyset,c,1}$), Quesenberry and Hurst's bound for the distribution of a multinomial proportion [20] can be used to guarantee that $\hat{\theta}_{S,c,|S|}$ lies within an interval centred at $\theta_{S,c,|S|}$ with probability $(1-\alpha)$. The interval can be expressed using Alzer's inequality [1] for the χ^2 cumulative density function in a closed form depending only on the values of ϵ, $\theta_{S,c,|S|}$ and N. Some algebra then yields the desired bound. For parameters lying deeper in the tree, the same approach is used, but only the subset of the rankings drawn that begin with sequence S can be used to learn the value of deeper parameter $\theta_{S,c,|S|}$. This results in a blowup of $\prod_{0<j<k}(\theta_{S_{1,j-1},S_j,j}(1-4\sqrt{\epsilon})^{-1})$ in the amount of data needed to produce the same bound as in the data for the parameters at the top level, and this is shown inductively.

PROOF. Suppose that N data points (i.e. total orderings) are sampled from a restricted Markov Tree T of order k as described above, parameterized with $\Theta_{C,\vec{k}}$. Let $N_{c,1}$ be a random variable representing the number of rankings sampled from T that rank candidate c highest. Let $\theta_{\emptyset,c,1}$ be the parameter in T corresponding to the true probability of generating a sequence that starts with c. Now, $N_{c,1}/N = \hat{\theta}_{\emptyset,c,1}$ is effectively distributed as a multinomial proportion. By Quesenberry and Hurst's (conservative) bound on the distribution of of a multinomial proportion [20], with probability $1-\alpha$, $\frac{N_{c,1}}{N} \in \frac{2\theta_{\emptyset,c,1}N+\chi^2_{1,\alpha/|C|}\pm\sqrt{\gamma}}{2(N+\chi^2_{1,\alpha/|C|})}$ where $\gamma = \chi^2_{1,\alpha/|C|}(\chi^2_{1,\alpha/|C|} + 4N\theta_{\emptyset,c,1}(1-\theta_{\emptyset,c,1}))$ and where $\chi^2_{1,\alpha/|C|}$ is the value at which the probability mass in the right tail of a χ^2 distribution with 1 degree of freedom is exactly $\alpha/|C|$. It follows that:

$$\frac{N_{c,1}}{N} \in \frac{\theta_{\emptyset,c,1} + \frac{\chi^2_{1,\alpha/|C|}}{2N} \pm \sqrt{\gamma/(4N^2)}}{1 + \frac{\chi^2_{1,\alpha/|C|}}{2N}}$$

.

Alzer's inequality [1], states that $\chi^2_{1,\alpha/|C|} < -2\ln((\alpha/|C|)^2)$. Let $\beta = \frac{-\ln((\alpha/|C|)^2)}{N}$. Then, with probability $(1-\alpha)$: $\frac{N_{c,1}}{N} \in \frac{\theta_{\emptyset,c,1}+\beta\pm\sqrt{\gamma/(4N^2)}}{1+\beta}$, from which it is easy to show that: $|\frac{N_{c,1}}{N} - \theta_{\emptyset,c,1}| < \beta + \sqrt{\gamma/(4N^2)}$

Now, assume that $\beta < \epsilon\theta_{\emptyset,c,1}$, which will be true for some $0 < \epsilon < 1$ for large enough N. Then one can derive $|\frac{N_{c,1}}{N} - \theta_{\emptyset,c,1}| < 4\theta_{\emptyset,c,1}\sqrt{\epsilon}$, which shows the desired bound for the case of the $P(R_i, 1)$.

The general case is now proven inductively. Suppose that the desired bound holds for a parameter $\theta_{S\setminus s,s,k-1}$. The goal is to show that it also holds for a parameter $\theta_{S,c,k}$, where $c \in C \setminus S$ is some other candidate. Let N_S be the total number of rankings drawn that begin with sequence S. By the inductive hypothesis, $|\theta_{S\setminus s,s,k-1} - \frac{N_S}{N_{S\setminus s}}| < 4\theta_{S\setminus s,s,k-1}\sqrt{\epsilon}$ with probability at least $(1-\alpha)^{k-1}$, for some $\frac{-\ln((\alpha/|C|)^2)}{N_{S\setminus s}\theta_{S\setminus s,s,k-1}} < \epsilon$. It follows that $N_S > N_{S\setminus s}\theta_{S\setminus s,s,k-1}(1-4\sqrt{\epsilon})$ with probability $(1-\alpha)^{k-1}$ also. Following the argument for the basecase above, if it is allowed that $\beta < \epsilon\theta_{S,c,k}$, then $|\theta_{S,c,k} - \frac{N_{S\cup c}}{N_S}| < 4\theta_{S,c,k}\sqrt{\epsilon}$ with probability at least $(1-\alpha)^k$. Since $N_S > N_{S\setminus s}\theta_{S\setminus s,s,k-1}(1-4\sqrt{\epsilon})$ it follows that N (the total number of sequences sampled) must also be larger by a factor of at least $\frac{1}{\theta_{S\setminus s,s,k-1}(1-4\sqrt{\epsilon})}$ than was required for the bound to hold for $\theta_{S\setminus s,s,k-1}$.

□

In more concrete terms, this result indicates that the total number of samples that must be drawn to accurately infer the true parameters of T grows exponentially in the depth of k, but linearly in the total number of parameters (i.e. in $|\Theta_{C,\vec{k}}|$). Further, errors will be concentrated in the least common sequences, which are unimportant for many applications. It may be helpful to see an example that illustrates the orders of magnitude for the data requirements. Suppose a second order Markov Tree is to be trained on ballots for an election with 10 candidates. It is expected *a priori* that all candidates have at least a 5% chance of following any two other candidates in a given ranking, and that ballots were generated by a second order Markov Tree or a similar process. If the error in the estimate of any parameter's value is desired to be at most 1% with probability 95%, then data requirements would be $N = \frac{-\ln(0.005^2)}{(0.05)^2(1-4\sqrt{0.01})^{-1}}$, or about $7,000$ example sequences of length $k+1$. This corresponds to about 875 complete rankings drawn from T, because a complete ranking of 10 candidates contains 8 subsequences of length 3. However, if some of the sequences are much more likely than others, these more likely sequences could be learned with far less data (since those sequences will appear disproportionately often in the data).

The runtime for parameter estimation is linear in the amount of data given (i.e. the number of contiguous subsequences provided), but there may be an enormous number of parameters in a deep model, such that simply enumerating the parameters requires more time than a pass through the data. To avoid this, the implementation renders training time independent of the total size of the tree being represented via the use of lazy evaluation and hash maps. The Markov Tree is represented as a nested data structure. The root holds a hashmap from C to the values of $\theta_{\emptyset,c_i,1}$ for all $c_i \in C$. Additionally, it holds a map from C to instances of the data structure that store distributions for the second candidate in sequence. Each of those nested data structures holds a mapping to distributions over the third candidate selected, and so on up to the k^{th}. However, vitally, the nested structures are only constructed if actual data is observed that necessitates their construction. Otherwise, they can be left un-constructed, and during inference and generation, an interchangable uniform distribution can be substituted anywhere that an undefined value is used. Since constructing a new node is a constant-time operation, even in the worst case, the total time required to construct and learn a model is linear in the number of observations (i.e. $O(kn)$ if n subsequences of length k are available for training).

3.2.1 Consistency

The results of the previous section also show consistency in the limit between a learned restricted Markov Tree T of order k, and a distribution from the same family. Although there are reasons to suppose at least some natural processes generate rankings in this manner, we can also show eventual consistency between the most probable sequence in a learned restricted Markov Tree and two common families of ranking models, the Mallows Family and the RUM Family with identical variances across utility distributions.

First, we define the most probable alternative at a given node of T at depth j (reached by traversing the tree with sequence S) to be $\hat{c}_S = \arg\max_{c \in C} \theta_{S,c,j}$, and the most probable sequence (MPS) for T at depth j to be $MPS_j(T) = MPS_{j-1}(T), \hat{c}_{MPS_{j-1}(T)}$, with $MPS_1(T) = \hat{c}_{\emptyset}$.

THEOREM 2. *Let M be a Mallows model with centroid $\mu = \mu_1, ..., \mu_{|C|}$ and dispersion parameter ϕ. As the number of total orders sampled from M goes to infinity, the most probable sequence $MPS_{|C|}(T)$ of a restricted Markov Tree T of order k trained on the drawn rankings converges to μ, provided* $0 \leq e^{-\phi} < 1$

The proof is omitted for space reasons, but can be found in the associated technical report, available on the authors' website.

THEOREM 3. *Let M be a Random Utility Model where the utility of candidate c_i is distributed as a Gaussian with mean $\eta_i \in \vec{\eta}$, and standard deviation σ, subject to the constraint that $\eta_i \neq \eta_j$ for all candidates $c_i \neq c_j$. The set of means $\vec{\eta}$ induce a ranking over the candidates $\mu = \mu_1, ..., \mu_{|C|}$, such that if $\mu_i = c_j$ and $\mu_l = c_k$, and $i < l$, then $\eta_j > \eta_k$. As the number of sample rankings drawn from M goes to infinity, the most probable sequence $MPS_{|C|}(T)$ of a restricted Markov Tree T of order k trained on the drawn rankings converges to μ.*

PROOF. The proof proceeds by induction. First, consider the distribution of first preferences among rankings drawn from M. For any $\mu_l \neq \mu_1$, the amount by which the subjective utility of μ_1 exceeds the subjective utility of μ_l during the generation of a particular ranking is a random variable distributed as $N(\eta_1 - \eta_l, 2\sigma)$. Since, by definition, $\eta_1 > \eta_l$, this amount is always positive in expectation, so $MPS_1(T)$ will converge to η_1 with infinite data.

Now, suppose that $MPS_j(T) = \mu_j \forall 0 < j < j+1$. It will be shown that $MPS_{j+1}(T) = \mu_{j+1}$. All $\mu_l \neq \mu_{j+1}$ that have not yet been fixed in $MPS_j(T)$ have the property $\eta_l < \eta_{j+1}$. If $j+1 < k$, then the amount by which the subjective utility of μ_{j+1} exceeds the subjective utility of μ_l during the generation of a particular ranking is a random variable distributed as $N(\eta_{j+1} - \eta_l, 2\sigma)$, always positive in expectation. □

An interesting implication of both results is that a Markov Tree of any order $k > 1$ will eventually converge so that its most probable sequence matches the centroid of a Mallows or the induced ranking of a RUM, although potentially large amounts of data will be required if there are many candidates involved because the proofs assume infinite data. This is significant because it demonstrates the ability of the Markov Tree model to accurately infer the parameters of other simpler models.

4. EVALUATION

To evaluate the proposed model, we implemented an algorithm for learning restricted Markov Trees. The system learns from top-t rankings using a sliding window of size k. The first $k-1$ preferences of the ranking are used to explicitly learn the parameters of the first $k-1$ nodes of the tree, while the k^{th} preference of the each window is used to learn the parameter $\theta_{S,c,k}$, where S is the first $k-1$ parameters of the window, and c is the k^{th}.

Figure 3 shows the convergence results for Mallows distributions with 5 candidates [2], and $e^{-\phi} = 0.5$, or 0.8. Each graph shows the convergence for four different values of k (1,2,3, and 4), starting with just 10 sequences drawn, and ending with a thousand or more after all or most of the models have fully converged. The figures are best viewed in colour. Figure 3 (left) show a very rapid convergence for all of the models when ballots are drawn from a fairly homogeneous population with a small number of candidates. Simpler models perform less well when there is insufficient data, but converge faster (likely because they have fewer parameters, and because they are able to extract more information per ballot). Figure 3 (right) shows a similar trend when the ballots are less homogeneous, though more data is required for convergence, and simpler models do not converge as quickly as before. However, a clear trend toward convergence is present even in the simplest (i.e. first and second) order models, bearing out the findings in Theorem 2. Similar results were produced when the experiments were repeated with various RUMs in place of the Mallows models.

[2] Similar results are readily obtained for 10 and 20 candidates also, but are not shown here for lack of space.

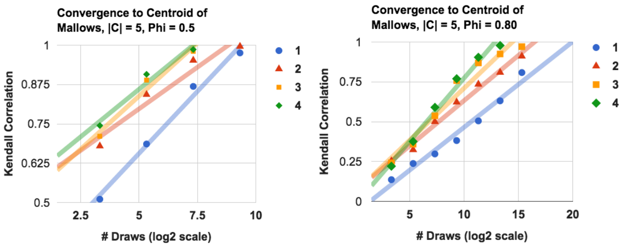

Figure 3: Convergence of $MPS_{|C|}(T)$ of Markov Trees of different depths to the centroid of a Mallows distribution with $|C| = 5$ and $e^{-\Phi} = 0.5$ (top) and $e^{-\Phi} = 0.8$ (bottom) as more data is drawn from the Mallows.

In addition, we evaluated the performance of the model on 10 data sets from the preflib.org repository of ranked data [16]. Seven sets are ranked ballots from elections of the leader Debian Project, a popular open source operating system. An eighth contains ranked ballots used to decide the project's logo. The remaining 3 sets were ranked ballots from the 2002 Irish National Election, from the Dublin North, Dublin West, and Meath districts. All sets consist of top-t rankings, with between 100 and 4000 usable ground truth rankings per set, and between 4 and 14 alternatives. We set $k = 3$ throughout, based on the results observed in the preceding experiments.

The sets do not contain a ground truth, since the winning ranking depends on the electoral system used and the treatment of missing preferences. As such, we evaluated our system by discarding all but the subset of the ballots which ranked all $|C|$ candidates. These ballots were then ablated in a way consistent with the original data (i.e. if x% of rankings in the original dataset ranked at least j candidates, then in the ablated version of the subset of complete rankings, x% of rankings also ranked at least j candidates, for every j), following the experiment design suggested by Doucette et al. [9]. The model was then trained on the ablated ballots, and used to impute them by generating the MPS completion of the top-t sequence starting with the ordering given by each ballot. The ranking of the candidates produced by applying particular electoral rules to the original, non-ablated, subset of complete data, was then compared to the ranking produced using the imputed rankings.

We compared performance under four popular social choice functions (Borda, $\frac{k}{2}$-Approval, Copeland, and Veto). Performance was measured both in terms of the Kendall correlation (i.e. a correlation proportionate to the Kendall-Tau Distance) between the ground-truth ranking and aggregate ranking produced by each model using the ablated ballots, and distance of the winning candidate in the ground truth from the top position in the aggregate rankings produced by each model. Comparisons are provided for the performance of a worst-case baseline and logistic regression approach (reproduced from [9]), and the Minimax Regret approach [13]. An approach based on the Maximum Likelihood Approach to voting [22] was also considered. We used an existing implementation [9] which samples possible completions under the assumption that (apriori) all completions are equally probable. This approach is denoted "random" in the results. Results are summarized for the Borda [3] social choice function in Figure 4, which shows mean error in the predicted position of the winner (top), and the overall Kendall correlation with the ground truth (bottom) for each of the 4 competing methods and the proposed Markov Tree approach. The Markov Tree performs best overall when the goal is to recover the true winning alternative, and as well as the other strong approaches when the goal is to recover the complete ranking of the candidates.

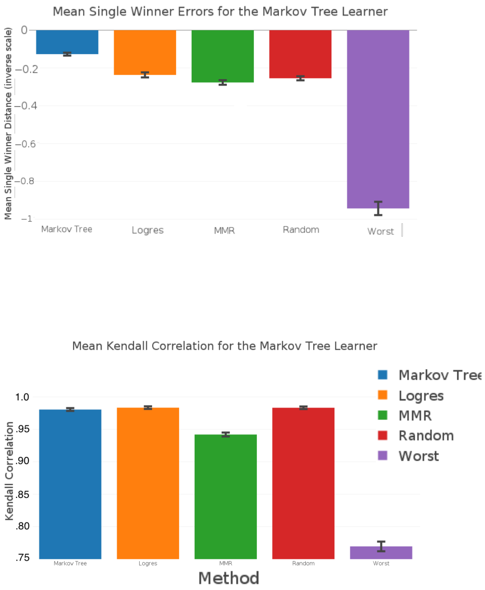

Figure 4: Summary of performance for the Markov Tree Learner compared with three competitors and the worst case model. (Top): Single Winner Error (inverse scale). Higher values indicate better performance, 0 is the best possible. (Bottom): Kendall Correlation. Higher values indicate better performance, 1 is the best possible.

As a final validation, we demonstrate a specific, simulated, agent society in which the newly proposed model offers a compelling advantage over existing approaches.

In many situations, the candidates an agent prefers may be intimately linked to the agent's general knowledge about the candidates, especially in applications like multiagent coordination. As a concrete example, we propose a scenario in which a swarm of robotic agents must collectively select a site to extract resources from. Mining sites and the headquarters of various mining companies were distributed on a two-dimensional grid. The companies wanted to mine resources from the sites, but are required to coordinate, and to operate their robots as a heterogeneous team or swarm. It is riskier to operate one's robots further from one's headquarters, but each company values the risk differently. Companies have different information about the quality of different sites. Suppose that viable mining sites are modelled as points distributed on the grid according to a two-dimensional Gaussian, centered at the origin, but that mining companies' headquarters are distributed uniformly at random throughout the space. Each site has an expected profit that will be received from mining it. The probability that a company knows enough about the site to estimate its expected profit is proportionate to the true value of the site, divided by distance between the company's headquarters and the site[4]. This creates a scenario where companies with headquarters near the origin will have lots of information, but companies further away will have much less. In

[3] Results on other rules omitted for space reasons. Relative performance of the Markov Tree was slightly better than on Borda under Copeland and Veto, and slightly worse under K/2-approval.

[4] This corresponds to the notion that information about more valuable sites will spread further, all else being equal.

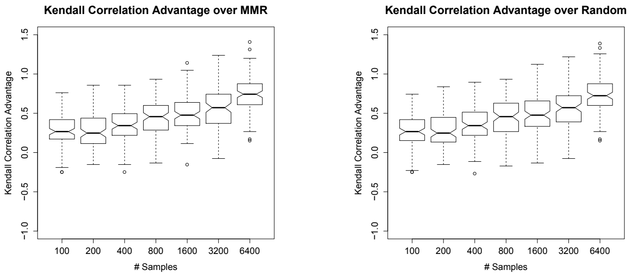

Figure 5: Improvement in Kendall Correlation from using the our approach in the coordination example problem under the K-Approval voting rule. Box plots show the distribution of advantage over 100 problem instances for different numbers of companies. Notches indicate a 95% confidence interval for the median.

this scenario, there is a strong correlation between the candidates agents prefer and the number of candidates agents are able to rank.

In our implementation, a space consisting of real-valued numbers between $(-5, 5) \times (-5, 5)$ was used for the grid. Mining sites were sampled from a Gaussian distribution with mean $(0, 0)$, and standard deviation of 1.5 along each major axis, with no covariance. If two sampled sites were within a Euclidean distance of 0.1 units of each other, one of the sites was re-sampled. The locations of company headquarters were sampled uniformly at random, with the constraint that no company could have headquarters within a Euclidean distance of 0.1 units of a mining site, or of any other company's headquarters. Each company had a utility function that placed the expected value of mining a site at $\frac{1}{r\delta}$, where r was the company's risk parameter, sampled from an exponential distribution with parameter $\lambda = 0.5$, and δ was the Euclidean distance between the company's headquarters and the mining site. A company knows the (correct) quality of the closest mining site with certainty. Given that a company knows the quality of the k closest sites already, it knows the quality of the next nearest site with probability $\frac{2}{1+e^{\delta}}$. For simplicity, no company knows the quality of more distant sites without also knowing the quality of less distant ones, and operates only 1 robot. A company's ballot ranks a mining site only if its quality is known. Candidates are ranked according to the company's utility function. A third order Markov Tree was used to impute the ballots. The **K-Approval** voting rule was used to decide the election. There were 15 mining sites, and the number of companies varied between 100 and 6400, with 100 problem instances being generated for each parameterization. The methodology from [9] was adopted to allow the Markov Tree to first learn a model of preferences, and then suggest an imputation, and then a winner. Results were also collected for MMR and the random approach. Figure 5 shows the advantage in Kendall Correlation for our approach over each competitor under $\frac{K}{2}$**-Approval**. The advantage is the Kendall Correlation between the outcome using the imputation approach and the ground truth, less the Kendall Correlation between the outcome using each competitor and the ground truth, on each of the 100 problems for each parameter setting. Values greater than 0 indicate advantage for our approach.

The advantage of using the Markov Tree approach is very pronounced in this example application, and for the 6,400 company case, begins to approach the distance between the correct outcome and an outcome sampled uniformly at random. Indeed, examining the raw results, the competing methods are often selecting outcomes that are worse than randomly selected ones. The reason for this is that the problem domain produces (by design) ballots with very extreme and asymmetric missingness. Most ballots will rank only one or two of the 15 candidates. The exception to this is for agents located near the centre of the grid, who will be close to many candidates, and so will rank many candidates. In the case of MMR, the extreme missingness gives wide latitude to make nearly any candidate win the election. This problem is not helped by having a larger number of ballots to work with, since these ballots are also highly incomplete, and while any candidates they order provide MMR with some information to constrain its selection of the winner, this is outweighed by the unconstrained choices for the ordering of the remaining candidates these extra ballots do not order. In essence, additional ballots add more degrees of freedom than they subtract, and MMR is picking a winner by solving a nearly unconstrained optimization problem. The random approach also fares rather poorly for this reason. There is a lot of noise in the relative positions of the candidates using only the given preferences. In contrast, our approach makes very efficient use of features in the given ballots to determine how agents would have voted. If two agents rank the same candidate highly, they must be close to one another in the grid, and so will have correlated opinions about more distant candidates. Effectively, each ballot improves the quality of the machine learning model, without increasing the complexity of the imputation task significantly. Given enough data, the Markov Tree will be able to predict an agent's preferences with a great deal of accuracy, given only their first few preferences. The example illustrates the great strength of our approach in domains where agents' preferences truly do exhibit structure of this kind.

5. CONCLUSION AND FUTURE WORK

In this paper we proposed a new model for learning preferences based on the notion of learning a distribution of *sequences* as a Markov model. This model is useful because it can both learn the most common sequences of preferences (allowing it to facilitate analysis of a preference distribution to make a social choice on its own), or generate completions of missing data elements (allowing it to be used in an imputation-based approach like [9], or to facilitate individual choices or recommendations). We provide a rigorous bound on the generalization error of the model when learning preferences that are drawn from a Markov model of similar depth, and proofs of eventual consistency with two common families of preference distributions. Our model is shown to recover the parameters of distributions of artificial data, and also to perform well when used to impute real world datasets for Social Choice. A simulation of a multiagent coordination problem also served to demonstrate a problem domain in which our model's advantages are highly pronounced. Our model is efficient and compact, allowing it to be learned quickly, and to be interpreted readily. In future, we plan to extend the model to include proofs of convergence to mixtures of Mallows distributions; and to automatically parametrize its own depth in response to the detected heterogeneity of rankings. We also plan to study the axiomatic properties of the model as a social choice function in its own right.

6. ACKNOWLEDGEMENTS

The authors gratefully acknowledge the support of the Natural Science and Engineering Research Council of Canada, and the Vanier Canada Graduate Scholars Program.

REFERENCES

[1] Horst Alzer. On some inequalities for the gamma and psi functions. *Mathematics of Computation of the American Mathematical Society*, 66(217):373–389, 1997.

[2] Kenneth J. Arrow. *Social choice and individual values*. Yale University Press, 12th edition, 2012.

[3] Kenneth J. Arrow, Amartya Sen, and Kotaro Suzumura. *Handbook of Social Choice & Welfare*. Elsevier, 2nd edition, 2010.

[4] Hossein Azari, David Parkes, and Lirong Xia. Random Utility Theory for Social Choice. In *Advances in Neural Information Processing Systems*, pages 126–134. NIPS Foundation, 2012.

[5] Hossein Azari Soufiani, David C. Parkes, and Lirong Xia. Preference Elicitation For General Random Utility Models. In *Uncertainty in Artificial Intelligence: Proceedings of the 29th Conference*. AUAI Press, 2013.

[6] Craig Boutilier, Ioannis Caragiannis, Simi Haber, Tyler Lu, Ariel D. Procaccia, and Or Sheffet. Optimal social choice functions: A utilitarian view. *Artificial Intelligence*, 227:190–213, 2015.

[7] Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel Procaccia. *Handbook of Computational Social Choice*. Cambridge University Press, 2015.

[8] Ioannis Caragiannis, Swaprava Nath, Ariel D. Procaccia, and Nisarg Shah. Subset Selection Via Implicit Utilitarian Voting. *Journal of Artificial Intelligence Research*, 58:123–152, 2017.

[9] John A. Doucette, Kate Larson, and Robin Cohen. Conventional machine learning for social choice. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*. AAAI Press, 2015.

[10] Allan Gibbard. Manipulation of schemes that mix voting with chance. *Econometrica: Journal of the Econometric Society*, 45(3):665–681, 1977.

[11] Maurice G. Kendall. A new measure of rank correlation. *Biometrika*, 30(1/2):81–93, 1938.

[12] Tyler Lu and Craig Boutilier. Learning Mallows models with pairwise preferences. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 145–152, 2011.

[13] Tyler Lu and Craig Boutilier. Robust approximation and incremental elicitation in voting protocols. In *Proceedings of the Twenty-First International Joint Conference on Artificial Intelligence*, pages 287–213. AAAI Press, 2011.

[14] R. Duncan Luce. *Individual choice behavior: A theoretical analysis*. Wiley, 1959.

[15] Colin L. Mallows. Non-null ranking models. I. *Biometrika*, 44(1/2):114–130, 1957.

[16] Nicholas Mattei and Toby Walsh. Preflib: A library for preferences http://www. preflib. org. In *Algorithmic Decision Theory*, pages 259–270. Springer, 2013.

[17] Marina Melia and Harr Chen. Dirichlet process mixtures of generalized mallows models. In *Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence (UAI2010)*, 2010.

[18] Robin L. Plackett. The analysis of permutations. *Applied Statistics*, 24(2):193–202, 1975.

[19] Ariel D. Procaccia and Nisarg Shah. Optimal Aggregation of Uncertain Preferences. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pages 608–614. AAAI Press, 2016.

[20] Charles P. Quesenberry and D. C. Hurst. Large sample simultaneous confidence intervals for multinomial proportions. *Technometrics*, 6(2):191–195, 1964.

[21] Mark Allen Satterthwaite. Strategy-proofness and Arrow's conditions: Existence and correspondence theorems for voting procedures and social welfare functions. *Journal of Economic Theory*, 10(2):187–217, 1975.

[22] Lirong Xia and Vincent Conitzer. Determining Possible and Necessary Winners under Common Voting Rules Given Partial Orders. 41:25–67, 2011.

[23] Lirong Xia and Vincent Conitzer. A maximum likelihood approach towards aggregating partial orders. In *Proceedings of the Twenty-First International Joint Conference on Artificial Intelligence*, pages 446–451. AAAI Press, 2011.