Messy Data, Robust Inference?

Navigating Obstacles to Inference with bigKRLS

Abstract

Complex models are of increasing interest to social scientists. Researchers interested in prediction generally favor flexible, robust approaches, while those interested in causation are often interested in modeling nuanced treatment structures and confounding relationships. Unfortunately, estimators of complex models often scale poorly, especially if they seek to maintain interpretability. In this paper, we present an example of such a conundrum and show how optimization can alleviate the worst of these concerns. Specifically, we introduce bigKRLS, which offers a variety of statistical and computational improvements to Hainmueller and Hazlett (2013)'s Kernel-Regularized Least Squares (KRLS) approach. As part of our improvements, we decrease the estimator's single-core runtime by 50% and reduce the estimator's peak memory usage by an order of magnitude. We also improve uncertainty estimates for the model's average marginal effect estimates - which we test both in simulation and in practice - and introduce new visual and statistical tools designed to assist with inference under the model. We further demonstrate the value of our improvements through an analysis of the 2016 presidential election, an analysis which would have been impractical or even infeasible for many users with existing software.

1 Introduction

Statistical performance and interpretability are desirable attributes for any modeling approach, particularly in social science research. As ongoing political commentary reminds us, both the academic and broader communities care about robust, prediction-oriented modeling approaches, with results that can be presented in a useful and interpretable fashion. In some applications, prediction may be a useful goal in and of itself, whether or not the model in question can help illuminate underlying causal mechanisms or estimate causal quantities of interest. However, even in these settings, interpretability is a helpful trait, allowing researchers to check assumptions and guard against overfitting.

Unfortunately - but unsurprisingly - modeling strategies that excel at these criteria often exhibit severe scalability constraints. For a concrete example, consider Hainmueller and Hazlett (2013)'s Kernel Regularized Least Squares approach. KRLS offers a desirable balance of interpretability, flexibility, and theoretical guarantees, primarily through the pointwise marginal derivative estimates produced by the estimation routine and their corresponding averages. However, these pointwise marginal derivatives are costly in time and memory to estimate, whether via KRLS or through related techniques such as BART (Chipman et al. 2010) or LASSOplus (Ratkovic and Tingley 2017). As a result, optimization (both in memory and speed) is important to make KRLS a practical choice in many applied settings.

In this paper, we present a series of improvements designed to improve KRLS's statistical performance, speed, and memory usage, which we implement in the *bigKRLS* package. Compared with the original *KRLS* library,[1] our algorithm decreases runtime by approximately 75%[2] and reduces peak memory usage by approximately an order of magnitude. These improvements allow users to straightforwardly fit models via KRLS to larger datasets (N > 3,500) on personal machines, which was not possible using existing implementations. We also develop an updated significance test for the average marginal effects estimates pro-

[1] In this manuscript, "KRLS" refers to the method of estimation whereas "*KRLS*" refers to the R package described in a companion piece by Ferwerda et al. (2017).

[2] Assuming parallelization is used. Single-core run-time is also approximately 50% faster with *bigKRLS*.

duced by the model - which we justify both theoretically and in simulation - and a new inferential statistic designed to help identify the presence of heterogeneous effects.

After describing our methodological contributions, we illustrate the practical utility of *bigKRLS* through an extended examination of the so-called "communities in crisis" explanation for the 2016 presidential election. The estimates we produce - which are robust both to our significance correction and to a series of cross-validated comparisons - straightforwardly address hypotheses advanced by post-election commentary. Due to sample size constraints, the model we estimate would have been impractical to fit on a personal computer using the original *KRLS* algorithm, but runs smoothly with *bigKRLS*, highlighting the importance of optimization work for applied political science tasks.

2 Data Science as Interpretability vs. Complexity

2.1 Desirable Estimator Properties

When selecting an estimator, there are an array of properties which we might value. For example, we might want our estimator to be unbiased or efficient, or we might want it to minimize some particular loss function (e.g., mean squared error). In theoretical settings, we generally assume that our model of interest captures the "true" data-generating process; however, in applied settings, we are usually - and, often, rightly - skeptical of such assumptions. As a result, we also want estimators to be robust against violations of potentially problematic modeling assumptions (e.g., incorrect functional form or omitted variables). For this reason, we sometimes place a premium on predictive accuracy against held-out test data.

Besides these traits, however, we also favor models whose results are *easily interpreted*. Compared with the traits described above, "interpretability" does not possess a particularly precise definition. However, we can colloquially view a model as more "interpretable" to the extent that its estimates allow researchers to answer useful questions with minimal additional effort. A model like linear regression, for example, for example, directly estimates coefficients that offer information about the marginal effect of some covariates $\mathbf{X}$ on a dependent variable $\mathbf{y}$. By contrast, prediction-oriented models like random forests (Breiman 2001) offer more limited options. If a researcher wishes to learn about the data-generating process using a random forest, her choices are either to inspect relatively uninformative summary statistics such as variable importance or to generate first difference estimates for particular values of interest through perturbations of the input data.

Many attributes of a model can influence its interpretability. For example, models which offer simple, familiar estimates such as average treatment effect estimates alongside more nuanced counterparts can make their contents more accessible, serving readers with different backgrounds and levels of experience. Similarly, modeling strategies which reduce the number of non-zero effect estimates or the complexity of their functional form tend to ease interpretation. Regularization constraints - which are explicitly designed allow researchers to ignore some parameters by shrinking their values to or near zero (Hastie et al. 2015) - offer a direct example of this kind of strategy.[3]

Importantly, we do not mean to suggest that these are the only traits that contribute to model interpretability, or that interpretability (however defined) is the only standard by which a model ought to be judged. Depending on the application, researchers might be willing to employ a less interpretable model in exchange for improved predictive performance or model fit. In general, however, we argue that all of these traits represent important modeling goals, which need to be balanced in context.

[3] Arguably, we might view Bayesian posterior probabilities as a good example of an "interpretable" procedure. In a direct sense, many regularization strategies can be justified as a particular prior structure (Wahba 1983; Tibshirani 1996). More broadly, as Gill (1999), Jackman (2009) and others argue, the frequentist null hypothesis testing paradigm is notoriously difficult to properly interpret. By contrast, researchers can straightforwardly calculate probabilities of interest such as $P(\beta > 0|X)$ under the Bayesian paradigm without reference to counterfactuals.

That said, many researchers find Bayesian priors confusing or arbitrary. To a certain extent, this disagreement is a question of whether one locates the primary interpretive dilemma at the beginning or the end of the analysis. Bayesian versions of kernel regularized regression are relevant to this discussion but beyond the scope of this paper. See, e.g., Zhang et al. (2011) for further discussion.

2.2 The Complexity Frontier

Unfortunately, estimators that excel at both interpretability and prediction in the face of challenging data-generating processes are often highly *complex*. Here, we use "complexity" in the algorithmic sense, referring to the CPU and memory resources needed to estimate a model given the size of the inputs (Papadimitriou 2003). Algorithmic complexity is usually represented using order notation: so, an $O(N)$ algorithm is one whose complexity grows linearly with N, and an $O(log(N))$ algorithm is one whose complexity grows logarithmically with N.[4] For example, linear regression with N observations and P covariates has complexity approximately $O(P^2N)$ (since calculating $\mathbf{X}'\mathbf{X}$ dominates other calculations involved in generating $\hat{\beta}_{OLS}$).[5] Since N is usually much larger than P, estimating an ordinary linear regression via OLS has complexity is approximately linear with respect to N.

Compared with other approaches, under appropriate assumptions linear regression directly calculates causally interpretable effects, but is sensitive to model specification choices and possesses poor predictive performance. On the other end of the spectrum, decision trees do not calculate causally interpretable effect estimates, but are highly flexible, make few assumptions about the data-generating process, and often possess excellent out-of-sample performance. In exchange for these desirable properties, however, decision trees are substantially more complex than ordinary linear regression. In rough terms, a single decision tree has complexity $O(Nlog(N)^2) + O(PNlog(N))$.[6] Generally, decision trees perform better when used in an ensemble approach such as a random forest (Breiman 2001), leading users to generate hundreds or thousands of such trees for any given application.

Models that attempt to optimize all of these traits simultaneously quickly encounter what we might call the *computational complexity frontier*. Flexibility with respect to func-

[4] Since order notation is designed to describe the limiting complexity of a given algorithm as the size of the inputs grows arbitrarily large, constants and lower-order terms are usually omitted from order-notation statements. However, if a high level of precision is necessary to compare a pair of algorithms (as in §4.1 of our paper), constant terms can be included in the complexity statement.

[5] Assuming N is substantially larger than P and a Cholesky decomposition of $\mathbf{X}'\mathbf{X}$ is used to calculate $\hat{\beta} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$ rather than inverting $\mathbf{X}'\mathbf{X}$ directly.

[6] With fairly pessimistic assumptions regarding tree growth rates (Witten et al. (2011), p.199-200).

tional form, sparsity constraints, and related modeling strategies all impose a substantial computational burden, rendering them impractical for particularly large datasets. Importantly, in many applications, interpretability also factors into this tradeoff. Again, regularization strategies (e.g., LASSO) and cross-validated parameter selection approaches offer canonical examples of this relationship.[7]

3 Complexity and Interpretability with KRLS

Kernel regularization techniques have a long history in the computer science and statistics literatures (Rifkin et al. 2003; Yu et al. 2009).[8] Hainmueller and Hazlett (2013) demonstrate that this approach is useful for inference as well as prediction, primarily through the derivative estimators they derive. However, the approach they present also offers a stark example of the tradeoffs between complexity, intepretability, and statistical performance we describe in the previous section. To build intuition, we briefly discuss key features of the KRLS approach in this section.[9]

3.1 An Overview of KRLS

Begin by considering the following model:

$$\mathbf{y}_i = c_1 k(\mathbf{X}_i, \mathbf{X}_1) + c_2 k(\mathbf{X}_i, \mathbf{X}_2) + ... + c_N k(\mathbf{X}_i, \mathbf{X}_N) + \epsilon_i$$

Where $\mathbf{c}$ represents a vector of coefficients and k represents a kernel function which quantifies the pairwise similarity between two vectors of data. Defining $\mathbf{K}$ as a matrix of such similarity

[7] The complexity frontier phenomenon has become increasingly relevant for applied political science work. For example, as Imai et al. (2016) document, workhorse political science ideal point models take days to run on standard datasets (e.g. Congressional roll-call votes), limiting researchers' ability to estimate these models in data-intensive settings. Imai et al. address this issue by proposing an EM estimator, which produces similar results to standard approaches two to three fewer orders of magnitude more quickly.

[8] For an interesting exampling involving facial expression recognition during televised debates, see Eleftheriadis et al. (2015).

[9] See Hainmueller and Hazlett (2013) for additional details.

values, we can express the model for the full sample as:

$$\mathbf{y} = \mathbf{K}\mathbf{c} + \epsilon$$

Viewed from this perspective, this model treats the dependent variable as a linear and additive combination of the pairwise similarity between a given observation and each other observation in the dataset, as calculated using the predictor matrix $\mathbf{X}$. These similarity values are then weighted by a set of so-called "choice coefficients" $\mathbf{c}$, which serve to weight observations based on their influence on the conditional expectation function.

In principle, any similarity kernel function k can be used to estimate the model, but we focus here on the Gaussian kernel:[10]

$$k(\mathbf{X}_i, \mathbf{X}_j) = e^{-||\mathbf{X}_i - \mathbf{X}_j||^2/\sigma^2}$$

Where $||\mathbf{X}_i - \mathbf{X}_j||$ denotes Euclidean distance, and σ^2 denotes a researcher-specified bandwidth parameter. In practice, Hainmueller and Hazlett (2013) recommend setting $\sigma^2 = P$ (the number of predictor variables), which we adopt throughout this paper.

Since this model involves estimating one parameter for each observation, to rule out degenerate solutions we replace $\mathbf{c}$ with $\mathbf{c}^*$, where $\mathbf{c}^*$ is defined using a Tikhonov regularization strategy:

$$\mathbf{c}^* = \underset{c \in \mathbb{R}^P}{\operatorname{argmin}} \Big[(\mathbf{y} - \mathbf{K}\mathbf{c})'(\mathbf{y} - \mathbf{K}\mathbf{c}) + \lambda \mathbf{c}'\mathbf{K}\mathbf{c} \Big]$$

Where λ is a regularization parameter, selected to minimize leave-one-out loss. As we document in §4.3, λ is computationally demanding to select; however, once λ is selected this

[10] See Hainmueller and Hazlett (2013) for a comparison of various kernel functions, and additional theoretical justification for this choice. Broadly, we view the choice of kernel as a preprocessing decision, which researchers can adjust based on their particular problem domain. Other kernels besides the Gaussian kernel are certainly justified in some settings; however, simulation results in Appendix E.2 and in Hainmueller and Hazlett (2013) suggest that this option represents a reasonable default choice.

approach yields a closed-form expression $\hat{\mathbf{c}}^*_{KRLS} = (\mathbf{K} + \lambda I)^{-1}\mathbf{y}$, which can be calculated straightforwardly. Under appropriate functional form and error structure assumptions, both $\hat{\mathbf{c}}^*_{KRLS}$ and $\hat{\mathbf{y}}^*_{KRLS}$ are unbiased and consistent estimators of their population equivalents $\mathbf{c}^*$ and $\mathbf{y}^*$, with closed-form expressions for both the estimators and their variances (Hainmueller and Hazlett (2013)).[11] In simulations, both we and Hainmueller and Hazlett (2013) show that KRLS is competitive with respect to out-of-sample predictive performance compared with related approaches (see Appendix E.2 for details).

3.2 Opening the Black Box

In contrast with other flexible modeling approaches, the kernel and regularized coefficients offer a natural way to express the effects of variables contained in the model. In particular, since $\hat{\mathbf{y}}^*$ has a closed-form expression, for continuous predictors we can estimate the marginal effect of a given predictor at any observed point $\mathbf{X}_{j,p}$ by taking the derivative[12] of the predicted values with respect to the point of interest:

$$\frac{\widehat{\delta \mathbf{y}^*}}{\delta \mathbf{X}_{j,p}} = \frac{-2}{\sigma^2} \sum_i^N \hat{c}_i^* \mathbf{K}_{i,j} (\mathbf{X}_{i,p} - \mathbf{X}_{j,p}).$$

Since the regularization constraints imposed on the estimated coefficient vector $\hat{\mathbf{c}}^*$ serve to shrink its values, many of the pairwise comparisons embedded in this expression have little or no effect on the final estimated derivative value. These shrinkage constraints therefore simplify and smooth the estimated derivatives, consistent with the logic of regularization. Because the distribution of the regularized coefficients is in general unknown, constructing reliable measures of uncertainty around each pointwise marginal effect is challenging. As

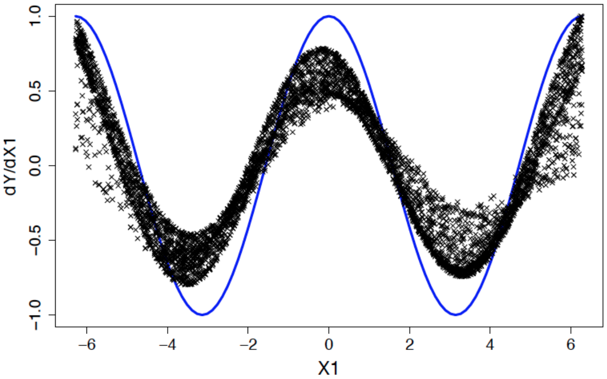
[11] In principle, if $\mathbf{K} = \mathbf{I}$ the KRLS coefficient estimates converge with those obtained via a ridge regression approach in which we regress $\mathbf{y}$ on $\mathbf{K}$ (up to a scaling factor). This relationship can be confirmed in R using glmnet(..., alpha = 0). However, the (multidimensional) Chebychev inequality implies $p(\mathbf{K} = \mathbf{I}) = 0$ at this or similar bandwidths. For observable kernels ($\mathbf{K} \neq \mathbf{I}$), the relationship between the KRLS coefficients and the ridge coefficients is not constant across iterations.

[12] In the discrete variable case, we estimate first differences rather than derivatives, yielding a different estimator; however, the interpretation of this quantity remains similar. For details, see §3.3.

a result, these estimates are best used for exploration rather than inference. However, for researchers who are mindful of these limitations, these estimates offer a useful summary of the observed effect structure.

Because these pointwise marginal derivatives can be expressed in closed form, we can straightforwardly produce summaries of their content. Defining $\hat{\Delta}$ as an $N \times P$ matrix of partial derivatives, we can define the average marginal effect (AME) of each variable as $\hat{\Delta}_{AME} = [\frac{1}{N} \sum_j \widehat{\frac{\delta \mathbf{y}^*}{\delta \mathbf{X}_{p,j}}} \forall p \in \{1, 2, ...P\}]$, or the column means of $\hat{\Delta}$. Hypotheses about $\hat{\Delta}_{AME}$ can be evaluated with standard hypothesis testing tools, and offer a highly interpretable summary of the overall effect of each variable.

Figure 1: "Actual" Marginal Effects



Note: the target function is $y = sin(x_1) + x_2 + N(0, 1)$, and its derivative $\frac{\delta y}{\delta x_1} = cos(x_1)$ is shown in blue. x_1 and x_2 have been drawn uniformly between -2π and 2π, with pointwise marginal effects plotted.

Unfortunately, the flexibility, statistical performance, and interpretability of this approach comes at a cost. This pairwise model is difficult to estimate with KRLS, even without estimating the marginals. In *bigKRLS* - our implementation of KRLS for this model - the algorithm's peak memory requirements are $O(N^2)$. Assuming derivatives are estimated, the original *KRLS* implementation has peak memory requirements of $O(PN^2)$; as a result, this figure offers a substantial improvement over the original estimation routine, but remains difficult to scale.[13] From a runtime standpoint, KRLS's requirements are similarly onerous, with

[13] Even when P is small, *bigKRLS*'s peak memory usage is lower since it is $O(5N^2)$ compared with $O((P + 10)N^2)$ plus an additional $O(11N^2)$ term if any of the predictors are binary for *KRLS*. In addition to changes

total runtime complexity $O(N^3)$. For comparison, decision trees have runtime complexity $O(Nlog(N)^2) + O(PNlog(N))$.

4 Algorithmic and Statistical Optimization using bigKRLS

4.1 Overview

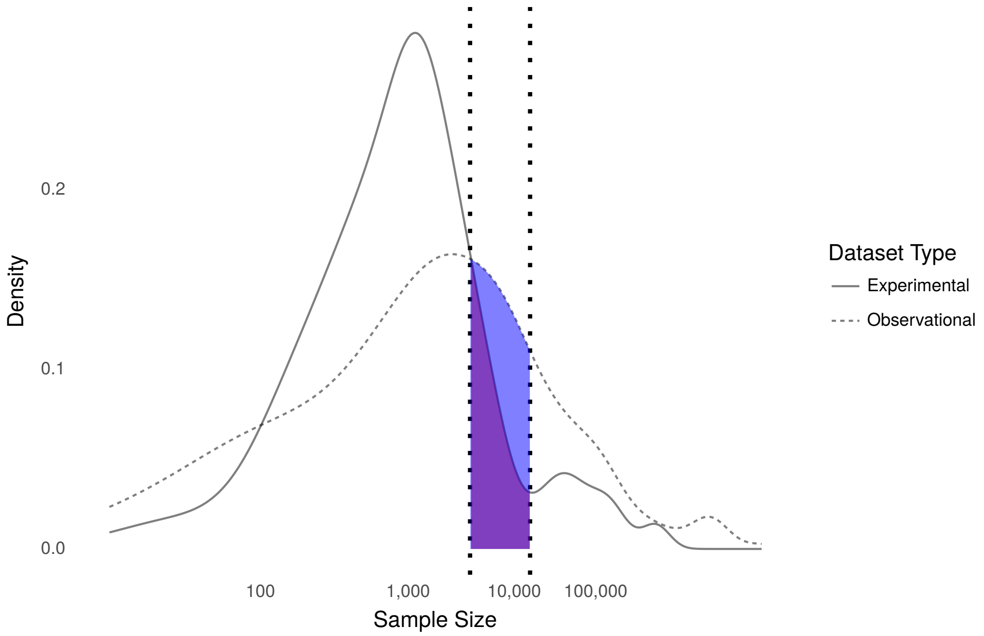
The improvements we present in this paper can be divided into two rough categories. First, from an algorithmic standpoint we re-implement all major functions using the bigmemory, Rcpp, and parallel packages in R, allowing researchers to easily parallelize model estimation. We also develop new algorithms for kernel regularization and first differences. Second, to improve inference, we propose a degrees-of-freedom correction for the model's average marginal effect estimates and a new measure of effect heterogeneity contained in the model. We justify our degrees-of-freedom correction both theoretically and in simulation, and find that it performs well in the settings we examine. The correction improves coverage most notably for complex data generating processes and smaller samples.

Put together, our algorithmic changes reduce peak memory consumption from $O((P + 21)N^2)$ to $O(5N^2)$ in our implementation. Crucially, unlike *KRLS*, the memory footprint of *bigKRLS* does not depend on P, the number of explanatory variables.[14] Runtime using *bigKRLS* and the original *KRLS* implementation is roughly comparable when N and P are small and all predictors are continuous. However, in most applied settings, *bigKRLS* is substantially faster. In simulation results for a dataset consisting of 10 binary and 10 continuous predictors, for example, we report approximately 50% decreased wall-clock time when running on a single core. When *bigKRLS* is set to use multiple processors (not an option with *KRLS*), a task that takes *KRLS* just over two hours can be done by *bigKRLS*

discussed in (§3.2), our algorithm also differs from the original implementation in that it constructs the simple distance matrices "just in time" for estimation and removes large matrices the moment they are no longer needed.

[14] If derivatives are not estimated, the two algorithms have a fairly similar memory footprint outside of the λ selection routine (see §4.2.2 for details). However, as we argue in §3.2, in most social science applications the model's derivative estimates are its most attractive quality.

Figure 2: Published sample sizes



Notes: Log-scaled sample sizes for datasets used in AJPS and APSR articles published January 2015-January 2017, separated by observational and experimental datasets. Shaded areas represent sample sizes that *bigKRLS* can estimate on a personal computer that *KRLS* cannot.

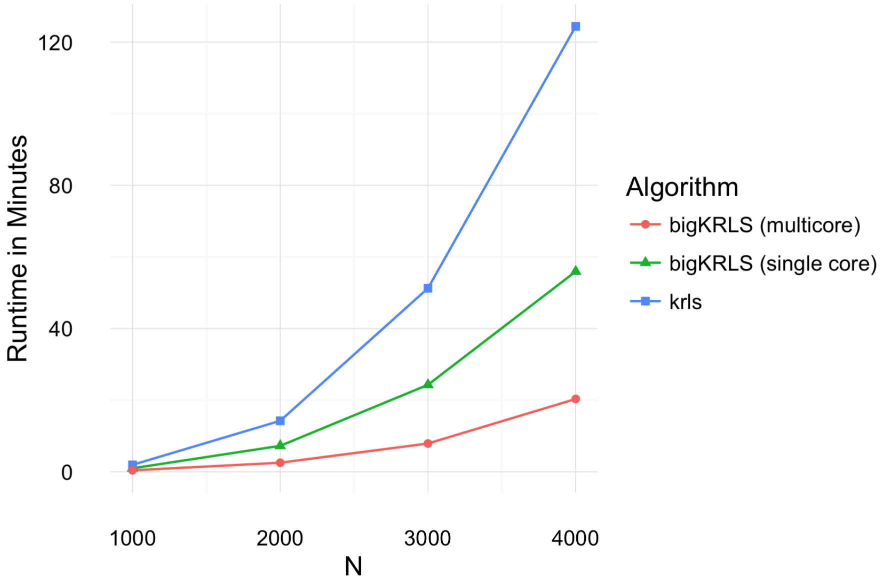
in twenty minutes (Figure 3).

How important are these improvements? For illustration purposes, we surveyed all empirical articles published in the *American Journal of Political Science* and the *American Political Science Review* from January 2015 to January 2017, and recorded sample sizes for each dataset used in those articles ($N = 279$). In the timeframe we surveyed, approximately 43% of datasets were too large for the original *KRLS* implementation. By contrast, with similar dimensions *bigKRLS* can handle datasets up to approximately $N = 14,000$ on a personal machine before reaching the 8 GB cutoff, opening an additional 20% of published datasets for estimation.[15]

These improvements, we argue, are substantial. Statistical models are only useful to the

[15] Real numbers require 8 bytes of storage each in most scientific computing languages including C, C++, and R. Assuming 8GB available memory, at least one binary predictor, and $P = 67$ (as in the 2016 election model in §5), the original *KRLS* implementation will have insufficient memory if $N > 3,500$. If only 4GB are available (R's default for many Windows laptops), the cutoffs for *KRLS* and *bigKRLS* would be $N = 2,500$ and $N = 10,000$, respectively, similarly suggesting *bigKRLS* can estimate 20% more publishable datasets.

Figure 3: Runtime comparisons



Notes: Runtime results for *KRLS*, *bigKRLS*, and parallelized *bigKRLS*. All models were estimated with 10 continuous and 10 binary predictors. Eigentruncation was not used. Two computers were used, a laptop (2012 MacBook Pro with 8 gigabytes of RAM) and a server (Xeon E5-2650 with 126 gigabytes of RAM). For the "*bigKRLS* (multicore)" test, 14 of the server's cores were used.

extent that they can be employed in practice. While high-complexity methods like KRLS are unlikely to be usable in truly "big" data settings, our improvements allow a noticeably greater proportion of applied researchers to use the modeling approach we present.

4.2 Algorithmic Improvements

4.2.1 A Leaner First Differences Estimator

For binary explanatory variables, KRLS estimates first differences.[16] The original algorithm for this procedure functions as follows. Suppose $\mathbf{X}_b$ is a column that contains a binary variable. Construct two copies of $\mathbf{X}$, denoted as $\mathbf{X}_{\{\mathbf{0}\}}$ and $\mathbf{X}_{\{\mathbf{1}\}}$, which are modified such that all observations in the b^{th} column of the copies are equal to 0 and 1, respectively. Combine the original matrix and the two (modified copies) into a new matrix $\mathbf{X}'_{new} = [\mathbf{X} \,|\, \mathbf{X}_{\{\mathbf{0}\}} \,|\, \mathbf{X}_{\{\mathbf{1}\}}]'$, and construct a new similarity kernel. This step is temporary, but has a memory footprint

[16] A nearly identical procedure is used for out-of-sample prediction.

of $9N^2$ (!). Finally, save the two submatrices of the kernel corresponding to the respective counterfactual comparisons between $\mathbf{X}_{\{0\}}$, $\mathbf{X}_{\{1\}}$, and the observed data $\mathbf{X}$.

Our leaner implementation can also be expressed in terms of potential outcomes (Keele 2015). The goal is to minimize the computational burden of obtaining the vector of differences for the scenario in which everyone was counterfactually assigned to one group vs. the other. Let $\mathbf{K}_{\{1\}}$ and $\mathbf{K}_{\{0\}}$ be the counterfactual kernels.[17] The first differences are:

$$\delta_{\mathbf{b}} = \mathbf{y}_{\{1\}} - \mathbf{y}_{\{0\}} = \mathbf{K}_{\{1\}}\mathbf{c}^* - \mathbf{K}_{\{0\}}\mathbf{c}^* = (\mathbf{K}_{\{1\}} - \mathbf{K}_{\{0\}})\mathbf{c}^*.$$

As with the AMEs of continuous variables, the mean $\bar{\hat{\delta}}_{\mathbf{b}}$ is used as the point estimate that appears in the regression table. The variance of that mean first difference is:

$$\hat{\sigma}^{\mathbf{2}}_{\delta_{\mathbf{b}}} = \mathbf{h}'(\mathbf{K_{new}}\hat{\mathbf{\Sigma}}_{\mathbf{c}})\mathbf{K}'_{\mathbf{new}}\mathbf{h}$$

where $\mathbf{h}$ is a vector of constants,[18] $\mathbf{K}_{new}$ is a partitioned matrix with the counterfactual kernels, and $\hat{\mathbf{\Sigma}}_{\mathbf{c}}$ is the variance co-variance matrix of the coefficients (Hainmueller and Hazlett 2013). Though highly interpretable, first difference calculations are computationally daunting because the peak memory footprint is $O(6N^2)$: $O(2N^2)$ for $\mathbf{K}_{new}$ and another $O(4N^2)$ for $\hat{\sigma}^{\mathbf{2}}_{\delta_{\mathbf{b}}}$. The following insight allowed us to derive a more computationally-friendly algorithm:

Consider the similarity score $\mathbf{K}_{i,j}$. We can manipulate this quantity as follows:

$$\begin{aligned}
\mathbf{K_{i,j}} &= e^{-||\mathbf{x_i}-\mathbf{x_j}||^{\mathbf{2}}/\sigma^{\mathbf{2}}} \\
&= e^{-[(\mathbf{x_{i,1}}-\mathbf{x_{j,1}})^{\mathbf{2}}+(\mathbf{x_{i,2}}-\mathbf{x_{j,2}})^{\mathbf{2}}+\ldots+(\mathbf{x_{i,b}}-\mathbf{x_{j,b}})^{\mathbf{2}}+\ldots]} \\
&= e^{-(\mathbf{x_{i,b}}-\mathbf{x_{j,b}})^{\mathbf{2}}/\sigma^{\mathbf{2}}} e^{-[(\mathbf{x_{i,1}}-\mathbf{x_{j,1}})^{\mathbf{2}}+(\mathbf{x_{i,2}}-\mathbf{x_{j,2}})^{\mathbf{2}}+\ldots]} \\
&= e^{-(\mathbf{x_{i,b}}-\mathbf{x_{j,b}})^{\mathbf{2}}/\sigma^{\mathbf{2}}} \mathbf{K}^*_{\mathbf{i,j}}
\end{aligned}$$

These manipulations allow us to re-express the quantity of interest in terms of $\mathbf{K}^*_{i,j}$, the observed similarity on dimensions other than b, and $\phi = exp(-\frac{1}{\sigma^2_{\mathbf{X}_b}\sigma^2})$, the (only non-zero) pairwise distance on the binary dimension where $\sigma^2_{\mathbf{X}_b}$ is the variance of the binary variable.

[17] How closely the first differences resemble an experiment depends on the entropy of $\mathbf{K}_{\{1\}}$ and $\mathbf{K}_{\{0\}}$ (Hazlett 2016).

[18] The first N entries of are $\frac{1}{N}$ and the next N are $-\frac{1}{N}$.

Figure 4: Re-expressed kernel for first differences estimation

$\mathbf{X}_{i,b}$	$\mathbf{X}_{j,b}$	$\mathbf{K}_{i,j}$	$\mathbf{K}_{\{1\},b}$	$\mathbf{K}_{\{0\},j}$	$\mathbf{K}_{\{1\},j} - \mathbf{K}_{\{0\},j}$
1	1	$\mathbf{K}^*_{i,j}$	$\mathbf{K}^*_{i,j}$	$\phi\mathbf{K}^*_{i,j}$	$(1-\phi) * \mathbf{K}_{i,j}$
1	0	$\phi\mathbf{K}^*_{i,j}$	$\phi\mathbf{K}^*_{i,j}$	$\mathbf{K}^*_{i,j}$	$\frac{(\phi-1)}{\phi} * \mathbf{K}_{i,j}$
0	1	$\phi\mathbf{K}^*_{i,j}$	$\mathbf{K}^*_{i,j}$	$\phi\mathbf{K}^*_{i,j}$	$\frac{(1-\phi)}{\phi} * \mathbf{K}_{i,j}$
0	0	$\mathbf{K}^*_{i,j}$	$\phi\mathbf{K}^*_{i,j}$	$\mathbf{K}^*_{i,j}$	$(\phi-1) * \mathbf{K}_{i,j}$

As part of the estimation of first differences, observation i is counterfactually manipulated and compared to each observation $j = 1, 2, ..., N$. The first difference for observation i ($\hat{\delta}_{\mathbf{b},\mathbf{i}}$) is a coefficient-weighted average of the final column.

This process facilitates re-expression wholly in terms of the observed kernel and the constant ϕ, as shown in Figure 4. As a result, our algorithm avoids constructing the costly temporary matrix required in the original implementation

Building on this observation, we took the following steps to make the variance covariance calculation more tractable.

1. Though $(\mathbf{K}_{new}\hat{\mathbf{\Sigma}}_{\mathbf{c}})\mathbf{K}'_{\mathbf{new}}$ is $2N \times 2N$ it is possible to focus the calculations on four $N \times N$ submatrices:

$$(\mathbf{K}_{new}\hat{\mathbf{\Sigma}}_{\mathbf{c}})\mathbf{K}'_{\mathbf{new}} = \begin{bmatrix}\mathbf{K}_{\{1\}}\mathbf{K}_{\{0\}}\end{bmatrix} \hat{\mathbf{\Sigma}}_{\mathbf{c}} \begin{bmatrix}\mathbf{K}'_{\{1\}} \\ \mathbf{K}'_{\{0\}}\end{bmatrix} = \begin{bmatrix}\mathbf{K}_{\{1\}}\hat{\mathbf{\Sigma}}_{\mathbf{c}}\mathbf{K}'_{\mathbf{\{1\}}} & \mathbf{K}_{\{1\}}\hat{\mathbf{\Sigma}}_{\mathbf{c}}\mathbf{K}'_{\mathbf{\{1\}}} \\ \mathbf{K}_{\{1\}}\hat{\mathbf{\Sigma}}_{\mathbf{c}}\mathbf{K}'_{\mathbf{\{0\}}} & \mathbf{K}_{\{0\}}\hat{\mathbf{\Sigma}}_{\mathbf{c}}\mathbf{K}'_{\mathbf{\{0\}}}\end{bmatrix}$$

   Each (sub)matrix in the final term functions as a weight on the observed variances and covariances in the various counterfactual scenarios. Partitioning the matrix in this fashion allows us to avoid constructing the full $2N \times 2N$ matrix directly.

2. Though $\mathbf{h}$ is simply an auxiliary vector that facilitates averaging, $\mathbf{h}'(\mathbf{K}_{new}\hat{\mathbf{\Sigma}}_{\mathbf{c}})\mathbf{K}'_{\mathbf{new}}\mathbf{h}$ presents different opportunities for factoring than $(\mathbf{K}_{new}\hat{\mathbf{\Sigma}}_{\mathbf{c}})\mathbf{K}'_{\mathbf{new}}$. Our algorithm factors out individual elements of $\hat{\mathbf{\Sigma}}_{\mathbf{c}}$ as far as possible. Along with an expanded version of Figure 4 that expresses all possible products of two counterfactual similarity scores,

we reduce the computational complexity by an order of magnitude by avoiding an intractable inner loop.

Other factorizations may exist that further optimize either speed or memory – but not both. Our first differences algorithm, for example, can be re-expressed as a triple-loop with no additional memory overhead; however, that formulation sacrifices vectorization speedups which our current setup exploits. In the implementation we present, we create two $N \times N$ temporary matrices, which is both an improvement over six and no worse than any other part of the algorithm. Consistent with our experience with *bigKRLS*, speed tests show that our algorithm is no slower than a purely linear algebra approach.

To illustrate why this advance is important, consider dyadic data. Because of the pairwise structure of the kernel, KRLS is tailor-made for international relations, which often encounters data in country-dyads. However, such analyses often require at least 150 binary variables for nation states. In §5, we use 50 binary variables for US states, which is similarly prohibitive on many machines with *KRLS*. With *bigKRLS*, this is no longer an issue.

4.2.2 Lowering the Cost of Kernel Regularization

For KRLS, the regularization parameter λ is represents the degree of skepticism towards idiosyncratic features of the data-generating process (see Appendix A). In *bigKRLS* we introduce a leaner version of the Golden Search algorithm as described by Rifkin and Lippert (2007) and adopted by Hainmueller and Hazlett (2013). *bigKRLS*'s approach to selecting λ offers a lower memory footprint and speed gains at the sample sizes where *bigKRLS* has the most to offer to political science researchers.

The Golden Search strategy is as follows. Though λ cannot be obtained analytically, it can be selected (to arbitrary precision) through an iterative procedure that depends primarily on the eigendecomposition of the kernel.[19] λ is selected to minimize the sum of squared leave-

[19] Mercer's Theorem enables regularization as the kernel's eigendecomposition takes a known form even in high dimensional space, ultimately enabling λ to be found in a finite, unidimensional space (Beck and Ben-Tal 2006; Hastie et al. 2008; El Karoui 2010).

one-out errors, $LOOE = \sum \left(\frac{\hat{\mathbf{c}}^*}{\mathbf{G}^{-1}_{i,i}}\right)^2$. $\mathbf{G} = \mathbf{K} + \lambda\mathbf{I}$ is the 'ridge' version of the kernel. The key computational challenge is $\mathbf{G}^{-1}$, which is used to calculate candidate coefficient values and $LOOE$ at each step.

For computational ease, $\mathbf{G}^{-1}$ (or even $\mathbf{G}$) is not obtained directly. Instead, the equivalent expression $\mathbf{Q\Lambda Q'}$ is substituted, where $\mathbf{Q}$ is a matrix containing the eigenvectors of $\mathbf{K}$ and $\mathbf{\Lambda}$ is a diagonal matrix of eigenvalues. Expanding this expression yields:

$$\mathbf{G}^{-1}_{i,j} = \sum_{k=1}^{N_Q} \frac{\mathbf{Q}_{i,k} * \mathbf{Q}_{j,k}}{\lambda + \mathbf{\Lambda}_{k,k}},$$

Where N_Q is the number of eigenvalues actually used in this computation.

Existing implementations of this approach usually begin by calculating the cross product $\mathbf{Q}(\mathbf{\Lambda} + \lambda\mathbf{I})\mathbf{Q}'$, and using its values to calculate candidate coefficient values. In our implementation, we instead build this matrix in column-by-column fashion. This strategy has two advantages. First, since the only purpose of constructing $\mathbf{G}^{-1}$ is to calculate $\hat{\mathbf{c}}^*$, we can avoid placing the full $\mathbf{G}^{-1}$ matrix in memory by simply accumulating each column of $\mathbf{G}^{-1}$ into the coefficient vector. Second, since $\mathbf{G}^{-1}$ is symmetric we can simply skip calculating the elements of each column that correspond to the upper triangle of the matrix. As shown in Figure 5, this implementation offers a noticeable performance boost over existing implementations.[20]

As implied by the preceding discussion, not all eigenvalues and vectors are necessary for selecting λ. In most applied settings, the vast majority of the eigenvalues are very small, and can be safely ignored during the calculations described above. *bigKRLS* facilitates two types of eigentruncation: specifying that (1) only N_Q eigenvectors and eigenvalues be calculated[21] and/or (2) defining an ϵ such that only those eigenvectors and eigenvalues will be used where the eigenvalue is at least $100\epsilon\%$ as large as the largest eigenvalue. Optimizing either of these

[20] See Appendix B for details. In practice, convergence for the applied problems we study in this paper takes 5-20 iterations. Without eigentruncation, at N = 5,000 each iteration takes 3.8 seconds with the new algorithm vs. 16.1 seconds. At N = 10,000, each iteration takes ≈ 8 minutes vs. ≈ 13 minutes.

[21] This option is also available in the original *KRLS* implementation.

Figure 5: Alternatives for the Spectral Decomposition's Cumulative Effect on Runtime

R Package	Full Decomposition	Eigentruncation	Estimating Fewer
KRLS	422.935	129.074	
bigKLRS	348.948	102.271	51.082

As an empirical benchmark, we isolate the portions of the algorithm that depend on the eigenvectors and eigenvalues (i.e., the eigendecomposition, λ search, and estimating the coefficients and their variance covariance matrix but not the kernel or marginal effects). We report the runtime of this portion on the 2016 election data presented in §5 ($N = 3106, P = 68$) in seconds on a 2017 Macbook Pro (2.3 GHz Intel Core i5 with 16 GB of RAM). For the eigentruncation case, $\epsilon = 0.01$ and, in the final column, $N_Q = 50$.

strategies *ex ante* is difficult; however, since runtime for this section of the algorithm depends on N_Q rather than the proportion of variance explained by the retained eigenvalues/vectors, even a conservative decision rule will often produce substantial speed gains. As a default in *bigKRLS* we set $\epsilon = 0.001$, which produces virtually identical results to those generated using the full decomposition in both simulated examples and our application in §5.

4.3 Inferential Tools

4.3.1 Degrees of Freedom for Average Marginal Effects

One of KRLS's key strengths is its ability to offer both nuanced effect estimates as well as high-level summaries of each effect. The key high-level quantities produced by the model are the average marginal effect (AME) estimates, for which Hainmueller and Hazlett (2013) derive closed-form expressions for both the values of the AMEs and their variances. They then use these quantities to derive a Student's t-test with $N - P$ degrees of freedom to assess statistical significance of each individual AME estimate. This approach works well for simple data-generating processes but not for more complex problems, as we show in simulation (introduced below and detailed in Appendix D). For many realistic cases this test yields overly narrow confidence intervals coupled with misleadingly low p-values.

To address this issue, we propose an uncertainty correction using the effective degrees of freedom from the Tikhonov penalty used by KRLS. Since KRLS estimates N choice coefficients, $\hat{c}^*$, each of which is a parameter with an L_2 penalty, the effective degrees of freedom for the model is:

$$N_{effective} = N - \sum_{k=1}^{N_Q} \frac{\mathbf{\Lambda_{k,k}}}{\mathbf{\Lambda_{k,k}} + \lambda},$$

where N is the original sample size, $\Lambda_{k,k}$ is the k^{th} eigenvalue of K and λ is the model's regularization parameter (Hastie et al. 2015, 61-68).[22] Provided N_Q (the number of eigenvalues and eigenvectors actually used) is large enough to select λ reliably, $N_{Effective}$ should yield essentially the same estimate as if $N_Q = N$ due the eigenvalues' skew and constraints.[23]

To test the performance of this approach, we conducted a series of simulation studies (see Appendix D for details). Our experiments suggest that this correction is most impactful when the true data-generating process is highly non-linear and the sample size is smaller, offering approximately a 10-point increase in empirical coverage in this scenario. When the true data-generating process is simpler, the difference in coverage rates between corrected and uncorrected approaches vanishes. However, since KRLS is most appealing for applied work when researchers suspect that the true effect structure is more complex, we argue that this result offers substantial justification for our correction.

4.3.2 Detecting Effect Heterogeneity

While useful, inspecting average marginal effect (AME) estimates alone can conceal substantial effect heterogeneity. To help bridge the gap between high-level AMEs and the more nuanced pointwise derivative estimates, we introduce two statistics, which we term the

[22] This quantity is often expressed in terms of squared singular values. However, since $\mathbf{K}$ is positive semi-definite the squared singular values and eigenvalues are equivalent in this case. We express $N_{Effective}$ as a function of the eigenvalues to remain consistent with *bigKRLS*' software architecture (which uses a decomposition of the symmetric kernel rather than SVD) and related work (see, e.g., Rifkin and Lippert 2007). In their paper, Hainmueller and Hazlett (2013) note this relationship, but do not use it as part of their degrees-of-freedom calculations.

[23] Unlike $\mathbf{XX'}$ and $\mathbf{X'X}$, $\mathbf{K}$ does not have exactly P non-zero eigenvalues (but $\mathbf{K}$'s eigenvalues are real, positive values that sum to N). For detail on the kernel's spectrum, see El Karoui (2010). For the application to this algorithm, see §4.2.2.

R^2_K and R^2_{AME}:

$$R^2_{AME} = cor(y, \hat{y}_{AME})^2,$$
$$R^2_K = cor(y, \hat{y})^2.$$

With $\hat{y} = K\hat{c}$ and $\hat{y}_{AME} = \mathbf{X}\hat{\Delta}'_{AME}$, where $\hat{\Delta}_{AME}$ denotes the vector of average marginal effects. Phrased differently, R^2_K denotes the pseudo-R^2 calculated with the kernel and all N coefficient estimates, while R^2_{AME} denotes the pseudo-R^2 calculated using predictions from $\mathbf{X}$ and its estimated partial derivatives alone. Intuitively, these quantities will be similar when $\mathbf{y}$ can be well-approximated by a linear combination of the columns of $\mathbf{X}$. When $\mathbf{y}$ is a more idiosyncratic function better modeled by the pairwise similarity of the observations, R^2_K will outperform the average marginal effects, often dramatically. Note that, since R^2_{AME} is not based on values selected to optimize any particular loss function, its performance can be unstable. Unlike R^2_K, which tends to be relatively consistent in and out of sample, R^2_{AME} is often "pessimistic" in the sense that it can be noticeably smaller on training than test data.

5 Application: The Trump Effect in "Communities in Crisis"

As an example application of *bigKRLS*, we analyze county-level results from the 2016 presidential election, with a focus on the so-called "communities in crisis" hypothesis (described in detail in the following section). This application highlights two key strengths of *bigKRLS*: scalability, and ability to gracefully handle binary predictors. Because we include state as a predictor, our resulting model contains more than 50 binary variables. Peak memory requirements in the original *KRLS* implementation scale with the number of predictors while with *bigKRLS* they do not, resulting in more than an order of magnitude decrease in memory consumption with the move to our implementation.

5.1 Overview

In both popular and academic discussions (e.g. Guo 2016; Siegel 2016; Monnat 2016), a number of commentators argued that Donald Trump's success in the 2016 election was partly attributable to his appeal in "communities in crisis". As shown by Case and Deaton (2015), suicides, drug overdoses, and other so-called "deaths of despair" rose sharply among non-Hispanic whites over the decades preceding the election, leading to a decrease in overall life expectancy within this population. Combined with declining economic opportunities, commentators argued, declining public health outcomes fostered a sense of dissatisfaction with traditional elites in afflicted areas. As a result, members of these communities may have been unusually inclined to vote for Trump relative to "establishment" Republican candidates.

To investigate this hypothesis, we use *bigKRLS* to model county-level voting patterns in the 2016 presidential election. Our dependent variable in this model is ΔGOP, defined as the difference between two-party vote shares for Donald Trump in 2016 and Mitt Romney in 2012 ($\%Trump - \%Romney$). We focus on county-level data for data availability reasons. Because of privacy considerations, county-level data is the most granular unit publicly available in relevant official U.S. data sources like the Census Bureau and the Center for Disease Control.

Our key independent variables are county-level age-adjusted all-purpose mortality rate (per 1,000 individuals) and difference in three-year mortality rates for the periods preceding the 2016 and 2012 elections. These variables are intended to capture the "communities in crisis" hypothesis, with a particular focus on highlighting communities in which public health crises emerged between election cycles. We also include standard racial, macroeconomic, and education variables, along with geolocation information for each county and state-level dummy variables (described in Appendix D). As we note at the outset of this section, including state-level dummies would have been impractical without the improvements we introduce in this paper, highlighting the utility of our approach.[24]

5.2 Average Effect Estimates

Average marginal effect (AME) estimates for this model are given in Figure 6. Unsurprisingly, the model fits the data. For our application, $R_K^2 = 0.83$, suggesting a reasonable level of in-sample fit.[25] Nearly all predictors easily reach conventional levels of statistical significance, with intuitive signs. On average, Trump performed better in whiter, older, poorer, and lower-education localities. As hypothesized, Trump also received a larger two-party vote share than Romney in higher-mortality counties, though the effect size is not particularly large. Averaged across the country, a one-standard deviation (≈ 1.47) increase in age-adjusted mortality is estimated to produce approximately a 0.25% increase in ΔGOP. These findings match the basic contours of the "communities in crisis" hypothesis: relative to previous Republican candidates, Trump performed particularly well in localities facing substantial hardships. ΔMortality is the main exception to this finding pattern, and does not reach conventional levels of statistical significance. Likely, this result is due to a lack of variability; since our study only covers a six-year period, large changes in mortality rates are rare.

As described in §4.3, uncorrected p-values for KRLS average marginal effects are suspect for more complex data-generating processes. In Figure 6, we give both the corrected and the uncorrected p-values for this model, calculated using the effective degrees of freedom correction given previously ($N_{effective} = 2,825$). Since the sample size and effects detected by this analysis are both reasonably large, implementing the degrees of freedom correction we propose does not change any conclusions regarding statistical significance. At least in this case, our sample size appears to be sufficiently large relative to the complexity of the data-generating process to limit the impact of our correction.

[24] Since the complexity of the original *R* implementations depended on both the number of predictor variables and the presence of binary variables, at $N > 3,000$ the original *KRLS* implementation is impractical to estimate with the predictor variables we include.

[25] See Appendix E.2 for model fit comparison between KRLS and other approaches.

Figure 6: Average marginal effect estimates

	Estimate	SE	t	p_{uc}	p_c
Mortality	0.176	0.035	4.983	< 0.001	< 0.001
Δ Mortality	-0.021	0.056	-0.379	0.705	0.725
Urban-Rural Continuum	0.052	0.016	3.336	< 0.001	0.002
Age	0.318	0.089	3.573	0.001	0.001
Median Household Income	-0.242	0.041	-5.849	< 0.001	< 0.001
Unemployment	0.227	0.027	8.322	< 0.001	< 0.001
Poverty	0.123	0.044	2.766	0.006	0.010
No High School Diploma	0.030	0.007	4.457	< 0.001	< 0.001
High School Graduate	0.140	0.006	24.792	< 0.001	< 0.001
Some College	0.112	0.008	13.434	< 0.000	< 0.001
College Graduate	-0.139	0.004	-35.830	< 0.001	< 0.001
White	0.022	0.002	9.574	< 0.001	< 0.001
Latino	-0.019	0.004	-5.131	< 0.001	< 0.001
Black	-0.032	0.003	-10.623	< 0.001	< 0.001
Asian	-0.165	0.017	-9.643	< 0.001	< 0.001

Estimates for latitude, longitude, and state omitted for brevity. The dependent variable is change in GOP vote share in the presidential election, 2012-2016, measured in percentage points. p_{uc} denotes uncorrected p-values generating using a t-test with $N - P$ degrees of freedom; p_c denotes corrected p-values with $N_{effective} = 2,892$, as described in Section 3.3. $N = 3,106$, $R^2_K = 0.83$, $R^2_{AME} = 0.31$.

5.3 Spatial First Differences

While useful, inspecting the average marginal effect estimates can conceal substantial effect heterogeneity. In this case, the R^2_{AME} is only 0.31 but $R^2_K = 0.83$ on the full sample. As we discuss in Appendix E.1, the out-of-sample differences between these values is likely smaller than their in-sample difference. However, as a binary indicator of effect heterogeneity, the gap between these quantities is suggestive.

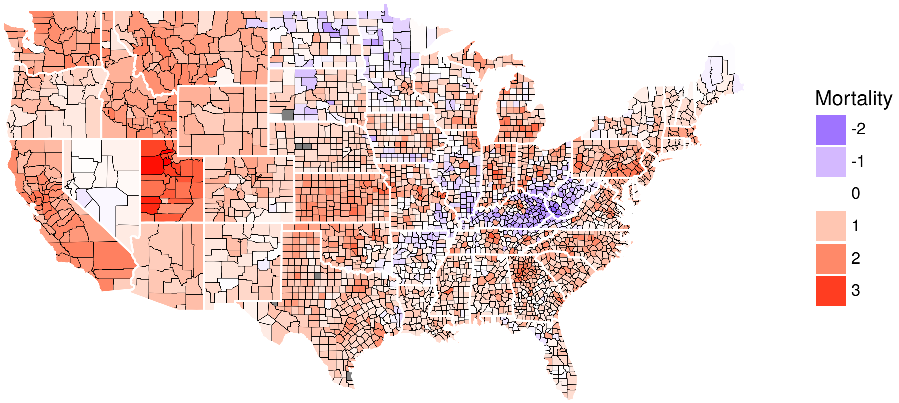
Since KRLS offers closed-form estimates for the variance of the predicted values produced by the model, a simple way to explore effect heterogeneity is to estimate perturbation-based first differences, in which we "perturb" the variable of interest and examine the perturbation's predicted impact (and the variance of that impact) on the dependent variable. In this section, we take precisely this approach. For each county, we calculate $\hat{\mathbf{y}}_{test} - \hat{\mathbf{y}}$, where $\hat{\mathbf{y}}_{test}$ represents the predicted value for the counterfactual scenario in which we perturb the mortality variable - our key variable of interest - by some fixed constant τ. We operationalize τ as the difference between the 95^{th} and 5^{th} percentile of the mortality variable (≈ 3.2 standard deviations).

Besides its interpretive benefits, this approach also offers an important theoretical payoff. Both $\hat{\mathbf{y}}_{test}$ and $\hat{\mathbf{y}}$ are distributed multivariate normal with known variance-covariance matrices (Hainmueller and Hazlett 2013). As a result, the marginal distribution of each predicted difference is distributed univariate Gaussian, with variance equal to the sum of the corresponding diagonal elements of the variance-covariance matrices for each set of predicted values. In Appendix F.1 we use these facts to derive a pointwise hypothesis test which distinguishes whether these first differences differ significantly from zero.

Figure 7 gives the results of this procedure. Nationwide, after applying a Benjamini-Hochberg correction approximately 23% of estimates are distinguishable from zero, with most significant county-level estimates clustered in the West, Mountain West, and Midwest (see Appendix F for details). Notably, though the counterfactual scenario we simulated involves a large change in mortality rate, many individual estimates are still insignificant, echoing the small average marginal effect size we note in the previous section. However, even with this small average effect size, the magnitude of our county-level estimates is remains highly variable. Our estimates are largest in the West and Mountain West, but we also estimate noticeable (and significant) effects in key swing states like Pennsylvania, Ohio, and Michigan. In these latter states, the mortality increase we model produces a predicted change of $\sim 1-2$ percentage points in ΔGOP, which is substantial given the small margins by which the presidential election was decided in these states. These results are consistent with Monnat (2016)'s findings, which suggest Trump's overperformance in high-mortality counties was regionally contingent.

Strikingly, in some regions of the country, the predicted relationship between mortality and ΔGOP Presidential vote share was actually *negative*. This effect is particularly pronounced (and statistically significant) in Kentucky and West Virginia, but is also noticeable in some neighboring Ohio and Illinois counties. One policy-driven explanation for this find-

Figure 7: Mortality first differences



Predicted effect of a ~ 3.2 standard deviation increase in all-purpose mortality (by county) on Δ GOP presidential vote share, 2012-16.

ing relates to state-level Medicaid expansion decisions following the passage of the Affordable Care Act. Under the "communities in crisis" hypothesis, the primary causal mechanism is a local dissatisfaction with political elites, and particularly with elite responses to poverty and poverty-related public health crises. In states like Kentucky and West Virginia that chose to expand Medicaid following the passage of the Affordable Care Act, high-mortality counties likely received a substantial portion of new Medicaid spending, which may have buttressed their faith in "establishment" politicians.

Though not conclusive evidence, we argue that these results are at least suggestive. While model we estimate clearly cannot distinguish the Medicaid expansion's effect on voter preference from unmeasured local predispositions, our results suggest that policy context matters. Far-reaching policy programs like the Affordable Care Act's Medicaid expansion provisions might plausibly condition voter receptiveness to the anti-elite message offered by the Trump campaign. Thus, KRLS's flexibility points to mechanisms worthy of future inquiry.

5.4 Interpreting Pointwise Marginal Effects

In addition to geographic heterogeneity, the “communities in crisis” hypothesis implies that mortality’s effect should be conditioned by two other factors. First, in line with most post-election commentary, Trump’s appeal should be strongest in *white* “communities in crisis”. In other words, we should expect the effect of increasing mortality rates to be strongest in communities with larger white populations. Figure 8 weakly supports this hypothesis; however, plotting pointwise marginal effects in this fashion suggests that the size of this relationship is small at best, with both majority-minority and majority-white counties displaying roughly similar effect sizes.

Second, we should also expect the marginal effect of mortality to be increasing. Marginal increases in mortality, in other words, should have a relatively small effect in low-mortality counties, and a much larger one in high-mortality locations (as mortality rates approach “crisis” status). However, as shown in Figure 9, the estimated marginal effect of mortality actually peaks in mid-mortality counties and declines as mortality increases. Based on these results, true “crisis” communities (those with the highest mortality rates) appear to have been less responsive to mortality differences of than their moderate-mortality counterparts, suggesting that the mortality effect is largely concentrated within the latter group of localities.

We hasten to emphasize that the discussion of pointwise marginal effects in this section necessarily possesses an exploratory quality. Unlike the first difference estimates we present in the previous section, pointwise hypothesis tests for these quantities are difficult to construct. Developing appropriate pointwise uncertainty estimators represents a fruitful direction for future research.

Figure 8: Marginal effect of age-adjusted mortality on Δ GOP presidential vote share, 2012-16, by proportion of white population in each county

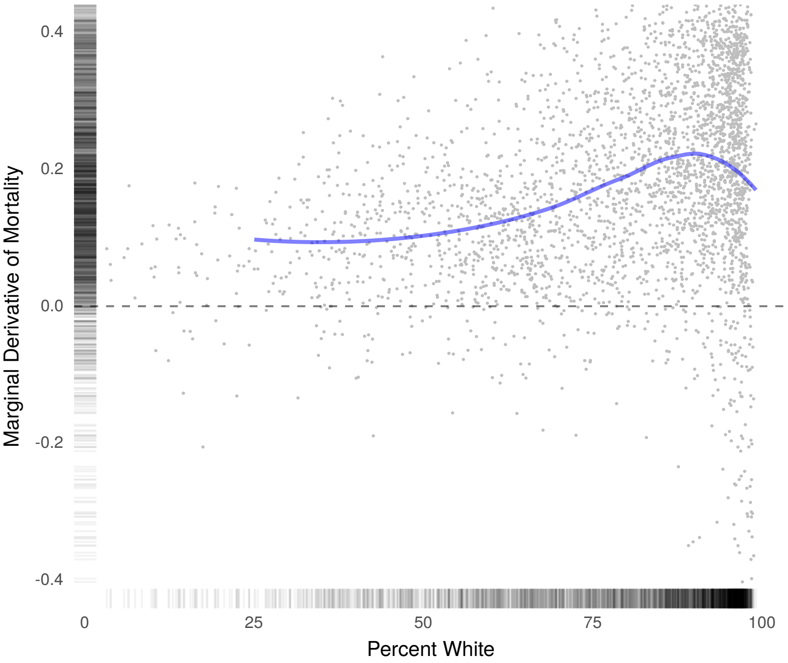
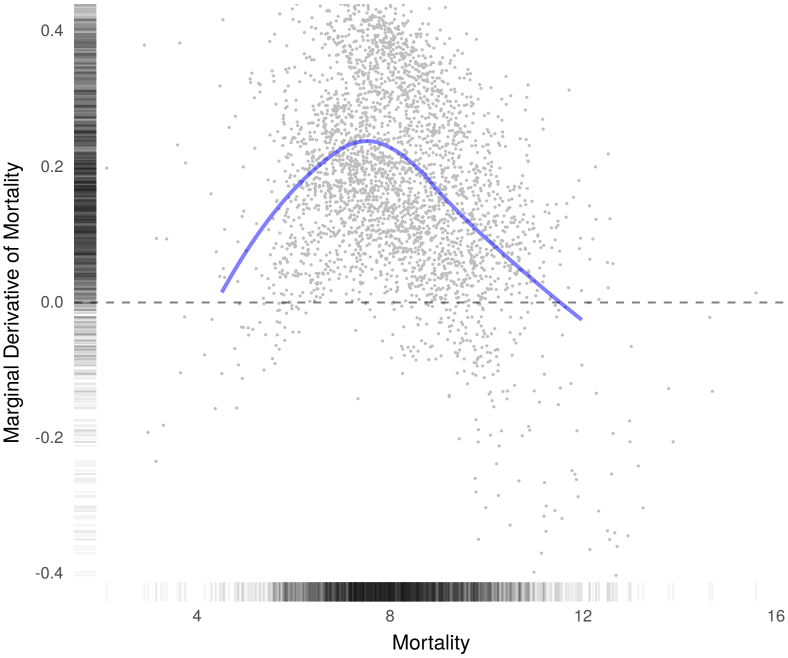


Figure 9: Marginal effect of age-adjusted mortality on Δ GOP presidential vote share, 2012-16, by mortality



6 Conclusion

In recent years, researchers have become increasingly interested in methods and models that combine canonical statistical properties with flexibility, robustness, and predictiveness. Some modeling approaches in this area also emphasize *interpretability*, which we argue should be viewed as a coequal goal with the other traits mentioned above. Hainmueller and Hazlett (2013) KRLS paradigm balances many of these concerns and has the capacity to contribute to social science research at a number of stages. Inevitably, KRLS offers no free lunch. By attempting to couple a flexible statistical model with interpretable effect estimates, KRLS encounters a steep *scalability* curve. We introduce *bigKRLS* not with the hopes of eliminating the computational burden of $N \times N$ calculations but rather in an effort to push the frontier (in terms of both N and P) for a variety of important political problems. For most applications, our improvements reduce runtime by about 75% and reduce memory usage by an order of magnitude. Our proposed p-value correction for the model's average marginal effect further improves on the model's statistical properties, particularly for complex data-generating processes estimated using smaller samples.

There are number of exciting areas for future work. Optimizing kernel regularization to scale to truly "big data" applications without compromising inference or interpretability is an open area of research. Though kernels are theoretically well suited to high dimensions (El Karoui 2010; Diaconis et al. 2008), large numbers of x variables still create practical problems, both computational and interpretative. Statistically, recent theoretical advances in selective inference (Taylor and Tibshrani 2015), risk estimation for tuning parameters (Tibshirani and Rosset 2016), and subsampling (Gu et al. 2013; Boutsidis et al. 2009; Homrighausen and McDonald 2016) offer paths forward in high dimensional space. In the meantime, our analysis suggests that KRLS can produce interpretable and theoretically useful estimates on a perennial topic of interest: the behavior of American voters.

References

Beck, A. and Ben-Tal, A. (2006). On the solution of the tikhonov regularization of the total least squares problem. *Journal of Optimization*, 17:98–118.

Boutsidis, C., Mahoney, M. W., and Drineas, P. (2009). An improved approximation algorithm for the column subset selection problem. In *Proceedings of the twentieth annual ACM-SIAM symposium on Discrete algorithms*, pages 968–977. Society for Industrial and Applied Mathematics.

Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.

Case, A. and Deaton, A. (2015). Rising morbidity and mortality in midlife among white non-hispanic americans in the 21st century. *Proceedings of the National Academy of Sciences*, 112(49):15078–15083.

Chipman, H. A., George, E. I., McCulloch, R. E., et al. (2010). Bart: Bayesian additive regression trees. *The Annals of Applied Statistics*, 4(1):266–298.

Demmel, J. W. (1997). *Applied numerical linear algebra*. Society for Industrial and Applied Mathematics.

Diaconis, P., Goel, S., and Holmes, S. (2008). Horseshoes in multidimensional scaling and local kernel methods. *The Annals of Applied Statistics*, 2:777–807.

Drineas, P. and Mahoney, M. W. (2005). On the nyström method for approximating a gram matrix for improved kernel-based learning. *journal of machine learning research*, 6(Dec):2153–2175.

El Karoui, N. (2010). The spectrum of kernel random matrices. *The Annals of Statistics*, 38(1):1–50.

Eleftheriadis, S., Rudovic, O., and Pantic, M. (2015). Discriminative shared gaussian processes for multiview and view-invariant facial expression recognition. *IEEE Transactions on Image Processing*, 24(1):189–204.

Ferwerda, J., Hainmueller, J., and Hazlett, C. (2017). Kernel-based regularized least squares in r (krls) and stata (krls). *Journal of Statistical Software, Articles*, 79(3):1–26.

Gill, J. (1999). The insignificance of null hypothesis significance testing. *Political Research Quarterly*, 52(3):647–674.

Grimmer, J., Messing, S., and Westwood, S. J. (2017). Estimating heterogeneous treatment effects and the effects of heterogeneous treatments with ensemble methods. *Political Analysis*, 25(4):413434.

Gu, C., Jeon, Y., and Lin, Y. (2013). Nonparametric density estimation in high-dimensions. *Statistica Sinica*, 23(3):1131–1153.

Guo, J. (2016). Death predicts whether people vote for donald trump.

Hainmueller, J. and Hazlett, C. (2013). Kernel regularized least squares: Reducing misspecification bias with a flexible and interpretable machine learning approach. *Political Analysis*, pages 1–26.

Hastie, T., Tibshirani, R., and Friedman, J. (2008). *The Elements of Statistical Learning*. Springer, second edition.

Hastie, T., Tibshirani, R., and Wainwright, M. (2015). *Statistical learning with sparsity: the lasso and generalizations*. CRC Press.

Hazlett, C. (2016). Kernel balancing: A flexible non-parametric weighting procedure for estimating causal effects. *arXiv.org*.

Homrighausen, D. and McDonald, D. J. (2016). On the nyström and column-sampling methods for the approximate principal components analysis of large datasets. *Journal of Computational and Graphical Statistics*, 25(2):344–362.

Imai, K., Lo, J., and Olmsted, J. (2016). Fast estimation of ideal points with massive data. *American Political Science Review*, 110(4):631–656.

Jackman, S. (2009). *Bayesian Analysis for the Social Sciences*. John Wiley & Sons, West Sussex.

Keele, L. (2015). The statistics of causal inference: A view from political methodology. *Political Analysis*, 23:313–35.

Kumar, S., Mohri, M., and Talwalkar, A. (2012). Sampling methods for the nyström method.

Journal of Machine Learning Research, 13(Apr):981–1006.

Monnat, S. M. (2016). Deaths of despair and support for trump in the 2016 presidential election.

Papadimitriou, C. H. (2003). Computational complexity. In *Encyclopedia of Computer Science*, pages 260–265. John Wiley and Sons Ltd., Chichester, UK.

Ratkovic, M. and Tingley, D. (2017). Sparse estimation and uncertainty with application to subgroup analysis. *Political Analysis*, 25(1):1–40.

Rifkin, R., Yeo, G., Poggio, T., et al. (2003). Regularized least-squares classification. *Nato Science Series Sub Series III Computer and Systems Sciences*, 190:131–154.

Rifkin, R. M. and Lippert, R. A. (2007). Notes on regularized least squares. *Computer Science and Artificial Intelligence Laboratory Technical Report.*

Siegel, Z. (2016). The trump-heroin connection is still unclear.

Taylor, J. and Tibshrani (2015). Statistical learning and selective inference. *Proceedings of the National Academy of Sciences*, 112:7629–7634.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of Royal Statistical Society*, 58:267–88.

Tibshirani, R. J. and Rosset, S. (2016). Excess optimism: How biased is the apparent error of an estimator tuned by sure? *arXiv preprint arXiv:1612.09415.*

Wager, S. and Athey, S. (2017). Estimation and inference of heterogeneous treatment effects using random forests. *Journal of the American Statistical Association*, (just-accepted).

Wahba, G. (1983). Bayesian “confidence intervals”. *Journal of the Royal Statistical Society*, 45(1):133–50.

Witten, I. H., Frank, E., and Hall, M. A. (2011). *Data Mining: Practical Machine Learning Tools and Techniques.* Elsevier, Burlington, MA, third edition.

Yu, K., Xu, W., and Gong, Y. (2009). Deep learning with kernel regularization for visual recognition. In *Advances in Neural Information Processing Systems*, pages 1889–1896.

Zhang, Z., Dai, G., and Jordan, M. I. (2011). Bayesian generalized kernel mixed models.

Journal of Machine Learning Research, 12:111–39.

A KRLS Steps

Figure 10: Overview of the KRLS estimation procedure

	Major Steps	**Runtime**	**Memory**
(1)	Standardize $\mathbf{X}_{\mathrm{N*P}}$, $\mathbf{y}$	—	—
(2)	Calculate kernel $\mathbf{K}_{N\times N}$	$O(N^2)$	$O(N^2)$
(3)	Eigendecompose $\mathbf{KE} = \mathbf{Ev}$	$O(N^3)$[i]	$O(N^2)$
(4)	Regularization parameter λ	$O(N^3)$[ii]	—
(5)	Estimate weights $\hat{\mathbf{c}}^* = \mathbf{f}(\lambda, \mathbf{y}, \mathbf{E}, \mathbf{v})$	$O(N^3)$	—
(6)	Fit values $\hat{\mathbf{y}} = \mathbf{K}\hat{\mathbf{c}}^*$	—	—
(7)	Estimate marginal effects, $\hat{\mathbf{\Delta}}_{\mathbf{N*P}} = [\hat{\delta}_{\mathbf{1}} \quad \hat{\delta}_{\mathbf{2}} \ldots \hat{\delta}_{\mathbf{P}}]$	$O(PN^3)$	$O(N^2)$

Letting i, j index observations such that $i, j = 1, 2 \ldots N$ ultimately captures all pairs and letting $p = 1, 2, \ldots P$ index the explanatory x variables. Note steps 4-6 are followed by uncertainty estimates, for which closed-form estimates also exist along with proofs of a number of desirable properties such as consistency (Hainmueller and Hazlett 2013).

[i] Using worst-case results for a divide-and-conquer algorithm, which we employ here (Demmel 1997, p.220-221). For runtime improvements via eigentruncation, see §4.2.1.
[ii] Using Golden Section Search given $\mathbf{y}$, $\mathbf{E}$ and $\mathbf{v}$. Note that this value also depends on a tolerance parameter, which is set by the user.

B C++ Kernel Regularization Code

This section provides code for the *RcppArmadillo* portion of the routine that the *bigKRLS* uses to obtain the coefficients, **c** (§3.5). The "extra" calls to *trans* (transpose) are computationally costless but enable computations on pointers to big matrices that could not otherwise be performed without non-trivial speed compromises.

```
template <typename T>
List xBigSolveForc(Mat<T> Eigenvectors,
                   const colvec Eigenvalues,
                   const colvec y,
                   const double lambda){

  int N = Eigenvectors.n_rows;
  int K = Eigenvectors.n_cols;
  // K at most N. Typically smaller (based on user eigentruncation input)
  double Le = 0;    //  leave one out error loss
  List out(2);

  // initializes coefficients to 0s
  colvec coeffs(N); coeffs.zeros();

  // initializes G inverse's diagonal (only)
  colvec Ginv_diag(N);
  Ginv_diag.zeros();

  // .memptr() expects data by column
  Eigenvectors = trans(Eigenvectors);

  for(int i = 0; i < N; ++i){

    // only length i to work on a triangle of Ginv
    colvec ginv(i);

    // .memptr() obtains raw pointer to particular elements
    mat temp_eigen(Eigenvectors.memptr(), K, i+1, false);

    ginv = (Eigenvectors.col(i).t()/
            (Eigenvalues + lambda)) * temp_eigen;

    Ginv_diag[i] = ginv[i];
    coeffs(span(0, i-1)) += ginv * y[i];
    coeffs[i] += sum(ginv * y(span(0,i)));
  }
  Eigenvectors = trans(Eigenvectors);

  for(int i = 0; i < N; ++i){
    Le += pow((coeffs[i]/Ginv_diag[i]), 2);
  }

  out[0] = Le;  // decision to accept lambda and use coeffs based on Le
  out[1] = coeffs;
  return out;
}
```

C Descriptive Statistics for 2016 Election Study

Figure 11 gives summary statistics for the variables used in our applied example in §5. All race and education variables are given in percentage point units. Units for other variables are given in table notes. For our applied example, we also include latitude, longitude, and state dummy variables as additional predictors. Replication materials are available here.

Figure 11: Descriptive statistics for Section 4

Variable	Mean	SD	Source
Δ GOP presidential vote share, 2012-16[i]	5.86	5.26	Townhall
Mortality[ii]	8.17	1.48	CDC
Δ Mortality[ii]	-0.04	0.71	CDC
Urban-Rural Continuum[iii]	4.98	2.70	USDA
Age[iv]	4.03	0.50	US Census
Income[v]	4.85	1.23	USDA
Unemployment	5.5	1.94	USDA
Poverty	3.13	1.17	USDA
No High School Diploma	14.60	6.63	USDA
High School Graduate	34.76	7.07	USDA
Some College	30.23	5.15	USDA
College Graduate	20.40	9.01	USDA
White	78.55	19.60	CDC
Latino	6.69	13.27	CDC
Black	8.93	14.71	CDC
Asian	0.97	3.14	CDC

All variables below and including "Unemployment" represent county-level percentages.
[i] The dependent variable is measured % Trump - % Romney via McGovern' s data.
[ii] All cause mortality per 1,000 individuals and age-adjusted. Mortality change subtracts 2013-2015 from 2009-2011. Data from counties with fewer than 10 deaths are suppressed by the CDC for privacy reasons, and are excluded from this analysis.
iii Ordinal variable, ranging from 1 (most urban) to 7 (most rural).
iv Average; measured in 10s of years.
v Median household income (in $10,000s).

D Simulations

In this Appendix, we describe a series of simulation experiments we reference throughout the text of our paper. We begin by describing the basic simulation setup we employ for most experiments, which we modify where necessary in each test.

D.1 Setup

Our simulation procedure is organized as follows. Using the county-level election dataset we use in our applied example presented in §4, we constructed a dataset containing eight variables: age-adjusted mortality, urban-rural continuum, age, income, unemployment rate, poverty rate, % college graduate, and % white. We then simulated a dependent variable using the following data-generating process:

$$y_i = \mathbf{X}_i\beta_1 + \mathbf{X}_i^2\beta_2 + \mathbf{X}_i^3\beta_3 + \mathbf{X}_i^4\beta_4 + \mathbf{X}_i\Theta_{z[i]} + \epsilon$$

Where $\mathbf{X}_i$ denotes the i^{th} row of $\mathbf{X}$, β_j represents a column vector of coefficients corresponding to the j^{th}-order polynomial of $\mathbf{X}$. Since county-level data possess a natural hierarchical grouping, we incorporated a hierarchical component into our data-generating process by grouping counties into the 8 US Census regions, and perturbing each linear effect based on district membership. In particular, we constructed Θ as a $9 \times P$ matrix of hierarchical effect disturbances, and $z_{[i]}$ as an auxiliary matrix denoting the census division to which the i^{th} county belongs. To place values on these coefficients, we set $\beta_1 \sim Uniform(0, 2)$, $\beta_{j>1} \sim Uniform(-4, 4)$, and $\Theta \sim Uniform(-2, 2)$, and $\epsilon \sim Normal(0, 3000)$. We selected our error covariance value in order to ensure that the in-sample R^2_K value remained reasonably close to its observed value in our dataset ($R^2_K \approx 0.8$). Finally, we standardized each coefficient based on the standard deviation of the variable in question, in order to ensure that derivative calculations were not dominated by any one term.

Our target for most simulations in this section is the population average marginal effect

(AME), defined as:

$$AME_{pop} = \beta_1 + \frac{2\beta_2}{n}\sum_i \mathbf{X}_i + \frac{3\beta_3}{n}\sum_i \mathbf{X}_i^2 + \frac{4\beta_4}{n}\sum_i \mathbf{X}_i^3 + \frac{1}{n}\sum_i \Theta_{z[i]}$$

Which simply represents the average derivative of the equation given above with respect to each row of $\mathbf{X}$.

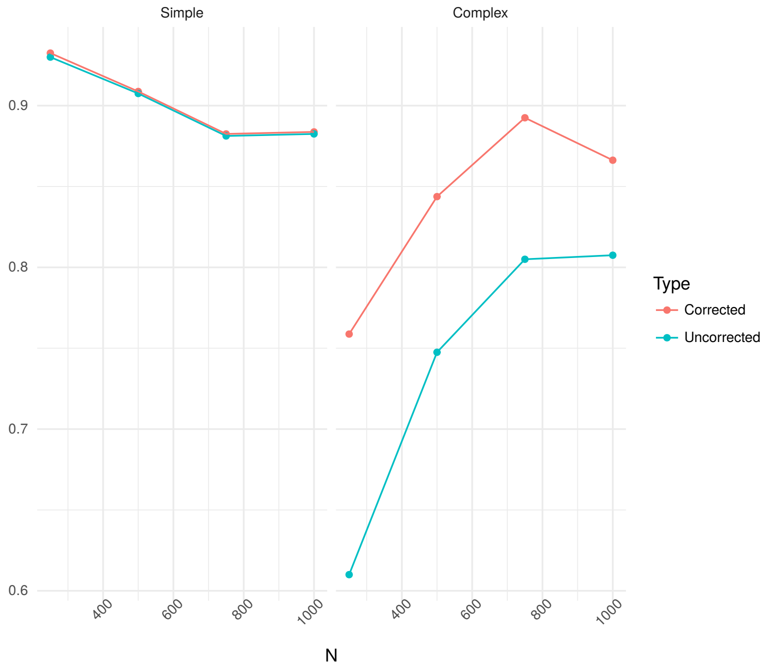
D.2 Degrees of Freedom Correction for Average Marginal Effects

Using the simulation setup described in the previous section, we examined the impact of our degrees of freedom correction at various sample sizes and under two different data-generating processes.

Our procedure functioned as follows. First, we drew values for all parameters besides ϵ using the procedure described above. Next, for each iteration, we drew ϵ and a random subsample of our dataset. We then generated our dependent variable according to two data-generating processes, which we label "complex" and "simple". The "complex" DGP is identical to the equation described in the previous section. The "simple" DGP consists of this same equation, but with all second-, third-, and fourth-order polynomial coefficients fixed at zero. Using these two DGPs, we fit two models using *bigKRLS* to the combined matrix consisting of the eight predictor variables and the eight US Census divisions for the observations selected into our sample, and calculated marginal and average marginal effects (AMEs) for the eight predictor variables. Finally, for each variable in each model we recorded AME estimates, standard error estimates (corrected and uncorrected), and other auxiliary information.

Coverage results of this experiment both with and without our degrees of freedom correction are are presented in Figure 12. In nearly all cases, empirical coverage results are somewhat lower than their nominal 0.95 value, which likely reflects the bias in our estimates

Figure 12: Simulated AME coverage results



Note: values represent average coverage rate across 100 iterations, with 8 coefficients in each iteration.

produced by the regularization strategy we employ. However, the impact of our correction is clear. In the "simple" DGP, both strategies return essentially identical results. But, in the "complex" DGP, our correction represents approximately a $10 - 15$ percentage point increase in empirical coverage rate, bringing the empirical rate substantially closer to the nominal value. Unsurprisingly, the difference in coverage rates is somewhat smaller at larger samples. However, at all sample sizes we examine, the correction we propose has a clear positive impact.

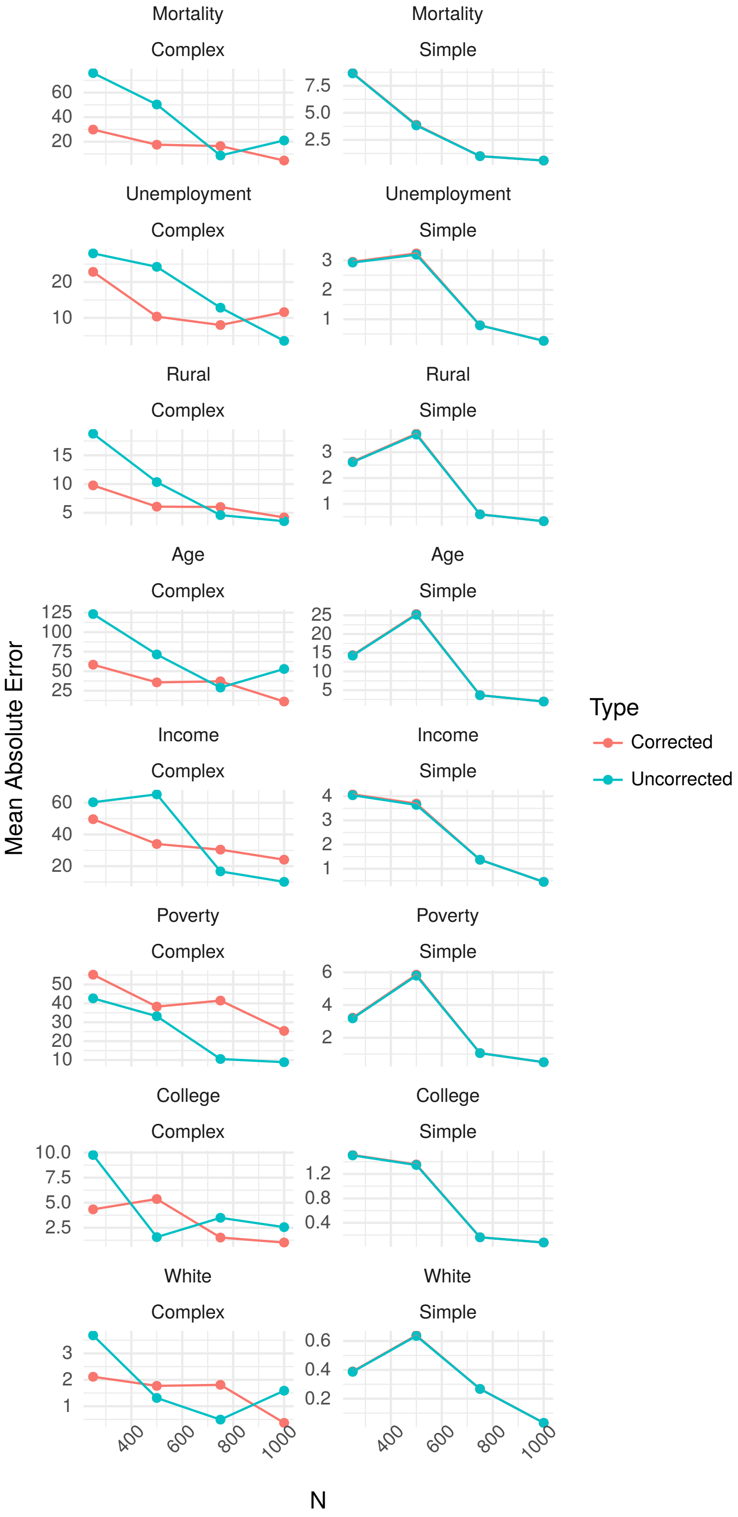
To probe these results more closely, an anonymous reviewer suggested the following additional test. Since the $KRLS$ estimates are biased due to the regularization procedure we employ, an alternative to examining standard coverage rates is to compare the standard error estimates produced by the corrected and uncorrected model to the "true" standard errors of the regularized AME estimates. In simulations, a straightforward resampling-style approach to estimate these these "true" standard error values is to calculate the cross-sample standard deviation of the AME estimates. We can then compare the corrected and uncorrected standard error estimates produced in each iteration to assess which procedure more faithfully reproduces the population standard errors for the regularized coefficient estimates.

The results of this comparison are given in Figure 13. As before, performance of the error estimates for the simple DGP are essentially identical across the two modeling strategies. By contrast, for the complex DGP the two methods produce noticeably different results. Generally, our corrected standard errors perform best at smaller sample sizes, with corrected standard error estimates for seven out of eight coefficients outperforming their uncorrected counterparts at $N = 250$. At $N = 1000$, by contrast, performance is split, with half performing better with our correction and half performing worse. However, performance is not constant across coefficients; for example, our correction consistently produces similar or superior performance for the Rural, Age, and Mortality variables, but similar or inferior performance for the Poverty variable.

These simulation experiments offer several basic conclusions. For data-generating processes involving essentially constant effect estimates, our correction produces nearly identical results to those generated using the uncorrected approach. As a result, in these cases the choice between the two approaches is not particularly consequential. But, for data-generating processes involving greater effect heterogeneity, our correction offers a noticeable performance boost. This improvement is most noticeable at smaller sample sizes, but remains substantial at larger samples at least as measured by coverage.

More generally, in an informal sense these results suggest that the impact of our correction will be larger to the extent that a particular user's data-generating process is more complex and their sample size is smaller. However, the extent to which our correction will affect a given user's results is clearly context-dependent. In future work, further simulations of this kind might help produce additional insight and advice for applied users. However, since no one paper can simulate all possible scenarios, we believe that these results offer reasonable justification for our choice to make the correction our default in *bigKRLS*.

Figure 13: Population SE versus estimated SE for regularized AME estimates



Note: per-coefficient performance measured via mean absolute error, across 100 iterations per sample size. Scales are allowed to float freely for each variable. Estimates for the "simple" DGP overlap nearly perfectly in all cases.

D.3 Subsample-Based AME Estimates

Estimating the kernel-based model we adopt in the KRLS framework is computationally demanding. However, as suggested by an anonymous reviewer, some of the simpler estimates contained in this model can be reasonably well-approximated using a subsampling scheme, which cuts computation time and resources substantially. The approach we adopt is particularly well-suited for the average marginal effect estimates (AMEs) generated by this model, which are the focus of this section. However, we also discuss the applicability of this approach for other quantities at the conclusion of this section.[26]

To generate subsample-based AMEs, we propose the following approach. First, divide the observations into M equally-sized groups.[27] Second, fit a model via KRLS to each subgroup and estimate AMEs for each model, denoted $\hat{\Delta}_{AME}^{(m)}$. Since the estimated AMEs across subgroup models are independent, we can leverage the closed-form expressions for their values and variances to produce an aggregated set of estimates:

$$\hat{\Delta}_{AME} \approx \frac{1}{m} \sum_{m} \hat{\Delta}_{AME}^{(m)}$$
$$V(\hat{\Delta}_{AME}) \approx \frac{1}{m} \sum_{m} V(\hat{\Delta}_{AME}^{(m)})$$

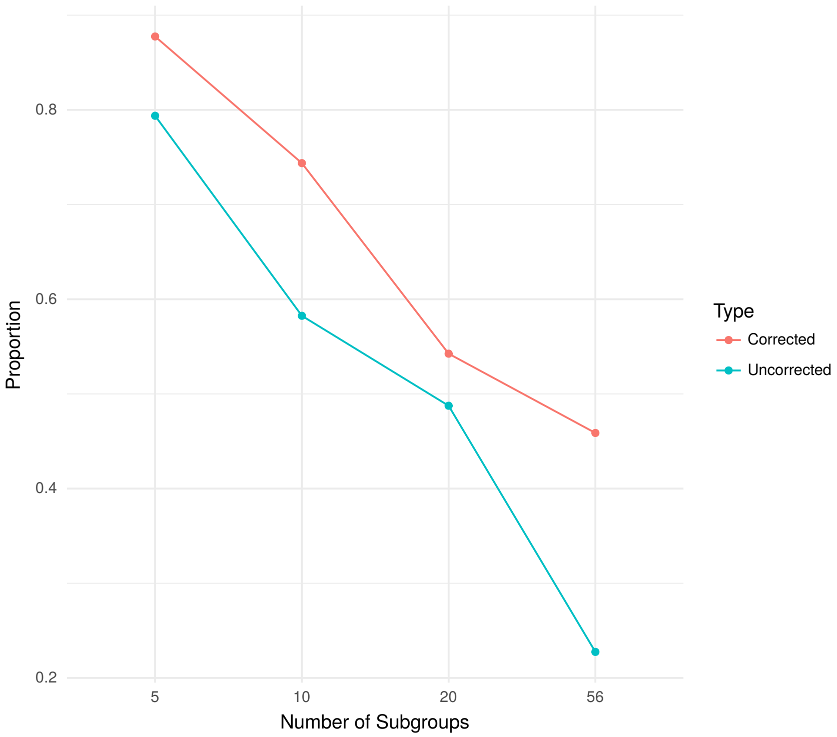
This approximation relies on the asymptotic normality of the AMEs. As a result, we should expect this approximation to behave better with large subgroups than with small ones.

To use these estimates as part of a hypothesis test, we can use a similar t-test framework to that proposed by Hainmueller and Hazlett (2013). However, since we now estimate P AMEs for each subgroup, the degrees of freedom for this test will be $N_{effective} - M \times P$, with $N_{effective}$ defined using the ridge correction we describe in §3.3. For small subsamples, then,

[26] This approach is related to ideas presented by, e.g., Drineas and Mahoney (2005); Kumar et al. (2012), which represent additional directions for future research.

[27] In this discussion, we assume that the subsampling scheme will produce subgroups which respect the original KRLS assumptions. In particular, we assume that no column of any subgroup data matrix $\mathbf{X}_m$ will be constant, which can be ensured by selecting an appropriately small M or adopting a stratified subgroup assignment scheme.

Figure 14: Empirical coverage for subsample-based AME estimates



Note: point estimates represent average coverage across 100 iterations. For all numbers of subgroups, the total number of observations is $N = 3,106$, the full set of counties in our dataset.

tests based on this value may be undefined, offering further motivation to adopt a larger subsample size when employing this approach.

To assess the performance of these estimates in practice, we returned to the simulation setup we describe at the outset of this section. Using the full dataset of 3,106 counties, we simulated effects using the "complex" (hierarchical and polynomial) data-generated process described in Appendix D.2, and subdivided the data, using 5, 10, 20, and $\sqrt{N} \approx 56$ subgroups. To ensure that no variables in any subgroup were constant, we stratified subdivision assignments by census division, and fit a model to each subdivision. Using these models, we aggregated our subgroup-level AME estimates, and assessed statistical significance for each estimate (using $\alpha = 0.05$ as a significance threshold). We repeated this process 100 times for each subsample size, and recorded coverage results for each iteration.

The results of this comparison are given in Figure 14. Following expectations, performance decays as the number of subgroups grows, with estimates generated using 5 subgroups performing roughly as well as the full model with $N = 1000$ observations as presented in Appendix D.2. As before, coverage results are noticeably better with our correction than without; however, with $\sqrt{N} \approx 56$ subgroups, neither approach offers adequate performance performance.

In our view, these results suggest that a subgroup-based estimation strategy for the AMEs can be useful. From a speed perspective, this approach is particularly appealing; with 5 subgroups, estimating models for all five subgroups takes a total of 50 seconds, compared with approximately 20 minutes to estimate the full model. Selecting an optimal subgroup size is a possible direction for future research. However, since this strategy will be most appealing for datasets with a large number of observations, opting for the largest feasible subgroup size is a sensible strategy, especially if (as we anticipate) the primary use case for this procedure is an exploratory one.

Adapting this strategy to more nuanced quantities (e.g., pointwise marginal derivatives) is a promising but more challenging avenue for future work. As Grimmer et al. (2017) note, runtime can be a limiting factor on the ability to include methods and models in an ensemble learner (419). Establishing the conditions under which the subsampled coefficients stabilize for complex data generating processes is sufficient to accurately generate $\hat{y}_{test}$ and therefore would be sufficient to adapt a subsampled version of KRLS to ensemble learner they propose.

E Crossvalidation

E.1 bigKRLS Out-of-Sample

To assess the stability of the regression estimates, we estimated a series of five-fold cross-validated replicates for a version of the model we present in-text. As a first cut, we estimated on a simplified version of the dataset we present in-text. This dataset - which is identical to the one we use in our simulation studies discussed in Appendix D - groups states into their eight census districts, and retains eight of the predictor variables included in the full model we present in-text (see Appendix D.1 for variable details).

The results of this experiment are presented in Figure 15. Across cross-validation replicates, performance statistics are stable, indicating that the full sample estimates (presented in the Section 4) are unlikely to have been influenced by outliers, subgroup-specific patterns, or by our specific geographic specification. In-sample, the full model of N coefficients consistently outperforms the portion which is a linear and additive function of the x variables, the Average Marginal Effects (AMEs). Out-of-sample, however, the performance gap is much smaller. As a result, we encourage users to employ cross-validation or a similar out-of-sample predictive strategy if precise estimates of effect heterogeneity are desirable.

E.2 Comparison to Other Approaches

To assess KRLS's out-of-sample performance relative to other modeling approaches, we compared cross-validated performance for KRLS to a variety of other related modeling approaches. As in the previous section, to ensure that no columns were constant we grouped the 50 states into 8 US Census divisions, which were included as dummy variables in place of the 50 state-level dummies included in our original model specification. For similar reasons, we also dropped the differenced mortality variable from our model specification in this section, since this variable was nearly constant for most counties in our dataset. All other variables we provide in text were retained unchanged for the tests in this section (see Appendix C

Figure 15: Overview of Crossvalidation Results

	In Sample	Out of Sample
RMSE	3.000 (0.196)	3.086 (0.109)
RMSE$_{AME}$	3.337 (0.139)	3.338 (0.154)
R^2	0.674 (0.0001)	0.660 (0.0009)
R$^2_{AME}$	0.104 (0.0001)	0.640 (0.0008)

Results of 75 five-fold cross-validation replicates (N_{train} = 80% = 2,485 observations; N_{test} = 20% = 621 observations). Average measures of fit provided along with standard errors. *AME* subscript indicates that only the Average Marginal Effects were used to obtain fitted values in sample or predicted values out of sample. For convenience, *crossvalidate.bigKRLS*, which also performs K folds cross validation, computes these measures of fit.

for details). As before, we use a 5-fold cross-validation procedure, with performance results generated by averaging root-mean squared error (RMSE) and root-mean squared predictive error (RMSPE) across each fold and replicated 100 times to generate uncertainty estimates. To generate uncertainty estimates, we replicated our cross-validation process 100 times, and presented mean, 2.5$^{\text{th}}$ and 97.5$^{\text{th}}$ percentile results for each performance statistic.

For our comparison models, we used a simple random forest (Breiman 2001), Wager and Athey (2017)'s Causal Forest procedure, and a series of penalized regression models trained via the glmnet package in R. Clearly, these models are not the only possible points of comparison; however, since no empirical test of this kind can examine all possible approaches, we selected this group as a representative subset of flexible, context-agnostic models commonly used in political science.

Specifications for all additional models are as follows. For our random forest models, we trained our models using 500 trees, with the parameter denoting the number of candidate variables to consider at each split selected by optimizing out-of-bag error (Breiman 2001).

For our causal forest models, we used 2000 trees, with our county-level age-adjusted mortality variable acting as the "treatment" variable.[28] For our penalized regression approaches, we trained two sets of models. In the first set, we only allowed the models to estimate linear and additive effects for each variable. In the second set, we additionally allowed the model to estimate coefficients corresponding to all two-way interactions between each variable included in our dataset. In both cases, we trained one model using a LASSO penalty, one model using a ridge penalty, and one using an elastic net penalty (with mixing parameter set to $\alpha = 0.5$). For all penalized approaches, we selected the regularization parameter λ by optimizing cross-validated error within the training set.

The results of this comparison are given in Figure 16. Compared with most other approaches, KRLS is prone to overfitting, with the largest gap between RMSE and RMSPE of the models we examine. However, KRLS's out-of-sample performance remains competitive. By RMSPE, KRLS is the second-best performer, producing approximately a 1% higher RMSPE than a simple random forest. In our view, this result is encouraging. In most predictive modeling contexts, random forests represent the best-performing general-purpose approach. As a result, KRLS's ability to essentially match a random forest's predictive performance is reassuring.

All other approaches we examine offer noticeably worse performance than simple random forests and KRLS. By RMSPE, elastic- and ridge-penalized regression approaches trained using all two-way interactions are the next-best performers, followed by Causal Forest predictions and the remaining penalized regression models. Causal Forests, in particular, offer a useful point of comparison with KRLS. Like KRLS, the goal in the Causal Forest paradigm is to estimate a heterogeneous effect. Unlike KRLS, under appropriate assumptions Causal Forests offer useful theoretical guarantees regarding causal interpretability of effect estimates for the treatment variable (Wager and Athey 2017). However, since our application in this

[28] We use "treatment" here only to denote the variable of interest in the causal forest specification. We do not claim that our data meet the assumptions required to place causal interpretations on our mortality variable.

Figure 16: In- and out-of-sample prediction results, comparing KRLS to related approaches

	RMSE	RMSPE
Random Forest[i]	2.389	**2.381**
	(2.383, 2.394)	(2.363, 2.399)
KRLS	**2.181**	2.415
	(2.177, 2.184)	(2.397, 2.432)
Elastic (two-way)[ii]	2.326	2.469
	(2.308, 2.349)	(2.454, 2.492)
Ridge (two-way)[ii]	2.333	2.470
	(2.313, 2.355)	(2.450, 2.492)
Causal Forest[iii]	2.623	2.618
	(2.624, 2.635)	(2.606, 2.631)
LASSO (two-way)[ii]	2.585	2.660
	(2.571, 2.599)	(2.644, 2.677)
Ridge[ii]	2.895	2.917
	(2.880, 2.909)	(2.902, 2.931)
Elastic[ii]	2.895	2.917
	(2.882, 2.913)	(2.904, 2.936)
LASSO[ii]	2.905	2.930
	(2.894, 2.917)	(2.916, 2.943)

Notes: Results of 100 five-fold cross-validation replicates (N_{train} = 80% = 2,485 observations; N_{test} = 20% = 621 observations). *RMSE* denotes in-sample root-mean squared error, and *RMSPE* denotes out-of-sample (predicted) root-mean squared error. Parenthetical values represent ±2 standard deviations.

[i] Estimated via the randomForest package, using the *tuneRF* function to select the number of (randomly-selected) candidate variables to consider at each split (Breiman 2001).
[ii] Estimated via the glmnet package, using the *cv.glmnet* function to select the regularization parameter λ. "Two-way" indicates models which were trained using all two-way interactions in addition to base effects. All other models were trained using simple linear effects only.
[iii] Estimated via the Generalized Random Forest package, using the *causal_forest* function and specifying age-adjusted mortality as the "treatment" variable (Wager and Athey 2017).

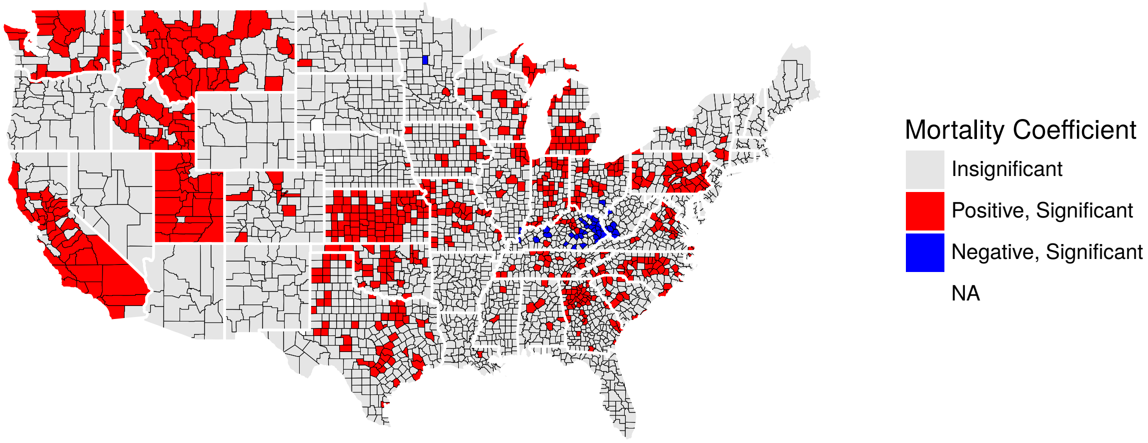
paper is more exploratory, KRLS's superior out-of-sample predictive performance and ability to estimate effects for each variable directly offer strong reasons to prefer KRLS in this context.

F Significance of predicted first difference estimates

If $\mathbf{y}_{test}$ and $\mathbf{K}_{test}$ represent the predicted values and kernel calculated using the test points generated by perturbing the original data matrix, then $Var(\hat{\mathbf{y}}_K) = \mathbf{K}'(\sigma_\epsilon^2 I(\mathbf{K} + \lambda I)^{-2})\mathbf{K}$ and $Var(\hat{\mathbf{y}}_{test}) = \mathbf{K}'_{test}(\sigma_\epsilon^2 I(\mathbf{K}_{test} + \lambda I)^{-2})\mathbf{K}_{test}$ (Hainmueller and Hazlett 2013). In practice, since σ_ϵ^2 is unknown we replace this quantity with $\hat{\sigma}_\epsilon^2 = \frac{1}{N_{eff}}(y - \mathbf{K}\hat{c}^*)'(y - \mathbf{K}\hat{c}^*)$, with N_{eff} defined using the degrees of freedom correction we propose elsewhere in this paper. Since both of these quantities are distributed multivariate normal, by standard identities their difference is also multivariate normal with variance $Var(\hat{\mathbf{y}}_K) + Var(\hat{\mathbf{y}}_{test})$. For the i^{th} difference, the marginal distribution of that difference is univariate normal, with variance $(Var(\hat{\mathbf{y}}_K) + Var(\hat{\mathbf{y}}_{test}))_{ii}$.

Using these facts, we propose a straightforward hypothesis test. Our null hypothesis is that the difference between the predicted and counterfactual value is zero. Since the marginal distribution of each difference is univariate normal, a straightforward test statistic for any given point is $\frac{(\hat{y}_K - \hat{y}_{test})_i}{\sqrt{(Var(\hat{\mathbf{y}}_K) + Var(\hat{\mathbf{y}}_{test}))_{ii}}} \sim t_{N_{eff}}$. In the body of our paper, we examine many such test points simultaneously; as a result, we correct for multiplicity by applying a Benjamini-Hochberg procedure to the p-values generated using this procedure.

Figure 17: Significance results for individual county-level predictions



Notes: Statistical significance for individual county-level first differences. P-values corrected via Benjamini-Hochberg procedure, with $\alpha = 0.05$.