Messy Data, Robust Inference?
Navigating Obstacles to Inference with bigKRLS

Pete Mohanty
pmohanty@stanford.edu

Robert Shaffer
rbshaffer@utexas.edu

April 1st, 2017

Abstract

Complex models are of increasing interest to social scientists. Flexible Bayesian approaches (e.g., infinite mixture models) often improve fit over conventional solutions, while causally-oriented studies must often incorporate complicated treatment structures or confounding relationships. Unfortunately complex models and their estimators 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 trade-offs 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 at least an order of magnitude. And though *bigKRLS* offers parallel processing, even on single processor, our algorithm decreases runtime by up to 50%. With political behavior examples, we show *bigKRLS* helps navigate obstacles to inference like treatment heterogeneity and unmodeled interactions. These applications would be difficult or impossible to estimate with the original implementation, but are straightforward with *bigKRLS*.

This project has already benefited immensely from feedback at the *Bay Area R Users Group* Official Meet Ups, hosted by Treasure Data in May 2016 and Santa Clara University in October 2016, *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, and the Stevens Institute of Technology's College of Liberal Arts on February 27th, 2017. Thanks in particular to Joseph Rickert, Susan Holmes, Christian Fong, Luke Sonnet, Chad Hazlett, Kristyn Karl, and Jacob Berman for additional suggestions, comments, and advice. Pete Mohanty would like to thank 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 the social science research community. As ongoing election coverage from data-oriented sites constantly 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 (e.g., forecasting terrorist events in Afghanistan using geolocation data), 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 helpful, allowing researchers to check assumptions and guard against overfitting more easily.

We view interpretability as a prerequisite of causal inference. A model that does not let researchers clearly describe a given data set cannot the next step: casually identifying x's effect on y. Decision trees are a canonical example of an approach that excels in prediction at the cost of interpretability, both in the sense that they cannot easily estimate causal effects and in a more informal, descriptive sense. As we argue later, models that are sparse, parsimonious, and directly estimate quantities of interest (e.g. causal effects) are generally more interpretable than those that do not.

Interpretability can also be viewed as complementary to traditional modeling goals such as robustness and flexibility.[1] Without robust estimators, the flexibility that allows models to consider diverse possibilities falls prey to false discoveries. Without interpretability, humans cannot easily extract relevant information from the estimated model. Unfortunately, of course, none of these traits imply any of the others. Scalability can be thought of as the ability to estimate large, complex models without added costs, whether primarily born in terms of hardware costs, run time, or labor time. Models that press against the "complexity frontier" can make choices between robustness, flexibility, and interpretability particularly stark.

In this paper, we illustrate this phenomenon through an extended complexity analysis and optimization exercise centered on Kernel-Regularized Least Squares (KRLS) (Hainmueller and Hazlett 2013). Compared with other approaches, KRLS offers a desirable balance of interpretability, flexibility, and theoretical properties. Unfortunately (and unsurprisingly), pairwise regression is also substantially more complex than many competing techniques. Here, we introduce bigKRLS, a re-implemented version of the original R package with algorithmic and implementation improvements designed to optimize speed and memory usage. These improvements allow users to straightforwardly fit models via KRLS to larger datasets (N > 2,500), which we illustrate through two applied examples.[2] Specifically, we explore treatment heterogeneity in a voter turnout experiment (Gerber et al. 2008; Green et al. 2009; Gelman and Zelizer 2015), and county-level voting patterns in the 2016 Presidential election).

[1] Like hierarchical models and random forests, the pairwise models that KRLS estimates make a 'variance-bias' trade off to flexibly capture the diversity of the data generating process.

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

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 *interpretable*. 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. For example, simple linear regression with N observations and P covariates has complexity $O(P^2N)$ (since calculating $\mathbf{X}'\mathbf{X}$ dominates other calculations involved in generating $\hat{\beta}$).[4] Since N is usually much larger than P, simple linear regression generally has complexity that is 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 violations of key assumptions 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, robust to violations of assumptions, and often possess excellent out-

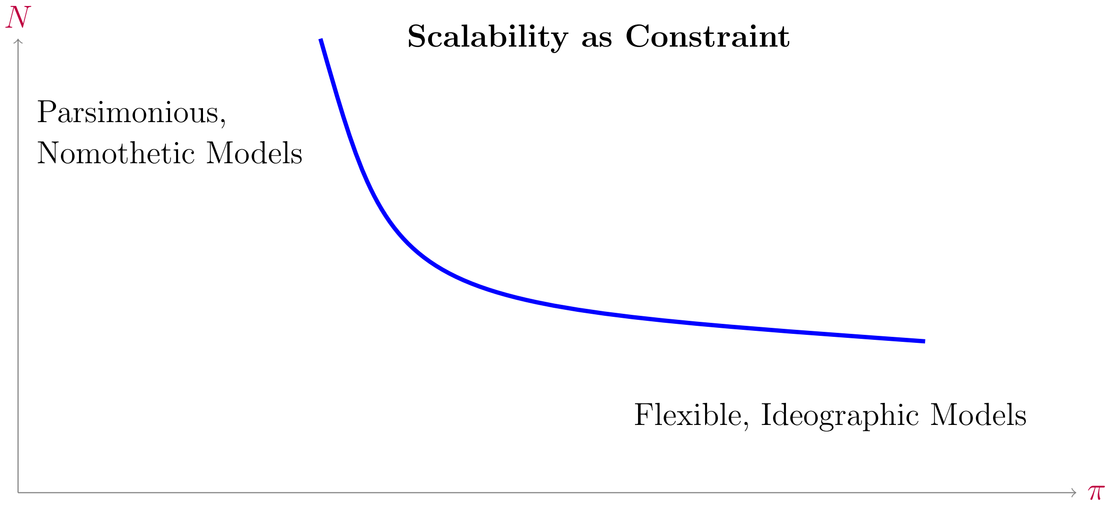
[3] Arguably, we might view Bayesian posterior probabilities as a good example of an "interpretable" procedure. As Gill (1999), Jackman (2009) and others argue, the frequentist null hypothesis testing paradigm is remarkably difficult to properly interpret. By contrast, researchers can straightforwardly calculate probabilities of interest such as $P(\beta > 0|X)$ under the Bayesian paradigm without reference to counterfactuals.

That said, many users find usage of priors in the Bayesian paradigm confusing (or arbitrary) compared with the prior-free frequentist approach. To a certain extent, then, the relative interpretability of the two paradigms depends on whether one locates the primary interpretive dilemma at the beginning or the end of the analysis, as well as the research question at hand. 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] Assuming N 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.

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 $O(Nlog(N)^2) + O(PNlog(N))$.[5] 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.

Figure 1: Computational Complexity Frontier



For any given model with N observations and π parameters, there is a computational cost; models above the line cannot be estimated even if they are statistically identified. Either hardware improvements (faster CPU, more RAM, etc.) or software improvements (better system architecture, more efficient algorithms) can shift the curve to the upper right. “Nomothetic” captures the drive for general, “law like” findings, while “ideographic” captures the drive for nuanced, complete description of complex political phenomena (Wallerstein 2000). The advent of distributed computing has enabled certain “big data” machine learning approaches like neural nets to estimate models where both N and π are massive (Hodge et al. 2016) but, because of comprises in interpretability, the social science research applicability remains to be seen.

Models that attempt to optimize all of these features simultaneously quickly encounter what we might call the *computational complexity frontier*, depicted in Figure 1. 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) offer one obvious example of this relationship, but many modeling approaches exhibit this relationship.

This 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

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

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 when the speed and memory gains are large.

3 bigKRLS as Case Study in Optimization

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 (not just average marginal effects; see 2), researchers can easily determine whether a treatment effect appears to be constant across the sample or assess whether or not the outcome is monotonic function of the treatment.

Predictably, however, KRLS is also computationally demanding. The source of the model's nuance–pairwise comparison–makes the linear algebra unusually memory intensive (Appendix 1). KRLS contains seven major steps, which we document (in somewhat simplified form) in Table 4. Compared with many workhorse methods, KRLS requires substantially greater resources to fully estimate, with total runtime complexity $O(N^3)$. By contrast, as noted in the previous section decision trees have complexity $O(Nlog(N)^2) + O(PNlog(N))$.

Figure 2: "Actually" Marginal Effects

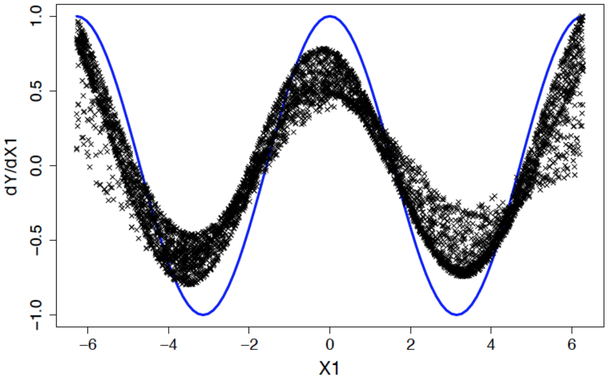
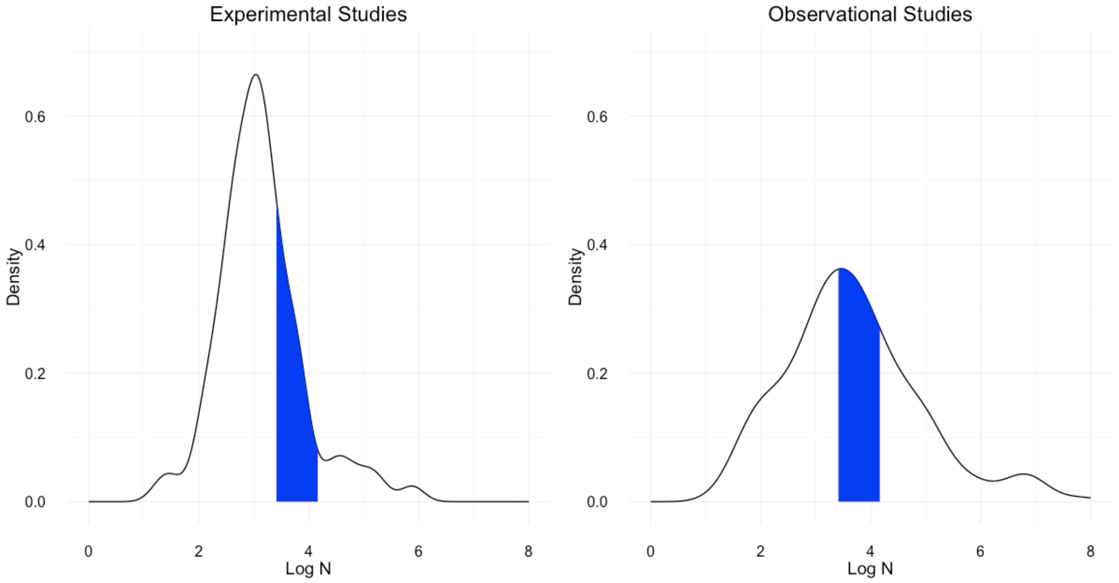


Figure: "Actually" Marginal Effects. The target function is $y = sin(x_1) + x_2 + N(0, 1)$; the derivative $\frac{\delta y}{\delta x_1} = cos(x_1)$ is shown blue. x_1 and x_2 have been drawn uniformly between -2π and 2π. No curves are modeled yet KRLS estimates the marginal effects well. Regularization 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.

Figure 3: Common log of sample sizes for datasets used in AJPS and APSR articles published January 2015-January 2017. The blue highlighted area represents sample sizes that *bigKRLS* can estimate on a personal computer that *KRLS* cannot.



Memory requirements for KRLS are similarly restrictive. In our optimized implementation, at peak runtime the 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.[6] 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 at least 200 MB of working memory plus any overhead for the underlying data structure. On a personal machine, then, estimating the full model on a dataset larger than $N \approx 15,000$ (1.8 GB each) is likely impractical.

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-January 2017, and recorded sample sizes for each dataset used in those articles ($N = 279$). As shown in Figure 3, approximately 48% of datasets are too large for the original (base R) *KRLS* (at least without substantial restrictions on the RHS). The *bigKRLS* improvements we present do not (and cannot) solve these problems entirely, but raise the cutoff for a personal machine to approximately $N = 15,000$, covering an additional 25% of datasets.[7]

[6] Even when P is small, *bigKRLS*'s peak memory usage is lower since it is $\approx 5N^2$ compared with $\approx (P+7) * N^2$ plus an additional $9N^2$ 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 big matrices the moment are no longer needed.

[7] 19% of experimental and 28% of observational datasets fall into this range. A handful of datasets were coded as too large for *bigKRLS* because researchers reported several hundred thousand country-year dyads though *bigKRLS* could likely estimate such models (for example, if a study involves 175 countries over 50 years, we take N to mean 175 * 50 = 8,750, not the larger pairwise figure).

Figure 4: 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 local derivatives, $\hat{\mathbf{\Delta}}_{\mathbf{N*P}} = [\hat{\delta}_{\mathbf{1}} \quad \hat{\delta}_{\mathbf{2}} \dots \hat{\delta}_{\mathbf{P}}]$	$O(PN^3)$	$O(N^2)$

Letting i, j index observations such that $i, j = 1, 2 \dots N$ ultimately captures all pairs and letting $p = 1, 2, \dots P$ 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.

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 the desirable interpretive and statistical properties of pairwise comparison.

3.2 Major Updates of bigKRLS

1. Leaner algorithm. Since KRLS is inherently memory-intensive, memory savings are important even for small- and medium-sized applications. We develop and implement a new first differences algorithm which, in tandem with an overall re-design, reduces peak memory usage by at least an order of magnitude (§3.4). We also cut the number of computations required for the numeric search for the regularization parameter, λ in half (§3.5).

2. Improved memory management. Many data objects in R perform poorly in

memory-intensive applications. We use a series of packages in the bigmemory environment to ease this constraint, allowing our implementation to handle larger datasets more smoothly.

3. Parallel processing. Since the marginal effects of each variable can be estimated independently, these calculations can be easily parallelized. *bigKRLS* uses snow (Simple Network Of Workstations) to take advantage of these speed gains.
4. Interactive data visualization. *bigKRLS* provides an easy-to-use *Shiny* visualization app (Appendix 2), allowing researchers to share results with collaborators and easily publish findings online for more general audiences.

Put together, these improvements offer a substantial reduction in peak memory usage, bringing peak memory consumption from $O((P+16)N^2)$ 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 for datasets 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* in twenty minutes (Figure 5).

Figure 5: bigKRLS vs. KRLS

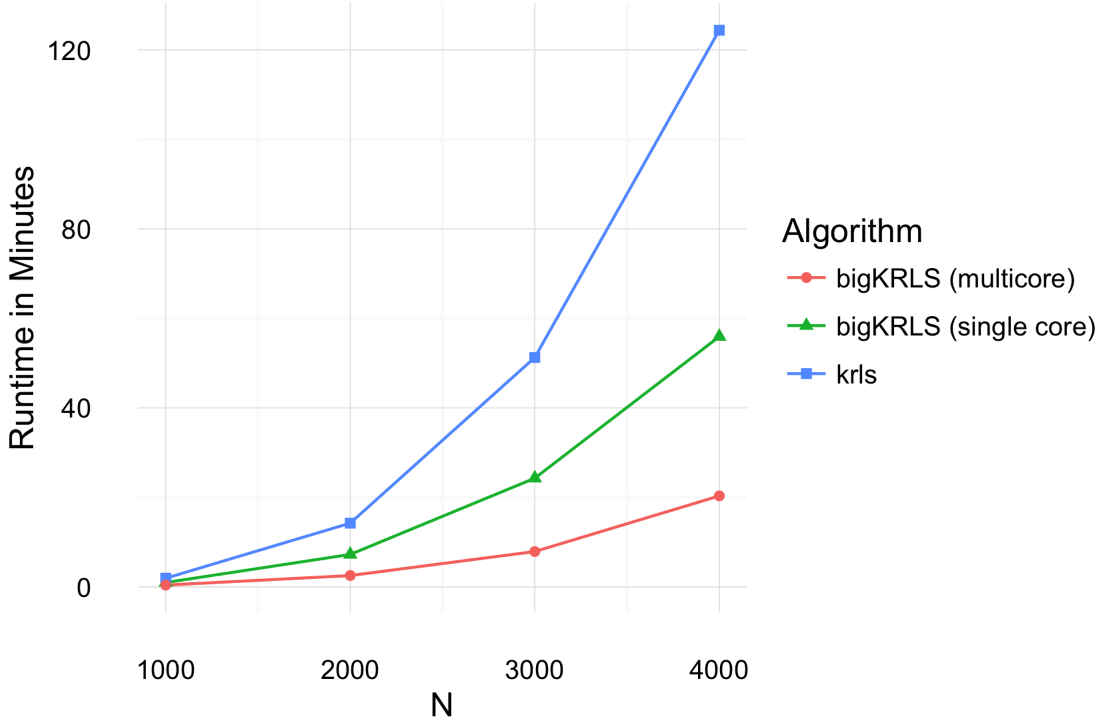


Figure: Runtime when $\mathbf{X}$ is simulated to contain 10 continuous and 10 binary predictors. Two computers were used, a laptop (mid 2012 MacBook Pro with 8 gigabytes of RAM) and a server (Xeon E5-2650 with 126 gigabytes of RAM). On a single core, run times were virtually identical. For the '*bigKRLS* (multicore)' test, 14 of the server's cores were used.

3.2.1 A Leaner First Differences Algorithm

For binary explanatory variables, KRLS estimates first differences.[8] 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 $\mathbf{X}$, $\mathbf{X}_{\{\mathbf{0}\}}$ (where $\mathbf{X}_b = 0$) and $\mathbf{X}_{\{\mathbf{1}\}}$ (where $\mathbf{X}_b = 1$). Compute a new kernel based on $\mathbf{X}_{new} = [\mathbf{X} \,|\, \mathbf{X}_{\{\mathbf{0}\}} \,|\, \mathbf{X}_{\{\mathbf{1}\}}]$. 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 $\mathbf{X}$ (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 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.[9] The first differences are:

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

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

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

As with for 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:

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

where $\mathbf{h}$ is a vector of constants,[10] $\mathbf{K}_{new}$ 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 $6N^2$: $2N^2$ for $\mathbf{K}_{new}$ and another $4N^2$ 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:

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

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 6.

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:

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

   Each (sub)matrix in the final term functions as a weight on the observed variances and covariances in the various counterfactual scenarios.

2. Though $\mathbf{h}$ is just 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

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

Figure 6: 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 $i\,(\hat{\delta}_{\mathbf{b},\mathbf{i}})$ is a coefficient-weighted average of the final column.

expanded version of Figure 3 that expresses all possible products of two counterfactual similarity scores, we are able to 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. The Boolean 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 2 $N \times N$ 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 the "Boolean" 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* once N crosses 2,000. 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. 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).[11]

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 $\hat{\mathbf{c}}^* = (\mathbf{K} + \lambda\mathbf{I})^{-\mathbf{1}}\mathbf{y}$. 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 $\mathbf{Q}$ contains the eigenvectors and $\mathbf{\Lambda}$ contains the eigenvalues on its diagonal (and zero otherwise).

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

The smaller 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 here identify individual outliers. 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 (Appendix 2). *bigKRLS*'s λ search runs anywhere from 40%-400% faster than the original implementation on personal computers.[12]

[11] 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).

[12] 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.

4 Applications

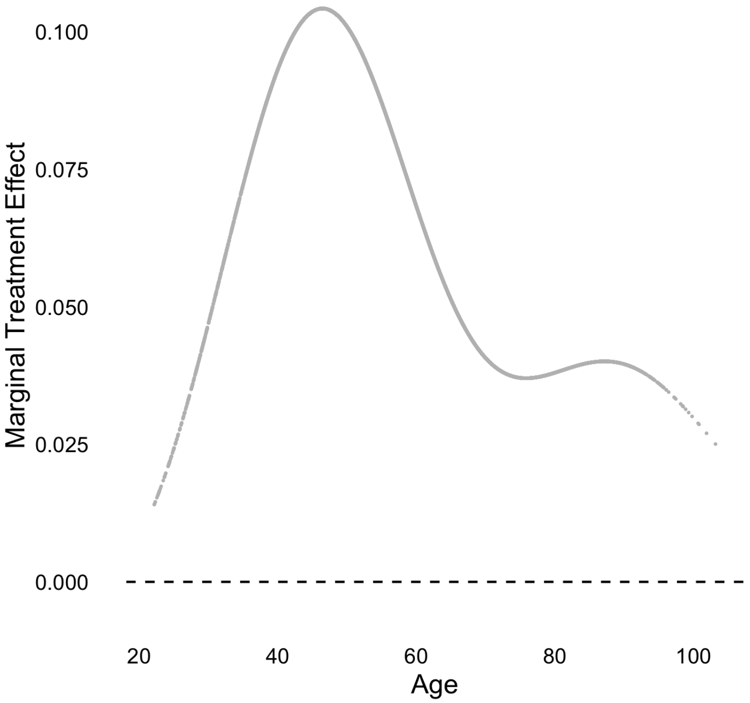
In this section, we re-analyze a voter turnout experiment that was conducted in 2006 and a recent Pew Research Survey on the 2016 Presidential Election. Both of these models involve datasets that would be too large to handle without the implementation and algorithmic improvements described above. The latter case, in particular, highlights one key strength of *bigKRLS*: gracefully handling binary covariates. Because we include voter state of residence as a predictor, our resulting model contains more than 50 binary predictors. 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.

Both datasets also highlight the need both for flexibility and for interpretability in complex modeling applications. Neither model assumes that y is normally distributed or that the data generating process reflects a particular functional form (e.g., probit in §4.1 and beta or truncated normal in §4.2), as would be required by a generalized linear model (GLM). We also do not specify a hierarchical structure, as would be particularly relevant in §4.2. We do not of course suggest that such functional form assumptions are irrelevant or incorrect, but rather that implementing these assumptions often involves trade-offs (whether computationally, in the availability of closed-form results, or in the overall tenability of the model). We briefly compare and contrast with plausible alternatives after giving our results.

4.1 Diagnosing Treatment Effect Heterogeneity

Gerber et al. (2008) conducted a field experiment based on 180,000 registered Michigan voters in 2006 to assess whether social pressure increases voter turnout. 5,074 respondents received the "neighbors" treatment, with 24,964 in the control group and the remaining respondents assigned to other treatments. The "neighbors" treatment consisted of a postcard which asked "what if your neighbors knew whether you voted?" followed by a list of the individuals in the neighborhood who voted in August and November 2004, which produced a positive effect on turnout. In a follow up study, Green et al. (2009) used their experimental finding as a benchmark for a study on polynomial regression. In particular, Green et. al include a fourth order polynomial on age and corresponding interactions with the treatment. In general, researchers are skeptical of higher-order polynomial terms in regression models; as a result, we might wonder whether we can replicate these findings without specifying a particular polynomial structure (Figure 7).

Figure 7: The Neighbors Treatment: Scatter Plot of Estimated Marginal Effects



	Average Marginal Effect	SE	t	p
Neighbor's Treatment	0.0735	0.0131	5.62	< 0.0001
Age (in Years)	0.001	0.0002	4.756	< 0.0001

$N = 10{,}000$, R^2: 0.0125, R^2_{AME}: 0.0059

The estimates suggest an average treatment effect of just over 7% increase in turnout; OLS, by contrast, estimates closer to 10%.[13] Though the treatment does at least appear monotonic (no one seems to have been offended enough to stay home because of the postcard), the effect is negligible on younger voters and quite modest on older ones.[14] The model fits (relatively) poorly for younger voters. Perhaps the effect of the treatment is conditional on paying attention to the mail (see Appendix 2).

[13] Obtaining 30,000 eigenvectors is non-trivial; here we analyze a simple random sample of 10,000 using the code found in Appendix 2 with seed 2017. Using seed 2016, the Treatment's AME is 0.0730 (as opposed to 0.0735) and the treatment effect plot is strikingly similar as well (Figure 7). The OLS estimate of the treatment effect is 9.79% in both samples as well as the full sample.

[14] Potential outcomes may or may not be monotonic functions of the treatment; on average, penicillin improves the health of those with bacterial infections but nevertheless harms the allergic.

4.2 Analyzing Interactions in the 2016 Presidential Election

As a further illustration, we use *bigKRLS* to examine voting patterns in the 2016 presidential election. In particular, we focus on what we might term the "communities in crisis" hypothesis. In both popular and academic discussions (e.g. Guo 2016; Siegel 2016; Monnat 2016), a number of commentators argued that Trump's success was partly attributable to his appeal in "collapsing" communities. 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 1000 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 Figure 8). 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]

Average marginal effects (AME) estimates for this model are given in Figure 9. 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 "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

[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.

[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).

Figure 8: Descriptive statistics for Section 4.3.

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.

Figure 9: Why did Trump Outperform Romney?

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

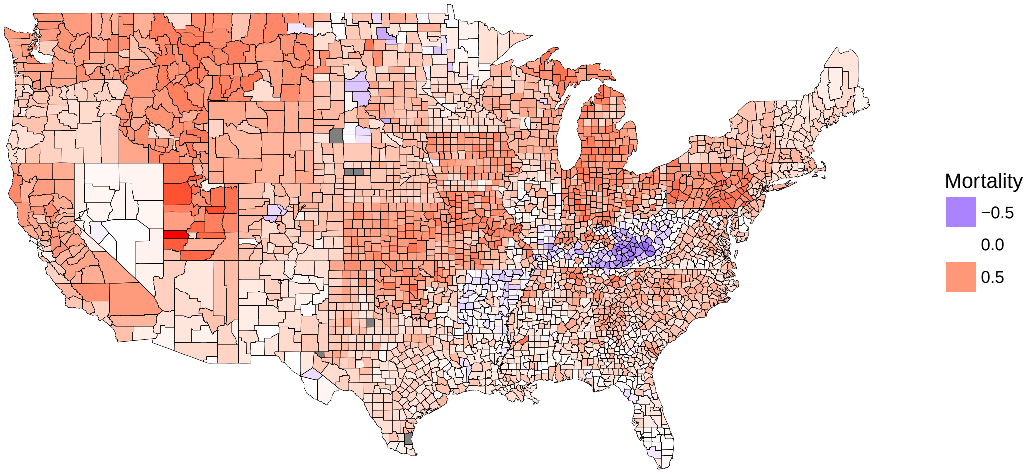
Average Marginal Effects (AMEs) with standard errors, t-values, and two-sided p-values (*bigKRLS* estimates). $N = 3,106$, $R^2 = 0.83$, pseudo-$R^2_{AME} = 0.31$. Estimates for latitude, longitude, and state omitted for brevity. The dependent variable is change in GOP vote share in the Presidential Election, 2012-2016, measured in percentages. For other variable definitions, see Figure 8.

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.

While useful, inspecting the average marginal effects conceals substantial effect heterogeneity: pseudo-R^2_{AME} is only 0.31 but $R^2 = 0.83$ meaning 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 10). 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 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 10 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

Figure 10: Predicted effect of a 1SD increase in all-purpose mortality (by county) on Δ GOP Presidential vote share, 2012-16.

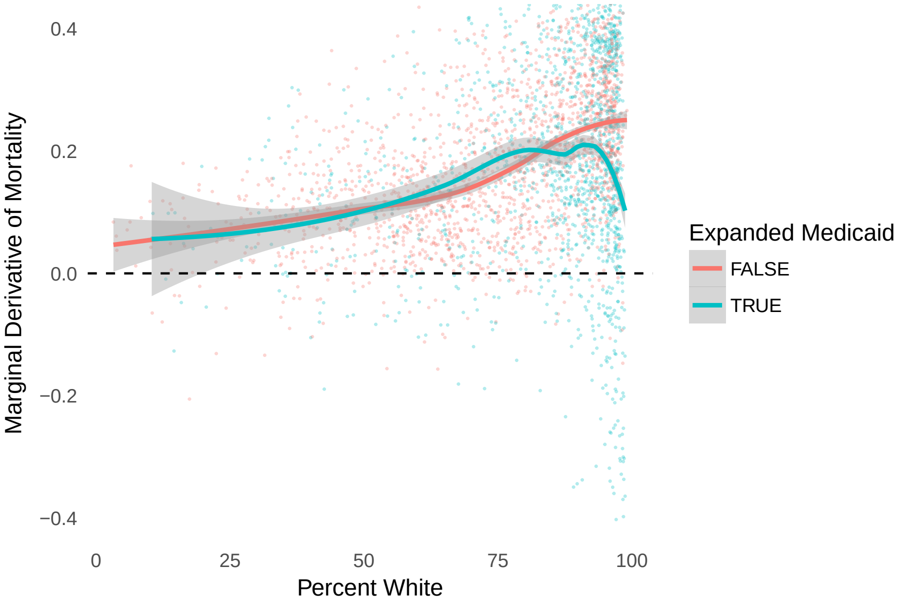


chose to expand Medicaid following the passage of the Affordable Care Act, high-mortality counties likely received a substantial portion of new Medicaid spending, which may have buttressed their faith in conventional elite politics. Figures 11 and 12 explore this explanation more directly by subsetting estimates by whether or not the state in which the county is contained chose to expand Medicaid, and find similarly provocative results. In Figure 12, 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 vs. 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 stricken communities in states where access to Medicaid was not expanded.

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 was 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. As shown in Figure 11, 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

Figure 11: Marginal effect of age-adjusted mortality on Δ GOP Presidential vote share, 2012-16, by proportion of white population in each county.

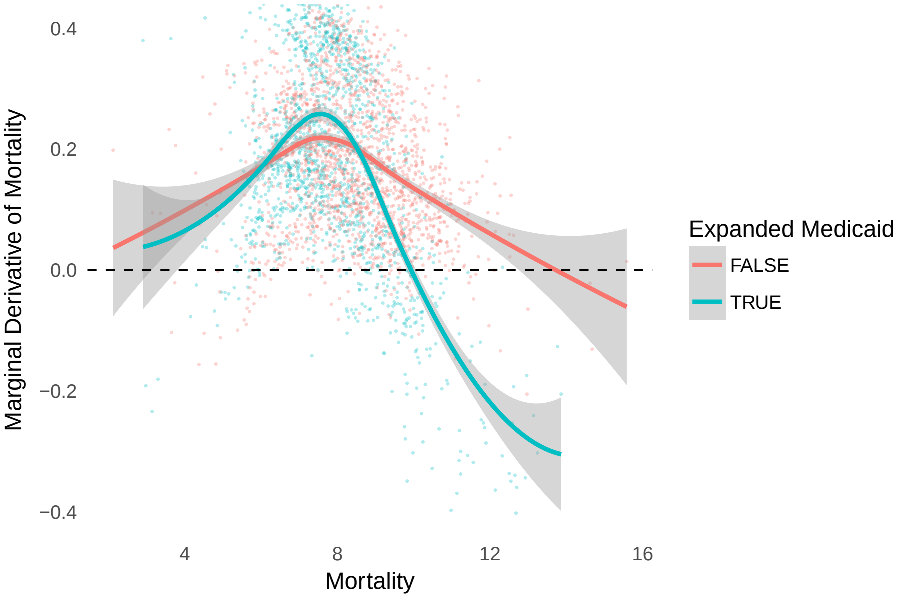


effect is moderated in Medicaid-expansion counties.

Second, based on post-election commentary, we would also likely 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 12, 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 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 battleground-state counties. 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.

Figure 12: Marginal effect of age-adjusted mortality on Δ GOP Presidential vote share, 2012-16, by mortality.



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 attempts to provide all of these characteristics, with the capacity to contribute to social science research at a number of stages. As the 2016 Presidential election example illustrates, KRLS yields a rich set of nuanced findings but they may tempt researchers with false discoveries. Selective inference may help further streamline exploratory and confirmatory analysis (Taylor and Tibshrani 2015).

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. In doing so, we aim to allow users to expand this particular model to a variety of important use cases, and to highlight the importance of optimization work in highly complex modeling scenarios.

There are number of exciting areas for future work. All statistical approaches are keen to demonstrate that their results are independent of the idiosyncrasies of the research process. Bayesian MCMC and non-parametric approaches often endeavor to do this in different ways. Sampling (particularly with uninformative priors) reflects a commitment to consider improbable possibilities. Regularized regression, by contrast, allows the research to include a large number of variables but makes each hypothesis test quite conservative. We are interested in the extent to which the implications of KRLS findings correspond with those of appropriately-specified Bayesian models, particularly for unusually challenging data.

6 Appendix 1: An Overview of KRLS

Variation in y is assumed to be some function of the similarity of the corresponding independent variables, observable in X. 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, the number of independent variables, similarity decreases as the average distance across observable dimensions increases.

The pairwise model we are interested in weights similarity:

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

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:

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

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

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

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 $\hat{\delta}_{\mathbf{p}} = \frac{-2}{\sigma^2}\mathbf{D_p}\mathbf{K}\hat{\mathbf{c}}^*$. 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.

7 Appendix 2: R Code for Neighbors Experiment

This section contains *R* code to replicate the analysis of the "neighbors" experiment on a simple random sample of the data ($N < 15,000$ recommended for personal computers). This code also shows how to make use of *bigKRLS*'s built-in save feature, which automatically handles the storage of big matrices contained in the output by writing them to a new folder "myresults". Next, the code calls *shiny.bigKRLS* which allows researchers to interact with the results in *RStudio* or their web browser.

```
load("Green_et_al_polanalysis_2009_BW5.RData")

set.seed(2017)
include <- sample(nrow(data1), 10000, replace=F)

y <- as.matrix(data1$voted[include])
X <- as.matrix(cbind(data1$treatmen,
                     data1$ageatelection + 55))[include,]

neighbors.out <- bigKRLS(y, X, model_subfolder_name = "myresults")

shiny.bigKRLS(neighbors.out,
          xlabs = c("Treatment", "Age"),
          main.label = "Marginal␣Effect␣of␣'Neighbors'␣Treatment")
```

8 Appendix 3: 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;
}
```

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.

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

Gelman, A. and Zelizer, A. (2015). Evidence on the deleterious impact of sustained use of polynomial regression on causal inference. *Research & Politics*, January-March:1–7.

Gerber, A. S., Green, D. P., and Larimer, C. W. (2008). Social pressure and voter turnout: Evidence from a largescale field experiment. *American Political Science Review*, 102:33–46.

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.

Green, D. P., Long, T. Y., Kern, H. L., Gerber, A. S., and Larimer, C. L. (2009). Testing the accuracy of regression discontinuity analysis using experimental benchmarks. *Political Analysis*, 17:400–17.

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.*

Hodge, V. J., O'Keefe, S., and Austin, J. (2016). Hadoop neural network for parallel and distributed feature selection. *Neural Networks*, 22:24–35.

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.

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.

Wallerstein, I. (2000). *Hold the Tiller Firm: On the Method of the Unit of Analysis.* The New Press, New York.

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.