

Conventional Machine Learning for Social Choice

John A. Doucette, Kate Larson, and Robin Cohen

David R. Cheriton School of Computer Science

University of Waterloo

200 University Avenue West

Waterloo, ON, Canada

{j3doucet,klarson,rcohen}@uwaterloo.ca

Abstract

Deciding the outcome of an election when voters have provided only partial orderings over their preferences requires voting rules that accommodate missing data. While existing techniques, including considerable recent work, address missingness through circumvention, we propose the novel application of conventional machine learning techniques to predict the missing components of ballots via latent patterns in the information that voters are able to provide. We show that suitable predictive features can be extracted from the data, and demonstrate the high performance of our new framework on the ballots from many real world elections, including comparisons with existing techniques for voting with partial orderings. Our technique offers a new and interesting conceptualization of the problem, with stronger connections to machine learning than conventional social choice techniques.

Introduction

Sometimes, as in national elections, hiring committees, and resource allocation problems, a group of agents (human or computational) will need reach common agreement on a course of action. This is often best accomplished by taking a *vote*, in which the agents (voters) express their preferences about which action to take in some common format. A *voting rule* is then applied to aggregate the preferences in some way, producing an *outcome*. For example, in an American Presidential election, voters cast a ballot indicating their most preferred choice, and the ballots are aggregated to produce a sum total of the ballots cast for each candidate (and then further aggregated to produce the electoral college tallies), and the outcome action is that the candidate with the most votes will be asked to serve as President for the next four years. There are many applications of voting within artificial intelligence, including multiagent resource allocation (voting on which resources to assign to which agents); combing ensembles in machine learning; and coordination strategies for multiagent systems in general.

Although voting is, in general, a desirable way to handle group decision making, some voting systems may yield better outcomes than others, and consequently, there exist a great many possible voting rules. Most of these rules (including all those considered in this work) are based on the

concept of *ranked ballots*. Ranked ballot voting systems require voters to specify not only their favourite alternative, but a linear ordering over all the alternatives. This format allows many nice properties to be satisfied. For example, with this extra information, one can select “compromise” alternatives, or avoid picking polarizing alternatives that are the first choice of one portion of the electorate, but the last choice of everyone else. However, this extra information comes at a price: voters must be able (and willing) to express their complete preferences.

In some domains, even though the extra ranking information could allow for a better outcome, voters may be unable to provide it. For instance, in multi-agent resource allocation, we might prefer compromise solutions, under the assumption that these will be less unequal than solutions which are the absolute favourites of individual voters. However, in multi-agent resource allocation problems, the set of *possible* allocations is usually combinatorially large. Most voters (humans or agents) would have difficulty expressing a linear ordering over such a large set. Similarly, certain voting rules have extremely desirable properties for national elections, and are able to operate efficiently, but require voters to rank hundreds of alternatives to work effectively (e.g. Skowron et al. 2013). Finally, in domains like committee meetings where votes are public, voters may not wish to supply complete information. Telling one of your colleagues that their pet proposal is your least preferred alternative may have greater political impact than merely abstaining.

Domains like the above examples have spurred considerable work on deciding elections *without* complete preference information. The simplest, and perhaps oldest, technique is just to treat missing information as abstentions. For instance, the popular Single Transferable Vote scheme (Tideman 1995) (notably used in Australian national elections) is often applied to incomplete ballots. If a voter only specifies their top three preferences, and all three are eliminated during runoffs, then the voter is treated as abstaining from the remainder of the election.

This approach is elegant in its simplicity, but leaves something to be desired in its implications. Voters who fail to complete their ballots are, in many cases, not so much ambivalent about the alternatives they omit as uninformed, or else unwilling to reveal their opinions. Rare indeed is the case where a person truly has no preferences at all regarding

the relative quality of two distinct items. Instead, we would like to decide the election in a way that is consistent with voters' true *hidden* (or unknown) preferences, using only the information they have chosen (or are able) to reveal.

We are not the first to consider this approach. Notably, Condorcet's Jury Theorem (See Black 1987 for a detailed summary) first proposed the idea of treating ballots as noisy observations of some hidden truth, and of using statistical techniques to infer the truth from these observations. This is sometimes called the *Maximum Likelihood Estimation* (MLE) approach to voting. Recently this approach has been developed to address partial preferences. Xia and Conitzer (Xia and Conitzer 2011) for instance, developed a model that assumes each partial ballot is a noisy observation of several pairwise comparisons of candidates. Thus, ranking A before B on one's ballot is treated as a noisy observation that A is a better alternative than B . However, determining who has won an election under this rule is computationally expensive, and the rule assumes that ranking A before B on one's ballot cannot provide information about whether A might also be preferred to some other, unranked, candidate C , when in practice it often can¹. Other related work includes Lu and Boutilier's work applying techniques from robust optimization to the problem (Lu and Boutilier 2011; 2013). In this approach, selecting a winner from an election with partial information becomes an optimization problem. The goal is to select a winner according to voting rule from a commonly applied family, such that an adversarial completion of the ballots produces the least dissatisfaction with the choice. This avoids selecting candidates that most voters (secretly or unknowingly) despise.

In this work, we study the possibility of using existing, standard, algorithms from machine learning to decide elections with missing information. These techniques have the advantage of being able to detect (in principle) arbitrary patterns contained in the ballots voters cast, and so avoid the assumptions of pairwise independence found in Xia and Conitzer's approach. They also aim to compute the most likely outcome, rather than the safest one, and so are less conservative, and should be suited to a somewhat different set of applications than Lu and Boutilier's system.

A New Approach

The proposed technique splits the problem into two portions, each of which has been studied extensively on its own. In the first portion, existing machine learning techniques are used to generate a completion of the partial ballots voters cast. In the second portion, existing voting rules are applied to the completed ballots to select the winner. This partitioning of the problem has several advantages. First, existing techniques in both fields are generally well studied and widely implemented. If the proposed system works well, then this would greatly simplify the problem of deciding elections with partial information by reducing it to the combination of existing approaches. Second, by phrasing the problem of

¹For instance, in national elections, a ranking of the form Communists \succ Socialists \succ Centrists actually says a great deal about what this voter thinks of the Conservative candidate

deciding elections with partial preferences as a more general machine learning problem, it allows us to consider the problem from a novel and interesting standpoint.

Completing Ballots

The process we propose for completing ballots is Imputation via Classification. Imputation is the process of replacing missing data (often referred to as "missingness") with a carefully selected guess at the missing value (Schafer 1999). For instance, if a person's age is missing from an otherwise complete questionnaire, a very simple imputation technique would replace the missing age with the average age of the other questionnaire-takers. A more sophisticated technique would be to use the age of another questionnaire-taker with similar demographic characteristics, especially those known to be correlated with age.

While imputation techniques are widely used in other domains with missingness, we are unaware of any direct application of imputation to the problem of missingness in election ballots. Intuitively, however, imputing the missing components of these ballots is possible because real-world votes have underlying structure: not every ordering of alternatives is equally likely to appear on a ballot. By exploiting this fact, we hope to provide outcomes more reflective of the electorate's desires than techniques that treat missing components independently of the content of the ballots.

Classification algorithms operate over a data matrix X and label vector Y , with an equal number of rows. The algorithm finds a model (a function) c such that $c(X)$ is a vector of labels, and a cost function $G(c, c(X), Y)$ is minimized. The set of possible functions from which c may be selected is called the *hypothesis space*, and is usually restricted in a way intended to ensure that c captures general patterns in the data that will allow it to accurately predict the labels of previously unseen datapoints in future. G often has some related component based on the structure of c as well as the number of differences (errors) between $c(X)$ and Y . For instance, G might penalize functions which are not very "smooth" (regularization).

In the context of this work, we say that a social choice problem consists of selecting from among a set of alternatives O , according to a set of N ballots collectively represented by the vote matrix R . The vote matrix is organized so that each row represents the preferences of a user, with their most preferred preference in the first column, and each following preference placed in a following column. For instance $R_{i,j}$ would be the j^{th} most preferred candidate of voter i , an element of O . If voter i has specified only j preferences, then $R_{i,k} = \emptyset, \forall k > j$.

A given ballot (row) $r_i \in R$ thus represents of a total ordering over an arbitrary subset of O . We assume the elements of O that are not on the ballot are of lower rank in the voter's preferences than those candidates that were ranked².

Finally, we denote the vote matrix formed by the first j columns of R with R_j . For instance, R_1 denotes the first preferences of every ballot, while R_2 denotes the first *and* second preferences of every ballot. We assume that there are

²This is equivalent to *top-t* orderings (Baumeister et al. 2012).

at least two candidates ($|O| \geq 2$), at least two ballots ($N \geq 2$), and that every ballot has at least one candidate ranked on it ($|r_i| \geq 1, \forall 1 < i \leq N$).

Our system begins by extracting R_2 , the first and second preferences of every voter's ballot. Some ballots may state only a single preference, while others state two. Taking the subset of R_2 which is complete ($R_{c_2} = \{r \in R_2 \mid |r| = 2\}$), we use a classification algorithm C to train a classifier $c_2 = C(R_{c_2})$, which predicts the second preference of each ballot from their first preferences. The data matrix X used for classification is some function Φ of the first preferences on every ballot in R_{c_2} . The label vector Y is the second preference of every ballot in R_{c_2} . Once c_2 has been computed, we use it to impute $R_2 \setminus R_{c_2}$, generating a complete ballot matrix of two columns. We call this imputed ballot matrix R'_2 .

The process is then extended to the next column of R . We take the ballot matrix of the first, second and third preferences (R_3), and build a classifier c_3 on R_{c_3} . We can then use c_2 to impute all missing second preferences, and c_3 to impute $R_3 \setminus R_{c_3}$ and generate R'_3 . We can iterate this process until the generation of $R'_{|O|} = R'$, an imputation of the entire ballot matrix. A winning alternative can then be selected by applying any standard voting rule S to the imputed matrix: $S(R') = o'$. We formalize this process in Algorithm 1.

Algorithm 1 Algorithm for selecting a winning alternative in an election with partial ballots using imputation.

```

1: function IMPUTE_BALLOTS(O,R,S,C)
2:   for all  $2 \leq j \leq |O|$  do
3:     LET  $c_j \leftarrow C(R_{c_j})$ 
4:     SET  $missing \leftarrow R_j \setminus R_{c_j}$ 
5:     for all  $2 \leq k \leq j$  do
6:        $missing \leftarrow c_k(missing)$ 
7:     end for
8:     SET  $R'_j \leftarrow R_{c_j} \cup missing$ 
9:   end for
10:  RETURN  $o' \leftarrow S(R'_{|O|})$ 
11: end function

```

The correctness of the imputations produced by our algorithms depend on the selection of classification algorithm C and on the amount of data available (i.e. the total number of *complete* preferences at each step), as well as the way in which voters' preferences are generated. For instance, learning a full joint distribution $P(R_{i,j} \mid R_{i,x} \mid 0 < x < j)$ for preferences will converge with certainty to the correct distribution of preferences irregardless of how voters' preferences are generated, but the error in its estimates will drop as $O((\frac{N}{|O||\sigma|})^{-0.5})$, which is impractically large for most problem domains. Using other classification algorithms allows convergence to some model with much less data, but assumes that the ballots were generated by a particular process. Consequentially, we can only guarantee the performance of the algorithm in terms of the closeness with which the chosen classification algorithm is suited to learning whatever process has generated the missing ballots. We provide a more formal description of this relationship in the Performance Guarantees section below.

Once ballots are completed, any voting rule capable of op-

erating over complete preferences may be applied to them to determine the winner. In this paper we primarily consider Borda's rule. Borda's Rule operates by selecting the candidate with the highest *mean* position on the ballots that were cast. Each candidate is given a Borda score computed as $BC(o) = \sum_i |O| - Pos(R_i, o)$ where $Pos(R_i, o)$ is the position of candidate o on ballot R_i (that is, if $R_{i,j} = o$, then $Pos(R_i, o) = j$). The candidate with the highest score wins. The rule is used to select compromise candidates; however, even if a majority of the electorate have the same first preference, that candidate will not necessarily be declared the winner (i.e. it does not satisfy the *majority* criterion). We selected the Borda rule because it is simple to implement, and because it degrades in an easily understandable way when errors are introduced into voters' ballots: swapping the positions of two candidates on a voter's ballot affects their scores in a way directly proportionate to how much the voter preferred one candidate over the other.

Imputation as Social Choice

Our motivation for Algorithm 1 comes from the interesting observation that imputation can be viewed as a form of social choice in its own right. The partial ballots can be viewed as voting in favour of certain potential completion policies over others, which provides a nice grounding for the use of imputation algorithms. We formalize this idea as the following theorem. (Proof omitted for space reasons)

Theorem 1 *Every classification algorithm, when used to impute missing information in ballots via the process described in Algorithm 1, is equivalent to a social choice function.*

The implication of Theorem 1 is that using classifiers for imputation entails holding a *vote* on the treatment of missingness in the data. Some chained classifiers are equivalent to voting rules with fairly intuitive interpretations (for instance, random dictator is a perfectly valid voting rule, and corresponds to a form of 'hot-deck' imputation).

A number of results follow directly from this theorem. For instance, every non-random classification algorithm will elicit strategic play from voters (by the Gibbard-Satterwaith theorem), but can also provide computational resistance to manipulation provided the scoring rule used for the actual election S is similarly resistant. Although we do not explore this result further here, there are many promising avenues of research following from it.

Performance Guarantees

In this section we formalize the process by which the algorithm works, and the various components of the algorithm, allowing us to provide a concrete performance bound.

We define a positional scoring rule as a voting rule with an $|O|$ -dimensional scoring vector \vec{v} . The score of a candidate o is equal to $S(o, R) = \sum_i v_{Pos(o, R_i)}$ where $Pos(o, R_i)$ is the position of candidate o on ballot R_i , and v_x is the x^{th} element of v . Many commonly used rules are positional scoring rules. For example the Borda count is a positional scoring rule with $v_i = |O| - i$. A monotonic positional scoring rule satisfies that $v_x \geq v_{x+1}$ for all $0 < x < |O|$.

Ballots are generated by a parameterized process \mathcal{M} . \mathcal{M} is defined by a set of distributions over possible ballots $\pi_o|o \in O$, and if the “correct” outcome of the election is \hat{o} , then ballots are generated by sampling i.i.d. from $\pi_{\hat{o}}$. A positional scoring rule S is *consistent* with a ballot generating process \mathcal{M} if, for every $o \in O$, applying S to an infinite set of ballots sampled from π_o causes S to select o as the winner of the election with probability one.

Top- t ballots are generated by applying a process \mathcal{N} to a profile of ballots. \mathcal{N} is defined by a distribution over possible ablations of individual ballots. We assume \mathcal{N} is *neutral*, meaning that the probability of ablating the bottom k positions on a ballot does not depend on the order of candidates on the ballot. We define $\mathcal{N}(k)$ to be the probability of ablating *at least* the last k positions from the ballot.

Under these assumptions, we can put a bound on the minimum performance of the classifier required to recover the correct outcome when S is a monotonic positional scoring rule. In particular, we show that if for any $o \in O, o \neq \hat{o}$, $S(\hat{o}, R) - S(o, R) + \sum_{1 \leq k \leq |O|} [P(\text{Pos}(\hat{o}, R_i) = k) \sum_{1 \leq j \leq k} (\mathcal{N}(j) - \mathcal{N}(j+1)) (v_k - \frac{v[\epsilon|O|+(1-\epsilon)j] + v[\epsilon|O|+(1-\epsilon)j]}{2})] - \sum_{1 \leq k \leq |O|} [P(\text{Pos}(o, R_i) = k) \sum_{1 \leq j \leq k} (\mathcal{N}(j) - \mathcal{N}(j+1)) (v_k - \frac{v[(1-\epsilon)|O|+\epsilon j] + v[(1-\epsilon)|O|+\epsilon]}{2})] \geq \text{bias}(\hat{o}, o, c, R)$, where $\text{bias}(\hat{o}, o, c, R)$ is the classifier’s bias for \hat{o} over o , and ϵ is a tolerance parameter, then our proposed technique will fail to find the correct outcome in expectation.

That is, if difference between classifier’s *bias* (the expected change in the score of a candidate as a result of ablating and then imputing the ballots with that classifier) for the winning candidate and every other candidate exceeds the margin of victory between those two candidates under assumptions about the damage ablation has done (parameterized with ϵ), then we can be sure the winner we selected would have been the true winner in the original ballot. This means that the technique’s effectiveness depends on the margin of victory in an election held using the true (hidden) preferences of the voters, and that the more accurate the classifiers are, the closer that margin can be before a mistake is possible. (Proof omitted for space reasons) We conjecture that a similar relationship exists for the process in Algorithm 1 for any *Generalized Outcome Scoring Rule* (Xia 2014), a much broader class of voting rules.

Validation

In this section, we present the application of our imputation based approach to social choice to datasets from the preflib.org repository (Mattei and Walsh 2013). We show that using imputation to select the winner produces accurate results under a missingness process \mathcal{N} derived empirically from the data itself.

We examined data from a total of eleven elections from the Irish and Debian datasets³, which are both comprised of real-world ballots with ranked preference formats. In both sets, voters were able to omit preferences if desired. The Debian set contains election data for the seven leadership elections from 2002-2012, and the vote on the Debian Project logo. The Irish set contains the ballots from the Dublin

³<http://www.preflib.org/election/{irish,debian}.php>

North, Dublin West, and Meath constituencies during the 2002 national election. Collectively, these elections provide good diversity both in terms of the number of candidates running, and the degree of missingness in the voters’ preferences. The Debian sets typically contain between 4 and 8 candidates, and between 50 100 and 400 complete ballots. The Irish sets contain between 8 and 14 candidates, and each have around 4000 complete ballots.

Preprocessing: Each dataset was processed as follows: First, all ballots in the original set which were incomplete were discarded, after learning an empirical distribution of missingness from them. The remaining ballots were ablated according to either an empirically learned distribution of missingness, and then converted into a set of ballot matrices $R_2 \dots R_{|O|}$. For each such matrix, we dropped all incomplete ballots, and used the last preference as a vector of labels for training a classifier. The other preferences on the ballot were converted into a numeric matrix of features, encoding for each candidate whether it appeared on the ballot, and at which position it appeared; and encoding for each pair of candidates what the relative ordering of the two candidates was (if known), and how far ahead of each other they were on the ballot.

Experiment: For each of the 11 elections, we first empirically measured the distribution of missing information (i.e. the fraction of voters with at least k candidates on their ballot for every k). We then dropped all the ballots with missing information to produce the ground truth set (the ballots where we know exactly how every voter ranked every candidate). Using the measured distribution of missingness, we generated 100 random ablations of this ground truth set. For each ablation, we used Algorithm 1 to decide the outcome of the election. We used one-vs-all classification (OVA) (Rifkin and Klautau 2004) with L1 regularized logistic regression as the base classifier to learn the imputation model for each position on the ballot. This is a standard and relatively simple approach to classification when there are more than two possible classes (in this case, each ballot can belong to one of $|O|$ possible classes: the possible values of the next candidate on the ballot). We name this combination *logres*.

For each dataset, we measured the performance of our imputation-based system in several different ways. First, we measured distance of the winner selected after applying our system from the true winner in the ground truth ranking. We call this the “single winner” distance. For example, a system that recommended the true third place candidate as the winner would have a “single winner” distance of 2, while one that recommended the correct winner would have a distance of 0. Second, we measured the Kendall correlation between the overall ordering of the candidates produced using our method, and the ordering produced using the ground truth rankings. The Kendall correlation is the number of pairwise comparisons between candidates on which the rankings agree, less the number on which they disagree, normalized by the total number of possible comparisons $(\frac{2 \sum_{x \in O} \sum_{y \in O \setminus x} I(t_1(x,y) \wedge t_2(x,y)) - I(t_1(x,y) \wedge t_2(y,x))}{|O|(|O|-1)})$ for rankings t_1 and t_2 . Although the Kendall correlation is a different distance metric than the one on which the

Borda count is based, we use it here because it provides information about the similarity of the two rankings on the whole, rather than just the positions of the first place candidates. The Kendall correlation ranges from -1.0 (one ranking is the reverse of the other), to 1.0 (the rankings are the same). A randomly generated ranking has an expected Kendall correlation of 0 with any other ranking. We also measured the performance of several controls, namely the minimax regret technique of (Lu and Boutilier 2011) (a state-of-the-art competing method) and a worst-case method that completed the ballots in a fashion exactly opposite to the correct overall ordering (i.e. if in the true election $A \succ B \succ C$, then unassigned positions on ballots are assigned preferentially first to C , then B , then A).

To verify that the ground truth sets were similar to the original datasets from which they were drawn, we also measured the single winner distances and Kendall correlations between the outcome selected using the original set and each ablated set, using the version of the Borda scoring rule where candidates not appearing on a partial ballot received no points from that ballot. The ablated version of the ground truth set produced exactly the same outcome as the ground truth on 10 of the eleven sets, with an error of 2 on the remaining Debian 2005 set. Kendall correlations were between 0.85 and 0.9 for the Irish sets, and averaged 0.85 on the Debian sets as well. This indicates that the ablated ground truth sets are highly representative of the original data from which they are generated, and thus representative of real world elections. All differences are statistically significant at a Bonferroni corrected confidence level of 95%.

Results: The proposed, imputation based, approach to social choice relies on machine learning algorithms providing reasonable imputations of user ballots. This is by no means certain, as the resulting machine learning problems are quite difficult. For example, the Dublin North election has twelve candidates and only about 40% of the total preference information. This results in a 12-class classification problem (i.e. a model needs to predict which of 12 possible candidates a given voter would prefer) with many features missing on any given record. Despite this, logres performs very well, strongly validating our approach.

The single winner distances of our new technique (logres), are compared to those of the two control techniques (MMR, worst-case) in Table 1. Sets where the winner was fully determined after ablation (where no possible imputation could change the outcome) are not shown. The new technique performs very well, finding the correct winner in 100% of cases on all three of the large Irish datasets, and all but one of the Debian sets. On the Dublin West and Meath datasets, we find the correct winner in 100% of cases, even though MMR consistently picks the second or third place candidates instead. The only set where our system performs less than perfectly is on the Debian 2007 set. Here, in about 25% of runs, we select the second place candidate instead of the first place one. Close examination of this set provides an explanation. The margin of victory for the first place candidate over the second place candidate averaged just 0.002% of their respective Borda Scores (equivalent to inverting adjacent pairs on approximately 10 ballots). This is well within

| | logres | MMR | worst-case |
|-------------|--------|-------|------------|
| Debian 2005 | 0.000 | 0.000 | 1.350 |
| Debian 2007 | 0.240 | 0.000 | 0.770 |
| Debian Logo | 0.000 | 0.000 | 0.050 |
| North 2002 | 0.000 | 0.000 | 6.200 |
| West 2002 | 0.000 | 1.990 | 2.000 |
| Meath 2002 | 0.000 | 1.000 | 11.700 |

Table 1: The single-winner distances for the proposed system (logres), and two comparison methods. 0 indicates a perfect performance.

the classifier’s reported error rate on the dataset, and so it is not surprising that errors can occur, given the performance bounds derived earlier. The reported classifier performance together with a narrow margin of victory in both the ablated and imputed ballots all indicate that this set would be a good candidate for the model to suggest a tie or request more information from the voters, allowing for an appropriate response and recovery when used in practice. It is notable that, despite the extreme difficulty of the set, we give the correct result in more than two thirds of cases where an error is possible, which is still acceptable for some applications, like votes on hiring committees, where occasionally picking a close second candidate is an acceptable outcome. The margin of the winner over the second place candidate is also very tight (under 1%) on the West and Meath sets, but here, our technique consistently finds the correct outcome while MMR and the worst-case model do not.

Table 2 shows the Kendall Correlation for our technique along with the comparison methods. Again, we omit the datasets where the entire ordering is unchangeable after ablation, and show only the remaining 8 sets where errors are possible. The new technique performs very well, and achieves higher Kendall correlation than MMR on four of the eight datasets, and ties with near perfect performance on two more. On the remaining two sets (Dublin North 2002 and Meath 2002), our correlation values are still very high, but are slightly worse than those for MMR. However, these are unquestionably the most difficult sets, as illustrated by the possibility for very poor performance by the worst-case method, and our performance is still very good (Kendall Correlations of 0.72 and 0.82 respectively indicate that 86% and 91% of the pairwise orderings of candidates are correct). MMR’s better performance likely results from the very high ablation rates for the lower positions on the ballots in these sets, which makes them challenging to learn. Dublin North offers about 400 ballots to learn the relationships between 12 candidates for many of the lower positions on the ballots, while Meath is even more severely ablated, and has 14 candidates. Since logistic regression needs sufficient data to learn patterns from, the combination of limited data availability, many candidates to choose from, and some candidates receiving very few votes at all from which to learn, all contribute to the performance reduction. Nevertheless, the new system still recovers the overwhelming majority of the correct ordering, even on these more difficult sets, including the correct ordering for all three of the closely ranked top candi-

| | logres | MMR | worst-case |
|-------------|--------|-------|------------|
| Debian 2005 | 0.999 | 0.998 | 0.991 |
| Debian 2006 | 0.999 | 0.999 | 0.939 |
| Debian 2007 | 0.985 | 0.982 | 0.843 |
| Debian 2010 | 1.000 | 1.000 | 0.996 |
| Debian Logo | 0.926 | 0.901 | 0.584 |
| North 2002 | 0.821 | 0.959 | -0.259 |
| West 2002 | 0.889 | 0.868 | -0.038 |
| Meath 2002 | 0.727 | 0.910 | -0.627 |

Table 2: The Kendall Correlation for the proposed system (logres), and two comparison methods. 1 indicates perfect correlation, -1 perfect anti-correlation.

dates which MMR and fails to order correctly. We conclude that logres may be most useful in applications where the exact ordering of relatively unpopular candidates is less vital, though generally it performs very well.

We also considered two other voting rules: $\frac{|O|}{2}$ -approval (in which voters may cast a single, equally weighted, vote for up to half the candidates) and Copeland (in which the candidate winning the most pairwise contents is declared the winner). Across all combinations of datasets and rules where errors appeared, our model has an average single winner distance 1.24 positions lower than MMR. We also recover more of the true ordering than MMR. The average Kendall correlation on sets where mistakes were made by either model was 0.83 for MMR and 0.90 for logres. This demonstrates that our model can function well under many voting rules, not just the Borda count. Run times were consistently under 5 minutes per ablated dataset, with up to 5,000 ballots and 14 candidates per set, on a contemporary desktop machine. Runtimes scale as the product of $|O|$ and N , and can be improved with the use of sampling techniques when N is large. In summary, imputation based social choice is shown to be a viable and fast technique, applicable to real world problems and capable of outperforming existing state-of-the-art methods on many datasets.

Related Work

While our approach to solving the problem of interest is novel, there exists considerable prior work on this problem, much of it couched in terms of *vote elicitation*.

Interest in vote elicitation extends back to at least the 2002 work of Conitzer and Sandholm (Conitzer and Sandholm 2002), who considered the problem of eliciting preferences from *strategic* voters, that might not wish to reveal their true preferences. A flurry of more recent work has examined the practical aspect of the problem, emphasizing elicitation of more informative preferences. Kalech et al.’s work (Kalech et al. 2011) showed that, although complete information is required to make optimal decisions in the worst case, many real world applications yield solutions with far less preference information. Similarly, Oren et al. (Oren, Filmus, and Boutilier 2013) considered the number of top-t style queries (where voters are repeatedly asked for their next highest preference) required to find the underlying global preference

ordering given certain assumptions about the underlying distribution of voter preferences, while Soufiani et al. (Soufiani, Parkes, and Xia 2013) examined a similar problem for general random utility models. Our work differs from this recent context insofar as it does not recommend a particular elicitation strategy for voters, but instead works with the preferences it has been given to accomplish the same goal.

The work most similar ours is that of Lu and Boutilier (Lu and Boutilier 2011), who proposed the use of minimax regret as a heuristic measure for selecting a winner from partial preferences. Here, each candidate is considered in turn. For each candidate, a completion of the ballots making the candidate as undesirable as possible is computed. The candidate most desirable in spite of their corresponding worst-case completion becomes the winner.

Irrespective of the performance obtained by this strategy, our system has an advantage in applications where it is important that skeptical voters *accept* the system’s result. Our system provides a completion of each ballot that is as *consistent* as possible with the patterns of voting that we observed, while MMR decides the election on the basis of a worst-case imputation, which may be quite unlikely, and which skeptical voters might have more difficulty accepting.

Conclusions and Future Work

We have presented and validated a novel approach to the problem of social choice with partial preferences. Our new approach imputes the missing components of the ballots using patterns inferred from the ballots themselves. This allows conventional voting rules for complete preferences to be applied directly, and provides a ranking based on a plausible completion of the ballots, rather than a conservative worst-case arrangement. We showed that the process of picking an imputation is itself a form of implicit social choice, which could allow many computational hardness results to be directly applied to the new model, and that it performs well on a large number of real-world election datasets. We also performed a direct comparison with the minimax regret system of Lu and Boutilier (Lu and Boutilier 2011), showing that our preliminary model picks the correct winner significantly more often, and exhibits generally low error rates on the rest of the ordering as well.

An especially interesting component of our work is the fusion of conventional social choice with standard techniques from machine learning. There are strong parallels between these fields, and much room for similar future work (See Xia’s visions paper Xia 2013.). Some interesting extensions might include the application of machine learning models that are specifically designed for problems with a large amount of class imbalance; controlled studies with artificial datasets to provide further insight into the best conditions to apply our new technique; integration of other of preference learning algorithms within our system (Kamishima, Kazawa, and Akaho 2011); and an exploration of the optimal policy for a tie to be declared when using our technique, utilizing the classifier’s error on validation data, and the margin of victory both before and after imputation.⁴

⁴We acknowledge NSERC Canada and the VCGS program.

References

- Baumeister, D.; Faliszewski, P.; Lang, J.; and Rothe, J. 2012. Campaigns for lazy voters: Truncated ballots. In *Proc. AAMAS-2012*, 577–584.
- Black, D. 1987. *The theory of committees and elections*. Springer.
- Conitzer, V., and Sandholm, T. 2002. Vote elicitation: Complexity and strategy-proofness. In *Proc. AAAI/IAAI-02*, 392–397.
- Kalech, M.; Kraus, S.; Kaminka, G. A.; and Goldman, C. V. 2011. Practical voting rules with partial information. *Autonomous Agents and Multi-Agent Systems* 22(1):151–182.
- Kamishima, T.; Kazawa, H.; and Akaho, S. 2011. A survey and empirical comparison of object ranking methods. In *Preference learning*. Springer. 181–201.
- Lu, T., and Boutilier, C. 2011. Robust approximation and incremental elicitation in voting protocols. In *Proc. IJCAI-11*, 287–293. AAAI Press.
- Lu, T., and Boutilier, C. 2013. Multi-winner social choice with incomplete preferences. In *Proc. IJCAI-13*, 263–270. AAAI Press.
- Mattei, N., and Walsh, T. 2013. PrefLib: A Library of Preference Data. <http://www.preflib.org>. [Online; accessed June 5th 2013].
- Oren, J.; Filmus, Y.; and Boutilier, C. 2013. Efficient vote elicitation under candidate uncertainty. In *Proc. IJCAI-13*, 309–316.
- Rifkin, R., and Klautau, A. 2004. In defense of one-vs-all classification. *The Journal of Machine Learning Research* 5:101–141.
- Schafer, J. L. 1999. Multiple imputation: a primer. *Stat. methods in medical research* 8(1):3–15.
- Skowron, P.; Yu, L.; Faliszewski, P.; and Elkind, E. 2013. The complexity of fully proportional representation for single-crossing electorates. In *Algorithmic Game Theory*. Springer. 1–12.
- Soufiani, H. A.; Parkes, D. C.; and Xia, L. 2013. Preference elicitation for general random utility models. In *Proc. UAI-13*, 596–605.
- Tideman, N. 1995. The single transferable vote. *The Journal of Economic Perspectives* 27–38.
- Xia, L., and Conitzer, V. 2011. Determining possible and necessary winners under common voting rules given partial orders. *Journal of Artificial Intelligence Research* 41(2):25–67.
- Xia, L. 2013. Designing social choice mechanisms using machine learning. In *Proc. AAMAS-13*, 471–474. International Foundation for Autonomous Agents and Multiagent Systems.
- Xia, L. 2014. Statistical properties of social choice mechanisms. In *Proc. Fifth International Workshop on Computational Social Choice (COMSOC-14)*.