APPENDIX: SUPPORTING INFORMATION FOR "TREE-BASED MODELS FOR POLITICAL SCIENCE DATA"

This supporting information (SI) appendix first provides additional details on the recursive binary splitting algorithm used to build individual trees. Second, it presents an additional discussion as to how these methods handle categorical outcomes. Third, it briefly provides additional discussion of k-fold cross validation. Fourth, it provides a more detailed discussion of several practical issues surrounding estimating and interpreting tree models. Finally, it presents additional details about the three applied examples in the main text.

SI-1. RECURSIVE BINARY SPLITTING

The essential elements of recursive binary splitting are shown in Table SI-1. First, for *each* covariate j we choose the value v that creates two new regions, denoted $R_1(j, v) = \{X : x_j < v\}$ and $R_2(j, v) = \{X : x_j : x_j \geq s\}$, such that they minimize $L(\cdot)$; this is a proposed split. By choosing a differentiable loss function, the optimization problem can be easily solved numerically. Second, the algorithm chooses the variable j (i.e., the splitting variable) and v (i.e., the splitting value) combination that minimizes the loss function for all observations, calculated using the average of the outcome variable in the proposed regions as the model's predicted values (e.g., $\bar{y}_i \forall i \in R_b \forall i \in R_1 = c_1$). The best splitting rule is then added to the tree, creating two new (potentially terminal) nodes.

Third, the algorithm checks these new terminal nodes against several stopping rules designed to reduce overfitting. Standard choices include specifying the minimum number of observations we wish to allow in each terminal node and the maximum "depth" of the tree (i.e., the number of splits encountered before reaching a terminal node). If these rules are not met, the algorithm then repeats the process to create an additional split. If these stopping rules are met, however, this indicates that the tree is becoming too complex, thereby increasing the potential for overfitting. Therefore, the algorithm stops and the proposed node becomes a permanent leaf associated with the prediction $c_b = \bar{y}_i \forall i \in R_b$.

	Purpose	Description
1	Calculate optimal splits	For each covariate j, calculate the optimal point (v) to create a new split.
2	Choose optimal covariate	Select the covariate and split rule that minimize $L(\cdot)$ using the average y_i in the corresponding regions as c_b.
3	Check stopping rules for new leaves	Check whether the tree has reached pre-specified level of complexity.
4	Repeat steps 1-3	For each new leaf, if the stopping rule has not been reached, add a new split.

Table SI-1. Building a tree via recursive binary splitting

SI-2. DISCRETE OUTCOMES

Although our discussion is centered around continuous outcomes, much of the logic behind each of the models we present applies to discrete responses with little modification. When dealing with a categorical outcome, the models are usually called *classification tree* (rather than regression tree) models. The main differences between classification and regression trees come in the definition of optimal terminal node values, c_b, and in the corresponding definition of an adequate loss function to compare the observed y_i in region b to the estimated c_b.

The most common choice of terminal node values in classification tree models is the modal category for all observations in the region (for CART models) and the "majority vote"—i.e. the most common prediction among the B trees in the ensemble (for sum-of-trees methods). It is also possible to use latent-variable approaches to produce predicted values—much in the same way some GLMs do. In BART, for instance, a logit link function can be applied to a linear predictor to estimate binary responses, which can then be compared to observed outcomes using any suitable loss function.

In turn, the three most commonly used loss functions (also known as measures of "node impurity") are the misclassification error E, the Gini index G, and the cross-entropy (or deviance) D. Letting $\hat{p}_{bk}$ be the proportion of observations in region R_b that are from class k, we can define each in turn as

$$
\begin{aligned}
E &= 1 - \max_k(\hat{p}_{bk}) \\
G &= \sum_{k=1}^{K} \hat{p}_{bk}(1 - \hat{p}_{bk}) \\
D &= -\sum_{k=1}^{K} \hat{p}_{bk}\log(\hat{p}_{bk})
\end{aligned}
$$

Although all three measures are similar, G and D are differentiable (thereby facilitating the minimization process) and more sensitive to node purity than E is—making them preferable criteria to use in recursive binary splitting. Yet another loss function that facilitates optimization in binary outcome ($y_i \in -1, 1$) contexts is $L\big(y_i, f(X_i)\big) = \exp\big(-y_i f(X_i)\big)$, which gives way to AdaBoost—popular boosting algorithm for binary outcomes.

Incidentally, the sensitivity of these criteria to node purity is the main reason why it is sometimes the case that certain adjacent regions of covariate space are assigned the same predicted category c_b, but are still treated as separate regions by the model; in general, this occurs if the split improves node purity for one of the resulting leaves (Hastie, Tibshirani and Friedman, 2009). However, it is important to note that although functions G and D

are usually preferred minimization criteria when it comes to *growing* a tree for a categorical response variable, the use of E in the pruning stage is often advocated—when the goal is classifying observations.

SI-3. K-FOLD CROSS VALIDATION

In k-fold cross validation (CV), a dataset is randomly partitioned into k equally sized groups (or *folds*). In turn, each fold is treated as a *test set*, and the model is fit using a variety of different settings for the tuning parameters to the remaining folds (i.e., the *training set*), which we use to make predictions for the test set. Using these (quasi) out-of-sample predictions for each test set, we evaluate the fit produced by the different tuning parameter combinations by applying some metric like root mean squared error (RMSE). The average of this fit statistic across all k estimations is known as the CV estimate of generalization error, and we generally choose the tuning parameter values that minimize this quantity. Thus, k-fold CV offers a principled way of choosing parameter values that avoids overfitting and improves model accuracy (Fox, 2000). Our replication archive contains a general-purpose function that performs k-fold CV for any model.

The choice of the number of folds is an important one, as different choices can yield different estimates of the so-called "generalization error": the expected out of sample error of any given model. In general, as the number of folds k approaches the number of observations n, the bias in the estimate of generalization error should be reduced. However, this gain in unbiasdness is offset by a corresponding increase in the estimate's variance: as all $n-1$ estimations are conducted on essentially the same dataset, estimates of error are highly correlated. When $k << n$, this correlation is reduced, and variance is thus reduced. The price, however, is an increase in bias. Thus, there is a tradeoff between bias and variance in the estimation of generalization error through k-fold CV that depends on the vakue of k. Although it will depend on the nature of the data at hand, values of $k = 5$ and $k = 10$ are common in the literature (see Hastie, Tibshirani and Friedman, 2009, chapter 7.).

SI-4. PRACTICAL CONSIDERATIONS IN FITTING, SELECTING, AND INTERPRETING TREE MODELS

First, we note that all of the models we review in the main text can be estimated using freely available packages for the `R` programming environment. The relevant functions, along with their main parameters, are listed in Table SI-2.

SI-4.1. *Choosing tuning parameters*

Tree models (like all nonparametric techniques) require researchers to choose values for pre-specified parameters, sometimes called *tuning parameters*. Table SI-2 lists these parameters for each model, along with common settings (Chipman, George and McCulloch,

Model	Tuning parameters	Used Values
Single tree (`rpart`)	Min. nr. of observations in any terminal node (`minbucket`)	3, 5, 10
	Max. tree depth (`maxdepth`)	3, 5, 15
	Minimum factor decrease in lack-of-fit measure (`cp`)	Inf., 0.1, 0.01, 0.001
Random forest (`randomForest`)	Nr. of trees (`ntree`)	50, 100, 500, 1000, 2500, 5000, 10000
	Nr. of candidate variables sampled at each split (`mtry`)	2, 10
	Min. nr. of observations in any terminal node (`nodesize`)	3, 5, 10
	Nr. of observations sampled when forming each tree (`sampsize`)	360, 720
Gradient boosting (`gbm`)	Nr. of trees M (`n.trees`)	50, 100, 500, 1000, 2500, 5000, 10000
	Max. tree depth (`interaction.depth`)	3, 5, 10
	Learning rate (or factor shrinkage ν) of each tree (`shrinkage`)	0.01
BART (`BayesTree`)	Degrees of freedom for error variance prior (`sigdf`)	3, 10
	Quantile for error variance prior (`sigquant`)	0.9, 0.99, 0.75
	Nr. of trees (`ntree`)	50, 200
	Factor multiplying the prior range of the outcome variable (`k`)	1, 2, 3, 4

Table SI-2. Four tree-based models, with their `R` packages, tuning parameters, and common parameter values.

2010). Tuning parameters should generally be chosen by cross-validation (CV), although some parameters can be treated as substitutes—such as parameters in charge of alternate regularization strategies (e.g. the number of trees M and the shrinkage parameter ν in GBMs). In such instances, it is common to hold one of the alternative parameters at an arbitrary (but sensible) value and cross-validate the other(s).

The tuning parameters for the BART model require some additional explanation. In the canonical implementation of BART, the tree-depth (B) prior probability distribution is given by $f(B) = \alpha(1+B)^{-\beta}$, with $\alpha \in (0,1)$ and $\beta \in [0,\infty)$. For splitting variables and splitting values, uniform distributions are adopted. Terminal node values are given a Normal prior, with zero mean and variance $\sigma_{bm} = \frac{0.5}{k\sqrt{M}}$ (where constant k and the number of trees M are supplied by the user). Finally, the error variance is given an inverse χ^2 distribution defined as $\frac{\nu\lambda}{\chi^2_\nu}$ (where the degrees of freedom hyper-parameter ν, different from the shrinkage parameter in GBMs, is user-supplied, and the scale λ is internally chosen so that the prior has a user-defined probability q of being *lower* than the observed sample variance of the data y). For a more detailed discussion, see Chipman, George and McCulloch (2010).

SI-4.2. *Categorical predictors and missing data*

Tree-based models can accommodate discrete and continuous *predictor* variables. However, discrete variables are typically turned into dummy (i.e., contrast-coded) variables for each category since ordering is much less important. Unlike regression models, however, tree models can include all categories (i.e., there is no reference category). While it is clear that this is in no way a *technical* deficiency of common regression models, the additional data processing step (as well as the additional steps required for making implicit comparisons between non-reference categories) is unnecessary when using tree-based models.

Similarly, all tree-based models can incorporate predictors with missing values. The most common approach makes use of proxy (or surrogate) splitting rules for observations that have missing values: for each chosen splitting rule, the tree-based model finds a set of alternative splitting rules that perform as similarly as possible in terms of loss reduction; this usually involves predictors that are highly correlated with the one on which the chosen split is based. When the algorithm is moving an observation down a tree to generate a predicted value, and it reaches a node for which the value is missing, it uses the next surrogate splitting rule for which a value *is* available (Faraway, 2005; Hastie, Tibshirani and Friedman, 2009). Discussing the use of BART when outcome variables are missing, Hill (2012) notes that—under the missing-at-random assumption—one can also simply fit a model using the complete cases and then make predictions for the full dataset.

SI-4.3. *Determining which model to use*

Each of these models may be more or less appropriate for different situations. Ensemble models generally outperform single-tree variants both in terms of out-of-sample predictive performance and in uncovering the relationship between covariates of interest and the outcome. RF models, for instance, perform quite well relative to CART models. In some sense, the bootstrapping procedure used to build the trees ensures some level of accurate out-of-sample performance. However, both RF and CART models perform relatively poorly when analyzing data with smooth additive relationships. We illustrate to this point in the first example presented in the main text.

A particular feature of the tree ensemble models is that, by combining many weak learners, they can recover both discontinuities *and* can approximate smooth relationships in the response surface through the combination of multiple step function. GBM has a clear advantage in terms of computation overhead and modeling flexibility: the standard `R` implementation can estimate normal, binomial, multinomial and Poisson models (among others)—in a matter of seconds. However, the major drawback of GBM, RF, and CART models is the lack

of uncertainty in its estimates.[1] This is particularly troubling since we are often interested in establishing whether a covariate (or a combination of covariates) is related to an outcome in a statistically discernible way. Although we will discuss graphical means of establishing covariate effects in the next subsection, measures of uncertainty around these effects are not readily available when using GBM, RF or CART.

BART offers results that are very similar to those produced by fine-tuned GBMs, and has been successfully used to study a variety of outcomes, including political phenomena (e.g. Green and Kern, 2012). Moreover, its default hyper-parameters perform extremely well under very diverse circumstances, making it an appealing contestant for the title of best "off-the-shelf" sum-of-trees model. Most importantly, perhaps, its default implementation produces both point estimates *and* measures of uncertainty around them. Its main drawback is its computational costs and the large number of tuning parameters that must be chosen to fully specify the priors. Moreover, the current `R` implementation accommodates only normal and binomial outcomes. On the other hand, BART offers the possibility of obtaining measures of uncertainty around predicted quantities and covariate effect calculations. Coupled with the fact that its predictive performance is comparable to that of boosting machines, this makes BART a very attractive default option.

SI-4.4. *Interpreting model results*

Once a model and its attendant tuning parameters have been selected, analysts face the task of interpreting the results. Like many statistical modeling approaches, we can break interpretation down into two separate tasks: First, does the variable contribute to the model's explanatory power? (i.e., How much does this variable improve the fit of the overall model?) Second, what is the relationship between the covariate and the outcome? (i.e., Does the conditional relationship between the variable and the outcome of interest match theoretical expectations?)

The machine learning literature places special emphasis on variable importance (often termed feature selection) —that is, which explanatory covariates are important in the model. In fact, one of the advantages of tree-based methods is its relative insensitivity to the inclusion of irrelevant covariates (Hill, 2012). Although out-of-sample predictions tend to become less accurate (and more uncertain) as the number of irrelevant predictors included increases, tree-based methods are particularly good at detecting low-dimensional signals in high-dimensional covariate spaces (Chipman, George and McCulloch, 2010).

In order to decide which variables play an important role in predicting the outcome of interest, several measures of variable importance have been developed. For example, we

[1] Measures of uncertainty could be computed using resampling techniques; see Hofner et al. (2014).

can leverage the tree-building process to produce relative importance measures for every covariate. Specifically, we can calculate the total reduction in the loss function induced by a covariate every time it is used as the splitting variable in a tree, and average these total reductions across the trees that form the ensemble. Letting K_{mj} be the set of internal (i.e., non-terminal) tree nodes of tree m that use j as a splitting variable, such a measure can be defined as

$$\mathcal{I}_j^{Improve} = \left(\frac{1}{M} \sum_{m=1}^{M} \sum_{k:\in K_{mj}} i_k^2 \right)^{0.5}, \tag{1}$$

where i_k is the difference between the total loss function evaluated before the split k and the total loss function evaluated after the split—that is, the improvement in fit induced by the split. To avoid the issues that come from the scale of the measure being highly application-specific, the largest observed importance is usually assigned a value of 1 (or 100), and all other values of $\mathcal{I}_j$ are rescaled accordingly (James et al., 2013).

Alternatively, and capitalizing on the fact that splitting variables are chosen to produce the maximum improvement in loss i_k at every split, a simpler measure of importance is given by the *relative frequency* with which a variable is used to generate binary splits. For instance, in the context of BARTs, such a measure can be defined as

$$\mathcal{I}_j^{Use} = \frac{1}{S} \sum_{s=1}^{S} z_{js}, \tag{2}$$

where s indexes tree ensemble samples (and S is the total number of samples taken), j indexes covariates, and z_{js} is the proportion of all splitting rules (i.e. *every* tree node across *all* trees in a sample) that use variable j. However, Chipman, George and McCulloch (2010) recommend that such estimates be estimated with a very small number of trees (e.g., 10), which often requires fitting a separate model for this specific step.

While not comparable, these two measures of variable importance should generally coincide in the *ranking* of variables, as they are both directly or indirectly based on the amount of loss reduction each covariate induces as part of a splitting rule. Nevertheless, since $\mathcal{I}_j^{Use}$ makes no explicit distinction between big and small loss reductions, it may tend to overstate the importance of certain covariates—specially if they are used most often in the lower branches of deep trees or as part of trees that come late to the ensemble. For this reason, and for ensembles that try to fit the prediction surface progressively (like GBM does) $\mathcal{I}^{use}$ provides the best measures of variable importance when the procedure is forced to make relatively large improvements at every split, which occurs when ensembles are small (i.e. when they have few trees) and when each tree in the ensemble is small. Finally, it is important to

note that there are no clear-cut criteria for deciding when a variable's measured importance is so low that the researcher should deem it unimportant.

Next, we turn to the task of evaluating the marginal and joint effects of covariates on the outcome. Given the possibility of having covariates that interact in complex ways to produce accurate predictions of the response surface and the presence of multiple trees, there is no one single coefficient that can be reported. However, methods have been developed to understand other, more common types of linear and non-linear models can also be applied in the context of regression trees.

In particular, we can empirically approximate the average marginal effect of a predictor by first calculating average predictions (Gelman and Hill, 2007), better known as *partial dependencies* in the machine learning literature (Friedman, 2001), but also discussed under different names in Political and social research (see, e.g., Hanmer and Ozan Kalkan, 2013; Long, 1997). In general, the goal is to describe the average relation between the outcome y and a predictor of interest after accounting for the effects of other predictors by marginalizing them out. More formally, we wish to estimate $E[f(x_u, \mathbf{x}_{-u})]$, where $\mathbf{x}_{-u}$ is the vector of values for variables other than the covariate of interest, and x_u are the various values that this covariate can take.

This expected value can be (roughly) approximated by taking the average over the *observed* values of variables other than the input predictor for observation i:

$$\bar{f}_u(x_u) = \frac{1}{N} \sum_{i=i}^{N} f(x_u, \mathbf{x}_{i,-u}). \tag{3}$$

Average predictions thus constructed can then be used to produce other related quantities of interest, such as average predicted *differences* between substantively interesting values of the input variable (like its maximum and minimum values)—what Gelman and Pardoe (2007) call "average predictive comparisons." Similarly, the approach can be implemented when considering the *joint* effect of multiple input variables: by using a vector $\mathbf{x}_u$ when creating the predicted values, and averaging over the observed values of all other predictors, the joint expectation can be approximated (Chipman, George and McCulloch, 2010).

While this seems superficially complex, creating and interpreting these diagnostics is straightforward in practice. Intuitively, we calculate the expected value of outcome y that corresponds to different values of x_u for each observation in the dataset. We make these predictions using the observed value of the remaining covariates for that individual ($\mathbf{x}_u$). Finally, we average these quantities across all observations in the dataset. The resulting plots, which can be created using standard functions discussed below, characterize how the outcome variable changes as of a function of x_u on average.

SI-4.5. *Evaluating convergence*

Finally, it is worth noting that, being an MCMC-based model, BART requires the researcher to evaluate whether a sufficient number of burn-in and posterior samples have been obtained. While it is hard to define a "sufficient number" to guarantee either convergence of a Markov Chain to its stationary distribution or its thorough exploration once reached, a number of diagnostic tools have been devised to evaluate several necessary (albeit not sufficient) conditions. Examples include Gelman and Rubin's (1992) scale reduction factor $\hat{R}$ (with values greater than 1 indicating lack of convergence), Geweke's (1991) Z score for comparing the means of a chain at different stages (or different chains altogether), and estimates of each chain's autocorrelation function (with highly autocorrelated chains indicating a slow exploration of the posterior distribution). As the current implementations of BART in `R` keeps sampled values for all quantities and parameters of interest, it is easy to use a package like `coda` to conduct such diagnostics.

SI-5. ADDITIONAL DETAILS FOR ILLUSTRATIONS

SI-5.1. *Synthetic data simulation*

Our four simulated datasets are generated as follows. First, we generate 500 observations for each of 40 explanatory variables as follows:

$$
\begin{aligned}
x_{1i} &\sim \text{Gamma}(8, 2) \\
x_{2i} &\sim \text{Gamma}(10, 1) \\
[x_3 \;\; x_4 \;\; x_5]_i' &\sim \text{MVN}\left([2 \;\; 3 \;\; 6], [1.5 \;\; 0.5 \;\; 3.3]' \cdot \mathbf{I}_3\right) \\
[x_6 \;\; x_7 \;\; x_8]_i' &\sim \text{Multinom}([0.33 \;\; 0.33 \;\; 0.33], n = 1) \\
[x_9 \;\; x_{10}]_i &\sim \text{MVN}\left([-0.3 \;\; 2], \begin{bmatrix} 1.5 & 0.685 \\ 0.685 & 5.5 \end{bmatrix}\right) \\
[x_1 1 \;\; x_{40}]_i &\sim \text{MVN}\left(\boldsymbol{\mu}, \mathbf{I}\right)
\end{aligned}
$$

We then generate responses using three different data-generating processes:

$$
\begin{aligned}
y_1 &= x_1 + x_2 + x_3 + x_{10} + \epsilon_1 \\
y_2 &= 1.5 + 3.2x_1 + 0.1x_2 - 2.8x_{10} + 1.2x_1x_2 \\
&\quad - 0.2x_2x_{10} + 2.4x_1x_2x_{10} + \epsilon_3 \\
y_3 &= \begin{cases} x_1 - x_1^2 - x_2^2 - 15x_1x_2x_{10} & \\ \qquad + P_3(x_{10}) \times [10 \;\; -5 \;\; 0.9] + \epsilon_4 & \text{if } x_{10} < 2.5 \\ 1750 + 350x_{10} + \epsilon_4 & \text{if } x_{10} \geq 2.5 \end{cases}
\end{aligned}
$$

where $\epsilon_\cdot \sim N(0, 1)$ and P_n is the polynomial-generating function.

As discussed in the main text, these specifications are used to generate 100 training sets and one test set per DGP.

SI-5.2. *Application to Negative Campaigning*

The full listing of variables used in our second empirical application are listed in Table 1 in the main text. In replicating the models in Blackwell (2013), for the numerator, the formula includes all time-invariant factors along with time-varying factors with which they interact in the denominator formula.

$$\begin{aligned}\text{logit}^{-1}(\Pr(y_{it}=1|\text{Covariates})) &= \beta_0 + \beta_1\text{DNeg}_{t-1} + \beta_2\text{DNeg}_{t-1} + \beta_3\text{DNegFrac}_{t-3} + \\ &\quad \beta_4\text{Week} \times \text{Length} + \beta_5\text{Length} + \beta_6\text{Base poll} + \\ &\quad \beta_7\text{y2002} + \beta_8\text{y2004} + \beta_9\text{y2006} + \\ &\quad \beta_{10}\text{Base Undec.} + \beta_{11}\text{Office}\end{aligned} \tag{4}$$

The GAM model for the denominator for incumbents was as follows:

$$\begin{aligned}\text{logit}^{-1}(\Pr(y_{it}=1|\text{Covariates})) &= \beta_0 + \beta_1\text{DNeg}_{t-1} + \beta_2\text{DNeg}_{t-1} + \beta_3\text{DNegFrac}_{t-3} + \\ &\quad \beta_4\text{Week} \times \text{Length} + \beta_5\text{Length} + \beta_6\text{Base poll} + \\ &\quad \beta_7\text{y2002} + \beta_8\text{y2004} + \beta_9\text{y2006} + \\ &\quad \beta_{10}\text{Base Undec.} + \beta_{11}\text{Office} + \beta_{12}\text{RNeg}_{t-1} + \\ &\quad \beta_{13}\text{RNeg}_{t-2} + \beta_{14}\text{RNegFrac}_{t-2} + \textstyle\sum_{j=1}^{P} s_j(\text{Poll}_{t-1})\end{aligned} \tag{5}$$

where $s_j(.)$ is a smoothed function of the polling indicator. For non-incumbents, the model is:

$$\begin{aligned}\text{logit}^{-1}(\Pr(y_{it}=1|\text{Covariates})) &= \beta_0 + \beta_1\text{DNeg}_{t-1} + \beta_2\text{DNeg}_{t-1} + \beta_3\text{DNegFrac}_{t-3} + \\ &\quad \beta_4\text{Week} \times \text{Length} + \beta_5\text{Length} + \beta_6\text{Base poll} + \\ &\quad \beta_7\text{y2002} + \beta_8\text{y2004} + \beta_9\text{y2006} + \\ &\quad \beta_{10}\text{Base Undec.} + \beta_{11}\text{Office} + \beta_{12}\text{RNeg}_{t-1} + \\ &\quad \beta_{13}\text{RNeg}_{t-2} + \beta_{14}\text{RNegFrac}_{t-3} + \textstyle\sum_{j=1}^{P} s_j(\text{Poll}_{t-1}) \\ &\quad \beta_{15}\text{RAds}_{t-1} + \beta_{16}\text{DAds}_{t-1} + \beta_17\text{Undec}_{t-1}\end{aligned} \tag{6}$$

For Table 2 in the main text, the formula in Blackwell (2013) for the Democratic candidate vote share is:

$$\begin{aligned}\text{DemPrct} &= \beta_0 + \beta_1\text{DNegRec} \times \text{DemInc} + \beta_2(\text{DNegDur - DNegRec}) \times \text{DemInc} + \\ &\quad \beta_3\text{Length} + \beta_4\text{DemInc} \times \text{Base poll} + \beta_5\text{DemInc} \times \text{Base Undec.} + \beta_6\text{Office}\end{aligned} \tag{7}$$

Note that the parameters reported in Table 1 of Blackwell (2013) are based on bootstrapping the weighted regression. For the purposes of exposition, we simply report the baseline model and standard errors. Further, we deviate from (Blackwell, 2013) in that we drop three additional observations from this final analysis due to missingness on the additional predictors used in calculating the weights. However, the parameter estimates reported in Table 2 of the main text do not change meaningfully when these three observations are included.

SI-5.3. *Application to estimating small-subpopulation quantities of interest*

Our post-stratified BART models use turnout and support for McCain as outcome variables. Both are continuous variables, and while they have a fixed range (viz. between 0 and 100), typical values are far enough from the theoretical extremes that we feel safe using a Gaussian distributional assumption when estimating our models (as GG originally did).

To expand the work done by GG and show the potential of post-stratified BART models, we use the following set of predictors: state (with 51 different categories, corresponding to all contiguous US states and DC), ethnicity (with categories 'black', 'white', 'latino' and 'other'), income (which is split into income quintiles and an 'NA' category), age (split into quartiles), sex (with 'male' and 'female' categories), education level (split into five levels), marital status (with 'married' and 'single' categories) and an indicator for whether the respondent has children or not (with an extra category for whether the answer is missing). Excluding a category for the binary predictors, this corresponds to 75 binary contrast-coded predictors.

References

Blackwell, Matthew. 2013. "A framework for dynamic causal inference in political science." *American Journal of Political Science* 57(2):504–520.

Chipman, Hugh A., Edward I. George and Robert E. McCulloch. 2010. "BART: Bayesian additive regression trees." *The Annals of Applied Statistics* 4(1):266–298.

Faraway, Julian J. 2005. *Extending the Linear Model with R: Generalized Linear, Mixed Effects and Nonparametric Regression Models*. New York: CRC press.

Fox, John. 2000. *Multiple and Generalized Nonparametric Regression*. Number 131 *in* "Quantitative applications for the social sciences." Sage Publishing.

Friedman, Jerome H. 2001. "Greedy function approximation: A gradient boosting machine." *Annals of Statistics* 29(5):1189–1232.

Gelman, Andrew and Donald B Rubin. 1992. "Inference from iterative simulation using multiple sequences." *Statistical science* pp. 457–472.

Gelman, Andrew and Iain Pardoe. 2007. "Average predictive comparisons for models with nonlinearity, interactions, and variance components." *Sociological Methodology* 37(1):23–51.

Gelman, Andrew and Jennifer Hill. 2007. *Data Analysis Using Regression and Multilevel/Hierarchical Models*. New York: Cambridge University Press.

Geweke, John. 1991. "Evaluating the accuracy of sampling-based approaches to the calculation of posterior moments.".

Green, Donald P. and Holger L. Kern. 2012. "Modeling heterogeneous treatment effects in survey experiments with Bayesian additive regression trees." *Public Opinion Quarterly* 76(3):491–511.

Hanmer, Michael J and Kerem Ozan Kalkan. 2013. "Behind the curve: Clarifying the best approach to calculating predicted probabilities and marginal effects from limited dependent variable models." *American Journal of Political Science* 57(1):263–277.

Hastie, Trevor, Robert Tibshirani and Jerome Friedman. 2009. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York: Springer Verlag.

Hill, Jennifer. 2012. "Bayesian nonparametric modeling for causal inference." *Journal of Computational and Graphical Statistics* 10(2):217–240.

Hofner, Benjamin, Thomas Kneib and Torsten Hothorn. 2014. "A unified framework of constrained regression." *Statistical Computing* DOI 10.1007/s11222-014-9520-y.

James, Gareth, Daniela Witten, Trevor Hastie and Robert Tibshirani. 2013. *An Introduction to Statistical Learning*. New York: Springer Verlag.

Long, Scott. 1997. *Regression Models for Categorical and Limited Dependent Variables*. Sage Publications.