Approximate Winner Selection in Social Choice with Partial Preferences

John A. Doucette
University of Waterloo
200 University Avenue West
Waterloo, Ontario, Canada
j3doucet@cs.uwaterloo.ca

Kate Larson
University of Waterloo
200 University Avenue West
Waterloo, Ontario, Canada
klarson@cs.uwaterloo.ca

Robin Cohen
University of Waterloo
200 University Avenue West
Waterloo, Ontario, Canada
rcohen@uwaterloo.ca

ABSTRACT

Deciding the outcome of an election when voters have provided only partial orderings over their preferences requires voting rules to accommodate missing data. In the worst case, the missing values from the data can often be completed in a way that renders the outcome selected by conventional social choice methods undesirable. Current work using minimax optimization techniques to select the winner can mitigate the impact of a worst-case completion, but may select winners that are *unlikely*. We propose a novel application of machine learning techniques to predict the missing components of ballots via latent patterns in the information that *is* provided. Although we do not offer a worst-case guarantee on performance, 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 ten real world elections, including comparisons with existing techniques for voting with partial orderings. Our technique offers a new conceptualization of the problem, with stronger connections to machine learning than conventional social choice techniques, and with a stronger emphasis on real world performance than worst-case bounds.

Categories and Subject Descriptors

J.4 [**Social and Behavioral Sciences**]: Economics

General Terms

Economics

Keywords

Social Choice, Imputation, Partial Preferences

1. INTRODUCTION

It is often necessary for a community of agents to reach a collective decision about a course of action, division of resources, or policy change. Agents have different beliefs about the relative quality of different alternatives, making it difficult to determine which alternative is most preferred overall. This is the origin of social choice problems, which arise both within multiagent systems (e.g. agent coordination) and the world at large (e.g. candidate rankings at a hiring committee). As recognised by public choice theory [1], casting a ballot in an election has an inherent cost. In this paper, we are especially interested in the part of that cost associated with the effort of determining and specifying one's own preferences, which may be considerable. Agents who do not know enough about the alternatives to formulate a complete opinion cannot report their true preferences on their ballots. Consequently, if gathering the required information becomes too expensive, rational voters may provide only partial information, or simply forego voting entirely [2], presenting a significant challenge for conventional social choice techniques. Addressing this amounts to determining the outcome of an election with only partial information, and can yield strange worst-case outcomes. For example, rules like Borda's count, which are designed to avoid selecting the Condorcet loser, may nonetheless select such a candidate on the basis of partial information. An extreme case for this would be an election where ordered preference ballots were truncated to include only first preferences. With so little information, Borda's count behaves like the popular plurality rule, and can select a Condorcet loser as the victor.

While existing techniques for partial ballot elections [11, 12, 19] have focused on circumventing the problems posed by missing information, we propose the use of classification algorithms to restore the missing components of each voter's ballot instead. Once the missing components have been restored, any of the ordinary, established, techniques for picking the winner (i.e. any social choice function) may be applied to these newly completed ballots to determine the outcome of the election.

A prototype implementation of our new system performs well on real world ballots. In addition to exhibiting a low error rate for the restoration of individual preferences, our new technique can recover the correct winner in nearly all instances, and close approximations of the entire ordering for use in multi-winner selection[1]. This illustrates the great potential role for machine learning in social choice [18], as well as providing a novel alternative technique for selecting a winner in elections with partial ballots.

2. SOCIAL CHOICE AS IMPUTATION

Social Choice problems arise when a set of agents wish to *collectively* agree on a common course of action. Each agent is assumed to have preferences about the relative quality of each alternative, which can be expressed as an ordering

Appears at: 1st Workshop on Exploring Beyond the Worst Case in Computational Social Choice. Held as part of the 13th International Conference on Autonomous Agents and Multiagent Systems. May 6th, 2014. Paris, France.

[1]i.e. correctly selecting the top k candidates in an election where there are to be k 'winners'

over the alternatives. We refer to an agent's actual beliefs about the relative order of candidates as their preference profile, and their stated ordering (i.e. what they report to others) as their ballot. In this work, we will usually assume that voters are truthful, and report a ballot that does not misrepresent their preferences. Consequently, we will use a voter's preference profile to refer to their hidden true beliefs, and their ballot to refer to their reported (likely partial) preferences.

Given the ballots of the voters, the problem of social choice is simply to select a 'winning' alternative that will appease them. Unfortunately, this is not as straightforward as it might appear. To select a winner, we must first agree on a method of picking one (a social choice function). Even a seemingly agreeable method like picking the alternative with the highest average approval can lead to wildly differing outcomes depending on something as subtle as the meaning of 'average'. Using the median alternative amounts to Condorcet's pairwise comparison rule, while using the mean yields Borda Scoring, and the mode Plurality [1]. The choice is not merely cosmetic — it could produce significant differences in the outcomes of the election. It is further complicated by a number of negative axiomatic results that demonstrate the mutual exclusion of various desirable properties for voting rules. In sum, these mean that the selection of an appropriate social choice function is itself a difficult exercise in social choice.

A further complication of the social choice problem is the introduction of *partial* preference orderings. In such a problem, voters may specify a more limited ballot, in which some alternatives do not have a specified position. The selection of a social choice function now entails a choice about how to treat the missing components of the ballots. For example, we might ignore the missing components [19], or fill them in with a conservative ordering in an effort to avoid picking a bad outcome [11]. In this paper, we present and argue for a new approach in which the votes themselves are used as the basis for determining how the missing components will be treated. We argue that insofar as this approach is equivalent to voting on a method for handling missing information, it inherently better reflects voters' desires than techniques which adopt a policy *a priori*. We expand further on this hypothesis in the next subsection.

2.1 Imputation

Imputation is the process of replacing missing data (referred to as "missingness" here, as in most imputation work, or as "incompleteness" in some social choice work) with a carefully selected guess at the missing value [16]. 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.

2.2 Our Model

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, which we treat as a set of ballots (rows). 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.

A given ballot (row) $r_i \in R$ thus consists 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 ranked[2].

The selection of the winning element of O is conducted according to some social choice function S. S maps sets of *complete* ballots ($R_c \in \mathbf{P}(O)^N$) to outcomes ($o' \in O$). 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 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, and it is this issue that our system will address. Taking the subset of R_2 which is complete ($R_{c_2} = \{r \in R_2 | \ |r| = 2\}$), we train a classifier $c_2 = C(R_{c_2})$ using a classification algorithm C, which predicts the second preference of each ballot from their first preferences. We then use c_2 to impute $R_2 \setminus R_{c_2}$, generating a complete ballot matrix of two columns. We call this imputed ballot matrix R'_2

Having accomplished this, it is naturally possible to extend the process to the next column of R. We simply take the ballot matrix of the first, second and third preferences (R_3), impute any missing values in the second column using c_2, and then build a classifier c_3 on R_{c_3}, which we can use 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 S to the imputed matrix: $S(R') = o'$. We formalize this process in Algorithm 1.

2.3 Imputation as Social Choice

It is natural to wonder what properties might follow from picking a winner in this fashion. Interestingly, we can show that the process is itself reducible to a social choice problem with respect to the model selected by our classification algorithm. A number of results follow naturally from this. For instance, it will be possible to manipulate the choice of model and, depending on the underlying voting rule S, may be more or less computationally difficult to do so.

We begin by formally defining a classifier as an imputation model which selects imputations for incomplete ballots

[2]Note that this is not equivalent to the usually assumed partial orderings, but is equivalent to the *top-t* orderings which are used by Lu and Boutilier [11].

O	Set of candidates (alternatives)
N	The number of ballots
R	Set of ballots
r_i	The i^{th} ballot in R
S	The social choice function
R_j	Top j preferences of every ballot
R_{c_j}	Top j preferences of every ballot with j or more.
R'_j	R_j after imputing any missing j^{th} preferences.
o'	The selected alternative.
C	A classification algorithm.
c_j	A classifier trained to impute the j^{th} column of R.

Table 1: Notation used throughout this paper.

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

```
function IMPUTE_BALLOTS(O,R,S,C)
    for all 2 ≤ j ≤ |O| do
        LET c_j ← C(R_{c_j})
        SET missing ← R_j \ R_{c_j}
        for all 2 ≤ k ≤ j do
            missing ← c_k( missing )
        end for
        SET R'_j ← R_{c_j} ∪ missing
    end for
    RETURN o' ← S(R'_{|O|})
end function
```

on the basis of complete ballots alone. A classification algorithm C maps a complete ballot matrix R_{c_j} to a particular imputation model according to a deterministic policy T. For example, least squares linear regression would select a regression model $c_j = C(R_{c_j})$ by minimizing the squared difference between the model's predictions of the j^{th} column, and the values actually present there.

A chained classifier is a full imputation model that takes a partially complete ballot matrix (R) and produces a fully imputed matrix R' using a series of classifiers on each column of the data in turn (i.e. the procedure in Algorithm 1 above). A scoring rule based social choice function operates by assigning each candidate a numeric score as a function of the ballots, and then selecting candidates based on their relative scores. For instance, the Borda Count scoring rule assigns scores as the sum of the position of the candidate on each ballot. We will show that every chained classifier is equivalent to using a scoring rule based social choice function to select the imputation — that is, that chained classifiers operate by treating voters' ballots as voting for particular imputation policies. We begin by showing that this is true of every (non-chained) classifier.

LEMMA 1. *Every classifier is equivalent to a scoring rule based social choice function.*

PROOF. A social choice function in this context is any function mapping a set of complete orderings over some finite set of outcomes (votes), to a single outcome. A classifier that predicts missing values in the j^{th} column of a $N \times |O|$ matrix functions by selecting and then applying a policy using a set of complete orderings over j of $|O|$ possible outcomes. Note that, if there are N_{ic} incomplete rows in the matrix provided (R), then there fewer than $(|O| - j + 1)^{N_{ic}}$ possible imputations that could complete R (each row has $j - 1$ elements already specified, out of $|O|$ outcomes). Call this the outcome space O_{impute} of our classifier, and note that this space is finite. Additionally, note that, although rows of R are not complete orderings over O_{impute}, we can design a mapping from the complete preference space to the format of an ordered ballot containing $j - 1$ elements. Represent the process of selecting an imputation policy c_j from the completed votes as a function T, and note that this function is deterministic and produces exactly one output (T is a deterministic model selection algorithm for classification). Define $t_{candidate} = T(R_c)$ as the classifier selected by applying T to the completed ballots in R to select an imputation policy, and $R_{candidate} = t_{candidate}(R_{ic})$ as the $N_{ic} \times 1$ vector produced by the application of this policy to the incomplete ballots ($R_{ic} = R \setminus R_c$). Then we can define a scoring rule S_j over candidate imputations: $S_j(R_c, R_{other}) = -\Delta(R_{other}, R_{candidate})$, where Δ is the number of vector components in which the two vectors have different values (i.e. the sum over component-wise delta functions). A social choice function over O_{impute} that selects the outcome with the highest score in S_j will produce identical output to a classifier selected via T, and thus, is equivalent. □

THEOREM 1. *Every chained classifier is equivalent to a scoring rule based social choice function.*

Proof Sketch: The proof is by induction on the number of candidates. It is easy to show using Lemma 1 that if $j = 2$, any chained classifier has an equivalent scoring rule. From this, it is possible to define a scoring function for larger values of j using the recurrence: $S_j(R_c, R_{other}) = -\Delta(R_{other_j}, R_{candidate}) + S_{j-1}(R_{c_{j-1}}, R_{other_{j-1}})$, where S_{j-1} is the scoring function for the chosen chained classification algorithm on ballots with $j - 1$ candidates.

The implication of Theorem 1 is that using classifiers for imputation entails holding a *vote* on the treatment of missingness in the data. A number of results follow directly from this theorem. For instance, every chained 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.

3. VALIDATION

In this section, we present the application of our imputation based approach to social choice to several sets of ballots from the `preflib.org` repository [13]. We show that using imputation to select the winner produces accurate results, and that the system is usable in the real world. Section 5 below provides a comparison with existing algorithms for the same data, as well as more qualitative comparisons. Our experimental design measures the ability of a classification algorithm to correctly predict missing preferences in each data set, using several different sets of features. As classifier performance is highly sensitive to the data used for training models, we generated many different randomized ablations of the data, and evaluated the quality of our models in terms of their average performance. The remainder of the section describes the datasets, our preprocessing and feature selection procedures, and the machine learning algorithms themselves. We then provide a precise description of our experimental design.

3.1 Data

We selected several elections from the Irish and Debian datasets[3], which are both comprised of real-world ballots with ranked preference formats. In both sets, voters were able to omit preferences if desired. From the Debian set, we took the election data for the seven leadership elections from 2002-2012, and the vote on the Debian Project logo. From the Irish set we took the ballots from the Dublin North and Dublin West constituencies. Collectively, these elections provide good diversity both in terms of the number of candidates running, and the degree of missingness in the voters' preferences. This is summarized in Figure 1a, which shows the percentages of information missing from ballots in each election. While the Debian sets are missing between 10 and 20% of the preference information (with the exception of the logo set), the Irish sets have much higher missingness, possibly reflecting the larger number of marginal candidates in national politics, or lower voter enthusiasm. Naturally, the Irish sets also have far more ballots than the Debian sets. While the former have tens of thousands of ballots, of which thousands are complete, the latter typically have only a few hundred completed ballots. Figure 1b provides an example of the distribution of missingness, showing that while nearly all users are able to specify their first few preferences, there is an extremely rapid drop afterwards , with only about 10% of voters completing their ballots. This illustrates how challenging the machine learning problem will be: correctly imputing a single ballot may entail making eight or nine correct classifications.

Although we selected these sets to provide a reasonable variety of different missingness levels and election sizes, they also provide some basis for comparison with prior work. The two Irish sets have also seen prior use benchmarking the performance of other voting systems [11, 19, 12], offering a compelling reason for their inclusion. However, because our proposed imputation based social choice relies on using numerical techniques to address the missingness in the data, some careful processing of the data is required prior to applying the new technique.

3.2 Preprocessing

Numerically coding the ballot data in our sets into a ballot matrix does not produce usable results in any classification algorithm we have applied. This data requires careful preprocessing and feature construction to provide good results[4].

The first preprocessing step consisted of measuring the empirical distribution of missingness for each dataset, that is, the proportion of ballots with at least k preferences specified for $1 \leq k \leq N$. We then discarded all ballots for which the complete preference data were not provided, following [11], to ensure that ground truth information would be available for our assessment of performance. The remaining data were ablated in a fashion consistent with the empirical distribution of missingness for the original dataset, to generate ballots with a level of missingness similar to that of real world data, but for which the ground truth was known.

The second phase of preprocessing involved extracting usable features from the data. First, we transformed each ballot from a relative format (e.g. $x \succ y \succ z$), to an absolute format (e.g. `x:1,y:2,z:3`). This provides a more consistent set of features for the classifier, since a high ranking in feature x indicates the importance of x as a candidate on its own. We then added two more sets of features to provide a clear quantification of the relative positioning of each set of candidates. For each pair of elements in the set of alternatives $o_1, o_2 \in O$, we defined one feature $I(o_1 \succ o_2)$ to be an indicator variable for whether $o_1 \succ o_2$, and another $\Delta(o_1, o_2)$ to be the difference in the relative positions of o_1 and o_2 in the ordering. For instance, in the ballot $x \succ z \succ y$, the feature $I(y \succ z)$ is 0 (false), and the feature $\Delta(x, y)$ is 2 (Since x is at position 1 and y is at position 3). When candidates did not appear on a ballot, we used the value "NA" for the absolute position features, and for any of the pairwise features in which that candidate appeared. The addition of these new features constitutes the injection of domain-specific knowledge into our classifier, which is necessary for the achievement of high performance on most machine learning tasks. Note that we are providing only features that can be computed from the *training data*, and that the classifier must still build a predictive model of partial preferences without using any information about the hidden preferences themselves (i.e. the *test data*).

The third step consisted of applying feature selection techniques to the data. We considered several feature selection approaches, including Principal Component Analysis (PCA) and Information Gain (IG) [7]. In both cases, we were interested primarily in the possibility of speeding up training of classifiers without significant reductions in performance. We used R implementations of both algorithms [15]. For PCA, we used the `prcomp` function to obtain the principal components, and discarded all components with eigenvalues less than 10% of the first principal component's. For IG, we used the *FSelector* package and discarded all but the 10 most predictive features from the original set. After performing the above procedures, we were left with three complete datasets: one containing the cleaned data with the original features, one with the PCA features, and one with the IG features. All three sets were used in our validation. Consistent with standard practice, we used only the data from our *training* set (the ablated ballots) to construct the alternative feature spaces. We then projected the test set data (i.e. the known user preferences from our ground truth) into the resulting space for the evaluation of any classifiers produced with the training data. The final step in our preprocessing was a standard normalization and centering of the data, to make it more suitable for use in classification. We then applied our classification algorithms.

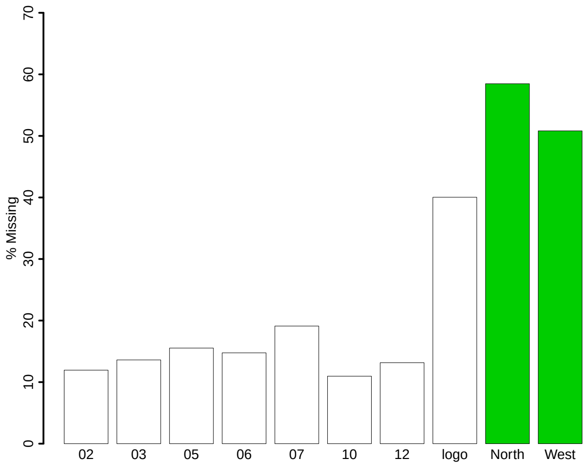
3.3 Classification

As there exist a variety of conventional classification algorithms, it is not immediately apparent which is best suited to a given application domain. This is particularly true in the case of voting data, because the problem has many classes, and often has great imbalance between the classes as well (e.g. some candidates are much more likely to appear as first preferences than others). For this reason we considered several classifiers.

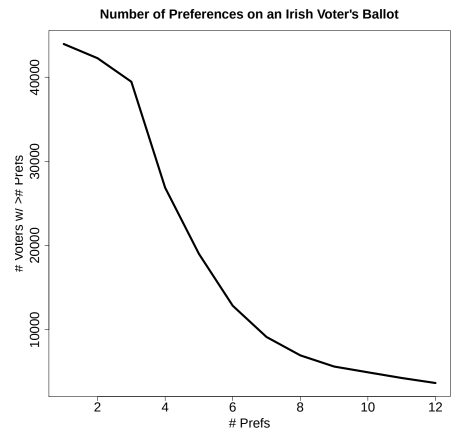
Most prominently, we used Support Vector Machines (SVMs).

[3] `http://www.preflib.org/election/`{irish,debian}.php

[4] We do not claim that the preprocessing steps we describe are the best available; they do at least provide a set of usable features in a data format which many common classifiers are able to process. We expect that further development of this area could provide better features, and as such, our results will provide only a lower bound on the performance of the proposed technique.



(a) Bargraph the percentage of missing preferences for each dataset. The Debian project leadership elections ('02 - '12) and logo election are shown in white. Green bars show the Dublin North and Dublin West voting districts from the Irish election dataset.



(b) Plot showing the distribution of completed preferences in the Dublin North dataset. The horizontal axis shows the number of preferences indicated, while the vertical axis shows the proportion of voters who indicated at least that many preferences. Note the rapid drop starting after the third preference.

Figure 1: Plots summarizing the datasets used for validation.

These classifiers select models from the set of possible hyperplanes that separate different classes (candidates in our context) in the feature space (preference space in our context). They differ from other classifiers of this sort by automatically projecting the data into a higher dimensional space using combinations of the provided features. This often causes the data to become linearly separable (or nearly so), providing high accuracy for hyperplanes selected from this larger space. We used the popular *libsvm* implementation [3] for our experiments. This implementation provides native support for multiclass problems and missing values, so the only additional work was converting our datasets into the application's data format[5].

As these classifiers are not purpose-built for the task at hand, they do not incorporate the constraint that no candidate may appear more than once on the same ballot. To address this problem, our classifiers provide a probability distribution over the classes as their output. We then select the most probable label that does not already appear on the given ballot as the classifier's proposed label.

3.4 Measurement of Imputation Quality

Our experiment consisted of repeatedly ablating a given dataset according to its measured distribution of missingness, and then using the preprocessing steps described above to render the data usable. We then applied our classifiers to each of the three resulting feature sets according to the procedure in algorithm 1, and compared the results to the ground truth. We assessed performance with two different measurements, based on the Borda Scores of candidates.

Our first measurement reflects the overall accuracy of the imputations produced by our classifiers. It consists of the sum of differences between each candidate's predicted and actual Borda Score, normalized by the total Borda Score for all candidates in the election. Thus, for a set of candidates O, and where $BC(o, R)$ is the Borda Scoring rule result for candidate o with imputed ballots R_c and ground truth ballots R_g, we write this measurement:

$$BC_{error}(O, R_c, R_g) = \frac{\sum_{o \in O} |BC(o, R_c) - BC(o, R_g)|}{\sum_{o \in O} BC(o, R_g)}$$

The measurement is equivalent to classification accuracy with errors weighted according to the difference between the correct and actual positions of a candidate on the ballot.

Our second measurement addresses the possibility of imputations which achieve low BC_{error} by assigning disproportionately many votes to one candidate at the expense of the others. For instance, if a candidate received only a few hundred votes in the ground truth, a reasonably good BC_{error} can be obtained by assigning all missing votes for this candidate to someone else. Although this produces a low error rate, the errors are not spread throughout the candidates proportionately. The outcome of the election could thus be significantly changed. This is analogous to the class imbalance problem in more typical machine learning applications [8].

To quantify this tendency, we measure the correlation between the error ($|BC(o, R_c) - BC(o, R_g)|$) in a candidate's Borda Count with the magnitude of their count in the ground truth. A strong correlation indicates that the classifier has a large bias (typically toward more popular candidates) while a lower value shows the opposite. A high bias is undesirable, because it indicates that the classifier is not learning the relationships between marginal candidates at all, and instead assigning their share of the imputed ballots to more popular candidates.

We generated 50 unique random ablations of each dataset, providing 50 directly comparable data points for each combination of dataset and feature set. Each point corresponds to

[5] We also considered a number of other classifiers including multinomial regression, Naive Bayes, the C4.5 decision tree and one-vs-all multiclass logistic regression. In general these techniques were outperformed by the SVM, even after careful parameter tuning. As such, we present only the SVM results here.

a single randomized ablation of the target dataset according to our preprocessing procedures. We selected SVM parameters automatically for each such point using a customized grid search over the C and γ parameter spaces, and the sigmoid and RBF kernels. Five-fold cross validation over the training data was used to guide the search.

3.5 Results

The proposed, imputation based, approach to social choice relies on the ability of machine learning algorithms to provide 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 twelve-class classification problem with many features missing on any given record. Despite this, our procedures yield classifiers with relatively high performance, strongly validating our approach.

Figure 2a shows the BC_{error} rates for the SVM classifiers on each combination of feature set and dataset (Plain denotes the full set of features, while IG and PCA denote the corresponding feature selection techniques). In general, this error rate is quite low. The information gain feature set, which provided the strongest performance, has error rates of 1% or less on every set of ballots. While the other feature sets exhibited somewhat worse performance, even they have error rates well under 2% on the more challenging sets. Also noteworthy is the relatively weak relationship between the amount of missing data and the magnitude of the BC_{error}. For instance, the 2002 Debian election set has one sixth the missingness of Dublin North, but only half the error rate.

Figure 2b provides a partial explanation for this. The plot shows the bias, measured as the correlation between a candidate's Borda Score and the candidate-specific component of the BC_{error} measure, for the SVM classifiers produced on each dataset/feature set combination. Notably, the information gain feature set exhibits significantly worse performance than the other two sets here, with a strong relationship (typically a negative one) between the number of votes a candidate received and the error rate in assigning votes to that candidate. The Debian '02, '10, and '12 sets provide evidence of semi-degenerate classifiers, which arise when data have a very high degree of class imbalance (i.e. when some candidates receive far more votes than others). These classifiers have poorly defined decision boundaries between the large and small classes in the set, but in practice, it appears that the boundaries between pairs of small classes are still learned — the overall ordering of the candidates is preserved (see Section 5). There are many techniques for addressing these issues (e.g. DSS [6]), and we expect that their application could enhance performance further on highly imbalanced sets.

Our results demonstrate the practicality of social choice through imputation. Models produced low error rates on all the datasets considered, and although some higher biases were present, on most datasets they are small enough to be acceptable, despite indicating somewhat lower accuracy for marginal candidates. Run times were 1–20 minutes for each data point gathered, on a single core of a contemporary desktop machine. Most of the runtime was spent on the parameter tuning of the SVM, an inherently parallelizable process. In summary, imputation based social choice is a viable and fast technique, applicable to real world problems.

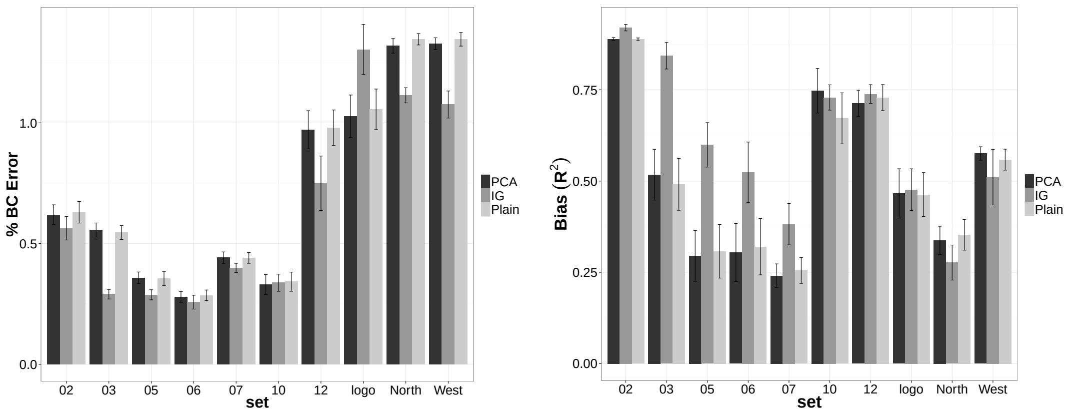
4. RELATED WORK

While our approach to solving the "truth-tracking" problem of social choice with partial preferences is novel, there exists considerable prior work on this problem, much of it coached in terms of *vote elicitation*. We begin with a summary of this work, before providing a more detailed comparison with the similar minimax regret approach to the problem [11, 12], including a comparison on the datasets used in our validation above.

Much of the earlier work is concerned with the determination of possible and necessary winners from a set of partial preferences [10]. For the intuition behind this problem, consider a set of ballots with only a few votes missing. Unless the election was exceptionally close, these missing votes may no longer matter, in the sense that even if the voters all colluded to sway the result, the margin of victory may exceed their collective influence. In such a case, the winner is completely determined by the completed ballots — there is a necessary winner implied by the ballots that have been cast. Even if the ballots do not imply a unique winner they may limit the set of possible winners. For example, a candidate may have received so few votes that they are out of the running without even counting the remainder of the ballots. Early work by Konczak and Lang [10] considered the problem of determining necessary and possible winners from partial preference ballots, notably establishing that a family of positional scoring rules allowed this determination to be made in polynomial time. More recent work by Xia and Conitzer [19] considered nine voting rules and showed a variety of NP- and coNP-completeness results for finding necessary and possible winners. However, like Lu and Boutilier [11], we differentiate our work from this area by virtue of it *picking* a winner (more precisely in our case, a winning ordering for the outcomes) on the basis of partial information, rather than providing a set of possible winners entailed by the preferences.

Interest in vote elicitation extends back to at least the 2002 work of Conitzer and Sandholm [5], 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 [9] 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. [14] 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. [17] 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 directly to accomplish the same goal[6].

[6]We note an interesting parallel however. While our imputation based techniques are akin to the application of traditional classifiers to the partial preference social choice problem, vote elicitation schemes are more similar to *active learning* [4], an alternative approach in which the algorithm automatically decides which pieces of information to ask for next, while at the same time inferring patterns about the



(a) Plot showing the BC_{error} measurement (percentage scale).

(b) Plot showing Bias measurement (correlation between # votes and magnitude of imputation error).

Figure 2: Results on datasets from the Debian leadership '02-'12, Debian logo , and Dublin North and west elections. Whiskers show a 95% confidence interval for the mean of each combination of feature set and dataset.

The work most similar ours is that of Lu and Boutilier [11], 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. Validation consisted of demonstrating that elicitation guided by minimax regret (MMR) could rapidly find correct single winners.

Irrespective of the performance obtained by this strategy, there are a number of reasons to suppose our imputation based approach should be preferred, at least in some situations. Chief among them is that the result in the minimax regret system is produced by the most pessimistic imputation of the ballots possible, while our system selects a result based on ballots that are as *consistent* as possible with the patterns of voting that we were able to observe. In applications where voters expect an outcome that is consistent with their beliefs, the suggestion that their own ballot was completed with an implausible preference ordering is an unpalatable one. Additionally, an imputation based approach can provide a more realistic assessment of the distance between candidates in the final result. While the minimax regret system uses *different* completions of the ballots to compute the worst-case regret for each candidate, our system uses the same imputation to produce an overall ranking of the candidates according to a provided scoring rule. Two candidates with similar scores on a given imputed set of ballots may have dramatically different scores on their respective worst-case sets, and vice versa. Imputation based social choice, by virtue of working with a single, comprehensive, imputation of the ballots, can take advantage of existing techniques for multi-winner social choice with complete preferences (A topic examined as well by Lu and Boutilier, again employing a minimax regret solution [12].).

5. COMPARISON WITH MMR

For a more direct comparison between the minimax regret and imputation based techniques, we ran the MMR algorithm on the same 50 ablations of the datasets used in our validation experiments (Section 3 above). We measured performance for both the single winner case (i.e. selecting the correct winner relative to the ground truth), and the Kendall Tau distance (bubble-sort distance) between the obtained rankings for each technique and the ground truth, a measure of performance in the multi-winner case. Our implementation of the minimax regret algorithm was based on the single winner MMR algorithm [11], with a customized extension to generate rankings of the candidates based on their minimax regret rankings. As in our earlier validation, we used the Borda Count scoring rule to determine winners.

In the single winner case, we measured the mean distance between the candidate selected by each method and its position the ground truth reference. For example, a method which always selected the correct winner has a value of 0, while one that always selects the third place candidate has a value of 2. Performance is summarized in Table 2. We find that the two algorithms are broadly similar. Both techniques select the correct winner 100% of the time on most of the sets. Imputation often selects the second place finisher as the winner on the Debian 2007 set, but this was an especially close race, decided by just 15–20 ballots (depending on the particular ablation). MMR consistently selected the second place finisher on the Dublin West set (indicated by the value of 1.0), where the election was decided by nearly 500 ballots. The other differences are not statistically significant.

In the multi-winner case, imputation usually finds high quality overall orderings, but sometimes inverts the order of narrowly separated candidates. In the Debian '02, '03, '10, and '12 sets, the two algorithms have statistically identical performance. The '05, '06, '07, logo, and Dublin West sets all show an advantage for MMR, but only a very slight one — often just a single swap of adjacent candidates separates them. On the Dublin North set however, an average of more than 9 adjacent swaps separates the algorithms. This appears to result from the extremely high missingness in the data, especially for marginal candidates. This lack of preference information, coupled with the presence of 12 candidates, made for some very difficult classification problems. Many of the classifiers were trained on 12-class problems

set.

Set	Imputation	MMR	$\Delta(\tau)$
02	0	0	0
03	0	0	0.04
05	0	0	**0.42**
06	0	0	**0.5**
07	**0.82**	0	**1.4**
10	0	0	0.04
12	0	0	0
logo	0.12	0	**0.56**
North	0.22	0	**9.46**
West	0.26	**1.0**	**1.68**

Table 2: Comparison of Imputation Based Social Choice with MMR for both the single and multi-winner cases. Bold entries show statistically significant differences. The first column shows the datasets used, the second and third show the mean position in the ground truth ordering of the candidate selected by each method. The fourth shows the difference between the Kendall Tau distances from MMR and Imputation to the ground truth for the multi-winner case.

with high imbalance and as few as 300 instances to work from. We anticipate that with a larger number of ballots, or a smaller number of candidates, imputation based social choice could perform well even on sets with such a high amount of missingness.

Collectively, the results are encouraging. Our prototype imputation based social choice system selects the correct single winner in the vast majority of cases across all of the datasets considered, and when mistakes are made, they tend to be small (e.g. selecting the second place finisher in a close election). Performance in completing the entire ordering is also surprisingly good. On a third of the sets we find the correct ordering essentially all the time. Most of the remaining sets typically exhibit only a single error. Only the Dublin North dataset, with its extreme missingness, proved difficult, but even here the most highly ranked candidates in the ordering are usually correct, exhibited by the low single-winner error rates. We expect that with additional refinement, imputation based social choice could provide consistently good performance, even in these extreme cases.

Overall we view imputation based social choice as an alternative approach to minimax regret. We solve a similar, but not identical, set of problems, in a way that may offer advantages when users want their ballot to provide a plausible completion of their (unknown) preferences, and where it is important to have a consistent view of the relative performance of different candidates. For instance, a voter who marks down a preference for a social democratic party might experience considerable regret if they discover their ballot has been completed to rank a nationalistic conservative party highly, even if this maximizes the difference between her first choice and the nearest opposition.The minimax regret approach may be more suitable when very limited ballot information is available. Minimax regret may also be more appropriate when preferences are being gathered gradually, instead of provided as a static set.

6. 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 an unlikely worst-case arrangement. We showed that the process of picking an imputation is itself a form of implicit social choice, which allows results like the Gibbard-Satterwaithe theorem 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 [11], showing that our preliminary model picks the correct winner at nearly the same rate, and exhibits generally low error rates on the rest of the ordering as well. A notable exception is in the case where the ballots have exceptionally high missingness, where our methods require additional refinement to perform 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 [18].). Some interesting extensions might include the application of machine learning models that are specifically designed for problems with a large amount of class imbalance; exploring the connection between active learning and vote elicitation more carefully; and experimenting with a broader set of features on the ballots.

There are also a number of extensions from the social choice direction. A good starting point would be extending the imputation model to accept general partial orders instead of just top-t style preferences. One possible approach to this would be representing the partial order as a set of features, and predicting the entire ballot, preference by preference, on that basis. This would still require at least some information about the absolute position of the elements of at least some of the preference profiles for training purposes however. The interaction between different scoring rules and the imputation based approach could also be interesting. In this work we have considered only the Borda scoring rule, but it is possible that other rules may be more or less sensitive to the types of systematic errors that classifiers sometimes produce.

7. ACKNOWLEDGMENTS

The authors gratefully acknowledge financial support from the Natural Science and Engineering Research Council of Canada.

8. REFERENCES

[1] D. Black. *The theory of committees and elections.* Springer, 1987.

[2] A. Blais and R. Young. Why do people vote? an experiment in rationality. *Public Choice*, 99(1-2):39–55, 1999.

[3] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[4] D. A. Cohn, Z. Ghahramani, and M. I. Jordan. Active learning with statistical models. *Journal of Artificial*

Intelligence Research, 4:129–145, 1996.

[5] V. Conitzer and T. Sandholm. Vote elicitation: Complexity and strategy-proofness. In *Proc. AAAI/IAAI-02*, pages 392–397, 2002.

[6] C. Gathercole and P. Ross. Dynamic training subset selection for supervised learning in genetic programming. In *Proc. PPSN-94*, pages 312–321. Springer, 1994.

[7] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *The Journal of Machine Learning Research*, 3:1157–1182, 2003.

[8] N. Japkowicz and S. Stephen. The class imbalance problem: A systematic study. *Intelligent data analysis*, 6(5):429–449, 2002.

[9] M. Kalech, S. Kraus, G. A. Kaminka, and C. V. Goldman. Practical voting rules with partial information. *Autonomous Agents and Multi-Agent Systems*, 22(1):151–182, 2011.

[10] K. Konczak and J. Lang. Voting procedures with incomplete preferences. In *Proc. IJCAI-05 Multidisciplinary Workshop on Advances in Preference Handling*, volume 20, 2005.

[11] T. Lu and C. Boutilier. Robust approximation and incremental elicitation in voting protocols. In *Proc. ICJAI-11*, pages 287–293. AAAI Press, 2011.

[12] T. Lu and C. Boutilier. Multi-winner social choice with incomplete preferences. In *Proc. IJCAI-13*, pages 263–270. AAAI Press, 2013.

[13] N. Mattei and T. Walsh. PrefLib: A Library of Preference Data. `http://www.preflib.org`, 2013. [Online; accessed June 5th 2013].

[14] J. Oren, Y. Filmus, and C. Boutilier. Efficient vote elicitation under candidate uncertainty. In *Proc. IJCAI-13*, pages 309–316, 2013.

[15] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2013.

[16] J. L. Schafer. Multiple imputation: a primer. *Stat. methods in medical research*, 8(1):3–15, 1999.

[17] H. A. Soufiani, D. C. Parkes, and L. Xia. Preference elicitation for general random utility models. In *Proc. UAI-13*, pages 596–605, 2013.

[18] L. Xia. Designing social choice mechanisms using machine learning. In *Proc. AAMAS-13*, pages 471–474. International Foundation for Autonomous Agents and Multiagent Systems, 2013.

[19] L. Xia and V. Conitzer. Determining possible and necessary winners under common voting rules given partial orders. *Journal of Artificial Intelligence Research*, 41(2):25–67, 2011.