Messy Data, Robust Inference?

Navigating Obstacles to Inference with bigKRLS

Pete Mohanty pmohanty@stanford.edu

Robert Shaffer rbshaffer@utexas.edu

August 16, 2017

Abstract

Complex models are of increasing interest to social scientists. Flexible prediction-oriented approaches (e.g., decision trees) often improve fit over conventional solutions, while causally-oriented studies must often incorporate complicated treatment structures or confounding relationships. Unfortunately, estimators of complex models often scale poorly. Though optimization, whether mathematical or software, cannot fully resolve this conflict, it can alleviate the worst of these concerns.

In this paper, we develop a conceptual framework with which to consider tradeoffs in this setting. We then present an example of this kind of optimization work by introducing bigKRLS, a memory- and runtime-optimized version of Hainmueller and Hazlett (2013)'s Kernel-Regularized Least Squares (KRLS). KRLS is a flexible yet interpretable approach which, like many penalized regression approaches, encounters substantial scalability challenges. Our improvements reduce peak memory usage by an order of magnitude and decrease single-core runtime by 50% (with additional improvements available via parallelization). By analyzing the 2016 presidential election, we show how *bigKRLS* can help researchers navigate obstacles to inference that would otherwise be difficult or impossible to address. We also develop an updated significance test for the average marginal effects estimates produced by the model, which we justify both theoretically and in simulation.

This project has benefited immensely from feedback at the Stanford University Statistics Seminar, Tuesday April 18th, 2017; *useR!* 2016, hosted by Stanford University; the *American Political Science Association* 2016 Annual Meeting in Philadelphia; the *International Methods Colloquium*, hosted by Justin Esarey on November 11, 2016; the Stevens Institute of Technology's College of Liberal Arts on February 27th, 2017; and the *Bay Area R Users Group* Official Meet Ups, hosted by Treasure Data (May 2016), Santa Clara University (October 2016), and GRAIL (June 2017). Thanks in particular to Joseph Rickert, Susan Holmes, Trevor Hastie, Christian Fong, Luke Sonnet, Chad Hazlett, Kristyn Karl, and Jacob Berman for additional suggestions, comments, and advice. Pete Mohanty thanks Stanford University's Vice Provost for Undergraduate Education for research leave.

1 Introduction

Robustness, predictive accuracy, and interpretability are desirable attributes for any statistical approach, particularly in social science research. As ongoing political commentary reminds us, both the academic and broader communities care about robust, predictively-oriented models, 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 more easily. As we argue, models that are sparse, parsimonious, and directly estimate quantities of interest are generally more interpretable than those that do not.

Perhaps unsurprisingly, models that possess these traits also exhibit severe scalability constraints. For a concrete example, take Hainmueller and Hazlett (2013)'s Kernel Regularized Least Squares model. Compared with other approaches, KRLS offers a desirable balance of interpretability, flexibility, and theoretical properties, primarily through the pointwise marginal effect estimates produced by the routine and their corresponding averages. However, these "actually" marginal effects are costly in time and memory to estimate, both via KRLS and through related techniques such as BART (Chipman et al. 2010) or LASSO-plus (Ratkovic and Tingley 2017). As a result, optimization (both in memory and speed) is important to make these models useful for applied researchers.

In this paper, we present a series of algorithmic improvements designed to improve KRLS's speed, memory usage, and statistical performance, which we implement in the *bigKRLS* package. Compared with the original *KRLS* estimation approach,[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

[1] In this manuscript, 'KRLS' refers to the estimator whereas '*KRLS*' refers to an R package.

[2] Assuming parallelization is used. Single-core run-time is approximately 50% faster with *bigKRLS* than the original *KRLS* implementation.

KRLS to larger datasets (N > 3,500) on personal machines, which was not possible using existing approaches. We also develop an updated significance test for the average marginal effects estimates produced by the model, which we justify both theoretically and in simulation.

Finally, we illustrate the practical utility of *bigKRLS* through an extended examination of the so-called "communities in crisis" explanation for the 2016 presidential election. As we demonstrate, 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 estimate 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 Model Interpretability

When constructing an estimator, there are an array of properties which we might find desirable. For example, we might want our estimator to be unbiased or efficient, or we might want our estimator to minimize some particular loss function (e.g., mean squared error). In the theoretical setting, we generally assume that our model of interest captures the "true" data-generating process; however, in applied settings, we are usually skeptical of these kinds of assumptions. For applied work, then, we also want our estimators to be robust against violations of potentially problematic modeling assumptions (e.g., incorrect functional form or omitted variables). At least in this context, predictive accuracy based on held-out testing data (an empirical, "data driven" property) might be more desirable than some kinds of theoretical guarantees.

Besides these traits, however, in applied settings we also favor models that are *inter-*

pretable. Compared with the traits described above, "interpretability" does not possess a particularly precise definition. Colloquially, we might view a model as "interpretable" if the values it estimates allow users to answer useful questions with minimal additional effort, which usually implies the need to be able to communicate results with others. A model like linear regression, for example, offers single coefficient estimates that offer information about the marginal effect of some covariates X on a dependent variable y.

We can usefully frame interpretability using the concept of *cognitive load*. As used in the cognitive science literature, cognitive load refers to the "demands on working memory" (Paas et al. 2003) imposed by a particular task or concept. High-dimensional tasks, which require users to simultaneously hold more ideas in working memory, place a larger cognitive load on users than lower-dimensional equivalents (Gerjets et al. 2004; Sweller 1994, 2010). In this sense, models that are parsimonious (few auxiliary/nuisance parameters) or sparse (few non-zero parameters) usually offer greater interpretability than their more parameter-rich counterparts (Hastie et al. 2015). Regularization constraints, in particular, are explicitly designed to reduce the effective dimensionality of a model, trading reduced flexibility for improved interpretability and (usually) better out-of-sample performance (James et al. 2013, 24).

An"interpretable" model, from this perspective, is one that possesses most (or all) of the following traits:

1. *Parsimony.* Models with few nuisance parameters (e.g. linear regression) are generally easier to interpret than their more complex counterparts (e.g. penalized regression, mixture models).
2. *Sparsity.* Sparsity constraints and shrinkage procedures (e.g. LASSO or elastic net) allow users to ignore a subset of parameters, reducing effective model dimensionality and easing interpretation.
3. *Direct estimation of quantities of interest.* In most applications, we favor methods and models that facilitate causal inferences as well as simple predictions. Methods that

either cannot produce these values or that require substantial post-estimation work to generate these values are less interpretable than those that estimate these quantities directly.[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 trait that researchers ought to seek. Depending on the application, researchers might be willing to employ a more cognitively demanding 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 depending on the setting of interest.

2.2 The Complexity Frontier

Unfortunately, improving the flexibility, robustness, and parsimony of a model generally involves increasing its *complexity*. 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, simple linear regression with N observations and P covariates has complexity $O(P^2N)$ (since calculating $\mathbf{X'X}$ dominates other calculations involved in generating $\hat{\beta}_{OLS}$).[5] Since N is usually much larger than P, simple linear regression therefore has complexity that is

[3] Arguably, we might view Bayesian posterior probabilities as a good example of an "interpretable" procedure. As Gill (1999), Jackman (2009) and others argue, the frequentist null hypothesis testing paradigm is remarkably 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).

[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, these terms can be included in the complexity statement.

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

approximately linear with the number of observations.

Compared with other approaches, under appropriate assumptions simple linear regression directly calculates causally interpretable effects, but is not robust to assumption violations and possesses poor predictive performance. On the other end of the spectrum, decision trees directly calculate very few quantities of interest, 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, assuming N observations and P independent variables, 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*. Complexity constraints, in other words, impose a tradeoff between flexibility, sparsity, and other traits that we might find desirable, rendering many approaches intractable for larger datasets. Importantly, in many applications, interpretability also factors into this tradeoff. Complexity penalties (e.g. LASSO) and cross-validated parameter selection offer obvious examples of this relationship, but many modeling approaches exhibit this tradeoff.

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. In this and similar situations, optimization work can offer a substantial benefit for applied researchers, especially

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

when the speed and memory gains are large.

3 Algorithmic and Statistical Optimization using bigKRLS

3.1 Overview

For a stark example of the complexity frontier phenomenon, consider Kernel-Regularized Least Squares (KRLS) (Hainmueller and Hazlett 2013). Kernel Regularized Least Squares (KRLS) is a kernel-based, complexity-penalized regression approach developed by Hainmueller and Hazlett (2013) intended to simultaneously maximize flexibility, robustness, and interpretive clarity. This mix of traits allows the model to easily incorporate heterogeneous treatment effects, which is helpful in most modeling settings. Since KRLS estimates marginal effects for each data point, researchers can easily determine whether the effect of a given variable appears to be constant across the sample or assess whether or not the outcome is a monotonic function of a predictor of interest.

Predictably, however, KRLS is also computationally demanding. The source of the model's nuance – pairwise comparison – makes the linear algebra required to estimate model parameters unusually memory intensive (see Appendix A for a detailed overview). Compared with many workhorse methods, KRLS requires substantially greater resources to estimate, with total runtime complexity $O(N^3)$. By contrast, as noted in the previous section decision trees have complexity $O(Nlog(N)^2) + O(PNlog(N))$.

Memory requirements for KRLS are similarly restrictive. In *bigKRLS* (our implementation of the KRLS model), at peak runtime the estimation algorithm still has $O(N^2)$ memory complexity. This figure is a substantial improvement over the $O(PN^2)$ requirements of the original algorithm, but remains difficult to scale.[7] In the C language, for example, double-precision numbers require 8 bytes of storage space, so a single $5,000 \times 5,000$ matrix requires

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

at least 200 MB of working memory plus any overhead for the underlying data structure. For reference, our applied example (described in detail in Section 4) uses $P = 67$ independent variables; a dataset with a similar number of predictor variables would require $\approx$ 8 GB of free memory to estimate with *KRLS*, the upper limit available for many personal machines.

How important are these limitations? 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.[8]

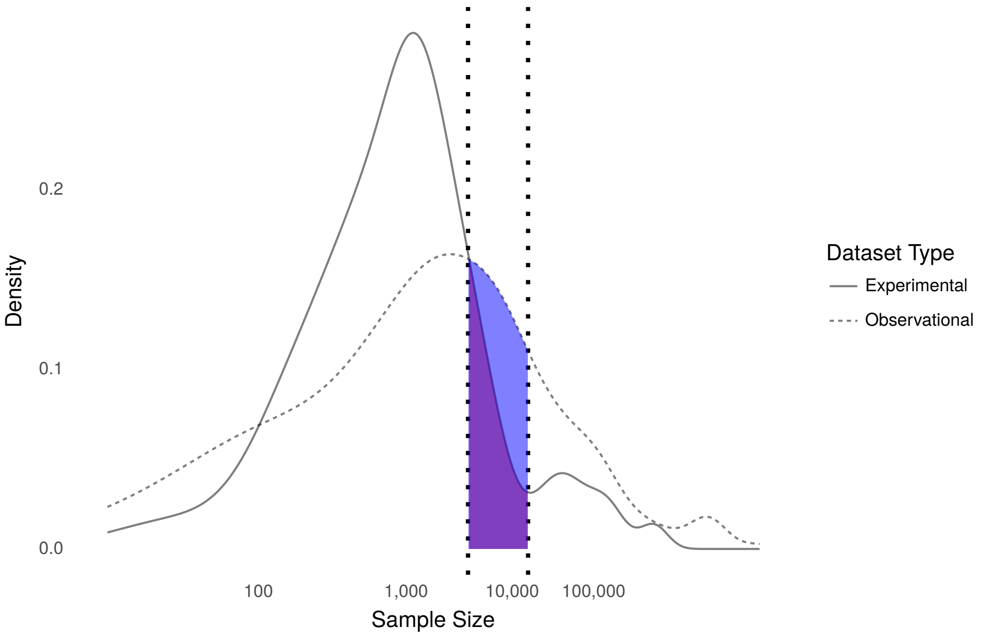
These improvements, we argue, are substantial. In methodological work, new estimators and models are only useful to the extent that they can be employed in practice. While high-complexity methods like KRLS are unlikely to be usable in truly "big" data settings, our improvements make KRLS much more accessible, allowing a noticeably greater proportion of applied researchers to take advantage of the desirable interpretive and statistical properties of pairwise comparison. Moreover, as we describe below, our algorithm also naturally extends to the distributed- and parallel-computing settings, allowing high-performance users to take advantage of their computational resources.

3.2 Algorithmic Improvements

The algorithmic improvements we present in this paper can be divided into two rough categories. First, we re-implement all major functions using the bigmemory, Rcpp, and parallel packages in R, allowing end users to easily parallelize and distribute model estima-

[8] Assuming $P = 67$, and using $N = 3,500$ as the practical cutoffs for a personal machine using the original *KRLS* implementation (assuming 8 GB available memory). If the working memory cutoff for a personal machine is lowered to 4 GB, the practical cutoffs for $P = 67$ for *KRLS* and *bigKRLS* are $N = 2,500$ and $N = 10,000$, respectively, yielding a similar 20% increase in estimable published datasets for *bigKRLS*.

Figure 1: Published sample sizes

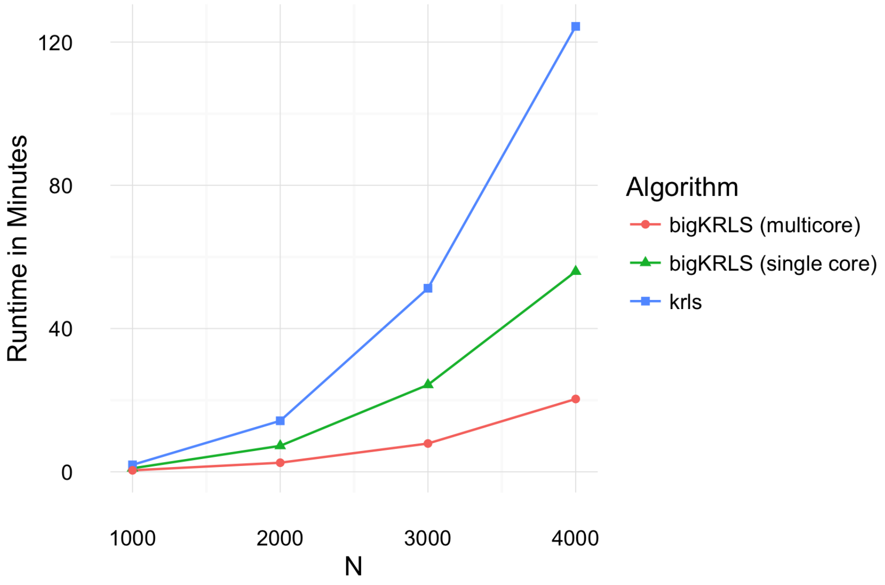


Log-scaled sample sizes for datasets used in AJPS and APSR articles published January 2015-January 2017, separated by observational and experimental datasets. Shaded areas represent sample sizes that *bigKRLS* can estimate on a personal computer that *KRLS* cannot.

tion. Second, we develop new first-differencing and kernel regularization algorithms. These changes, which we describe in detail in the following sections, provide substantial complexity reductions in all cases, but provide particularly large improvements for users with large numbers of discrete predictors.

Put together, these 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. 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*

Figure 2: bigKRLS vs. KRLS



Runtime when **X** is simulated to contain 10 continuous and 10 binary predictors. 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). On a single core, run times were virtually identical. For the '*bigKRLS* (multicore)' test, 14 of the server's cores were used.

in twenty minutes (Figure 2).

3.2.1 A Leaner First Differences Algorithm

For binary explanatory variables, KRLS estimates first differences.[9] 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 **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. Vertically concatenate 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 similarity kernel based on this concatenated matrix. This step is temporary, but has a memory footprint of $9N^2$ (!). Finally, save the two submatrices of the kernel corresponding to the respective counterfactual comparisons between $\mathbf{X}_{\{\mathbf{0}\}}$, $\mathbf{X}_{\{\mathbf{1}\}}$, and the observed data **X** (not the similarity of $\mathbf{X}_{\{\mathbf{0}\}}$ vs. $\mathbf{X}_{\{\mathbf{1}\}}$).

Our leaner implementation can also be expressed in terms of potential outcomes (Keele

[9] A nearly identical procedure is used for out-of-sample prediction given a pre-estimated model.

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.[10] 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 marginal effects 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 point estimate for first differences 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,[11] $\mathbf{K}_{new}$ 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:

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

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. This process facilitates re-expression wholly in terms of the observed kernel and the constant ϕ, as shown in Figure 3. As a result, our algorithm avoids constructing the costly temporary

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

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

Figure 3: Re-expressed kernel for first differences estimation.

$\mathbf{X}_{i,b}$	$\mathbf{X}_{j,b}$	$\mathbf{K}_{i,j}$	$\mathbf{K}_{\{1\},j}$	$\mathbf{K}_{\{0\},j}$	$\mathbf{K}_{\{1\},j} - \mathbf{K}_{\{0\},j}$
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 $j = 1, 2, \ldots\ N$. The first difference for observation i ($\hat{\delta}_{\mathbf{b,i}}$) is a coefficient-weighted average of the final column.

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 3 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 our example in §4.2, 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.

3.2.2 Lowering the Cost of Kernel Regularization

In the KRLS context, the regularization parameter λ is designed is to determine the appropriate degree of skepticism regarding outliers' impact on estimates of marginal effects (see Appendix A). Since the kernel's similarity scores range between 0 and 1 and $E(\mathbf{c}) = 0$, $\mathbf{c}'\mathbf{K}\mathbf{c}$ captures outliers weighted by their degree of similarity (perhaps to a hidden subpopulation). All model estimates ultimately depend on $\hat{\lambda}$. Though $\hat{\lambda}$ cannot be obtained analytically, it can be approximated with closed-form functions of the eigendecomposition of the kernel (Hainmueller and Hazlett 2013; Hastie et al. 2008).[12]

The key computation at each iteration depends on the kernel's eigendecomposition and our working hypothesis for λ (Rifkin and Lippert 2007). We are ultimately interested in

[12] 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 and hypotheses can be investigated in Reproducing Kernel Hilbert Space (very roughly, continuous functions can be analyzed even though observations are inevitably discrete in small enough spaces) (Beck and Ben-Tal 2006; Hastie et al. 2008; Rifkin and Lippert 2007).

$\hat{\mathbf{c}}^* = (\mathbf{K} + \lambda\mathbf{I})^{-\mathbf{1}}\mathbf{y}$. Let $\mathbf{G} = \mathbf{K} + \lambda\mathbf{I}$. We do not obtain $\mathbf{G}$ (or $\mathbf{G}^{-1}$) directly but rather substitute $\mathbf{Q\Lambda Q'}$ for the kernel, where $\mathbf{Q}$ is a matrix containing the eigenvectors of $\mathbf{K}$ and $\mathbf{\Lambda}$ is a diagonal matrix of eigenvalues. Using this formulation, we can express each element of $\mathbf{G}$ as:

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

The smaller the entries of $\mathbf{G}_{i,}^{-1}$ are, the closer $\mathbf{G}_{i,}^{-1}\mathbf{y}$ will be to y's midpoint, 0. Suppose the unit of observation is individual respondents. Since the eigenvectors may be positive or negative, $\mathbf{Q}_{i,k} * \mathbf{Q}_{j,k}$ ultimately captures whether respondent i and j exhibit similar pairwise correlations with all respondents. Since the kernel is symmetric and positive semi-definite, the eigenvalues are non-negative and serve to identify outlier points in the dataset. By construction, $\lambda > 0$. The higher λ is, the more skeptical we will ultimately be of coincidences in the independent and dependent variables in the entire dataset.

Though each entry of $\mathbf{G}^{-1}$ can be computed in linear time, the entire matrix is still $O(N^3)$ and so slows down quickly as N grows. Even with *Rcpp* and *bigmemory*, obtaining $\hat{\lambda}$ can easily take over half an hour on a typical laptop once $N > 7,500$, which often makes it the most time-consuming portion of the algorithm. *bigKRLS* performs this numeric search more efficiently because, unlike earlier approaches, it never solves for $\mathbf{G}^{-1}$ as the taxing cross product $\mathbf{Q}(\mathbf{\Lambda} + \lambda\mathbf{I})\mathbf{Q}'$. Instead of constructing the auxiliary $\mathbf{G}^{-1}$, *bigKRLS* updates the coefficients as it goes, saving only the bare minimum for subsequent error calculation. *bigKRLS* also takes advantage of $\mathbf{G}^{-1}$'s symmetry, halving the computational burden (see Appendix B for details). *bigKRLS*'s λ search runs anywhere from 40%-400% faster than the original implementation on personal computers.[13]

[13] Convergence takes 5-20 iterations. 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.

3.3 Degrees of Freedom for Average Marginal Effects

One key strength of KRLS is its ability to estimate "actual" marginal effects, which allow researchers to capture nuanced, potentially heterogeneous effects. However, to aid interpretability, a high-level summary of these individual estimates is often helpful. In their original paper, Hainmueller and Hazlett (2013) propose the average marginal effect (AME) as such a summary. They derive closed-form estimators for both the average marginal effect of a given variable and its variance, which are used for a Student's t-test with $N - P$ degrees of freedom to assess statistical significance. 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 C). 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:

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

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).[14] Intuitively, statistical tests derived using this quantity should take the initial loss of degrees of freedom incurred by estimating $\hat{c}$ into account, leading to $N_{effective} - P$ degrees of freedom for the t-tests.

We conducted a simulation using actual county-level data similar to that presented below for the 2016 presidential election but with artificial data generating processes (see Appendix C for details). We allow slopes to vary both hierarchically and with cubic polyno-

[14] This quantity is often expressed in terms of the squared singular values. In line with (Hainmueller and Hazlett 2013, 12), we focus on the here equivalent eigenvalues. Though eigenvalue calculations are computationally costly the kernel's spectral decomposition is already part of this regularization algorithm so calculating $N_{effective}$ takes no extra time.

mials at a range of sample sizes. The results suggest for a relatively modest loss in discovery (true positives decrease from 94.9% to 87.7%), the p-value correction markedly decreases in false positives (20.3% down to 3.1%). The reduction of false positives appears to be relatively independent of sample size. By contrast, the correction induces fewer and fewer false negatives as N grows. At $N = 1500$, the correction eliminates about six false positives for every false negative it adds. At least in this dataset, then, the p-value correction we propose improves inference across the board but optimal performance requires $N > 750$.

Since regularization involves an inherent bias-variance trade-off, we should not expect regularized confidence intervals generated using these quantities to perform perfectly. In particular, based on general shrinkage patterns induced by regularization we should expect most estimates to be biased downwards (in magnitude), reducing coverage in exchange for greater robustness and fewer false positive results. In our view, this trade off is worth it since it reliably avoids spurious statistical significance. However, to address potential shortcomings of this statistic, we encourage researchers to provide and discuss "actual" marginal effects as well as averages, in order to gain a fuller picture of the relationships present in their data.

4 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.

4.1 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 2016 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 last several decades, leading to an 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 previous Republican candidates.

To investigate this hypothesis in more detail, we used *bigKRLS* to fit a model of the 2016 presidential election that is based on the pairwise similarity of each county. Our dependent variable is the difference between two-party voting shares for Donald Trump in 2016 and Mitt Romney in 2012 ($\%Trump - \%Romney$). We focus on county-level data for data availability reasons.[15]

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, as well as communities in which public health crises emerged between election cycles. We also include standard racial, macroeconomic, and education variables (described in Appendix D). Each county's geolocation and state dummy variables further facilitate pairwise similarity measurement. Note that including state dummies would not have been possible without $bigKRLS$.[16]

[15] 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.

[16] 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 earlier implementation crashes with half this many predictors.

4.2 Average Effect Estimates

Average marginal effects (AME) estimates for this model are given in Figure 4. Unsurprisingly, the model fits the data, with a pseudo-R^2 of 0.83.[17] Nearly all predictors reach conventional levels of statistical significance, with intuitive signs. As predicted, Trump received a larger two-party vote share than Romney in higher-mortality counties. On average, Trump also performed better in whiter, older, poorer, and lower-education localities. 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. Δ Mortality is the main exception to this overall pattern of findings, 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 section 3.3, uncorrected p-values for KRLS average marginal effects are suspect for more complex data-generating processes. In Figure 4, 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 p-value correction we propose does not change any conclusions regarding statistical significance. However, in absolute terms, this correction increases the size of most p-values in the model noticeably, with a median increase of approximately an order of magnitude for the non-geographic covariates included in the model.[18]

[17] For comparison, a random forest fit to the same dataset produced an in-sample pseudo-R^2 of 0.81. Relative to a random forest, KRLS overfits the data slightly, with in-sample/out-of-sample MSEs of 4.37/5.69 compared with 5.51/5.02 (based on an 80-20 train/test split).

[18] To assess the stability of these results, we studied out-of-sample performance of these estimates on 100 five-fold cross-validated replicates of our original dataset (seep Appendix E for details). Mortality was statistically significant at the 0.05 level, with or without the p value correction, in all training sets. With the correction and the stricter cutoff of p = 0.005, the effect is still significant in 489 of 500 (97.8%) of training models. λ correlates with $N_{effective}$ at $r = 0.909$.

Figure 4: Average Marginal Effects 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. $N = 3,106$, $R^2 = 0.83$, pseudo-$R^2_{AME} = 0.31$.

4.3 Spatial Variation in Mortality Effect Estimates

While useful, inspecting the AME estimates conceals substantial effect heterogeneity: pseudo-R^2_{AME} is only 0.31 but $R^2 = 0.83$, suggesting that the majority of the explained variance is not explained by a linear combination of the X variables. Though mortality's AME is positive, the magnitude of its marginal effect varies substantially (Figure 5). Strikingly, mortality's effect is estimated to be at or near its strongest states like Pennsylvania and Michigan, which most forecasters rated a coin toss at best for Trump. In many Upper Midwest and Mountain West counties, a one-standard deviation increase in mortality produces approximately 0.5% increase in GOP presidential vote share; however, for counties in the South and Northeast, the predicted effect is substantially smaller. These results are consistent with Monnat (2016)'s findings, which suggest Trump's overperformance is broadly regional. Our analysis, however, also suggests a more local form of spatial dependence. In Kentucky–and some surrounding Appalachian areas–high mortality predicts Trump *under-*

performance.

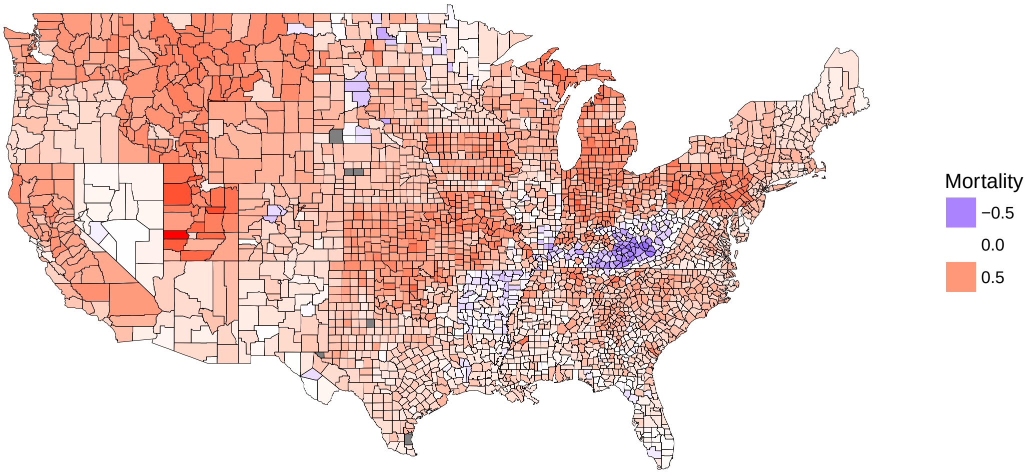
One policy-driven explanation suggested by Figure 5 relates to state-level Medicaid expansion decisions following the passage of the Affordable Care Act. Under the "communities in crisis" hypothesis, the primary causal mechanism is a local dissatisfaction with political elites, and particularly with 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 conventional elite politics. Figures 6 and 7 explore this explanation more directly by subsetting estimates by whether or not the state in which the county is contained chose to expand Medicaid, and find similarly provocative results. In Figure 7, in particular, we find that high-mortality counties in states that expanded Medicaid demonstrated a sharply negative relationship between mortality and Δ GOP Vote Share. High-mortality counties in other states, by contrast, generally retain a positive coefficient value.

We hasten to emphasize the speculative nature of this discussion, but we argue that these results are at least suggestive. The choice to expand Medicaid was not solely driven by partisanship, as Kentucky exemplifies. While viewing the results this way cannot, of course, distinguish between the Medicaid policy's effect on voter preference versus unmeasured local predispositions, there is a suggestion that policy context matters. Policy context may inform why homogeneous white counties behave so differently even after conditioning on observables. Based on our model, Trump appears to have been better positioned to win communities in crisis in states where access to Medicaid was not expanded.

4.4 Mortality Interactions

In addition to geographic heterogeneity, the "communities in crisis" hypothesis implies 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";

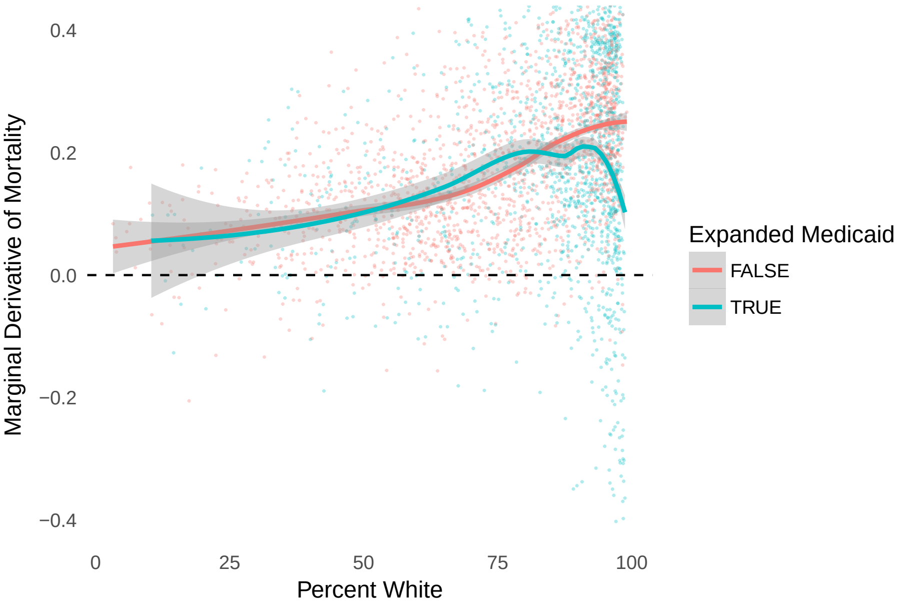
Figure 5: Predicted effect of a 1SD increase in all-purpose mortality (by county) on Δ GOP presidential vote share, 2012-16.



in other words, we should expect the effect of increasing mortality rates to be strongest in communities with larger white populations. As shown in Figure 6, the data weakly support this prediction, with $\geq 80 - 90\%$ white counties exhibiting the largest estimated effects of increasing mortality. Though much fainter, mortality's predicted effect is in the same direction in majority-minority counties. However, this effect is moderated in Medicaid-expansion states. Counties with 95% white residents or more experience most noticeable drop, with the predicted effect cut nearly in half for counties in Medicaid-expansion states.

Second, based on post-election commentary, we should also expect the marginal effect of mortality to be increasing. Marginal increases of 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 7, the estimated marginal effect of mortality actually peaks in mid-mortality counties and declines as mortality increases. In Medicaid-expansion counties, this relationship even reaches negative values in some of the highest-mortality localities. These results complicate the "communities in crisis" hypothesis substantially. Based on this model, true "crisis" communities (those with the highest mortality rates) appear to have been less supportive of Trump than their

Figure 6: Marginal effect of age-adjusted mortality on Δ GOP presidential vote share, 2012-16, by proportion of white population in each county.



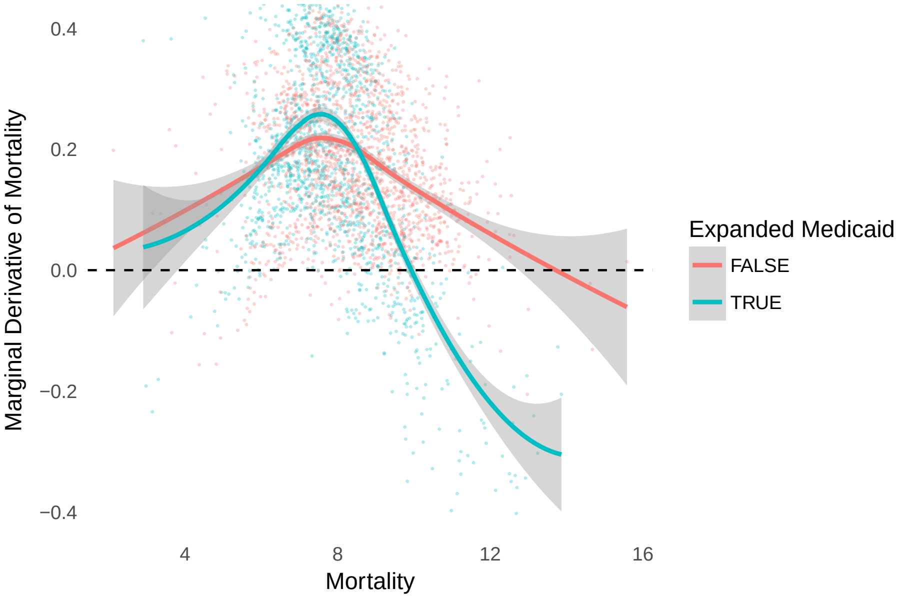
moderate-mortality counterparts, suggesting that the mortality effect is largely concentrated within the latter group of localities.

Apart from Florida, where there is little effect on Δ GOP Vote Share, the "communities in crisis" hypothesis is at least weakly supported in the vast majority of counties in battleground states. That at least raises the possibility that the mechanism by which crisis affects vote choice is priming and the ways issues are framed and presented during campaign season. However, apart from Nevada, large non-competitive swaths of the West appear similar.

5 Conclusion

In recent years, researchers have become increasingly interested in methods and models that combine the standard desirable mathematical properties with flexibility, robustness to violation of assumptions, and out-of-sample predictive accuracy. 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. KRLS offers one example of a method which

Figure 7: Marginal effect of age-adjusted mortality on Δ GOP presidential vote share, 2012-16, by mortality.



attempts to provide all of these characteristics, with the capacity to contribute to social science research at a number of stages. Unfortunately (and unsurprisingly), KRLS offers no free lunch. By attempting to maximize so many desirable properties, 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 approximately 75% and reduce memory consumption by approximately an order of magnitude.

There are number of exciting areas for future work. As the 2016 presidential election example illustrates, KRLS yields a rich set of nuanced findings but they may tempt researchers with false discoveries. Our proposed p-value correction for the model's average marginal effect estimator improves performance in this area, but selective inference techniques and further analytical work might offer opportunities for improvement (Taylor and Tibshrani 2015). Comparisons with non-parametric Bayesian schemes and other regularized regression approaches on real-world problems (such as the 2016 election data we examine in this paper) offers another avenue for future work. We particularly are interested in the extent to which the implications of KRLS findings correspond with those of appropriately-specified Bayesian models, especially for challenging, realistic data and research problems.

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.

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.

Gerjets, P., Scheiter, K., and Catrambone, R. (2004). Designing instructional examples to reduce intrinsic cognitive load: Molar versus modular presentation of solution procedures. *Instructional Science*, 32(1-2):33–58.

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

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*.

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.

James, G., Whitten, D., Hastie, T., and Tibshirani, R. (2013). *Introduction to Statistical Learning.* Springer, sixth edition.

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

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

Paas, F., Renkl, A., and Sweller, J. (2003). Cognitive load theory and instructional design: Recent developments. *Educational psychologist*, 38(1):1–4.

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. 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.

Sweller, J. (1994). Cognitive load theory, learning difficulty, and instructional design. *Learning and instruction*, 4(4):295–312.

Sweller, J. (2010). Element interactivity and intrinsic, extraneous, and germane cognitive load. *Educational psychology review*, 22(2):123–138.

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

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

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

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

A An Overview of KRLS

In this section, we provide a brief discussion of the KRLS model (see Figure 8 for an overview of major algorithm steps). This section presents key features of the method and model introduced in Hainmueller and Hazlett (2013), which interested readers may wish to consult for additional details.

Consider the model:

$$\mathbf{y}_i = \delta_{i,1}\mathbf{x}_{i,1} + \delta_{i,2}\mathbf{x}_{i,2} + \ ... \ + \delta_{i,P}\mathbf{x}_{i,P} + \epsilon_i$$

The model allows for the possibility that the marginal effects, $\delta_{i,1}$, $\delta_{i,2}$, and so on, may vary at each point (and so it has too many parameters for OLS). Nonetheless, KRLS can estimate this model. Under KRLS, we assume y is a function of the similarity of the P independent x variables. Consider two respondents, A and B, where $\mathbf{x}_A$ and $\mathbf{x}_B$ are respective vectors of standardized observables (age, ideology, etc.). The Gaussian kernel function is defined as:

$$k(\mathbf{x}_A, \mathbf{x}_B) = e^{-||\mathbf{x}_A - \mathbf{x}_B||^2/\sigma^2}$$

where $||\mathbf{x}_A - \mathbf{x}_B||$ denotes Euclidean distance. Once squared,

$$||\mathbf{x}_A - \mathbf{x}_B||^2 = (Age_A - Age_B)^2 + (Ideology_A - Ideology_B)^2 + ...$$

Intuitively, if $\mathbf{x}_A$ and $\mathbf{x}_B$ are identical, $||\mathbf{x}_A - \mathbf{x}_B||$ is 0 and so the similarity score $k(\mathbf{x}_A, \mathbf{x}_B) = 1$. The more dissimilar they are, the greater the distance between them is and so the smaller $k(\mathbf{x}_A, \mathbf{x}_B)$ becomes. Since the bandwidth σ^2 is chosen to be P, similarity decreases as the average distance across observable dimensions increases.

The pairwise version of the model we are interested in weights similarity:

$$\mathbf{y} = \mathbf{Kc}$$

Figure 8: 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, $\hat{\boldsymbol{\Delta}}_{\mathbf{N*P}} = [\hat{\delta}_{\mathbf{1}} \quad \hat{\delta}_{\mathbf{2}} \ldots \hat{\delta}_{\mathbf{P}}]$	$O(PN^3)$	$O(N^2)$

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).

[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.

To prevent overfitting, the model penalizes estimated weights such that $\hat{\mathbf{c}}^* = (\mathbf{K} + \lambda\mathbf{I})^{-\mathbf{1}}\mathbf{y}$, where λ is the regularization parameter chosen to minimize leave-one-out-error loss. This estimate results from a Tikhonov regularization problem:

$$\underset{f \in H}{\operatorname{argmin}} \sum_i^N (f(\mathbf{x}_i) - \mathbf{y}_i)^2 + \lambda ||f||_K^2$$

Like the kernel, the structural equation relies on a squared L_2 penalty, $||f||_K^2$. The minimization can be rewritten:

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

Including the kernel in the regularization ($\mathbf{c}'\mathbf{Kc}$) weighs outliers by similarity (see §3.2). Regarding the "actually" marginal effects, on each dimension the goal is to estimate $\hat{\delta}_{\mathbf{p}}$, an N x 1 vectors of the marginal effect of x_p at each observation. For continuous variables, if $\mathbf{D_p}$ contains pairwise simple distances, then $\hat{\delta}_{\mathbf{p}} = \frac{-2}{\sigma^2}\mathbf{D_p K}\hat{\mathbf{c}}^*$. The average marginal effect (AME) is the mean of this vector. The AME tends to coincide with slope estimates of typical linear regression models to extent the underlying data generating process is linear and additive. For the effect of binary variables, see §3.3.

Figure 9: "Actually" Marginal Effects

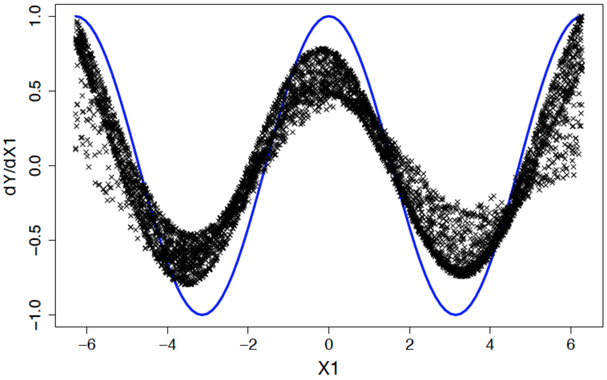


Figure: "Actually" Marginal Effects. The target function is $y = sin(x_1) + x_2 + N(0, 1)$; the derivative $\frac{\delta y}{\delta x_1} = cos(x_1)$ is shown blue. x_1 and x_2 have been drawn uniformly between -2π and 2π. No curves are modeled yet KRLS estimates the marginal effects well. Regularization is also apparent: $\approx$ 86% of $N = 5{,}000$ point estimates of the marginal effects are closer to 0 in magnitude than the true value.

B C++ Kernel Regularization Code

This section provides code the 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; List out(2);
  //  leave one out error loss
  double Le = 0;

  // 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(), N, 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);
  }

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

C Simulation of Average Marginal Effects

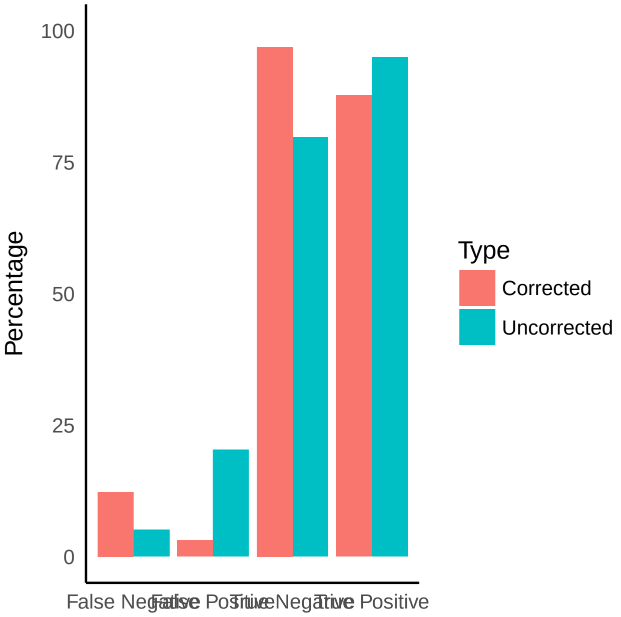
This Appendix describes and discusses the simulation introduced in Section 3.3. For each round of simulation, we took a random sample of US counties. We recorded the values of eight of the 2016 county-level covariates used in our applied example in Section 4: age-adjusted mortality, urban-rural continuum, age, income, unemployment rate, poverty rate, % college graduate, and % white. Each variable was randomly assigned to be either null or to have one of two types of effects: a linear effect which varied by hierarchical unit (here, US Census division), or a hierarchical effect as given previously with a cubic polynomial term included. Across all simulations, one-third of variables had a null effect, while two-thirds were non-zero. Of the variables with non-zero effects, half had a polynomial component. In sum, one third of the effects are null, one third were linear with a varying hierarchical component, and one third contained both hierarchical and polynomial terms.

Details on effect sizes were as follows (summarized in Figure 10). For the linear component, we simply simulated a small positive coefficient (distributed $\beta_j \sim (0.75, 1.25)$). The hierarchical component depends on US Census division. For each simulation, half of the census divisions were set to be constant and half to have a random slope disturbance (distributed $\gamma_{z[j]} \sim Uniform(-1, 1)$), where $z_{[i]}$ is an auxiliary matrix denoting the census division to which the i^{th} county belongs. Variables with a polynomial component were set to contain an additional cubic term cubic (distributed $\eta \sim Uniform(1, 3)$). Finally, all effects were standardized based on the standard deviation of the variable in question, in order to place effect sizes on the same scale.

Figure 10: Effect Summaries

Effect Type	Effect	Average Marginal Effect
Null	0	0
Hierarchical	$\beta_j x_{ij} + \gamma_{z[i]} x_{i1}$	$\beta_j + \frac{1}{n} \sum_i \gamma_{z[i]}$
Polynomial, Hierarchical	$\beta_j x_{ij} + \eta_j x_{ij}^3 + \gamma_{z[i]} x_{ij}$	$\beta_j + \frac{1}{n} \sum_i \left(3\eta_j x_{ij}^2 + \gamma_{z[i]}\right)$

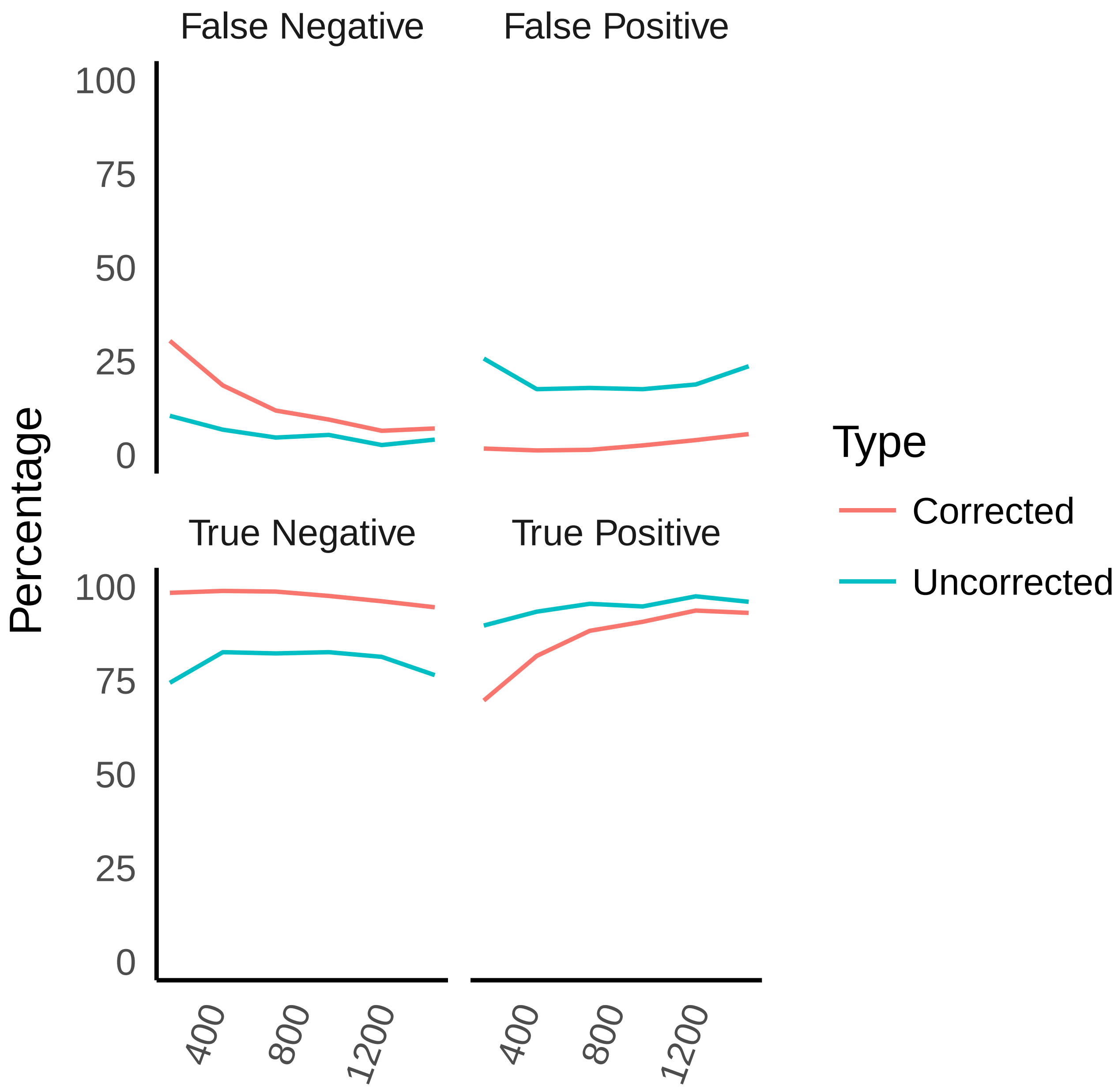
Figure 11: Simulated Accuracy of AMEs



Simulated Accuracy of Average Marginal Effects. A random subset of the county data was drawn and a new data generating process was simulated 200 times at each sample size ($N_{Sample} \in 250, 500, 750, 1000, 1250, 1500$). Eight AMEs are tested for each regression. The figure reflects a grand total of 9,600 AMEs, on each of which a two sided t-test was performed with and without the p value correction that is based on the effective degrees of freedom, $N_{effective} = N - N_{effective} = N - \sum_{k=1}^{N} \frac{\mathbf{\Lambda_{k,k}}}{\mathbf{\Lambda_{k,k}}+\lambda}$.

The results suggest for a relatively modest loss in discovery (true positives decrease from 94.9% to 87.7%), the p value correction there is a marked decrease in false positives (20.3% down to 3.1%). Put differently, averaging across sample sizes and effect types, the p value correction eliminates about 2.5 false positives for each false negative it adds. Breaking the results down by sample size suggests the p value reduces false positives relatively independent of sample size. Though the correction makes the estimates more conservative, the number of true positives increases (and the number of false positives decreases) with sample size. At $N = 1500$, the correction eliminates about six false positives for every false negative it adds. In summary, we simulated a complicated data generating processes such as the ones KRLS is intended for and found that the p value correction improves inference across sample sizes but optimal performance requires $N > 750$.

Figure 12: Simulated Accuracy of AMEs by Sample Size



D Descriptive Statistics for 2016 Election Study

Figure 13 gives summary statistics for the variables used in our applied example in Section 4. All race and education variables are given in percentage point units. Units for other variables are given in table notes. For the simulation study in Section 3.3, we use all predictor variables given in Figure 13 (i.e. all variables besides Δ GOP presidential vote share) to simulate a set of dependent variables, based on various error and effect specifications (as described in-text). Spatial errors and spatially correlated effects are generated using distances between counties, as measured in thousands of kilometers. For the applied example in Section 4, we also include latitude, longitude, and state dummy variables as additional predictors. The simulation study used in Section 3.3 employs eight region dummies instead to avoid zero variance x columns (US Census divisions e.g., "New England", "Middle Atlantic").

Figure 13: Descriptive statistics for Section 4.

Variable	Mean	SD	Source
Δ GOP presidential vote share, 2012-16[a]	5.86	5.26	Townhall
Mortality[b]	8.17	1.48	CDC
Δ Mortality[b]	-0.04	0.71	CDC
Urban-Rural Continuum[c]	4.98	2.70	USDA
Age[d]	4.03	0.50	US Census
Income[f]	4.85	1.23	USDA
Unemployment[e]	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

[a] The dependent variable is measured % Trump - % Romney via McGovern' s data.

[b] Mortality is used to measure the 'Communities in Crisis' hypothesis. 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.

c Ordinal variable, ranging from 1 (most urban) to 7 (most rural).

d Average; measured in 10s of years.

e Median household income (in 10,000s).

f Unemployment–and all variables that appear below it–are county-level percentages.

E Crossvalidation Results

To assess the stability of the regression estimates, we estimated 100 five-fold cross-validation replicates, and averaged out-of-sample performance across each replicate. 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. The estimates are remarkably 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. 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). That said, the AMEs do remarkably well out of sample, capturing over two third of the variance that the less parsimonious coefficients do.

Figure 14: Overview of Crossvalidation Results

	In Sample	Out of Sample
MSE	4.755	5.832
	(4.739, 4.770)	(5.747, 5.917)
MSE_{AME}	97.758	98.270
	(94.874, 100.643)	(95.437, 101.105)
R^2	0.828	0.791
	(0.826, 0.829)	(0.789, 0.794)
R^2_{AME}	0.089	0.550
	(0.087, 0.091)	(0.542, 0.557)

Results of 100 five-fold cross-validation replicates (N_{train} = 80% = 2,485 observations; N_{test} = 20% = 621 observations). Average measures of fit provided along with 95% confidence intervals. 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.