Bayesian Rule Set: A Quantitative Alternative to Qualitative Comparative Analysis

Albert Chiu and Yiqing Xu*

First version: 7th August, 2020
This version: 18th January 2022

The Journal of Politics, forthcoming

Abstract

We introduce Bayesian Rule Set (BRS) as an alternative to Qualitative Comparative Analysis (QCA) when data are large and noisy. BRS is an interpretable machine learning algorithm that classifies observations using *rule sets*, which are conditions connected by logical operators, e.g., IF (condition A AND condition B) OR (condition C), THEN Y = TRUE. Like QCA, BRS is highly interpretable and capable of revealing complex nonlinear relationships in data. It also has several advantages over QCA: It is compatible with probabilistically generated data; it avoids overfitting and improves interpretability by making direct trade-offs between in-sample fitness and complexity; and it remains computationally efficient with many covariates. Our contributions are threefold: We modify the BRS algorithm to facilitate its usage in the social sciences, propose methods to quantify uncertainties of rule sets, and develop graphical tools for presenting rule sets. We illustrate these methods with two empirical examples from political science.

Keywords: Bayesian Rule Set, Qualitative Comparative Analysis, interpretable machine learning

*Albert Chiu, Graduate Student, Department of Political Science, Stanford University. Email: altchiu@stanford.edu. Yiqing Xu, Assistant Professor, Department of Political Science, Stanford University. Email: yiqingxu@stanford.edu. We are indebted to Adrian Duşa for incredibly generous support of the `QCA` package in `R` (including its developmental version) and for helpful comments and suggestions. Some of the computing for this project was performed on the Sherlock cluster. We thank Stanford University and the Stanford Research Computing Center for providing computational resources and support that contributed to these research results. We are grateful to Lead Editor Vera Troeger and three anonymous referees for their constructive comments.

1. Introduction

Complex phenomena pervade the social sciences, and correspondingly complex explanations are sometimes required. For example, guided by Lipset (1959)'s seminal work, modernization theorists cite a multitude of factors when explaining democratic transition and consolidation: industrialization, urbanization, income, and education are each a facet of the socioeconomic development that ultimately drives democratization. The relationship between any single variable and democracy is simple to characterize, but a potential third or higher order interaction is difficult to represent in an interpretable manner using regression or other common methods. *Rule sets* offer a more interpretable alternative. They are conditions connected by the logical operators "and" and "or" that classify data using an "if-then" statement—e.g., if (A and B) or C, then Y = TRUE, where A, B, and C are conditions such as "high level of income."[1]

Qualitative Comparative Analysis (QCA) is a commonly used method amongst social scientists for discovering and describing patterns from data using rule sets. Because rule sets are a useful way of representing complex relationships in data, QCA's popularity has been trending upwards for the past two decades, and there are now hundreds of articles that employ QCA with tens of thousands of citations (Roig-Tierno, Gonzalez-Cruz and Llopis-Martinez 2017). Although QCA was originally designed for small-N settings, scholars are increasingly using it to analyze large-N datasets (Rihoux et al. 2013; Greckhamer, Misangyi and Fiss 2013; Fiss, Sharapov and Cronqvist 2013). Yet it remains poorly understood how QCA performs in these settings, as studies have continued to focus on QCA's small-N properties (Hug 2013; Krogslund and Michel 2014; Lucas and Szatrowski 2014; Seawright 2014; Rohlfing 2018). We provide new evidence that QCA can struggle with performance, interpretability, and computational tractability when data are large and heterogeneous, and

[1] In QCA literature, this is sometimes referred to as "multiple conjunctural": conjunctural as in a conjunction of conditions, and multiple as in multiple such conjunctions.

we explain the origins of these struggles.

In this paper, we propose using Bayesian Rule Sets (BRS), first described by Wang et al. (2016), as an alternative that preserves QCA's appeal while avoiding these limitations, and more broadly as an interpretable classifier for modeling nonlinear relationships, high-order interactions, and other complexities. We make three main contributions to the literature: (1) we modify the original BRS algorithm, including the prior and method for choosing tuning parameters, to facilitate its usage in the social sciences; (2) we propose methods that quantify the uncertainties of BRS results, as rule sets in general; and (3) we develop graphical tools for presenting rule sets and their relationships with data.

QCA operates under the premise of necessary and sufficient conditions (Schneider and Wagemann 2010). This "if and only if" logic implies a degree of determinism that is incompatible with the high degree of heterogeneity in micro-level, large-N data. To find a solution when data contains probabilistic errors, QCA must discard information. The original formulation ignores cases with errors. With the adoption of the concept of "consistency," QCA now ignores the errors themselves by recoding outcomes to mimic deterministic data.

From an algorithmic perspective, QCA seeks to find a parsimonious rule set. By parsimony, we refer simultaneously to the length of rules (i.e., the number of conditions in an "and" statement) and the number of rules in a rule set. Parsimony is desirable because simpler rule sets are easier to interpret and tend to be less overfitted. Problematically, parsimony is always a secondary objective of QCA: The set of feasible rule sets contains only those that achieve perfect in-sample accuracy, and the QCA solution is the most minimal among them. Hence, it is possible that all feasible solutions are too large, and even the most minimal among them is uninterpretable. Moreover, the computational price of QCA can be exorbitant if there are even moderately many variables. We find these concerns to be particularly salient in settings where the data are large and heterogeneous.

BRS, which is another method for finding rule sets, is a natural alternative to QCA that avoids many of the latter's drawbacks. First, BRS is more compatible with heterogeneous

data. Premised on a probabilistic data generating process (DGP), BRS makes full use of the variation in outcomes and does not discard data. Second, BRS is better at finding parsimonious solutions, because it can trade off performance and parsimony. Third, BRS is computationally tractable. We use Monte Carlo simulations to show that BRS quickly recovers parsimonious and accurate solutions in any setting, whereas QCA witnesses its interpretability break down and ceases to be computationally feasible as heterogeneous data grow large.

Obstacles to integration into the social sciences remain, but this paper suggests some solutions. First, the original description of BRS employs penalties that depend on data and many hyper-parameters in unpredictable and unintuitive ways. We modify an alternative version of the method to improve interpretability, computation, and ease of use. We also propose a new method for hyper-parameter tuning to achieve a desired level of parsimony.

Second, methods producing rule sets face unique challenges in quantifying uncertainty and finding stable solutions. Regression employs confidence or credible intervals that efficiently summarize subsets of the parameter space using the end points of an interval, but this is not possible for rule sets, which have no sense of ordering. Instability is also more pernicious. Small changes in numerical results typically do not carry substantive implications, but small changes to rule sets may. We propose using prevalence, defined as the proportion of times a rule is part of the solution when bootstrapping the data, to quantify uncertainty. Bootstrapping can also be used to produce frequentist confidence intervals for measures of performance. To create more stable rule sets, we propose aggregating high-prevalence rules from bootstrapping into a single rule set by maximizing performance subject to constraints on complexity. The solutions we provide are not specific to BRS and can be applied to other methods that generate rule sets.

Last, existing graphical methods are poorly suited to representing non-numerical results. To fill this gap, we create a set of tools that can help researchers visualize and present their results. First, we propose plotting the empirical distribution of rules obtained when

bootstrapping BRS alongside the number of true and false positives covered by each rule, which we call "coverage." Second, we propose using chord diagrams to visualize interactions between variables. Third, we propose using t-distributed stochastic neighbor embedding (t-SNE) plots to show rule set performance and coverage. This plot helps to visualize the clusters covered by each rule, as well as true and false positives and true and false negatives.

Our efforts do not overcome all shortcomings. For one, BRS requires discrete data. If the data are naturally continuous, then discretizing it costs information. Clarke (2020) also points out that rule sets can capture a limited selection of social science theories. For example, rule sets cannot describe central tendency, variation, or quantifiers (i.e., "for all" or "there exist"). Still, we argue that rule sets excel in important areas where other statistical methods fall short, and BRS is an effective way of learning them from data.

It is also helpful to address which role BRS would play in quantitative research. BRS is meant to complement, not to replace, regression and other popular methods by helping to uncover complex relationships. We argue it is best suited for data exploration and description and theory building—theories that can then be tested on separate data—especially when we do not already have a specific variable of interest. Thus, although BRS is not suited to making causal inferences on its own, it can help inspire causal inquiries. QCA plays a similar role in large-N research (Greckhamer, Misangyi and Fiss 2013).

The rest of this paper is organized as follows. In Section 2, we explain the utility of rule sets. We then provide more a detailed critique of QCA in Section 3. In Section 4, we describe BRS and introduce our proposed changes and additions. Sections 5-7 serve as a practical guide. In Section 5, we introduce three graphical tools and use our running example to demonstrate each of them. We then present evidence of BRS's utility both vis-à-vis QCA and popular quantitative methods such as regression. In Section 6, we presents the results of our simulation study, which we use to argue that BRS retains QCA's ability to model complex relationships while being better suited to large and heterogeneous data. In Section 7, we apply BRS to a large-N empirical example previously analyzed using regression and

uncover novel descriptions of turnout in the United States. The last section concludes.

2. Rule Sets (versus Regression)

This paper's main utility is premised on the fact that BRS has advantages over QCA when data are large and heterogeneous. This point is moot, however, if methods currently in social scientists' toolbox can already better fill BRS's proposed role. This section explains that existing methods cannot: Rule sets have distinct advantages over other function classes.[2]

Algorithmically, QCA and BRS are simply different ways of learning rule sets from data. One key advantage of rule set based methods is that they are interpretable. Methods like random forest, neural networks, and other black box learners are generally considered uninterpretable—it is a mystery why a model makes any given prediction, and the researcher cannot explain which characteristics of the observation are responsible for its classification. In contrast, "if-then" conditions are easy to understand. For any given observation, the researcher knows exactly why it was classified as either positive or negative and can clearly articulate the reasoning to a reader. Moreover, decision-making is often naturally rule based. For example, a judge might use the following procedure to determine whether to deny bail: "If the defendant has a history of failing to show in court and is a flight risk, or the defendant committed a violent crime and is a threat to the public, then I will deny bail." This interpretability is an advantage whenever the goal is to understand the data.

Amongst interpretable methods, generalized linear regression methods are the most prominent in political science. These methods estimate the marginal "effect" (not necessarily causal) of each variable on the outcome, holding all other variables constant. Their common shortcoming is that they do not naturally model heterogeneity and instead assume that the coefficient of one variable does not depend on the values of other variables.

[2]The debate over whether QCA offers anything of substance on top of what regression already offers is not new (e.g., Achen 2005; Seawright 2005; Ragin 2005). Rather than rehashing the history of this debate, we offer a more general defense of rule sets tailored to skeptical quantitative social scientists.

There are at least two reasons why rule sets are more suitable than regression in settings where the outcome Y is dichotomous, the explanatory variables X are discrete, and their relationships are complex. First, although the inclusion of interaction terms can make a regression model more flexible, such models pay a much higher cost in interpretability than rule sets. Consider the interaction model with just three variables,

$$\mathbb{E}[Y|X_1, X_2, X_3] = \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_{1,2} X_1 X_2 + \beta_{1,3} X_1 X_3 + \beta_{2,3} X_2 X_3 + \beta_{1,2,3} X_1 X_2 X_3.$$

This is complicated but perhaps manageable when we are interested in only one specific variable (which we seldom are at the data exploration phase), but the totality of the model's substantive implications are arcane. Rule sets, on the other hand, retain interpretability for higher order interactions by using intuitive logical operators—for example, "if ($X_1 = 1$ and $X_2 = 1$) or ($X_3 = 1$), then $Y = 1$." This model tells us several things: (1) the relationship between X_1 and Y depends on X_2, and vice versa; and (2) the relationship between X_3 and Y does not depend on X_1 and X_2, and vice versa. To represent this simple statement using a regression model, we will need not only $\beta_{1,2} = \beta_3 = 1$ and $\beta_1 = \beta_2 = \beta_{1,3} = \beta_{2,3} = 0$, but also $\beta_{1,2,3} = -1$ to avoid "double-dipping"—that is, the "effect" of having both $(X_1, X_2) = (1, 1)$ and $X_3 = 1$ is the same as that of just having one be true.

The gap in interpretability quickly becomes chasmic. Consider the rule set "if ($X_1 = X_2 = X_3 = 1$) or ($X_4 = X_5 = X_6 = 1$), then $Y = 1$." To model this, regression uses a sixth-order interaction: The coefficient for $X_1 X_2 X_3 X_4 X_5 X_6$ must be negative to correct for "double dipping." Adding a third rule will complicates things further: The terms that correct for "double dipping" will over-correct for when all three rules are satisfied, and we will need a ninth-order interaction to remedy this if the third rule is also of length three. In greater generality, if there are J variables, regression would use up to LM-order interactions and $\sum_{m=1}^{M} \sum_{k=1}^{m} \binom{J}{k}$ non-zero terms to model a rule set with M rules each of at most length L. Regression uses very high-order interactions and many non-zero terms to model simple rule sets, and this complexity grows very quickly with the length of each rule and the number of rules. A benign description of some data by rule sets becomes grotesque by regression.

In addition to uninterpretability, another drawback of using such high-order interactions is the increase in the dimension of our data, which provokes the curse of dimensionality. For example, if we have J variables and want to include up to LM-order multiplicative interaction terms, the dimension of our data becomes $\sum_{k=1}^{LM} \binom{J}{k}$. Any regression method will inevitably struggle to select the right parameters. In the extreme, a fully saturated model will perfectly explain the data, or some models become inestimable without regularization. With this many terms, model selection is difficult, and overfitting is a concern. In online Supplementary Materials (SM), we explore this phenomenon in more detail. We also apply regularized regression, a LASSO, to our large-N empirical example and discuss the differences with the result from BRS using the ideas from this section.

The second reason is more subtle. Rule sets are classifiers, which directly predict the value of Y given X, whereas regression predicts $\mathbb{E}[Y|X]$. When Y is binary, these are not the same because the value of Y is much more restricted than that of $\mathbb{E}[Y|X]$. Consider the example described in Table 1. There are two binary explanatory variables, X_1 and X_2, and the sample has 150 observations. Regression will tell us, rightly, that X_2 has a positive

Table 1. Rule Set vs. Regression: A Toy Example

X_1	X_2	$\mathbb{E}[Y]$	N
0	0	0	50
1	0	1	50
0	1	0.5	50
1	1	–	0

effect on $\mathbb{E}[Y]$, holding X_1 constant. However, a rule set method will only include one rule: if $X_1 = 1$ then $Y = 1$. The reason why X_2 does not enter the rule set is that knowing the value of X_2 gives us no better chance at guessing Y.[3] Since a high-performing regression

[3]Conditional on knowing $X_1 = 1$, we already know the value of Y. Conditional on $X_1 = 0$, X_2 does give us some information: if $X_2 = 0$, then also $Y = 0$. However, we would do just as well by guessing $Y = 0$ whenever $X_1 = 0$, regardless of the value of X_2, as the best we could do by distinguishing between the cases when $(X_1, X_2) = (0, 0)$ and $(X_1, X_2) = (0, 1)$; we could correctly guess at most 125 of the cases either way. In both cases, we would also include the rule "$X_1 = 1$" and no other rules.

model will include both X_1 and X_2, the rule set is more parsimonious than regression in this case. Rule sets and regression have different goals and differentially reward models for the information they provide, and consequently provide non-redundant descriptions of the data.

3. Qualitative Comparative Analysis

Qualitative Comparative Analysis, or QCA, was originally developed by the sociologist Charles Ragin (1987). QCA focuses on combinations of conditions and their relation to the outcome, and rather than attempting to isolate the effects of a single variable, each case is viewed as a "configuration" of properties that must be considered as a whole (Ragin 2005).

Take Lipset (1959), for example. Using interwar Europe as a case study, the author argues that socioeconomic factors determine the likelihood of democratic consolidation. He categorizes countries as either "stable democracies" or "unstable democracies and dictatorships" and identifies wealth, industrialization, education, and urbanization as the common factors that characterize the former. Berg-Schlosser and Cronqvist (2005) study this setting using QCA. They use one discretized measure for each factor Lipset identifies: gross national product per capita for wealth, industrial labor force for industrialization, literacy for education, and population in towns with more than 20,000 inhabitants for urbanization. Wealth is split into three categories: high ($>$ \$850), medium (between \$550 and \$850), and low ($<$ \$550). All other variables are split into high and low, with the cut-offs being 30%, 75%, and 50% for industrialization, education, and urbanization, respectively.

We can associate a set with each value of each variable, such as the set of countries with high wealth or the set of countries with low industrialization. A configuration is an intersection of four sets, one associated with each variable. The collection $\{wealth_{v_w} \cap industrialization_{v_i} \cap education_{v_e} \cap urbanization_{v_u}\}$, in which $v_w \in \{low,\ medium,\ high\}$ and $v_i, v_e, v_u \in \{low, high\}$, exhaustively describes all possible configurations.

The configuration is, in effect, QCA's unit of analysis. If all cases in a configuration

have the same outcome, then the configuration is assigned that outcome. Otherwise, the configuration is "contradictory." Assuming there are no contradictions, QCA searches for a solution by taking the union of all configurations with positive outcomes and reducing it using Boolean algebra. Solution criteria can be framed in terms of (within-sample) necessity and sufficiency: QCA seeks to find a "minimally necessary disjunction of minimally sufficient conjunctions" (Schneider 2018, p. 249). Solutions are disjunctions of conjunctions of conditions, or a set union of set intersections (e.g., $(A \cap B) \cup C$), such that a case satisfies the solution if and only if it has a positive outcome. They must also be minimal in the sense that they cannot be further simplified using Boolean algebra.

Contradictions give rise to complications, because QCA is ill-equipped to deal with errors. In the modernization case, Czechoslovakia and Germany are a part of the same configuration but have different outcomes. The original formulation of QCA simply ignores both countries. Considering only non-contradictory configurations, the QCA solution is "(high wealth) OR (medium wealth AND low industrialization)."[4] Publications employing QCA may present data using a truth table, which is a table whose rows are unique configurations and whose columns are variables, the outcome, and possibly some other information on each configuration. Table 2 shows the first five rows of the table for this example.

TABLE 2. PARTIAL EXAMPLE OF A TRUTH TABLE

Wealth	Urbanization	Education	Industrialization	Outcome	n	Consistency
0	0	0	0	0	5	0
0	0	0	1	?	0	-
0	0	1	0	0	3	0
0	0	1	1	?	0	-
0	1	0	0	?	0	-

Note: Each row represents a configuration. For wealth, 0 = "low," 1 = "medium," and 2 = "high." For all other variables, 0 = "low" and 1 = "high." For outcome, "?" is used when there are no cases belonging to that configuration. n represents the number of cases in the data set for each configuration. *Consistency* is defined as the proportion of cases in a configuration that have a positive outcome. We present the full table, which has $3 \times 2^3 = 24$ rows, along with summary statistics of the data, in SM.

[4]This solution includes "logical remainders," which are often included to produce a more parsimonious solution (Rihoux and Ragin 2008). Including logical remainders allows the program to make assumptions about the outcomes of configurations with no data.

In response to criticism of QCA's handling of errors, scholars have proposed various solutions (Maggetti and Levi-Faur 2013). Ragin (2006) introduces the measure of "consistency" to handle the type of errors that most interest us—that is, the errors that produce heterogeneous outcomes or contradictions. Consistency is the proportion of cases in a configuration that have a positive outcome. Though modern discussions replace the language of contradictions with that of consistency, the two are intimately related: A configuration will have the maximum consistency score of 1 if and only if it is not contradictory, and consistency decreases proportionally to the number of contradictions.

For the purposes of finding a solution, the researcher sets a consistency threshold, above which a configuration is consistent enough to be treated as having a positive outcome, and below which it is treated as having a negative outcome. Consistency is thus a generalization of QCA's original framework, which uses a consistency threshold of 1. Though an improvement, this process still discards information by reducing many possible consistency scores to just two possible states. Consider the typical consistency threshold of .75. A configuration with 80% positive cases is treated the same as a configuration with 100% positive cases. More jarringly, a configuration with 70% positive cases is treated the same as one without any.

QCA also struggles with constraining the size of its solutions. QCA solutions are *more* parsimonious than all other potential solutions that are logically equivalent, but they are still not guaranteed to be parsimonious *enough* to be interpretable. If there exists a complex and irreducible combination of conditions that perfectly categorizes the data, then it will be the unique QCA solution, even if there exists a much simpler combination of conditions with marginally worse predictions. In other words, parsimony is a secondary objective.

This tendency to overlook parsimonious yet still highly predictive solutions hampers both interpretability and out-of-sample prediction, because complex rules are often overfitted. Consider a simple example with 100 cases, two binary variables A and B, and a binary outcome. Suppose 99 cases are in the configuration $A \cap B$ and have positive outcomes, and the remaining case is in the configuration $A \cap B'$, where B' denotes the complement of B,

and has a negative outcome. For any consistency threshold greater than zero, $A \cap B$ is the unique QCA solution. QCA ignores the possibility that A is the only important variable and that B's explanatory power may be spurious. This is most likely to be problematic when the data are heterogeneous and there are many variables, because then it becomes likely that there is a complex split of the data that explains within-sample patterns better than a simple one. We demonstrate this phenomenon with simulations comparing QCA and BRS.

QCA is also computationally expensive when data are large. The size of the problem grows exponentially, and there does not exist an efficient algorithm for solving the problem. If we have J variables with just two possible values each, then we have $\sum_{j=1}^{J} 2^j$ possible "and" statements. The QCA solution is the smallest collection of such statements that covers the positive class exactly (i.e., their union must equal the positive class). This is an instance of the set cover problem, which Karp (1972) famously proves to be NP-complete. As a consequence, the fastest known algorithm for solving the problem runs in exponential time. We show that QCA encounters computational problems for reasonable datasets.[5]

4. Bayesian Rule Set

In this section, we describe Bayesian Rule Sets (BRS), an alternative algorithm for learning rule sets. We alter BRS's priors to ease computation and improve interpretability, derive the penalties placed on complexity as functions of hyper-parameters, give recommendations on hyper-parameter specification and tuning, propose novel measures of uncertainty and stability, and propose a means of aggregating results from different bootstraps while preserving interpretability.

[5]Using the most recent and most efficient algorithm of which we know, consistency cubes (in particular, using version 3.13 of the `R` package `QCA`, published to CRAN in September of 2021, Dusa 2018; Dusa et al. 2019), we found that QCA is not computationally feasible at 250 observations and 20 variables, and possibly for fewer variables as well.

4.1. Setup

Because our framework is Bayesian in nature, our goal is to find a maximum a posteriori (MAP) solution, or the rule set that maximizes the posterior distribution over the set of all possible rule sets $\mathbb{A}$. That is, given data S and hyper-parameters H, our goal is to find a rule set $\hat{A} \in \mathbb{A}$ such that $\hat{A} \in \arg\max_{A \in \mathbb{A}} p(A|S;H)$.

Suppose we have N observations, J discrete variables, and a binary outcome. Let $y_n \in \{0,1\}$ be the outcome of the nth observation, let $x_{j,n} \in \{w_{j,1}, w_{j,2}, ..., w_{j,K_j}\}$ be the value of jth variable for the nth observation, where K_j is the number of total possible values of the jth variable, and let $x_n = (x_{1,n}, ..., x_{J,n})$. Let $V_{j,w_{j,k}}$ denote the set of observations with $x_{j,n} = w_{j,k}$. Let Y_+ denote the positive class and Y_- the negative class.

A rule a of length $l \in \mathbb{N}$ is the intersection of l subsets, $a = \bigcap_{j:j \in J_l} V_{j,w_{j,k}}$, where $J_l \subseteq \{1, ..., J\}$ and $|J_l| = l$. A rule set A is a union of one or more rules. We say an observation satisfies (or is covered by) a rule or rule set if it is an element of that rule or rule set. To satisfy a rule, it is necessary to satisfy all conditions making up that rule. To satisfy a rule set, it suffices to satisfy at least one rule making up the rule set. Let $\mathbb{A}_{L^{max}} = \{\bigcap a_i : length(a_i) \leq L^{max}\}$ denote the collection of all possible rule sets composed of rules of at most length L^{max}. A rule set classifies all the observations it covers as positive and all others as negative. The natural interpretation is, "if $x_n \in A$, then $y_n = 1$." We can also translate membership in A into words: The "and" operator is the semantic counterpart of intersection, and "or" is the counterpart of union.

4.2. Priors

BRS's Bayesian framework combines a prior that encourages parsimony with a likelihood that encourages accuracy.[6] Our prior is that we first draw the number of rules M from

[6] Wang et al. (2017) propose two versions of BRS: one that uses Beta priors (BRS-BetaBinomial), and one that uses Poisson priors (BRS-Poisson). Wang et al. (2017) focus on BRS-BetaBinomial, but the behavior of the priors depends on the data in use and is less intuitive. It also requires more hyper-parameters,

Poisson(λ). If $M = 0$, then we have the empty set. Otherwise, we then draw the length of each rule L_m, which must be between 1 and a preset maximum length L^{max}, independently from Truncated-Poisson(η), $m = 1, 2, ..., M$. Then for each rule we choose L_m variables uniformly at random, and finally we choose the value for each variable uniformly at random. This prior is uninformative about the variables in our rule set and the values these variables take on, but it is informative about the complexity of our rule set. We can manipulate λ and η to guide parsimony: If they are small, then the probability that the number or length of rules is large will also be small.

4.3. Likelihood

Given a rule set A, let $\rho_+ \sim$Beta(α_+, β_+) be the probability that an observation satisfying A has a positive outcome, and let $\rho_- \sim$ Beta(α_-, β_-) be the probability that an observation not satisfying A has a negative outcome. This is to say that the probability of an observation being in either the positive or the negative class is distributed differently conditional on whether the observation is part of the rule set. Formally,

$$y_n | x_n, A \sim \begin{cases} \text{Bernoulli}(\rho_+) & \text{if } x_n \in A \\ \text{Bernoulli}(1 - \rho_-) & \text{if } x_n \notin A. \end{cases}.$$

The hyper-parameters $\alpha_+, \beta_+, \alpha_-$, and β_- guide the solution's performance. If we choose them such that $\mathbb{E}[\rho_+]$ and $\mathbb{E}[\rho_-]$ are close to one, then the algorithm will penalize misclassification. Such choices would make our hyperpriors informative about our model's performance.

complicating tuning. We present a modified version of BRS-Poisson that constrains the maximum length of a rule so that the problem is more computationally feasible and the solutions are more interpretable. If we do not constrain the maximum length of a rule, the size of the candidate rule pool grows exponentially with the number of variables, and some of these rules will be too long to be interpretable.

4.4. Algorithm for Approximate MAP Estimation

The posterior distribution is the product of the prior, the likelihood, and a standardizing constant. Maximizing the posterior is thus equivalent to maximizing the sum of logs of the prior and the likelihood. We refer to this sum as our objective function $F(A, S; H) = \log p(A; H) + \log p(S|A; H)$.[7]

The solution is easy to recover through an exhaustive search if J is small, but because the size of $\mathbb{A}$ grows quickly with J, this is not computationally feasible when J is even moderately large. Wang et al. (2017) provides two bounds that give us tractability. They show that the support of each rule in the solution (i.e., the number of cases that each rule covers) is bounded from below, and that the number of rules in the solution is bounded from above. This allows us to first mine for a relatively small set of feasible rules and to then search for the rule set in a smaller search space.

Intuitively, a small number of rules must collectively cover a large number of positive cases, and thus each rule must cover a sufficient number of cases. Define the support of a rule as the number of observations the rule covers, $supp(a) = |\{n : x_n \in a\}|$. We can first construct a sequence of upper bounds $\{M^{[t]}\}_t$ on the number of rules in the optimal rule set $M^* = |A^*|$. Each $M^{[t]}$ is a decreasing function of the "room for improvement" remaining over the best rule set found by time t—that is, the difference between the maximum of the objective function and the maximum value of the objective function achieved by time t. We can then lower bound the support of each rule in the optimal rule set with $C^{[t]}$, an increasing function of $M^{[t]}$. As t increases and better solutions are found, the room for improvement decreases and the minimum support increases. These bounds enable exact inference but require updating the pool of candidate rules at each step, because our bounds are a function

[7] The prior is $p(A; \lambda, \eta) \propto \text{Poisson}(M; \lambda) \prod_{m=1}^{M} \text{Poisson}(L_m; \eta) / \binom{L^{max}}{L_m} \prod_{k=1}^{L_m} 1/K_{v_{m,k}}$, where $K_{v_{m,k}}$ is the total number of possible values for the variable $v_{m,k}$. The likelihood is $p(S|A; \alpha_+, \beta_+, \alpha_-, \beta_-) = \frac{B(TP+\alpha_+, FP+\beta_+)}{B(\alpha_+, \beta_+)} \frac{B(TN+\alpha_-, FN+\beta_-)}{B(\alpha_-, \beta_-)}$, where TP is the number of true positives, FP is the number of false positives, TN is the number of true negatives, and FN is the number of false negatives.

of t. Wang et al. (2017) proposes to cull all rules below a pre-specified minimum support threshold C and use the same pool of candidate rules at each step.[8] This results in an "approximate" but more efficient algorithm. There is no theoretical guarantee that the resulting solution will be the MAP, but this is not concerning as long as the solution is still interpretable and high performing; we can view the Bayesian framework as a means of attaining such solutions, and the quality of maximizing the posterior is not necessarily an inherently valuable trait.

To conduct the search, Wang et al. (2017) proposes using a simulated annealing algorithm. Here, we provide a brief sketch of the algorithm. We begin by generating the pool of candidate rules, $\mathcal{A} = \{a : supp(a) \geq C,\ length(a) \leq L^{max}\}$. Next, simulated annealing generates a discrete state, discrete time Markov chain $(A^{[t]})_{t=0}^{N_{\text{iter}}}$, where N_{iter} is the maximum allowed number of steps. We choose the initial rule set $A^{[0]}$ at random, and with each subsequent step, we update the rule set $A^{[t+1]}$ by probabilistically choosing between the previous rule set $A^{[t]}$ and its neighbors, which are rule sets that differ by one rule. Specifically, we randomly select a misclassified point and consider only the neighbors that correctly classify that point. If the point is in the positive class, then the algorithm would either add a new rule to $A^{[t]}$ or replace a rule in $A^{[t]}$ with one that covers the point. If the point is in the negative class, then the algorithm selects a rule in $A^{[t]}$ that covers the misclassified point and either removes it or replaces it with one that does not cover the point.

When picking $A^{[t+1]}$, the algorithm probabilistically chooses between greediness and exploration to avoid getting trapped at local maxima. Conditional on the choice of misclassified point and action (ADD, CUT, or REPLACE), the greedy choice is to choose the neighbor that maximizes the objective function. The alternative options are to either choose a feasible neighbor uniformly at random or to keep the current rule set. The probability of each

[8] If there are very many variables, the search space may still be too large even after culling rules with low support. In this case, Wang et al. (2017) proposes further screening rules based on information gain. If plotted on an ROC (receiver operating characteristic) curve, these rules would be the upper leftmost rules.

choice depends on how much the objective function would increase and how many iterations have passed. Choices that increase the objective function more are more likely to be chosen. Exploration is prioritized earlier in the chain, and greediness is emphasized later on. We give more details of the algorithm, including exact probabilities used and explanations of the ADD, CUT, and REPLACE operations, in the SM.

4.5. Hyper-Parameters

BRS influences complexity and performance using hyper-parameters, but existing work gives little guidance on how to tune parameters to meet context-specific needs. We remedy this. In our formulation of the problem, we define the penalty for the length of a rule as the difference in log-priors incurred from adding a variable v to a rule of length $l \in \{1, 2, ..., L^{max} - 1\}$, $L^{max} \geq 2$, to create a rule of length $l + 1$. We show this to be $\log \frac{K_v(L^{max}-1)}{\eta}$, where K_v is the number of values that v can take on. For any l and any v, this is a strictly decreasing function of η. That is, the smaller our choice of η, the more the model penalizes longer rules. To guarantee the penalty is positive, we should choose $\eta < 2$.

We define the penalty for the number of rules as the difference in log-priors incurred from adding a rule of length L_{M+1} to a rule set with M rules. This function is more complex, but we can lower bound it using $\log \frac{M+1}{\lambda} \frac{2L^{max}}{Poisson(1;1)}$. If we choose $\lambda < 4/exp(1) \approx 1.47$, then this penalty is always positive. The penalty is always a strictly decreasing function of λ, and it is also a strictly decreasing function of η whenever $\eta \in (0, 2)$. This implies we can simultaneously manipulate the penalty for both the number of rules and the length of each rule by shifting η.

The likelihood, on the other hand, guides performance. To encourage accurate solutions, we should choose α_+, α_-, β_+, and β_- such that $\mathbb{E}[\rho_+] = \frac{\alpha_+}{\alpha_+ + \beta_+}$ and $\mathbb{E}[\rho_-] = \frac{\alpha_-}{\alpha_- + \beta_-}$ are close to 1. If these expectations are greater than half, then penalty for misclassification is positive, with the penalty growing with the expectations. In the SM, we give more details on hyper-parameter tuning and describe an automated procedure that only requires the user

to specify a parsimony-based stopping criteria (e.g., the number of rules or the sum of rule lengths, which can be either a specific value or a range of values).

4.6. Addressing Uncertainty and Stability Using the Bootstrap

In this section, we propose using bootstrapping to address two concerns with using rule sets: difficulty quantifying uncertainty, and result instability. Bootstrapping gives us measures of uncertainty, and aggregating rules from different bootstraps improves the algorithm's stability and allows us to be more explicit about performance criteria.

Researchers often care not just about point estimates, but also about uncertainty. In Bayesian statistics, the typical measure of uncertainty is the credible interval or region, which is the region of the parameter space in which a parameter has some probability of being. In the continuous case, we can summarize an interval using its end point. For a discrete parameter space with no ordering such as $\mathbb{A}_{L_{max}}$, however, we cannot define an interval or region. Therefore, a confidence interval is inappropriate. An analogous concept might be to construct a "credible collection" of rule sets $CC(\alpha) \subseteq \mathbb{A}_{L_{max}}$ such that the posterior over the rule sets in $CC(\alpha)$ sum to at least $1-\alpha$, $\sum_{A \in CC(\alpha)} p(A; S, H) \geq 1-\alpha$, but there are several challenges to this approach. Chief among them is that this collection would be difficult to interpret. It would also be expensive to find a highest posterior density collection, since this would involve sorting exponentially many rule sets based on their posterior.

Alternatively, then, we propose turning to the popular frequentist tool, the bootstrap, and a novel measure of uncertainty, prevalence, which we define as the proportion of bootstrapped rule sets in which a rule appears. Prevalence gives us a sense of how small perturbations in the data change our result, but note that this is more of a heuristic than a formal test. We measure the prevalence of individual rules as opposed to rule sets because the exact same rule sets may appear too infrequently to be informative.

We can also use bootstrapping to construct frequentist confidence intervals for various statistics of interest. In our examples, we investigate a statistic similar to support, which we

call coverage, and use quantiles of the statistic calculated on each bootstrapped dataset to estimate confidence intervals. In small sample settings, this can further help identify rules that may have spurious predictive power.

Rule sets can also suffer from instability. Even if improvements in the objective function tend to quickly approach zero, there is always a chance that we will find a marginally better rule later in the search. Perturbations in the data can also alter results. This is also true for regression and other methods, but small changes in numerical estimates tend not to carry substantive meaning, whereas changes to rule sets may. To stabilize results, we propose using high-prevalence rules from bootstrapping as a pool of candidate rules and picking a rule set based on some performance- and parsimony-related criteria. As an added benefit, it is easier to customize the objective function. Our aggregation method based on bootstrapped results takes two simple steps. First, we take the five most prevalent rules of each length with at least 5% prevalence as candidate rules. Setting a minimum threshold for prevalence helps guard against spurious rules. Another potential precaution (which we do not take in our examples) is to exclude rules whose confidence interval for correct coverage includes zero. Second, we conduct an *exhaustive* search to find the rule set with at most three rules that maximizes in-sample accuracy.[9] Because the search space is small, an exhaustive search is computationally inexpensive. We can also use this method on a training sample and reserve a test data set to estimate out-of-sample accuracy.

[9]This is equivalent to empirical risk minimization with a 0-1 loss on the class of rule sets $\{\cap_{i=1}^{k} a_i : a_i \in \mathcal{M}, k \leq 3\}$, where $\mathcal{M} = \{a_i : prev(a_i) \geq \max\{.05, prev_{(5)}^{\{length(a_i)\}}\}$ is the set of rules whose prevalence $prev(\cdot)$ are at least 5% and in the top five for rules of the same length, and $prev_{(k)}^{\{l\}}$ is the kth highest prevalence amongst rules of length l. Alternatively, we could consider minimizing some weighted sum of false positives and false negatives to reflect unequal misclassification costs, or including a penalty term for rule length.

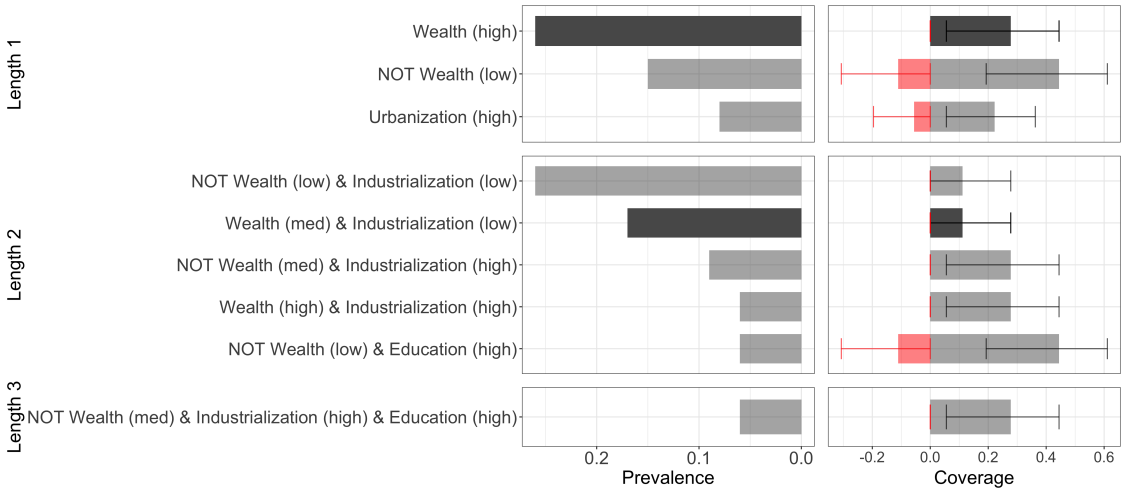
5. Graphical Tools and Quantifying Uncertainties

Existing tools in political science are designed to represent numerical results, such as those from regression, and there is a dearth of methods for presenting rule sets. To help BRS thrive in a research setting, we introduce tools for effectively and efficiently communicating its results. In particular, we propose to using bar plots, chord diagrams, and t-SNE plots.

First, we propose using bar plots to distinguish high prevalence rules and to present how well these rules perform. This method presents uncertainty about both the rules themselves and the rules' performance. We define *coverage* as the proportion of observations that a rule covers and separately plot correct and incorrect coverage, which are positive and negative cases covered by the rule. We use error bars to indicate the 2.5th and 97.5th quantiles of both correct and incorrect coverage from the bootstrap.

To demonstrate this method, we run BRS on 100 bootstraps on the Lipset data, using a maximum length of three, $\alpha_+ = \alpha_- = 500 \times 10^2$, $\beta_+ = \beta_- = 1 \times 10^2$, and $\lambda = \eta = 1$. We run the algorithm for two chains of 1,000 iterations each time. The left panel of Figure 1

FIGURE 1. PREVALENCE AND COVERAGE OF RULES: DEMOCRATIC CONSOLIDATION



Note: Includes up to five most prevalent rules of each length that appear in at least 5% of 100 iterations. True positives are represented by positive values for coverage and false positives by negative ones. Error bars mark the 2.5th and 97.5th percentiles. Rules used in the aggregated rule set are highlighted with darker colors.

shows the prevalence of each rule, and the right panel shows coverage. Rules used in the

aggregated rule set are highlighted with darker colors. Wealth on its own appears to be a powerful predictor of democratic survival, although there seems to be ambiguity as to what level of wealth is required. All countries with high wealth were in the positive class, but not all countries in the positive class have high wealth. Conversely, almost all (8 of 10) countries with either medium or high wealth—NOT Wealth (low)—were also in the positive class, and all countries in the positive class have either medium or high wealth, giving the rule set composed of just that rule 89% in sample accuracy.

We can achieve slightly better accuracy by using a more complex rule set. Using our method of aggregation, we identify the rule set that most accurately classifies the data as "[Wealth (high)] OR [Wealth (med) & Industrialization (low)]."[10] This is the same as the QCA result and achieves just over 94% accuracy.

Figure 2(a) presents our aggregated rule set using a chord diagram.[11] A chord diagram is a graphical method used for visualizing connections (called "edges" in graph theory) between entities (called "nodes" or "vertices"). Nodes are represented as arcs that together form a circle, and edges are represented using "chords." We represent variables with nodes, and the chords represent rules. In this case, wealth appears by itself in one rule and again in a different rule alongside industrialization.

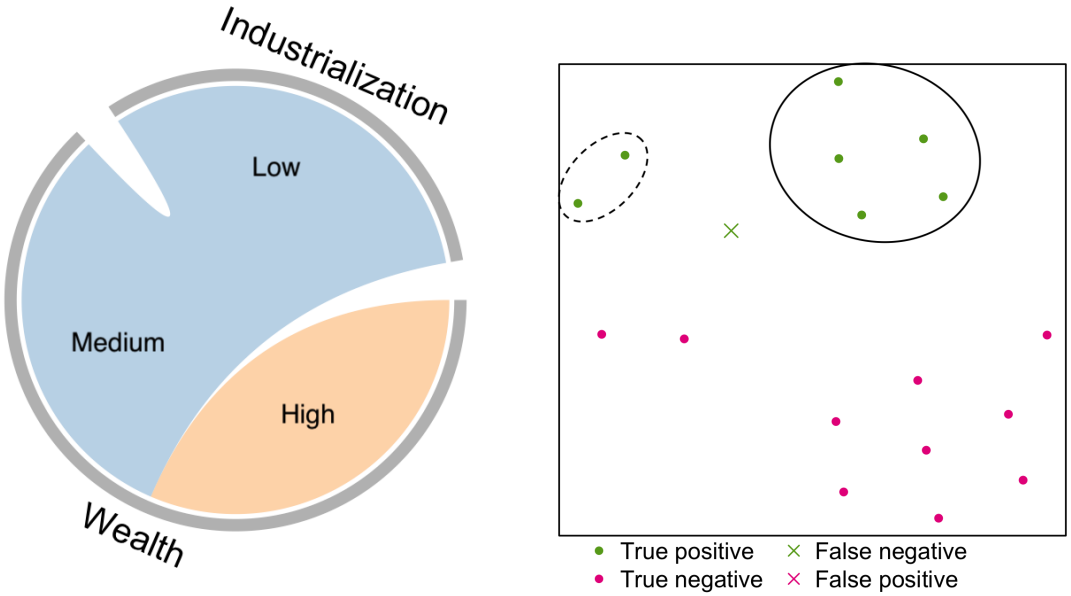
Last, we propose using t-SNE plots to visualize the data and rule set coverage. t-SNE is a method that embeds high-dimensional data on low-dimensional space, allowing for easier visualization (van der Maaten and Hinton 2008).[12] The t-SNE plot allows us to visualize

[10] We use parsimony (number of rules, then average length of rules) to break ties, and then arbitrarily amongst rules with equal performance and parsimony. Even amongst rule sets of the same size, there are several rule sets that cover the same points. For example, we can use the rule "Wealth (not low) & Industrialization (low)" instead of "Wealth (med) & Industrialization (low)."

[11] We construct it using the `circilize` package in R developed by Gu et al. (2014).

[12] t-SNE improves and computationally simplifies an earlier method, stochastic neighbor embedding (SNE) (Hinton and Roweis 2002), while maintaining the original framework for preserving neighborhood identity. SNE and t-SNE both model the probability of two points picking each other as neighbors based on how dissimilar they are in high-dimensional space. The model also defines a corresponding probability of being

FIGURE 2. INTERACTIONS AND RULE SET COVERAGE: DEMOCRATIC CONSOLIDATION



(a) Chord diagram representing interactions between variables. Colors of chords are purely for aesthetic clarity.

(b) t-SNE plot depicting rule set coverage. Solid line encircles points covered by "Wealth (high)," and dashed line encircles points covered by "(Wealth (med) & Industrialization (low))."

our rule sets and the points they cover. Figure 2(b) is a t-SNE plot of the Lipset dataset constructed using only the outcome variable and the continuous measures of wealth and industrialization. Points with $Y = 1$ are colored green, and points with $Y = 0$ are colored red. Points covered by the rule set are circles, and points not covered are crosses. As expected, points classified by the same rules are also clustered together: The two cases covered by "medium wealth and low industrialization" form one cluster, and the five covered by "high wealth" form another cluster. Only one point is misclassified—the rule set does not cover one positive case.

In summary, we propose three visualization tools for BRS, namely bar graphs to commu-

neighbors in low-dimensional space, but this is not immediately obtainable since we do not observe any point's location in low-dimensional space. Instead, the algorithm minimizes a cost function that penalizes divergences in these probabilities to find a low-dimensional representation of the data. It is shown to be more capable of demonstrating nonlinear and complex relationships in data than other dimension reduction methods such as the Principle Component Analysis.

nicate rule prevalence and coverage; chord diagrams to present interactions; and t-SNE plots to visualize data and rule set coverage and performance. They are also compatible with rule sets produced by any method, including QCA. In the next two sections, we present evidence that BRS has unique advantages over QCA and regression methods in modeling complex relationships, especially when data are large and noisy, using both Monte Carlo simulations and an empirical example.

6. Monte Carlo Simulations

In this section, we study the performance of QCA and BRS using simulations. We examine the complexity and out-of-sample predictions of both methods using 36 variants of our DGP by varying the number of observations, the number of variables, and the conditional probability of an observation having a positive outcome given its membership in the "true rule set."

The general setup for our DGP is as follows. We randomly generate a value for each observation and each variable, $x_{j,n} \sim Bernoulli(.5)$ for $n = 1, ..., N$ and $j = 1, ..., J$. We set the true rule set A^* to $(V_1 \cap V_2) \cup (V_3 \cap V_4 \cap V_5')$, where V_5' is the complement of V_5. Then we generate values for each observation's outcome probabilistically, conditioning on whether the observation satisfies the true rule set A^*. That is, $P(x_n \in Y_+ | x_n \in A^*) = p_+$ and $P(x_n \in Y_+ | x_n \notin A^*) = p_-$. Note that when $p_+ = 1$ and $p_- = 0$, outcomes are generated deterministically, and an observation is a member of the positive class if and only if it satisfies the true rule set. We let $N \in \{25, 100, 250, 500, 750, 1000\}$, $J \in \{5, 10, 20\}$, and $(p_+, p_-) \in \{(1, 0), (.75, .25)\}$. For each choice of J and (p_+, p_-), we generate a population with $100,000$ observations and take 100 random samples each of size N for each choice of N. We apply QCA and BRS to each sample and use the resulting rule set to classify the observations outside of the sample. For each setting and method we record the average number of rules, the average length of each rule, and the average out-of-sample accuracy.

For QCA, we use the R package developed by Dusa et al. (2019) and include logical remainders. We do not need to set a minimum consistency threshold in the deterministic setting (i.e., $(p_+, p_-) = (1, 0)$), but we do in the probabilistic setting (i.e., $(p_+, p_-) = (.75, .25)$). In our simulations, we set the threshold at .67 to allow QCA to recognize positive configurations even when random error pushes consistency slightly below $p_+ = .75$.

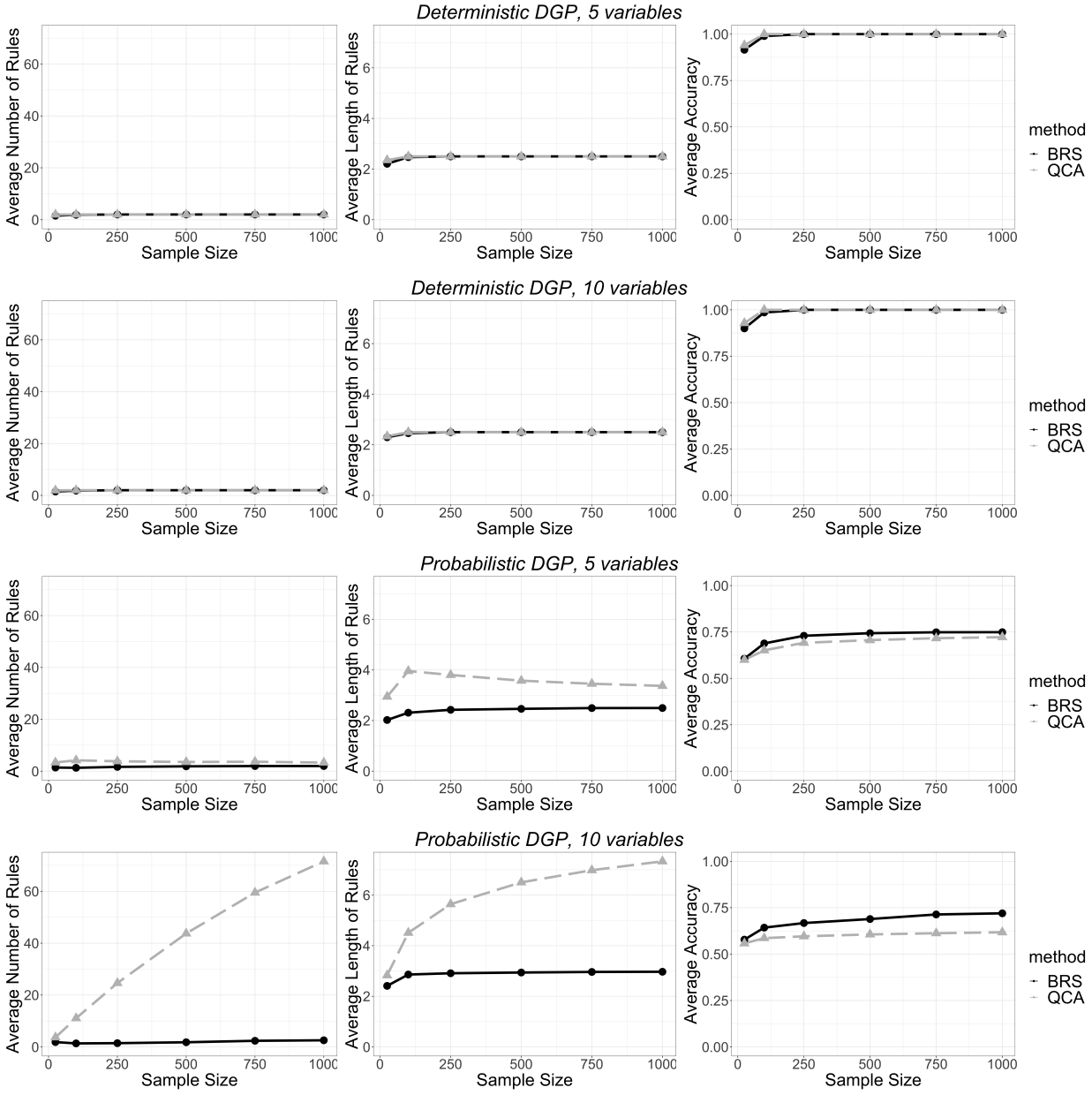
For BRS, we let the maximum length of a rule be three, and we scale hyper-parameters based on the setting. In particular, we scale the strength of the likelihood with the number of variables, $\alpha_+ = \alpha_- = 500 \times 10^2/J$, $\beta_+ = \beta_- = 1 \times 10^2/J$, to reflect the inverse relationship between the value of in-sample fitness and dimensionality: The more variables there are, the more likely that there are rule sets that spuriously perform well. In line with our own recommendations, we scale the priors with sample size to prevent them from being overwhelmed by the likelihood as the sample size grows, $\lambda = 1$ and $\eta = (N/10)^{-5}$. We run BRS for five chains of 2,000 iterations each and set the minimum support to 5% and the maximum number of rules to include in the candidate pool to $5,000$.

Figure 3 presents the results of our simulations. We find that in deterministic DGP settings, both QCA and BRS reproduce the true rule set if the sample is sufficiently large. When the sample is small, QCA performs marginally better, but the difference is minute. In probabilistic settings, however, we find that BRS outperforms and is more interpretable than QCA. BRS achieves higher out-of-sample accuracy than QCA and preserve interpretability, whereas QCA solutions explode in size as N increases when there are moderately many variables. We observe interpretability break down starting at $J = 10$ and $N = 100$. That QCA strays further from the true rule set as N increases is an extremely concerning property for quantitative researchers who expect their methods to converge to the truth as N grows.

We also attempt simulations with 20 variables. The results reaffirm the trends in interpretability and performance that we observe with 10 variables. We also find that QCA ceases to be computationally feasible with a sample size of 250 or more.[13] Tables from all

[13]We were unable to produce a solution in ten hours using Sherlock, the High-Performance Computing cluster

FIGURE 3. AVERAGE SOLUTION PARSIMONY AND OUT-OF-SAMPLE PERFORMANCE



Note: Results from Monte Carlo simulations. For each DGP setting, we apply each method to the same 100 samples and calculate average statistics for parsimony and performance. Left column plots average number of rules in each set, middle column plots average number of conditions in each rule, and right column plots average out-of-sample accuracy.

settings and plots with 20 variables can be found in the SM.

In summary, in deterministic settings, QCA and BRS are comparable, but in probabilistic settings, BRS is advantaged. In such probabilistic settings, BRS is the only practical option

at Stanford. For comparison, the cluster completed all 3,600 simulations for BRS in five hours.

when there are even a moderate number of variables and observations for reasons of predictive power, computational feasibility, and interpretability. Since all large-N settings are also probabilistic, we may instead say that QCA is only well suited to small-N settings, whereas BRS is decidedly advantaged when N is large.

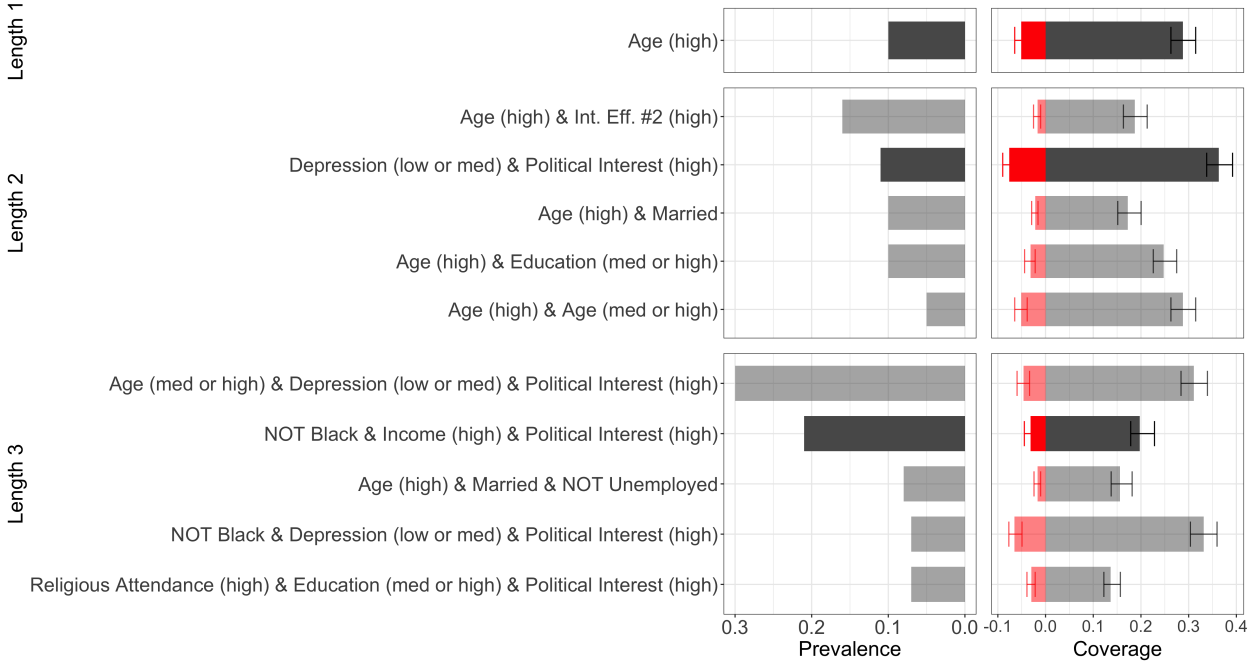
7. A Large-N Example: Depression and Voter Turnout

In this section, we apply BRS to a large N empirical example. In the appendix, we contrast the BRS results with results from other methods, such as the LASSO and Random Forest, and explain why they serve very different goals. In brief, other popular methods do not recover a description of the data that is substantively similar to the descriptions we present in this section.

Landwehr and Ojeda (2021) use survey data and regression to estimate the effect of depression on voter turnout, controlling for other variables. In this section, we present results from BRS to demonstrate how the method offers a unique perspective on the authors' data. The survey asked 1,014 respondents 13 questions concerning their demographics (gender, age, education, income, marital status, underemployment, Black, and Hispanic) and behavior or beliefs (voter turnout, two measures of "internal efficacy," political interest, and religious attendance), plus a battery of questions to measure depressive symptoms. Survey data are particularly suitable for BRS, since responses tend to be categorical. In this case, responses were either ordinal or binary. We created binary features that broke up ordinal variables into fewer, overlapping, incremental categories (e.g., "low," "low or medium," and "high"). We set the maximum length of a rule to be $L = 3$, the minimum support to 5%, the maximum number of rules to 5,000, $\alpha_+ = \alpha_- = 500$, $\beta_+ = \beta_- = 1$, and $\lambda = 1, \eta = 1/1000$. We ran the algorithm for 10 chains of 10,000 iterations each.

Figure 4 presents the most prevalent rules of each length from 100 bootstraps. The most prevalent rule of length one is "high age," and the rule is a relatively good predictor

FIGURE 4. PREVALENCE AND COVERAGE OF RULES: EDUCATIONAL ATTAINMENT



Note: Includes the five most prevalent rules of each length appearing in at least 5% of 100 bootstraps. True positives are represented by positive coverage values and false positives by negative ones. Error bars mark the 2.5th and 97.5th percentiles. Rules used in the aggregated rule set are highlighted with a darker color.

of turnout: Most respondents covered by the rule voted, and the rule also covers a large portion of voters. The cut-off for high age is 57 years old, which is the top tercile for age and which also roughly corresponds to the demarcation between baby boomers and younger generations. Previous work has documented that there exists a sharp drop-off in voter turnout at this generational shift, making the cut-off both mathematically convenient and substantively meaningful (Rubenson et al. 2004). This result also coincides with Landwehr and Ojeda (2021)'s original finding that age is one of the most important predictors of turnout. Age frequently appears in rules of length two and three, but not consistently or more prevalently with the same variable. Combined with the fact that age often appears by itself, the instability in age's interactions suggests that these interactions may be spurious.

The more novel results pertain to other variables. "High political interest" and "low or medium depression" are blunt rules (not shown in Figure 4 due to low prevalence) in the sense that they misclassify many non-voters. Each condition can be significantly improved upon, at the cost of some additional complexity and marginally less coverage of voters, by

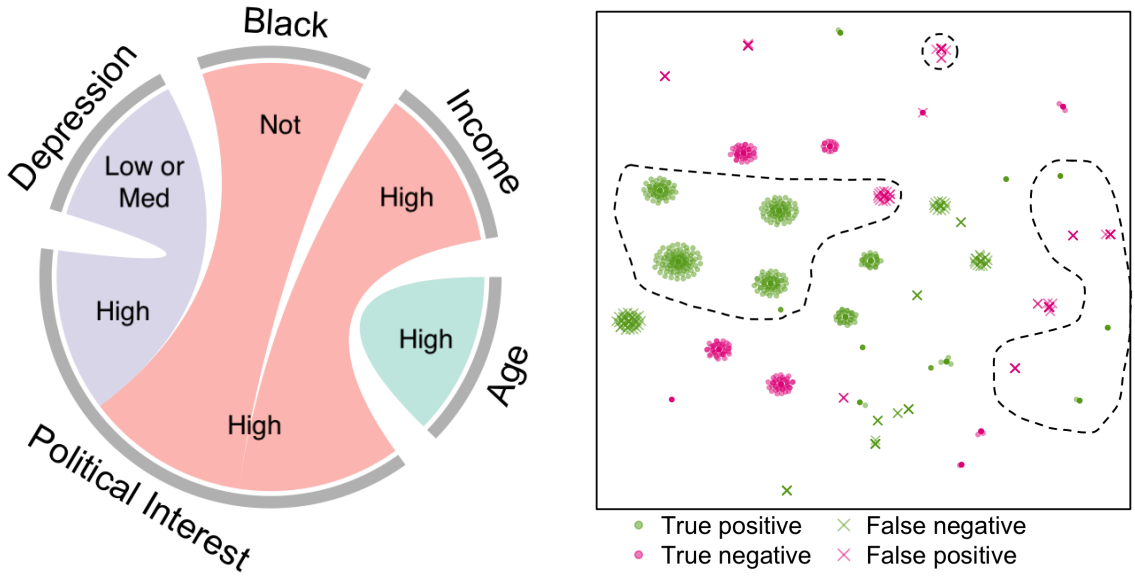
combining both into a single rule. Indeed, their conjunction is the second most prevalent rule of length two. One interpretation of this result is that depression prevents many respondents who are interested in politics and want to participate from doing so; it blocks the conversion of political interest into political participation. Not only does BRS suggest that depression has a different "effect" on different groups of people, but it also identifies political interest as an especially salient dimension of heterogeneity. For a comparison, if we condition on high political interest, the "effect" of moving from low/medium to high depression on a respondent's probability of voting is 25% higher than if we instead condition on medium/high age and over twice as large if we condition on high age.[14] Figure 5(b) is a t-SNE plot of the data, with observations covered by this rule encircled using dashed lines—these observations are located in three seemingly disconnected "neighborhoods" because t-SNE takes into account similarities in all five variables that appear in the rule set.

We aggregate the rules that appear in Figure 4, again with the goal of maximizing accuracy with at most three rules. The resulting rule set is "age (high); or depression (low or med) and political interest (high); or not Black, income (high), and political interest (high)." We highlight the component rules. Figure 5(a) further visualizes the rule set.

We can think of adding rules to a rule set as a means of covering residual cases, i.e., cases not covered by rules that are already part of the rule set. The first two rules, "age (high); or depression (low or med) and political interest (high)," collectively cover most (70.6%) of the positive class, and, as discussed, are relatively precise. A rule set composed of just these two rules would already have about 70.4% accuracy. The only way to further increase accuracy would be to add a rule that covers more previously uncovered positive cases than it does previously uncovered negative cases; adding a rule cannot expand the negative class by reclassifying cases already covered by other rules. Improvements are also likely to be

[14] By this we mean $P(Y = 1|polint = high, depression \in \{low, med\}) - P(Y = 1|polint = high, depression \notin \{low, med\})$=0.247; $P(Y = 1|age \in \{med, high\}, depression \in \{low, med\}) - P(Y = 1|age \in \{med, high\}, depression \notin \{low, med\})$=0.2; and $P(Y = 1|age = high, depression \in \{low, med\}) - P(Y = 1|age = high, depression \notin \{low, med\})$=0.11.

FIGURE 5. INTERACTIONS AND RULE SET COVERAGE: DEPRESSION AND VOTER TURNOUT



(a) Chord diagram representing interactions between variables. Colors of chords are purely for aesthetic clarity.

(b) t-SNE plot depicting rule set coverage. Dashed lines encircle points covered by the rule "depression (low or med) and political interest (high)."

marginal, because, in general, adding rules to a rule set has diminishing returns: Even if the new rule's precision and coverage remain high, its coverage will likely overlap with the coverage of existing rules.

In this case, we can marginally improve upon the first two rules by adding on a third rule, "not Black, income (high), and political interest (high)." This rule covers an additional 6.4% of the positive class and brings accuracy to 73.5%. A respondent satisfies this third rule but neither the first nor the second rule if she is interested in politics, young, depressed, non-Black, and high income. That is, being non-Black and high income is often enough for young respondents with an interest in politics to vote, even if they are depressed. Non-Blackness and high income shield citizens from popular mechanisms of disenfranchisement, such as voter ID laws and felon disenfranchisement, and political parties are more likely to target wealthier individuals with their mobilization efforts (Franko, Kelly and Witko 2016; Cobb, Greiner and Quinn 2010; Behrens, Uggen and Manza 2003). It would thus be unsurprising for race and income to appear somewhere in our rule set, but what is interesting

is that blackness and income appear together in this high-prevalence rule and seldom apart elsewhere, suggesting an interaction between these two forms of privilege.

We can also rewrite the union of our second and third rules as "political interest (high) and [(depression (low or med)) or (not Black and income (high))]." In a sense, depression and privilege are exchangeable for the purposes of predicting voter turnout. One interpretation of this result is that interest in politics might increase the utility an individual derives from voting, and this utility is more likely to outweigh the costs of voting for individuals not experiencing depression and individuals with a particular form of racial and class privilege, for whom voting is less costly or more rewarding.

Alternatively, we can view both the second and third rules as additions to the first rule, "high age," and as a characterization of post-Gen X voters. Older generations are very likely to vote, regardless of other factors. Younger voters, however, are almost as likely to not vote as they are to vote, and individual characteristics become important for predicting turnout. In particular, interest in politics, which appears in both the second and third rules, is very informative: Over 65% of younger respondents who say they are not interested in politics did not vote (compared with just over 25% of older respondents). But interest in politics alone is still a rather crude classifier. Depression and racial and class background further refine our understanding of the behavior of young respondents with an interest in politics. Even if they are interested in politics, the benefit of voting is often not significant enough for young respondents fighting depression and race- and class-based barriers to vote, but freedom from at least one of these is frequently sufficient to catalyze voting.

Using rule sets, we have explored the complex ways in which age, interest in politics, depression, race, and class interact to explain voter turnout. This would be difficult to uncover using regression, for which model specification and interpretation quickly become challenging as the order of interactions grows, especially if we do not already have a theory or specific variable in mind. Using BRS, we have explored a large-N, moderate-J setting that was previously inaccessible to QCA users. Though BRS is primarily a tool for classification

and data description, it can lead to causal inquiries. In particular, it has the potential to discover treatments with heterogeneous effects. The results for this example might lead to an investigation of the effect of political interest on voter turnout in different subpopulations. We might theorize that the effect is smaller than average on high-age individuals (who are highly likely to vote, independent of other variables) and design an experiment to test our hypothesis. The effect might also be smaller on individuals who are depressed, Black, or not high income than on those who are either not depressed or both non-Black and high income.

8. Conclusion

In this paper, we introduce BRS as an alternative to QCA when data are large and heterogeneous. BRS is an interpretable and high-performing classifier and merits inclusion in the social scientist's toolkit. By using set membership to represent relationships, BRS models high-order interactions and other complexities in an interpretable fashion. This contrasts the prevailing trend in machine learning toward black box methods like neural networks and Random Forest, which are ill-suited for comprehension. BRS also divorces rule sets from their roots in determinism in the social sciences and casts learning rule sets as a statistical and machine learning problem. In doing so, BRS resolves QCA's inability to handle contradictory cases without discarding information, forfeiting interpretability, or overfitting to the data. Monte Carlo simulations support these claims. BRS also takes more seriously the potential for random errors in the data and, with our additions, attempts to quantify the resulting uncertainty. Furthermore, BRS always remains computationally feasible, whereas QCA becomes too expensive to solve when there are many variables.

We make three contributions to the literature. First, we identify and propose solutions to problems specific to BRS. To facilitate computation, interpretability, and ease of use, we propose amendments to BRS's priors, derive penalties for complexity as functions of hyper-parameters, and provide guidance in hyper-parameter tuning for practitioners. Second,

we address more general challenges with using rule sets. We propose using bootstrapping and prevalence to quantify uncertainty, and aggregating high-prevalence rules from different bootstraps. Third, we introduce and showcase graphical tools to aid with result and data visualization. We provide an open-source `R` package to facilitate implementation of BRS, our modifications, and the proposed visualization tools.

We apply BRS to two empirical examples: one application, which occurs in a small-N macro-level setting with few variables, was previously analyzed using QCA; another one, which occurs in a large-N micro-level setting with moderately many variables, is unable to be analyzed by QCA. Using our proposed method of aggregating bootstraps, we replicate the findings of QCA using BRS in the first example. In the second example, BRS produces interpretable results using data with highly complex interactions, demonstrating a key advantage of using rule sets over regression.

Our recommended workflow with BRS takes three steps. First of all, BRS requires discrete data and a binary outcome. We recommend discretizing data by creating overlapping, increasing sets (e.g., if X is continuous, we can create indicator variables for "$X < percentile(X, 25\%)$," "$X < median(X)$," "$X < percentile(X, 75\%)$," etc.). This method allows us to increase the number of categories without making the data matrix sparse. Once data are prepared, researchers need to tune hyper-parameters. Using our proposed method with a Poisson prior, users only need to set a criteria for interpretability, such as the maximum number of rules and the maximum length of each rule. To help quantify uncertainty and create a more stable rule set we recommend researchers run BRS with bootstrapping. If the moderator with which a variable is interacted varies a lot by bootstrap, this may indicate the interaction is spurious. If the confidence interval for a rule's coverage includes zero, this may indicate that the rule's explanatory power is spurious.[15] Finally, researchers can use the graphical tools we provide to interpret BRS results. The primary goal

[15] We can opt to exclude these rules from aggregation, or create separate aggregated rule sets with and without them. We may also evaluate the performance of our rule set on a hold-out set.

is to uncover interesting, and interpretable, descriptions of the data or discover dimensions of heterogeneity, which can in turn inspire causal inquiries. At this task, BRS holds a distinct advantage over other methods currently available in political science.

References

Achen, Christopher H. 2005. "Two cheers for Charles Ragin." *Studies in Comparative International Development* 40(1):27.

Behrens, Angela, Christopher Uggen and Jeff Manza. 2003. "Ballot manipulation and the "menace of Negro domination": Racial threat and felon disenfranchisement in the United States, 1850–2002." *American Journal of Sociology* 109(3):559–605.

Berg-Schlosser, Dirk and Lasse Cronqvist. 2005. "Macro-quantitative vs. macro-qualitative methods in the social sciences—an example from empirical democratic theory employing new software." *Historical Social Research/Historische Sozialforschung* pp. 154–175.

Clarke, Kevin A. 2020. "Logical constraints: the limitations of QCA in social science research." *Political Analysis* 28(4):552–568.

Cobb, Rachael V, D James Greiner and Kevin M Quinn. 2010. "Can voter ID laws be administered in a race-neutral manner? Evidence from the city of Boston in 2008." *Quarterly Journal of Political Science* 7(1):1–33.

Dusa, Adrian. 2018. "Consistency Cubes: a fast, efficient method for exact Boolean minimization." *R Journal* 10(2).

Dusa, Adrian, Vasil Dinkov, Dmitry Baranovskiy, Emmanuel Quentin, Jimmy Breck-McKye and Alrik Thiem. 2019. "Package 'QCA'.".

Fiss, Peer C, Dmitry Sharapov and Lasse Cronqvist. 2013. "Opposites attract? Opportunities and challenges for integrating large-N QCA and econometric analysis." *Political Research Quarterly* pp. 191–198.

Franko, William W, Nathan J Kelly and Christopher Witko. 2016. "Class bias in voter turnout, representation, and income inequality." *Perspectives on Politics* 14(2):351–368.

Greckhamer, Thomas, Vilmos F Misangyi and Peer C Fiss. 2013. "The Two QCAs: From a small-N to a large-N set theoretic approach." *Research in Sociology of Organizations* 38:49–75.

Gu, Zuguang, Lei Gu, Roland Eils, Matthias Schlesner and Benedikt Brors. 2014. "circlize implements and enhances circular visualization in R." *Bioinformatics* 30(19):2811–2812.

Hinton, Geoffrey E and Sam Roweis. 2002. "Stochastic neighbor embedding." *Advances in neural information processing systems* 15:857–864.

Hug, Simon. 2013. "Qualitative comparative analysis: How inductive use and measurement error lead to problematic inference." *Political analysis* 21(2):252–265.

Karp, Richard M. 1972. Reducibility among combinatorial problems. In *Complexity of computer computations*. Springer pp. 85–103.

Krogslund, Chris and Katherine Michel. 2014. "A larger-N, fewer variables problem? The counterintuitive sensitivity of QCA." *The Counterintuitive Sensitivity of QCA (January 20, 2015). Qualitative & Multi-Method Research* 12(1). Available at SSRN: https://ssrn.com/abstract=2552940.

Landwehr, Claudia and Christopher Ojeda. 2021. "Democracy and depression: a cross-national study of depressive symptoms and nonparticipation." *American Political Science Review* 115(1):323–330.

Lipset, Seymour Martin. 1959. *Political man: The social bases of politics*. Vakils, Feffer and Simons Private.

Lucas, Samuel R and Alisa Szatrowski. 2014. "Qualitative comparative analysis in critical perspective." *Sociological Methodology* 44(1):1–79.

Maggetti, Martino and David Levi-Faur. 2013. "Dealing with errors in QCA." *Political Research Quarterly* pp. 198–204.

Ragin, Charles C. 1987. *The comparative method: Moving beyond qualitative and quantitative strategies*. Univ of California Press.

Ragin, Charles C. 2005. "Core versus tangential assumptions in comparative research." *Studies in comparative international development* 40(1):33.

Ragin, Charles C. 2006. "Set relations in social research: Evaluating their consistency and coverage." *Political analysis* pp. 291–310.

Rihoux, Benoît and Charles C Ragin. 2008. *Configurational comparative methods: Qualitative comparative analysis (QCA) and related techniques*. Vol. 51 Sage Publications.

Rihoux, Benoît, Priscilla Álamos-Concha, Damien Bol, Axel Marx and Ilona Rezsöhazy. 2013. "From Niche to Mainstream Method? A Comprehensive Mapping of QCA Applications in Journal Articles from 1984 to 2011." *Political Research Quarterly* 66(1):9–18.

Rohlfing, Ingo. 2018. "Power and false negatives in qualitative comparative analysis: foundations, simulation and estimation for empirical studies." *Political Analysis* 26(1):72–89.

Roig-Tierno, Norat, Tomas F Gonzalez-Cruz and Jordi Llopis-Martinez. 2017. "An overview of qualitative comparative analysis: A bibliometric analysis." *Journal of Innovation & Knowledge* 2(1):15–23.

Rubenson, Daniel, André Blais, Patrick Fournier, Elisabeth Gidengil and Neil Nevitte. 2004. "Accounting for the age gap in turnout." *Acta Politica* 39(4):407–421.

Schneider, Carsten Q. 2018. "Realists and Idealists in QCA." *Political Analysis* 26(2):246–254.

Schneider, Carsten Q and Claudius Wagemann. 2010. "Standards of good practice in qualitative comparative analysis (QCA) and fuzzy-sets." *Comparative Sociology* 9(3):397–418.

Seawright, Jason. 2005. "Qualitative comparative analysis vis-à-vis regression." *Studies in comparative international development* 40(1):3–26.

Seawright, Jason. 2014. "Comment: Limited diversity and the unreliability of QCA." *Sociological Methodology* 44(1):118–121.

van der Maaten, Laurens and Geoffrey Hinton. 2008. "Visualizing data using t-SNE." *Journal of machine learning research* 9(Nov):2579–2605.

Wang, Tong, Cynthia Rudin, Finale Doshi-Velez, Yimin Liu, Erica Klampfl and Perry MacNeille. 2017. "A Bayesian framework for learning rule sets for interpretable classification." *The Journal of Machine Learning Research* 18(1):2357–2393.

Wang, Tong, Cynthia Rudin, Finale Velez-Doshi, Yimin Liu, Erica Klampfl and Perry MacNeille. 2016. Bayesian rule sets for interpretable classification. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*. IEEE pp. 1269–1274.

A. Supplementary Materials *for*

Bayesian Rule Set: A Quantitative Alternative to Qualitative Comparative Analysis

Table of Contents

A.1. Implementation Details of BRS

A.1.1. BRS Algorithm

A.1.2. Penalty Functions

A.1.3. Hyper-Parameter Tuning

A.2. Rule Sets versus Regression

A.2.1. Describing Rule Sets Using Regression

A.2.2. Model Selection Using the LASSO

A.2.3. Regression versus Classification

A.3. Monte Carlo Evidence

A.4. Additional Information on the Empirical Examples

A.4.1 Truth Table of Lipset (1959)

A.4.2 Analyzing Voter Turnout With Tree Based Methods

A.1. Implementation Details of BRS

A.1.1. BRS Algorithm

Here, we provide pseudocode and further details for the BRS algorithm.

Algorithm 1 MAP Estimation Algorithm

procedure SIMULATED ANNEALING(N_{iter}, C)
  $\mathcal{A} \leftarrow \text{GenerateRules}(S, C)$
  $A^{[0]} \leftarrow$ a randomly chosen rule set
  for t = 0 to N_{iter} **do**
    $(x_n, y_n) \leftarrow$ a random point such that $a(x_n) \neq y_n$
    if $y_n = 1$ **then**
      $A^{[t+1]} \begin{cases} ADD(A^{[t]}, x_n) \text{ with probability } \frac{1}{2} \\ REPLACE(A^{[t]}, x_n) \text{ with probability } \frac{1}{2} \end{cases}$
    else
      $A^{[t+1]} \begin{cases} CUT(A^{[t]}, x_n) \text{ with probability } \frac{1}{2} \\ REPLACE(A^{[t]}, x_n) \text{ with probability } \frac{1}{2} \end{cases}$
    $A_{\max} \leftarrow \underset{A \in \{A_{\max}, A^{[t+1]}\}}{\arg\max} F(A)$
    $\alpha = \min\{1, exp(\frac{F(A^{[t+1]}) - F(A^{[t]})}{T(t)})\}$
    $A^{[t+1]} \begin{cases} A^{[t+1]} \text{ with probability } \alpha \\ A^{[t]} \text{ with probability } 1 - \alpha \end{cases}$
  return $A_{\max}$

Let $\mathcal{R}_1(x_n)$ and $\mathcal{R}_0(x_n)$ denote the set of rules that x_n does and does not satisfy, respectively. The three possible actions are performed as follows:

- ADD:
  1. With probability p, choose $z \in \mathcal{R}_1(x_n)$ uniformly at random. With probability 1-p, choose $z = \underset{w \in \mathcal{R}_1(x_n)}{\arg\max} F(A^{[t]} \cup w)$.
  2. Then, $A^{[t+1]} \leftarrow A^{[t]} \cup z$.
- CUT:
  1. With probability p, choose $z \in \mathcal{R}_0(x_n) \cap A^{[t]}$ uniformly at random. With probability 1-p, choose $z = \underset{w \in \mathcal{R}_0(x_n) \cap A^{[t]}}{\arg\max} F(A^{[t]} \setminus w)$.
  2. Then, $A^{[t+1]} \leftarrow A^{[t]} \setminus z$.
- REPLACE: first CUT, then ADD.

A.1.2. Penalty Functions

Rule length. First, we describe the penalty incurred for using longer rules. We define this as the difference in the log-prior for some rule A and for the rule A' obtained by adding a condition to one of the rules belonging to A, all else equal.

Let $A_{\{a_l\}}$ denote an arbitrary rule set with at least one rule of length l, a_l. Let $A^{+v}_{\{a_l\}}$ denote the rule set obtained by adding a condition on the variable v to a_l. In other words, we replace a_l with $a_l \cap V_{v,w}$ for some value $w \in \{w_{v,1}, ..., w_{v,K_v}\}$.

The penalty for using $A^{+v}_{\{a_l\}}$ instead of $A_{\{a_l\}}$ is,

$$\begin{aligned}
\phi(A_{\{a_l\}}, v; S, H) :=& \log p(A_{\{a_l\}}; \eta, L^{max}) - \log p(A^{+v}_{\{a_l\}}; \eta, L^{max}) \\
=& \log Poisson(l; \eta) - \log \binom{L^{max}}{l} \\
& - \log Poisson(l+1; \eta) + \log \binom{L^{max}}{l+1} + \log K_v \\
=& \log \frac{l+1}{\eta} + \log \left(K_v \frac{L^{max} - l}{l+1} \right) \\
=& \log \frac{K_v (L^{max} - 1)}{\eta}.
\end{aligned}$$

For any l and any v, this is a strictly decreasing function of η. That is, the smaller our choice of η, the more the model penalizes longer rules. Note that $K_v \geq 2$ for any v, so $K_v(L^{max} - 1) \geq 2$. Thus, to guarantee that the penalty is always positive, we should choose $\eta < 2$. We can tailor this bound for a specific dataset (setting K_v to be $\min_u K_u$), but in practice we will seldom want $\eta \geq 2$.

Number of rules. Next, we describe the penalty incurred for using more rules. We define this to be the difference in the log-prior for some rule A and for some rule A' obtained from adding a rule to A.

Let $A_{[M]}$ denote a rule set with M rules, and let $A^{+\mathbf{v}_{M+1}}_{[M]}$ denote the rule set obtained from adding a rule composed of the variables $\mathbf{v}_{M+1} = (v_{i_1}, v_{i_2}, \ldots, v_{i_{L_{M+1}}})$. That is, $A^{+\mathbf{v}_{M+1}}_{[M]} = A_{[M]} \cup (\bigcap_{v_i \in \mathbf{v}_{M+1}} v_i)$.

The penalty for using $A^{+\mathbf{v}_{M+1}}_{[M]}$ instead of $A_{[M]}$ is,

$$\begin{aligned}
\psi(A_{[M]}, \mathbf{v}_{M+1}; S, H) :=& \log p(A_{[M]}; \lambda, \eta, L^{max}) - \log p(A^{+\mathbf{v}_{M+1}}_{[M]}; \lambda, \eta, L^{max}) \\
=& \log Poisson(M; \lambda) - \log Poisson(M+1; \lambda) \\
& - \log Poisson(L_{M+1}; \eta) + \log \binom{L^{max}}{L_{M+1}} + \sum_{v_i \in \mathbf{v}_{M+1}} \log K_{v_i}.
\end{aligned}$$

This function is considerably more complex than $\phi(\cdot)$, but we can bound it from below:

$$\psi(A_{[M]}, \mathbf{v}_{M+1}; S, H) \geq \log \frac{M+1}{\lambda} - \log Poisson(1;1) + \log L^{max} + \log K_{v_1},$$

with equality being achieved when $\eta = 1$ and $L_{M+1} = 1$. Note that $K_{v_1} \geq 2$, and so if we choose $\lambda < 4/exp(1) \approx 1.47$, then the penalty is always positive. We can create an adaptive threshold based on the data and choice of η, but as before, we will seldom want larger λ. Note that $\psi(\cdot)$ is always a strictly decreasing function of λ, and it is also a strictly decreasing function of η over the part of the domain $\{\lambda \in \mathbb{R}^+, \eta \in (0, 2)\}$.

A.1.3. Hyper-Parameter Tuning

The values of the hyper-parameters will vary by a number of factors, including the sample size, the dimension, the distribution of your data, and the desired level of complexity. Thus, it is somewhat difficult to give rules of thumb for hyper-parameters.

The amount the posterior depends on the likelihood is a function of the sample size N. In the limit, the likelihood dominates the prior and the MAP is equal to the maximum likelihood estimate. This is not desirable, because the priors are what preserve interpretability. Thus, as N increases, we should scale the priors (i.e. decrease η) to combat this effect. Towards the other extreme, when N is very small, the prior might have an outsized influence and force the solution to be more parsimonious than necessary at the expense of performance. In this case, rather than increasing λ and η above the bounds we derive (which would make the penalties negative and reward complexity instead of penalizing it), we recommend that users instead strengthen the likelihood. The criteria for interpretability might also vary from context to context. We may also be more worried about overfitting as the dimensionality of the data increases, which might mean we want to place less emphasis on performance or more emphasis on parsimony.

One way of tuning hyper-parameters is simply through trial and error. If the solution is too complex, we can penalize complexity more. If it is "too" parsimonious (i.e., you are willing to sacrifice parsimony for performance), we can penalize misclassification more.) To increase the penalty for complexity, we recommend either decreasing η by itself or both η and λ by a common factor less than one. We might prefer to manipulate only η, because simultaneously decreasing λ and η results in a double jeopardy situation. The number of rules will be penalized very heavily, which can result in rule sets with very few rules that are each as long as the maximum rule length L^{max}. To increase the penalty for misclassification, we suggest increasing α_+ and α_- or $\alpha_+, \alpha_-, \beta_+$ and β_- by common a factor greater than one.

We recommend starting with relatively weak priors that do not heavily penalize complexity (e.g., $\lambda = \eta = 1$) and iteratively decreasing η until the model is sufficiently parsimonious for the application at hand. We advise having a stopping condition, such as a maximum number of rules or sum of rule lengths, before starting the search. For the likelihood, we suggest either using

$\alpha_+ = \alpha_- = 500$ and $\beta_+ = \beta_- = 1$, or determining them based on sample size with a formula similar to the one we use in our simulations.

Trial and error may be a bit involved, and there may be concerns over overfitting. Guided by the same ideas, we can create a recursive procedure for automatically tuning hyper-parameters. The procedure begins with hyper-parameters that fall within the bounds stated above, e.g., $\alpha_+ = \alpha_- = 500$, $\beta_+ = \beta_- = 1$, and $\lambda = \eta = 1$. There are three possible outcomes:

1. **Solution is too complex:** We increase the penalty for complexity. Then, we recurse.
2. **Solution is exactly as complex as the stopping criteria:** The procedure terminates.
3. **Solution is too parsimonious:** We compare this solution to the one obtained from recursing with increased penalties for misclassification. If the new solution performs better, we accept it. Otherwise, we accept the original solution. The procedure terminates.

Some concerns are that this can take a long time, and there is no guarantee a recursive algorithm will terminate. We can add in some other stopping conditions to ease these concerns. We can also turn it into a linear search. To do so, we first run the algorithm once to determine which of the three situations above our initial hyper-parameters create. If we are in the first or third case, we iteratively increase the respective penalties until we either satisfy or exceed the stopping criteria. If the latter, we pick the more parsimonious of the last two solutions (the one that is too parsimonious; the other will be too complex).

If using the bootstrap, we recommend tuning hyper-parameters for the original (not bootstrapped) data and using those hyper-parameters for all bootstraps.

A.2. Rule Sets versus Regression

Readers may wonder whether regression can accomplish what BRS does. We demonstrate here that it cannot do so in a way that is sufficiently interpretable. That is, although we can construct a similar description of the data using a linear model with multiplicative interaction terms, we would need to use interactions of order greater than the maximum length of each rule and many more nonzero terms than the number of rules. Moreover, popular methods like LASSO will not recover substantively interpretable descriptions of the data. This is partly due to the difference in philosophy; regression is not naturally a classifier and will identify variables as important for reasons different than those of a classification method like BRS. Below we explain these three points sequentially.

A.2.1. Describing Rule Sets Using Regression

Consider the aggregated rule set used for our example on voter turnout,

(1) Age (high); or
(2) Depression (low or med) and political interest (high); or
(3) Not black, income (high), and political interest (high).

This implies that the "effect" of being depressed is positive, but only for individuals who *both* (I) are interested in politics and (II) are not *either* (i) high age or (ii) both not black and high income.

To describe the positive effect of depression conditional on high political interest, we could use a positive slope for the term $Dep_{lm} \times PolInt_h$. But to ensure that the effect is zero for those who satisfy rules (1) or (3), we would also need negative slopes for the terms $Age_h \times Dep_{lm} \times PolInt_h$ and $Black_0 \times Income_{high} \times Dep_{lm} \times PolInt_h$. If we had two rules each of length three and with no variables in common, we would need to use a term of order six.

To make things even more complex, this description only works if we know the rule set a priori and thus know to include the term $PolInt_h$ instead of $PolInt_l$, or $Black_0$ instead of $Black_1$, or Dep_{lm} instead of Dep_h.[A1] If we instead used $Black_1$, we would then need to $Income_{high} \times Dep_{lm} \times PolInt_h$ positive but $Black_1 \times Income_{high} \times Dep_{lm} \times PolInt_h$ negative so that people who satisfy rule (3) do not have their effects over corrected.

Regression can be more flexible than rule sets. For example, the effect of rule (2) can be smaller for people with high age but be nonzero. However, this flexibility comes at considerable expense. Models are much more difficult to interpret when the order of interactions is even moderately high, and if rule sets can describe a relationship, it will be much easier and more interpretable to do so using rule sets than regression.

A.2.2. Model Selection Using the LASSO

To include the entire class of models that BRS can describe using rules of at most length three, we would need sixth order interactions. This would result in a matrix with over 3 billion entries, which is larger than the theoretical maximum that a program like `R` can store. Instead, then, we attempt the simpler task of selecting variables from up to third order interactions using regression.

If we generate up to third order interaction terms for our data and apply the LASSO (using cross-validation to select a penalty parameter), we end up with 46 nonzero terms. Because we wish to use the method for discovery and do not have a specific variable of interest, this is far too many to interpret. We can subsequently use OLS on the nonzero terms selected by LASSO, and seven of them will be significant at the 5% level. They are,

1. $Age_{lm} \times PolInt_l$ (< 0),
2. $Age_l \times Black_1 \times Motivation_{mh}$ (< 0),
3. $Age_l \times Educ_l \times Inc_h$ (< 0),

[A1] We cannot include both variables from each pairing due to multicollinearity.

4. $Age_{lm} \times Educ_{lm} \times Inc_{lm}$ (< 0),

5. $Attend_{mh} \times Dep_h \times IntEff1_{mh}$ (< 0),

6. $Educ_l \times Inc_{mh} \times Motivation_{lm}$ (< 0), and

7. $Inc_l \times IntEff1_l \times PolInt_1$ (< 0),

where (< 0) denotes that the slope was negative.

Without question, regression recovers a description of the data that is *different* for BRS. This observation alone substantiates the utility of rule sets: Although it is theoretically possible to construct a linear model that has similar implications as a rule set, regression methods will generally not recover said model. Additionally, we argue that the regression result gives a less interpretable description of the data: The substantive implications of this hodgepodge of significant interactions is rather arcane.

A.2.3. Regression versus Classification

Part of the reason that regression and BRS tend to recover disparate descriptions of the data are because they were designed with different goals in mind. Regression and classification both seek to explain variation in the outcome, but regression predicts a continuous outcome, whereas classification predicts a binary outcome. If we give regression data with a binary outcome, it will often be interpreted as predicting $\mathbb{E}[Y]$, which is continuous.

Although we can always turn a regression model into a classifier by picking some cutoff value for $\hat{Y}$, above which we predict $Y = 1$ and below which we predict $Y = 0$, this is not the objective the algorithm has in mind when fitting the model. Instead, most algorithms will reward the ability to predict $\mathbb{E}[Y]$, which is not always helpful for predicting Y. Below we provide more detailed explanation of the toy example presented in the main text.

Table A1. Toy Example in the Main Text

X_1	X_2	$\mathbb{E}[Y]$	N
0	0	0	50
1	0	1	50
0	1	0.5	50
1	1	–	0

As stated in the mann text, regression will tell us, rightly, that X_2 has a positive effect on $\mathbb{E}[Y]$, holding X_1 constant. However, rule sets will not identify X_2 as important $Y = 1$. "X_2" or "not X_2" by themselves are clearly not good rules: knowing the value of just X_2 gives us no better chance at guessing the outcome, because $P(Y = 1) = P(Y = 1|X_2 = x) = 0.5$ for $x \in \{0, 1\}$. Conditional on knowing $X_1 = 1$, we already know the value of Y. Conditional on $X_1 = 0$, X_2 does give us some

information: if $X_2 = 0$, then also $Y = 0$. However, we would do just as well by guessing $Y = 0$ whenever $X_1 = 0$, regardless of the value of X_2, as if we did not distinguish between the cases when $(X_1, X_2) = (0, 0)$ and $(X_1, X_2) = (0, 1)$: we could correctly guess at most 125 of the cases either way. In both cases, we would also include the rule "$X_1 = 1$" and no other rules.

Because rule sets classify points as negative by omission rather than inclusion, a natural follow-up question might then be whether reversing the coding of the outcome changes this result, but in this case, it does not. If we reverse the outcome, then indeed the rule "$X_1 = 0$ and $X_2 = 0$" has high precision, but using just this rule will still only correctly classify 125 cases (the first two rows of Table 1 and half of the cases in the third row), and there is no way to improve upon this by adding more rules. Using only the rule "$X_1 = 0$" achieves the same accuracy: the only difference is that it classifies the third row as positive instead of negative, but this is no worse of a choice. In short, knowing the value of X_2 does not help us make better guesses about the data.

We can generalize this result to other classifiers. The central idea here is that even though knowing $(X_1, X_2) = (0, 0)$ gives us information, that information does not help us make better predictions. This is because when $(X_1, X_2) = (0, 1)$, we are no better off guessing one way or the other. Thus, we can effectively ignore this scenario, fit our model with only the remaining data in mind, and simply accept whatever that model ends up saying about these cases. And when we ignore this scenario, X_2 does not offer any additional explanatory power on top of what X_1 already tells us about the data. In fact, there would be no variation at all in X_2.

This example is clearly quite contrived, but it sheds light on an important distinction between regression and classification. We reward regression for telling us that $\mathbb{E}[Y|X] = 0.5$, but from a classification point of view, this information is useless. For values not exactly 0.5, it is helpful for classification to know $\mathbb{E}[Y|X]$, but regression and classification will place different values on this knowledge.

A.3. Monte Carlo Evidence

FIGURE A1. AVERAGE SOLUTION PARSIMONY AND OUT-OF-SAMPLE PERFORMANCE FROM 100 DATASETS IN EACH DGPS

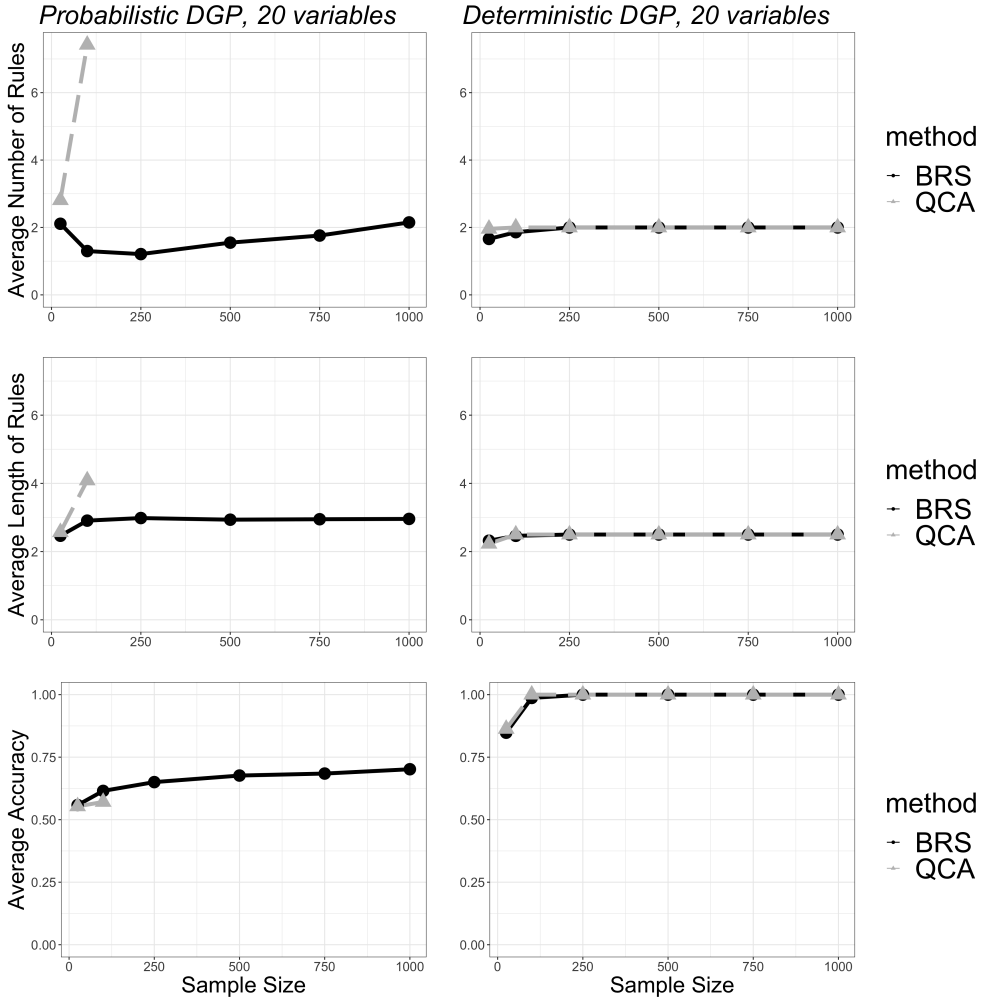


Table A2. Average solution parsimony and out-of-sample performance from 100 datasets with deterministically generated outcomes

(a) 5 variables

	BRS			QCA		
N	Num. Rules	Rule Len.	Accuracy	Num. Rules	Rule Len.	Accuracy
25	1.54	2.21	0.92	1.96	2.35	0.94
100	1.89	2.47	0.99	2.00	2.50	1.00
250	2.00	2.50	1.00	2.00	2.50	1.00
500	2.00	2.50	1.00	2.00	2.50	1.00
750	2.00	2.50	1.00	2.00	2.50	1.00
1000	2.00	2.50	1.00	2.00	2.50	1.00

(b) 10 variables

	BRS			QCA		
N	Num. Rules	Rule Len.	Accuracy	Num. Rules	Rule Len.	Accuracy
25	1.51	2.30	0.90	1.93	2.34	0.93
100	1.85	2.46	0.99	2.00	2.50	1.00
250	2.00	2.50	1.00	2.00	2.50	1.00
500	2.00	2.50	1.00	2.00	2.50	1.00
750	2.00	2.50	1.00	2.00	2.50	1.00
1000	2.00	2.50	1.00	2.00	2.50	1.00

(c) 20 variables

	BRS			QCA		
N	Num. Rules	Rule Len.	Accuracy	Num. Rules	Rule Len.	Accuracy
25	1.66	2.32	0.85	1.96	2.23	0.86
100	1.86	2.46	0.99	2.00	2.50	1.00
250	2.00	2.50	1.00	2.00	2.50	1.00
500	2.00	2.50	1.00	2.00	2.50	1.00
750	2.00	2.50	1.00	2.00	2.50	1.00
1000	2.00	2.50	1.00	2.00	2.50	1.00

Note: Results in each row are based on 100 simulated datasets with deterministically generated outcomes. "Num. Rules" and "Rule Len" represent the average number of rules and average rule length, respectively. Accuracy = 1.00 is the theoretical upbound based on the DGP.

TABLE A3. AVERAGE SOLUTION PARSIMONY AND OUT-OF-SAMPLE PERFORMANCE FROM 100 DATASETS WITH PROBABILISTICALLY GENERATED OUTCOMES

(a) 5 variables

	BRS			QCA		
N	Num. Rules	Rule Len.	Accuracy	Num. Rules	Rule Len.	Accuracy
25	1.39	2.02	0.61	3.35	2.94	0.60
100	1.29	2.31	0.69	4.18	3.96	0.65
250	1.65	2.42	0.73	3.82	3.80	0.69
500	1.86	2.46	0.74	3.53	3.58	0.71
750	1.97	2.49	0.75	3.63	3.45	0.72
1000	1.98	2.49	0.75	3.30	3.37	0.72

(b) 10 variables

	BRS			QCA		
N	Num. Rules	Rule Len.	Accuracy	Num. Rules	Rule Len.	Accuracy
25	1.85	2.41	0.58	3.67	2.83	0.56
100	1.35	2.86	0.64	11.05	4.51	0.59
250	1.44	2.91	0.67	24.62	5.64	0.60
500	1.79	2.94	0.69	43.73	6.50	0.61
750	2.35	2.96	0.71	59.49	6.98	0.61
1000	2.55	2.97	0.72	71.56	7.33	0.62

(c) 20 variables

	BRS			QCA		
N	Num. Rules	Rule Len.	Accuracy	Num. Rules	Rule Len.	Accuracy
25	2.11	2.46	0.56	2.81	2.57	0.55
100	1.30	2.91	0.62	7.42	4.09	0.57
250	1.21	2.98	0.65	NA	NA	NA
500	1.55	2.94	0.68	NA	NA	NA
750	1.76	2.95	0.68	NA	NA	NA
1000	2.15	2.96	0.70	NA	NA	NA

Note: Results in each row are based on 100 simulated datasets with probabilistically generated outcomes. "Num. Rules" and "Rule Len" represent the average number of rules and average rule length, respectively. Accuracy = 0.75 is the theoretical upbound based on the DGP.

A.4. Additional Information on the Empirical Examples

A.4.1. Truth Table of Lipset (1959)

Table A4. Full Truth Table of example used in table 2

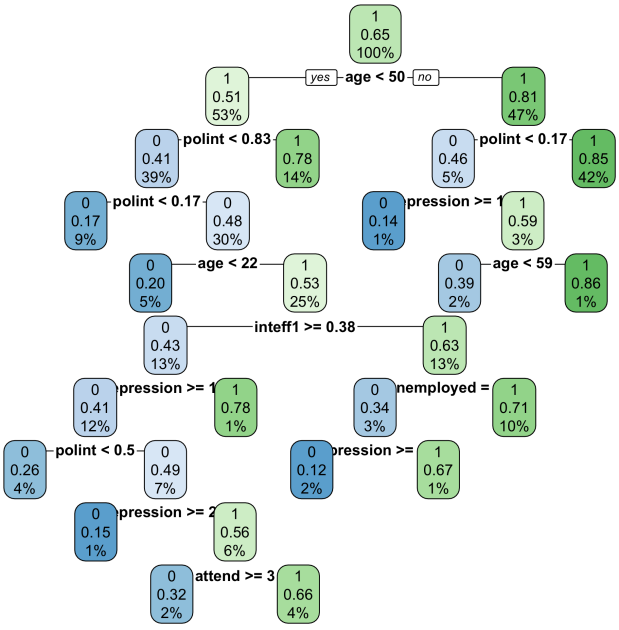
Wealth	Urbanization	Literacy	Industrialization	Outcome	n	Consistency
0	0	0	0	0	5	0
0	0	0	1	?	0	-
0	0	1	0	0	3	0
0	0	1	1	?	0	-
0	1	0	0	?	0	-
0	1	0	1	?	0	-
0	1	1	0	?	0	-
0	1	1	1	?	0	-
1	0	0	0	?	0	-
1	0	0	1	?	0	-
1	0	1	0	1	2	1
1	0	1	1	0	1	0
1	1	0	0	?	0	-
1	1	0	1	?	0	-
1	1	1	0	?	0	-
1	1	1	1	0	2	0.5
2	0	0	0	?	0	-
2	0	0	1	?	0	-
2	0	1	0	?	0	-
2	0	1	1	1	2	1
2	1	0	0	?	0	-
2	1	0	1	?	0	-
2	1	1	0	?	0	-
2	1	1	1	1	3	1

Key: "?" indicates no observations belong to that configuration.

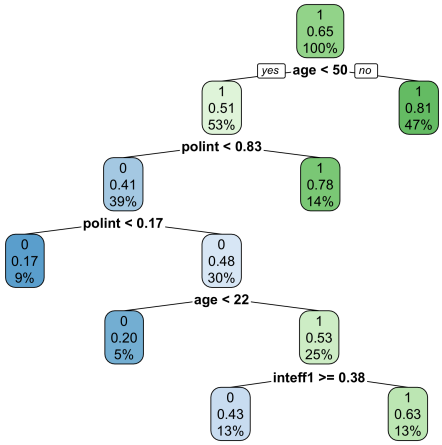
A.4.2. Analyzing Voter Turnout With Tree Based Methods

Aside from regression, we can use other methods to learn about voter turnout. In this section, we will discuss the results obtained from decision trees and Random Forest. We pick these two methods in particular because decision trees are actually another way of constructing rule sets, and Random Forest, its bagged counterpart, is popular in political science. Wang et al. (2017) has more extensive comparisons of BRS with other classification methods.

FIGURE A2. DECISION TREES



(a) Complexity chosen by cross-validation

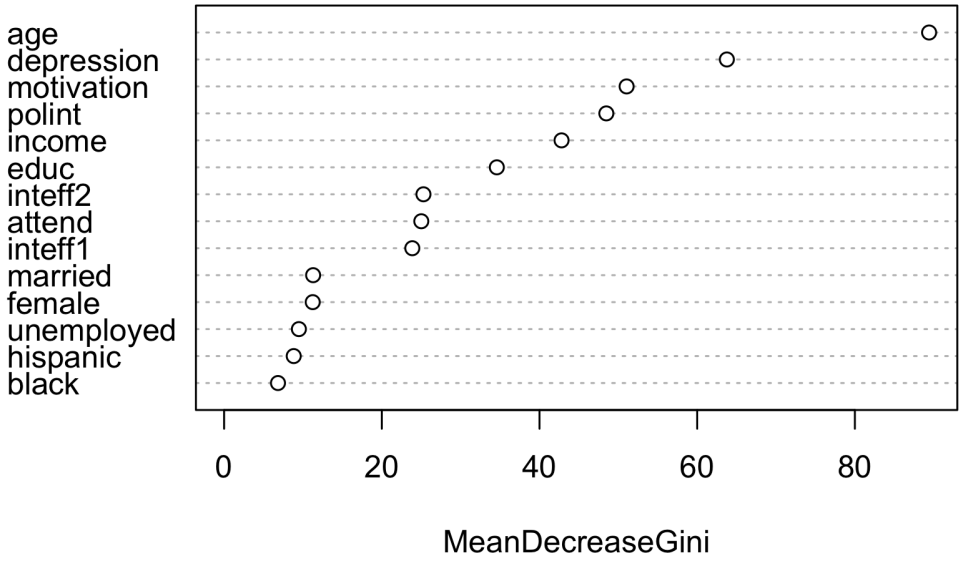


(b) Complexity similar to aggregated-BRS solution

The branches of a decision tree that terminate with a "1" equate to a rule, and the union of all these rules is a rule set. For example, the tree in Figure 2(b) can be read as: "(Age $\geq$ 50) or (Political interest $>$ .83) or (political interest $<$ .17 and age $<$ 22 and internal efficacy (#1) $\geq$ .38)). This tree, which was constructed by picking a complexity that would result in a rule set with a similar number of rules and average rule length as the aggregated-BRS solution, has some similar implications as BRS: high age and high political interest are both considered important for predicting voter turnout. However, we do not advocate using decision trees for learning rule sets because they tend to be less parsimonious, and because for a given level of parsimony, they tend not to perform as well. This first point is made evident by Figure 2(a). When we pick complexity using cross-validation, we end up with a very complex model that is not very interpretable and is possibly overfitted. If we instead pick complexity with the goal of keeping the model interpretable, we might end up with a model like the one in Figure 2(b). This will tend to produce worse rule

sets than BRS, though, because tree algorithms are typically greedy: they make the best split at each given juncture, but this seldom has a chance at finding the global optimum.

FIGURE A3. IMPORTANCE OF VARIABLES ACCORDING TO RANDOM FOREST



To address the issue of overfitting, we might think to turn to Random Forest. This method uses bootstrap aggregation or "bagging" to combine the prediction of many trees fitted to various bootstrapped datasets by using the modal prediction of these trees as its own prediction. While this method has benefits from a predictive performance standpoint, bagging comes at the cost of interpretability. We can no longer interpret the model; we do not know what about a given observation led to its classification outcome. There are some proposed measures of importance, but these do not tell us about the interactions between variables. Figure A3 plots one measure of importance for each of the variables. We could manually interact variables, but this is somewhat redundant (since trees implicitly interact variables) and puts us in the same position as with regression.