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

Pete Mohanty
pmohanty@stanford.edu

Robert B. Shaffer
rbshaffer@utexas.edu

International Methods Colloquium
Friday November 11th, 2016
Watch Live @ Noon Eastern

Working Paper, Kindly Do Not Cite without Permission

Abstract

Complex models are of increasing interest to social scientists. Flexible Bayesian models (e.g., infinite mixture models) often improve fit over conventional solutions; causal models must often reflect 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 regressions, encounters substantial scalability challenges. Our improvements decrease runtime by up to 50% and reduce peak memory usage by approximately an order of magnitude. With political behavior examples, we show *bigKRLS* helps navigate obstacles to inference like treatment effect heterogeneity or, in the observational setting, 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 comments and suggestions at the *Bay Area R Users Group* Official Meet Ups, hosted by Treasure Data in May and Santa Clara University in October, *useR!* 2016, hosted by Stanford University, and the *American Political Science Association* 2016 Annual Meeting in Philadelphia.

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 a helpful, allowing researchers to check assumptions and guard against overfitting more easily.

In general, we view interpretabality as a prerequisite of causal inference. A model that fails to describe what happened in a given data set cannot take 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. 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 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 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), the model is also substantially more complex than many competing techniques. Here, we introduce bigKRLS, a re-implemented version of the original R package with KRLS R package with algorithmic and implementation improvements designed to optimize speed and memory usage Mohanty and Shaffer (2016). These improvements allow users to straightforwardly fit models via KRLS to larger datasets (N > 2,500), which we illustrate through two applied examples (specifically, detecting treatment heterogeneity in a voter turnout experiment (Gerber et al. 2008; Green et al. 2009; Gelman and Zelizer 2015), and unit-based non-linear effects in 2016 two-party polling data).

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 generally skeptical of these kinds of assumptions. For applied work, then, we might also want our estimator of interest 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 usually 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 that estimate relatively 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 models that make causal inferences as well as simple predictions. Models 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.[1]

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 $\hat{\beta}$)[2]. Since N is usually much larger than P, simple linear regression generally has complexity that is nearly linear with the number of observations.

Compared with other approaches, simple linear regression directly calculates causally interpretable effects (under appropriate assumptions), 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

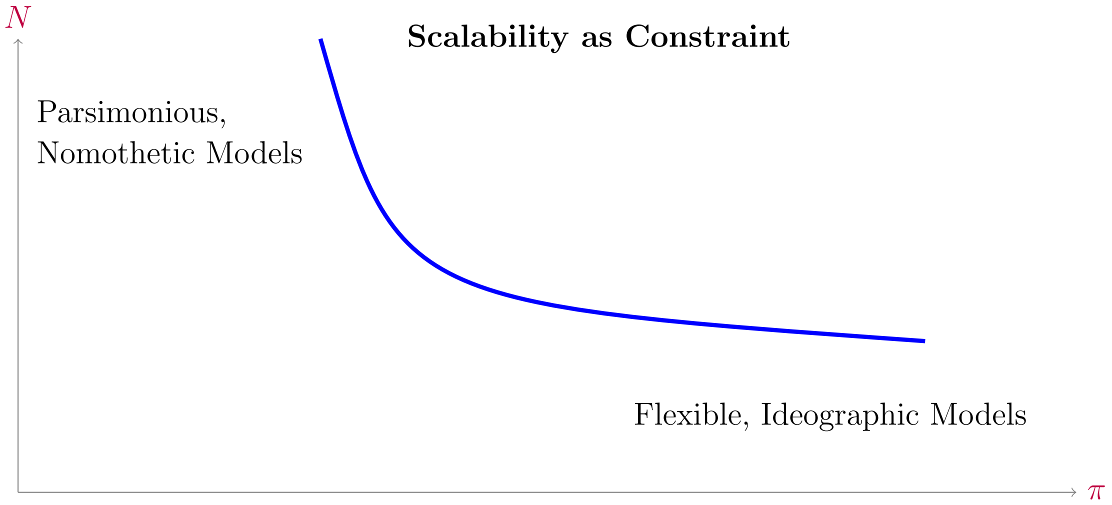
[1] Arguably, we might view Bayesian posterior probabilities as a good example of an "interpretable" procedure. As Gill (1999), Jackman (2009) and others argue, the frequentist null hypothesis testing paradigm is remarkably difficult to properly interpret. By contrast, researchers can straightforwardly 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).

[2] 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.

highly flexible, robust to violations of assumptions, and often possess excellent out-of-sample properties. 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 $O(Nlog(N)^2) + O(PNlog(N))$.[3] 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).

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.

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

3 bigKRLS as Case Study in Optimization

For a stark example of the complexity frontier phenomenon, consider Kernel-Regularized Least Squares (KRLS) (Hainmueller and Hazlett 2013). KRLS has has received substantial attention among political methodologists for its mix of flexibility, interpretability, and desirable theoretical properties. However, as we describe, KRLS also represents a model at the outer edge of the complexity frontier for medium-to-large applications. KRLS and models like it, we argue, offer a prime opportunity for optimization work, opening high-value techniques to new users and applications.

3.1 Overview

Kernel Regularized Least Squares (KRLS) is a promising 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), researchers can easily determine whether a treatment effect appears to be constant across the sample, calculate the variance of the treatment effect across sample subsets, or assess whether or not the outcome is monotonic function of the treatment.

To explain the variation in y, the model begins by calculating a Gaussian similarity kernel across the dataset. 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,

$$\begin{aligned} ||\mathbf{x}_A - \mathbf{x}_B||^2 = &(Age_A - Age_B)^2 \\ &+ (Ideology_A - Ideology_B)^2 \\ &+ (Female_A - Female_B)^2 \\ &+ ... \end{aligned}$$

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, the similarity score declines as the average distance across observable dimensions increases.[4]

[4] Expanding and re-arranging suggests an interpretation of distance as "unsynchronized" variance:

$$\begin{aligned} ||\mathbf{x}_A - \mathbf{x}_B||/\sigma^2 &= \frac{\left(\sum_1^P \mathbf{X}^{\mathbf{2}}_{\mathbf{A,p}} + \sum_1^P \mathbf{X}^{\mathbf{2}}_{\mathbf{B,p}} - 2 \sum_1^P \mathbf{X}_{\mathbf{A,p}} * \mathbf{X}_{\mathbf{B,p}} \right)}{P} \\ &= s_A^2 + s_B^2 - 2 * cov(\mathbf{x}_A, \mathbf{x}_B) \end{aligned}$$

The model we are interested in is:

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

The outcome $\mathbf{y}$ is hypothesized to be the product of the kernel $\mathbf{K}$ (an N x N matrix containing the similarity of each pair of observations)[5] and $\mathbf{c}$, a mean-zero weights vector which controls tightness of model fit to each individual point.

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.[6] This estimate results from a Tikhonov Regularization problem, which can be expressed as:

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

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

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

Including the kernel in the regularization ($\mathbf{c}'\mathbf{Kc}$) effectively weighs outliers by similarity to overall model fit. Suppose that a researcher hypothesizes that some $\frac{\delta y}{\delta x_p} > 0$. If that effect is actually only present in some subpopulation, the regularization scheme will make the algorithm skeptical of a $\hat{\mathbf{c}}$ that suggests $\frac{\delta y}{\delta x_p} > 0$ if that effect is confounded by other variables (e.g. group structure).

Regarding the local derivatives, 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. If $\mathbf{D_p}$ is an N by N matrix of simple distances on dimension p ($\mathbf{D_{p(i),p(j)}} = \mathbf{X_{i,p}} - \mathbf{X_{j,p}} \; \forall \; i, j$), then $\hat{\delta}_{\mathbf{p}} = \frac{-2}{\sigma^2}\mathbf{D_p K}\hat{\mathbf{c}}$. For binary variables, first differences are estimated according to a different procedure discussed in §3.4. In this sense, KRLS trades parsimony for flexibility, as the marginal effects may suggest many curves. However, in our view KRLS is still quite interpretable because KRLS estimates average marginal effects, the canonical quantity of interest. Readers new to KRLS may wish to skip to §4 and circle back to the rest of §3.

[5] In related work on such kernels, Hazlett (2016) obtains unbiased ATT estimates when matching fails and Diaconis et al. (2008) illustrate data reduction properties using Congressional roll call data.

[6] The kernel is positive semi-definite and has a number of useful properties. Hypotheses H (i.e., possible $\hat{\mathbf{c}}$) 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). Mercer's Theorem enables regularization as the kernel's Eigendecomposition takes a known form even in high dimensional space. Though λ cannot be obtained analytically, λ exists in a finite, unidimensional space and can be approximated with closed-form functions (Beck and Ben-Tal 2006; Hainmueller and Hazlett 2013; Hastie et al. 2008; Rifkin and Lippert 2007).

3.2 Complexity

KRLS contains seven major steps, which we document (in somewhat simplified form) in Table 2. Compared with many workhorse modeling approaches (e.g. ordinary regression setups, decision trees), KRLS requires substantially greater resources to fully estimate, with total runtime complexity $O(N^3)$ (compared with, for example, the $O(Nlog(N)^2) + O(PNlog(N))$ decision tree result given earlier).

Unfortunately, these results cannot be easily remedied. Eigendecomposition of the Gaussian distance kernel and estimation of the variance-covariance and pairwise local derivative matrices are inherently high-complexity operations, which cannot be straightforwardly reduced. Parallelization is helpful for some computations in the algorithm; for example, the divide-and-conquer eigendecomposition algorithm is straightforwardly parallelizable, and the local derivatives calculations for each individual predictor can be conducted independently. However, for most steps in the algorithm, parallel implementations do relatively little to improve overall complexity results.

Memory requirements for KRLS are similarly challenging. 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.[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 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.

3.3 Major Updates of bigKRLS

1. C++ integration. We re-implement most major computations in the model in C++ via Rcpp and RcppArmadillo. These changes produce up to a 50% runtime decrease compared to the original R implementation.

2. Leaner algorithm. Because of the Tikhonov regularization and parameter tuning strategies used in KRLS, the model is inherently memory-heavy ($\mathrm{O(N^2)}$), making marginal memory savings important even in small- and medium-sized applications. We develop and implement a new local derivatives algorithm, which reduces peak memory usage by approximately an order of magnitude.

3. Improved memory management. Many data objects in R perform poorly in memory-intensive applications. We use a series of packages in the bigmemory

[7] 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 (§3.4), 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.

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

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.

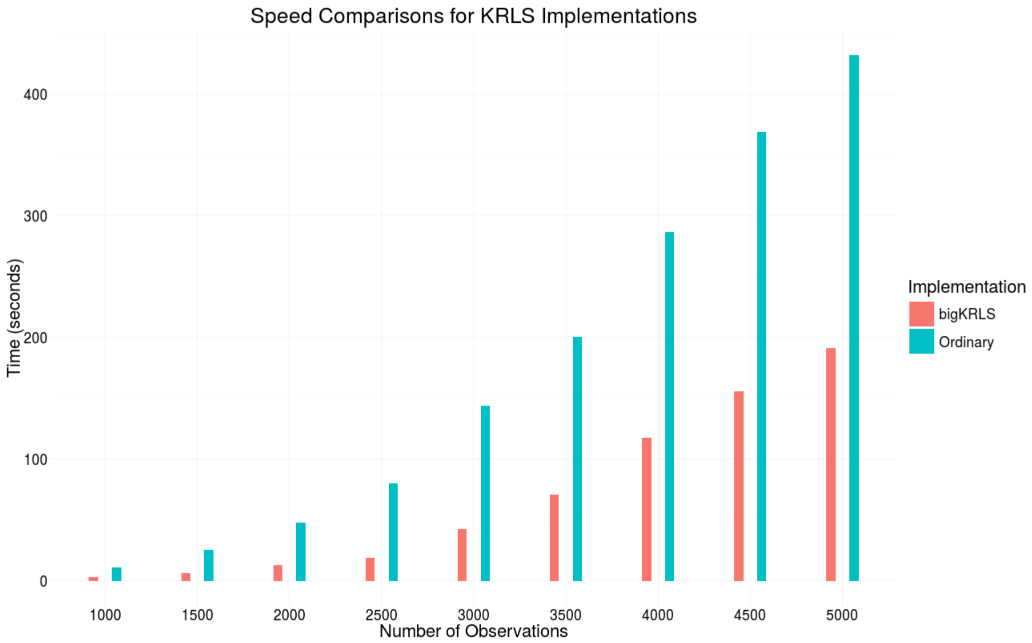
environment to ease this constraint, allowing our implementation to handle larger datasets more smoothly.

4. Interactive data visualization. We've designed an R *Shiny* app that allows users to explore marginal effects, share results with collaborators, and easily publish results online for more general audiences, a prototype of which is available online. The analysis of the Pew Research survey are presented with screen shots of a lightly customized version of what will be distributed along with *bigKRLS*.

5. Parallel Processing (*coming soon*). Many of the calculations are best done on a single core when (particularly when working with bigmemory) however parallel processing (with snow) offers substantial speed gains for the marginal effects.

Put together, these improvements offer a substantial reduction in peak memory usage, bringing peak memory consumption from $O(9PN^2)$ (in the binary variable case) to $O(5N^2)$ in our implementation. Runtime for datasets consisting entirely of continous predictors is roughly comparable across the two implementations; however, our new first differences algorithm (see below) is substantially faster than the previous implementation. In simulation results for a dataset consisting of 10 binary predictors (shown in Figure 3), for example, we report approximately 30-40% decreased wall-clock time at all sample sizes tested.

Figure 3: Performance in binary prediction case



3.4 A Leaner First Differences Algorithm

For binary variables, KRLS estimates first differences in lieu of of marginal derivatives.[8] The original algorithm for this procedure functions as follows. Suppose X_b is a column of X corresponding to a binary variable. Construct two copies of $\mathbf{X}$ as $\mathbf{X^{(0)}}$ and $\mathbf{X^{(1)}}$. Assign $\mathbf{X_b^{(0)}} = \mathbf{0}$ and $\mathbf{X_b^{(1)}} = \mathbf{1}$. Compute a new kernel based on $\mathbf{X}_{new} = [\mathbf{X}_{observed} \,|\, \mathbf{X_b^{(0)}} \,|\, \mathbf{X_b^{(1)}}]$. This step is temporary but has a memory footprint of $9N^2$! Finally, save the two submatrices of the kernel corresponding to the two counterfactual comparisons between $\mathbf{X_b^{(0)}}$, $\mathbf{X_b^{(1)}}$, and the observed data $\mathbf{X}_b$.

Our leaner implementation proceeds as follows. Let $\mathbf{K}_{new} = [\mathbf{K}_{\{1\}} \,|\, \mathbf{K}_{\{0\}}]'$, that is, let $\mathbf{K}_{new}$ be a partitioned matrix containing (just) the counterfactual matrices. The first differences are:

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

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 the average derivatives for first differences that Hainmueller and Hazlett (2013) derive 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 (the first N entries are $\frac{1}{N}$ and the next N are $-\frac{1}{N}$) and $\hat{\mathbf{\Sigma}}_{\mathbf{c}}$ is the variance co-variance matrix of the coefficients. 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 $\mathbf{K}_{i,j}$.

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

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 4.

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

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

Figure 4: 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 $\hat{\mathbf{c}}$-weighted average of the final column.

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.

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 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 which further optimize either speed or memory usage (but not both) are also possible, which may be useful in certain situations. The Boolean algorithm, for example, can be re-expressed as a triple-loop with no additional memory overhead; however, this 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*, preliminary speed tests show the "Boolean" algorithm is no slower than a purely linear algebra approach.

To take a particular example of why this advance is important, consider the dyadic data case. Because of the pairwise structure of the kernel, KRLS is tailor made for international relations, which often encounters data in country-dyad years. However, such analyses often require at least 150 binary variables for nation states. With *bigKRLS*, such binary variables no longer induce additional computational burden.

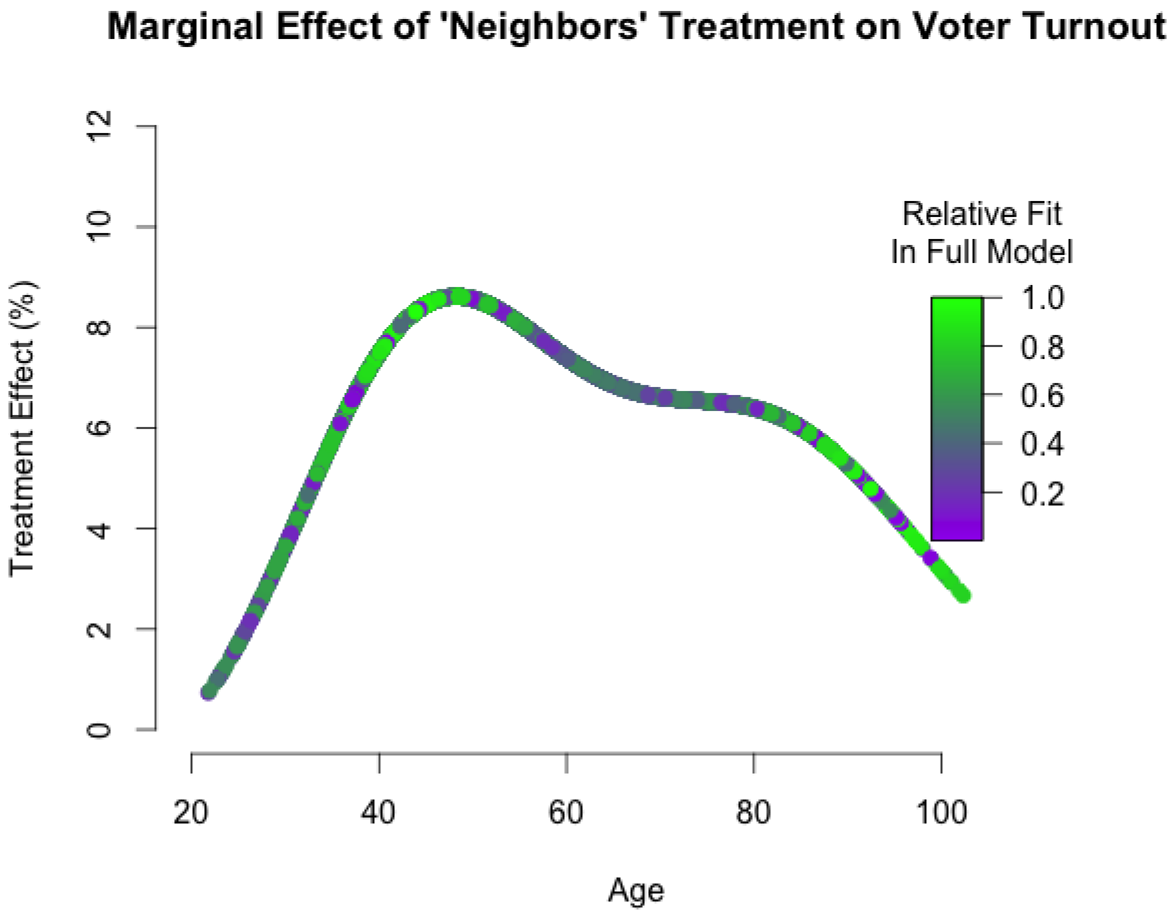
4 Applications

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 *KRLS*, the R 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. Additionally, the counterfactual calculations in the original algorithm are even more computationally taxing, whereas with *bigKRLS* they are not (see 3.4).

Both datasets also highlight the need both for flexibility and for interpretability in complex modeling applications. In both cases, we do not assume that y is normally distributed nor that the data generating process reflects a particular functional form (e.g., probit in the experiment and ordinal, or multinomial in the observational study), as would be required by a generalized linear model (GLM). We also do not specify a hierarchical structure, as would be particularly relevant in the observational study. 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, Green, and Larimer (GGL) conducted a field experiment based on 180,000 registered Michigan voters in 2006 to assess whether social pressure increases voter turnout (Gerber et al. 2008). 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.



	Average Marginal Effect	SE	t	p
Neighbor's Treatment	0.0730	0.0132	5.5489	< 0.0001
Age (in Years)	0.0005	0.0002	2.2358	0.0254

$N = 10{,}000$, R^2: 0.0093

The estimates suggest an average treatment effect of just over 7% increase in turnout; OLS, by contrast, estimates closer to 10%.[9] 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.[10] We have also colored each marginal effect according to its relative fit (green means fits well, purple poorly) by ranking the squared coefficients (see Appendix 1). The model fits (relatively) poorly for younger voters. Perhaps the effect of the treatment is conditional on paying attention to the mail.

[9] As mentioned earlier, obtaining more than 10,000 eigenvectors is non-trivial; here we have simply presented analysis of a simple random sample of the 30,000.

[10] 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 those who are allergic. That said, if the average marginal effect is statistically significant but the marginal effects span positive and negative values, careful analysis of underlying conditions is required.

4.2 Analyzing Interactions

To further illustrate the model, consider whether Americans prefer Donald Trump over Hillary Clinton, which we estimate below with Pew Research Center's January 2016 survey. The dependent variable, $\mathbf{y}_i \equiv \mathbf{Q22E}_i - \mathbf{Q22D}_i$, that is, relative preference for Trump as found by taking the difference of two five point Likert responses.[11] The independent variables in the model are standard sociodemographics (including whether the interview was conducted in Spanish), liberalism, approval for Obama, all 50 states plus DC, and region. Pew Research does not release zip code or county but population density is included.

The model fits the data. As Figure 4 shows, it explains about two thirds of the variance ($R^2 = 0.676$). The R^2 is the proportion of the variance explained by $\mathbf{K}\hat{\mathbf{c}}$, not the average marginal effects reported in the regression table. Computing a pseudo-R^2 with just the average marginal effects suggests the model is reasonably linear and additive in that pseudo-$R^2_{AME} = 0.548$ (a figure we will add to $bigKRLS's\,summary$). Put differently, the average marginal effects provide approximately 80% of the explanatory power in the model.

Even in January, most of the patterns that have dominated media coverage of the election are evident. Self-identified liberals, African Americans, Latinos, and women all favor Clinton, while whites (particularly in the South) tilt slightly towards Trump. As Figure 5 shows, we can explore these results further additional results with the *Shiny* app to be released with *bigKRLS*.[12] The first map shows that Clinton's advantage with women varies considerably by state. States where she has a clear advantage are shown in blue; the colors in red indicate less a Trump preference than lack of a gender effect. For example, if Wisconsin is in play, Hillary should feel confident she'll enjoy a bit of added support from women, but somewhat less so about Virginia or North Carolina. Like an analogous map plotting the average marginal effect of liberalism by state, KRLS has likely detected an omitted variable: namely, support for Sanders (as the red-colored Vermont indicates).

The next plot shows the effect of liberalism by age. The plot suggests that, while more liberal voters are less likely to support Trump, liberalism (at least in the self-reported sense) has a weaker relationship with vote preference among younger voters than older voters. The final scatter plot of the marginal effect of being Hispanic Latino by age is included mainly for contrast. Unlike in the experiment, where the marginal effects of the treatment fall neatly on a curve, the marginal effects from the

[11] "Regardless of who you currently support, Id like to know what kind of president you think each of the following would be if elected in November 2016? First, [INSERT NAME; RANDOMIZE]. If [INSERT NAME] were to become president do you think (he/she) would be a great, good, average, poor, or terrible president?"

[12] Shiny by RStudio allows users to interact with *R* results. Due to the variability in the way geographic data is labeled and sorted, we won't be able to generate the map in a single command, however we can enable researchers, their collaborators, and their peers to easily explore the other graphs via a function which generates basic *RShiny* code. We will also post examples demonstrating how to incorporate maps.

observational study are considerably noisier. Thus KRLS picks up on a number of interesting trends, particularly considering the data come from just one survey and no contextual information is included.

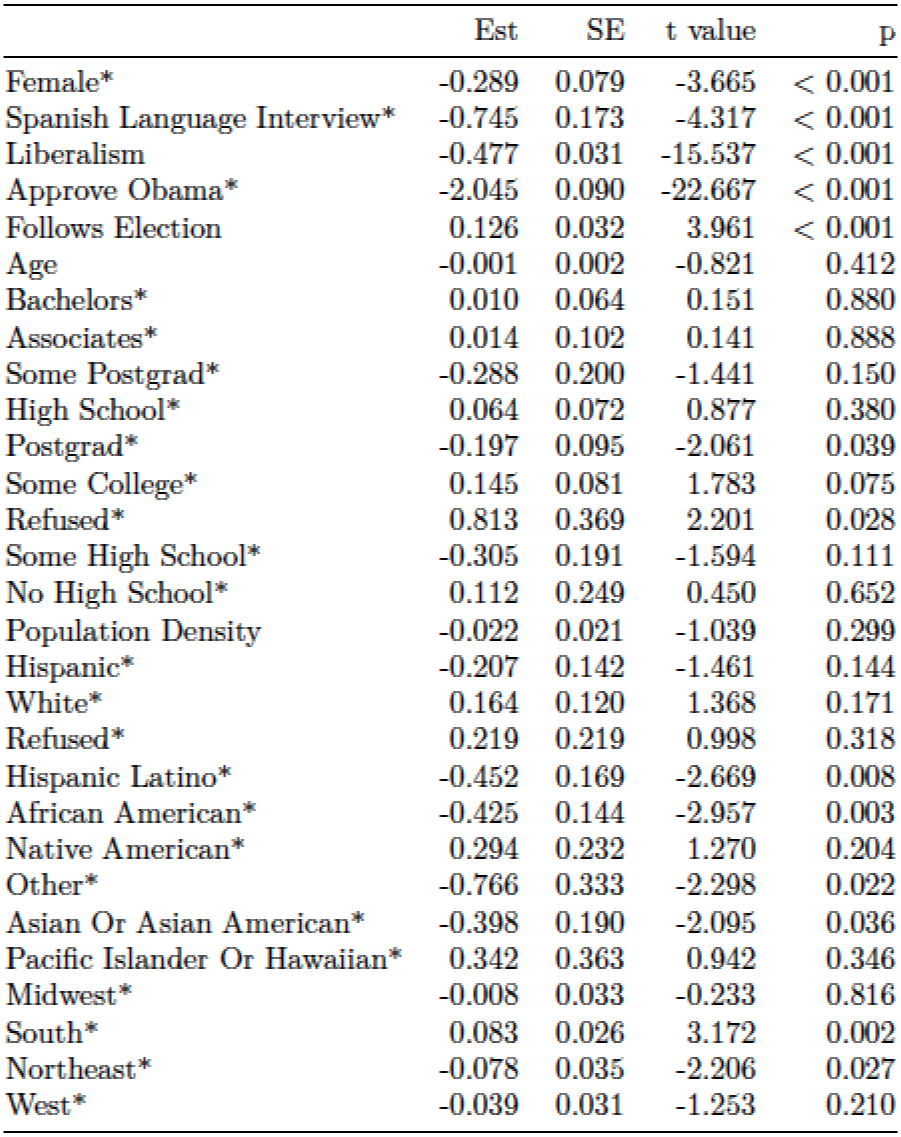
5 Conclusion

In recent years, researchers have become increasingly interested in 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 model which attempts to provide all of these characteristics, with the capacity to contribute to social science research at a number of stages. As the Trump vs. Hillary example illustrates, KRLS has 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 MCMC, particularly for unusually challenging data.

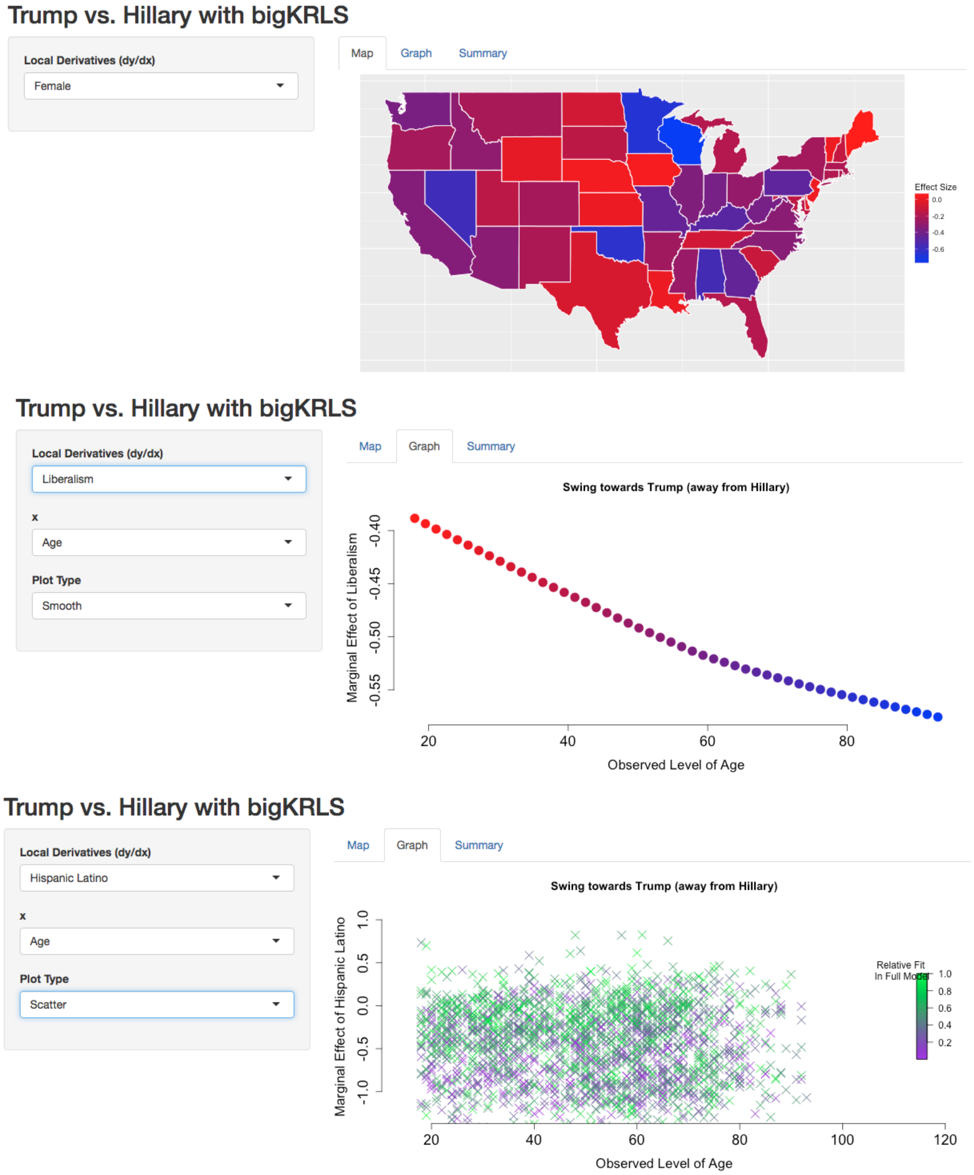
Figure 5: Model 2 Regression Results. Dependent Variable = Relative Preference for Trump vs. Clinton. Source Data: Pew Research Foundation January 2016 Survey.



	Est	SE	t value	p
Female*	-0.289	0.079	-3.665	< 0.001
Spanish Language Interview*	-0.745	0.173	-4.317	< 0.001
Liberalism	-0.477	0.031	-15.537	< 0.001
Approve Obama*	-2.045	0.090	-22.667	< 0.001
Follows Election	0.126	0.032	3.961	< 0.001
Age	-0.001	0.002	-0.821	0.412
Bachelors*	0.010	0.064	0.151	0.880
Associates*	0.014	0.102	0.141	0.888
Some Postgrad*	-0.288	0.200	-1.441	0.150
High School*	0.064	0.072	0.877	0.380
Postgrad*	-0.197	0.095	-2.061	0.039
Some College*	0.145	0.081	1.783	0.075
Refused*	0.813	0.369	2.201	0.028
Some High School*	-0.305	0.191	-1.594	0.111
No High School*	0.112	0.249	0.450	0.652
Population Density	-0.022	0.021	-1.039	0.299
Hispanic*	-0.207	0.142	-1.461	0.144
White*	0.164	0.120	1.368	0.171
Refused*	0.219	0.219	0.998	0.318
Hispanic Latino*	-0.452	0.169	-2.669	0.008
African American*	-0.425	0.144	-2.957	0.003
Native American*	0.294	0.232	1.270	0.204
Other*	-0.766	0.333	-2.298	0.022
Asian Or Asian American*	-0.398	0.190	-2.095	0.036
Pacific Islander Or Hawaiian*	0.342	0.363	0.942	0.346
Midwest*	-0.008	0.033	-0.233	0.816
South*	0.083	0.026	3.172	0.002
Northeast*	-0.078	0.035	-2.206	0.027
West*	-0.039	0.031	-1.253	0.210

$N = 2009$. * indicates binary variable for which first differences are computed (coefficient estimates for state and DC omitted for brevity); $R^2 = 0.6755$.

Figure 6: Screen shots from *bigKRLS Shiny* app, with results from the Clinton/Trump preference data.



Trump vs. Hillary with bigKRLS

Local Derivatives (dy/dx): Female

Trump vs. Hillary with bigKRLS

Local Derivatives (dy/dx): Liberalism

x: Age

Plot Type: Smooth

Trump vs. Hillary with bigKRLS

Local Derivatives (dy/dx): Hispanic Latino

x: Age

Plot Type: Scatter

6 Appendix 1: Code for Neighbors Experiment

This section contains *R* code to replicate the analysis of the neighbors experiment. At N 10,000, the run time is a few hours on a machine with 16 gigs of RAM. *bigKRLS* can be installed via GitHub. The weights, $\hat{\mathbf{c}}$, appear in the code below as out$coeffs.

```
library(bigKRLS); library(fields); library(grDevices)
load("Green_et_al_polanalysis_2009_BW5.RData")

set.seed(2016)

include <- sample(1:nrow(data1), 10000, replace=F)
y <- as.matrix(data1$voted[include])
X <- as.matrix(cbind(data1$treatmen, data1$ageatelection))[include,]

out <- bigKRLS(y=y,X=X)
summary(out)

N <- nrow(out$X)

plot(x=(out$X[,2]+55), # age comes centered at 55

        y=(100*out$derivatives[,1]), ylab="Treatment Effect (%)",

        xlab="Age", pch = 19, bty = "n", ylim = c(0, 12), xlim = c(20, 115),

        main="Marginal Effect of 'Neighbors' Treatment on Voter Turnout",

        col = colorRampPalette(c("green", "purple"))(N)[rank(out$coeffs^2)])

image.plot(legend.only = T, zlim=c(1/N, 1),
                          legend.cex = 0.75,legend.shrink = .5,
                          col = colorRampPalette(c("purple", "green"))(nrow(out$X)))

text(x = 107, y = 11.5, "Relative Fit\nIn Full Model"))
```

WARNING: even at this sample size, the full output is several gigabytes and bigmemory objects cannot be saved "as usual" in *R*; to save the objects of interest (which are quite small), first run

```
vignettes("bigKRLS_basics")
```

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.

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

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

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.

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*.

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.

Mohanty, P. and Shaffer, R. B. (2016). bigkrls, available at https://github.com/rdrr1990/bigkrls.

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*.

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.