Messy Data, Robust Inference?

Navigating Obstacles to Inference with bigKRLS

Pete Mohanty
pmohanty@stanford.edu

Robert Shaffer
rbshaffer@utexas.edu

May 16, 2018

Abstract

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

Supplemental Materials prepared for *Political Analysis*

Full replication materials to appear on Harvard's *Dataverse*

A C++ Kernel Regularization Code

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

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

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

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

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

  // .memptr() expects data by column
```

```
  Eigenvectors = trans(Eigenvectors);

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

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

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

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

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

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

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

B Descriptive Statistics for 2016 Election Study

Figure 4 gives summary statistics for the variables used in our applied example in §5. All race and education variables are given in percentage point units. Units for other variables are given in table notes. For our applied example, we also include latitude, longitude, and state dummy variables as additional predictors. Replication materials are available here.

Table 4: Descriptive statistics for Section 4

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

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

C Simulations

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

C.1 Setup

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

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

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

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

(AME), defined as:

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

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

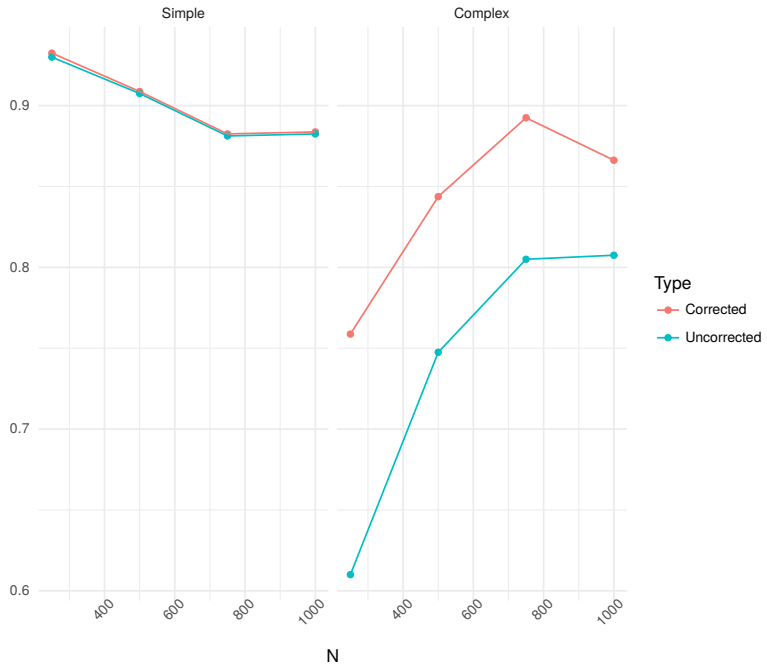
C.2 Degrees of Freedom Correction for Average Marginal Effects

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

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

Coverage results of this experiment both with and without our degrees of freedom correction are are presented in Figure 8. In nearly all cases, empirical coverage results are somewhat lower than their nominal 0.95 value, which likely reflects the bias in our estimates

Figure 8: Simulated AME coverage results



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

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

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

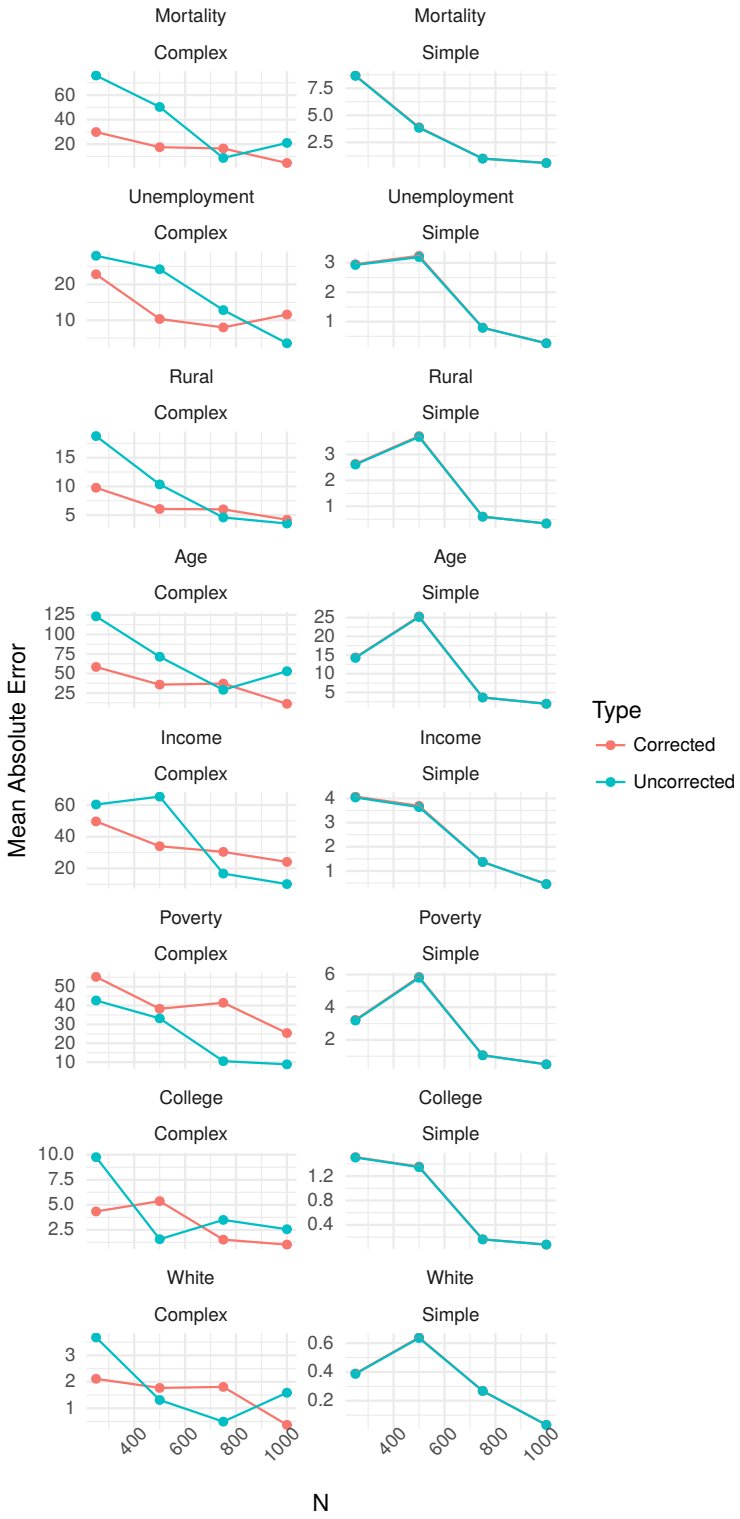
rected standard error estimates produced in each iteration to assess which procedure more faithfully reproduces the population standard errors for the regularized coefficient estimates.

The results of this comparison are given in Figure 9. As before, performance of the error estimates for the simple DGP are essentially identical across the two modeling strategies. By contrast, for the complex DGP the two methods produce noticeably different results. Generally, our corrected standard errors perform best at smaller sample sizes, with corrected standard error estimates for seven out of eight coefficients outperforming their uncorrected counterparts at $N = 250$. At $N = 1000$, by contrast, performance is split, with half performing better with our correction and half performing worse. However, performance is not constant across coefficients; for example, our correction consistently produces similar or superior performance for the Rural, Age, and Mortality variables, but similar or inferior performance for the Poverty variable.

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

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

Figure 9: Population SE versus estimated SE for regularized AME estimates



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

C.3 Subsample-Based AME Estimates

Estimating the kernel-based model we adopt in the KRLS framework is computationally demanding. However, as suggested by an anonymous reviewer, some of the simpler estimates contained in this model can be reasonably well-approximated using a subsampling scheme, which cuts computation time and resources substantially. The approach we adopt is particularly well-suited for the average marginal effect estimates (AMEs) generated by this model, which are the focus of this section. However, we also discuss the applicability of this approach for other quantities at the conclusion of this section.[28]

To generate subsample-based AMEs, we propose the following approach. First, divide the observations into M equally-sized groups.[29] Second, fit a model via KRLS to each subgroup and estimate AMEs for each model, denoted $\hat{\Delta}_{AME}^{(m)}$. Since the estimated AMEs across subgroup models are independent, we can leverage the closed-form expressions for their values and variances to produce an aggregated set of estimates:

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

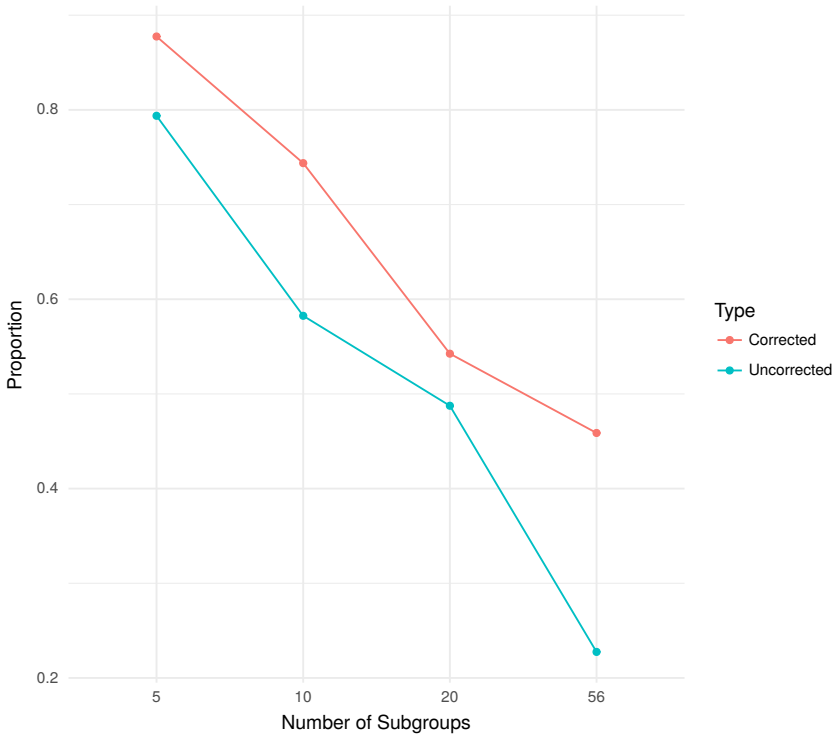
This approximation relies on the asymptotic normality of the AMEs. As a result, we should expect this approximation to behave better with large subgroups than with small ones.

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

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

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

Figure 10: Empirical coverage for subsample-based AME estimates



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

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

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

The results of this comparison are given in Figure 10. Following expectations, perfor-

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

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

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

D Crossvalidation

D.1 bigKRLS Out-of-Sample

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

The results of this experiment are presented in Figure 5. Across cross-validation replicates, performance statistics are stable, indicating that the full sample estimates (presented in the Section 4) are unlikely to have been influenced by outliers, subgroup-specific patterns, or by our specific geographic specification. In-sample, the full model of N coefficients consistently outperforms the portion which is a linear and additive function of the x variables, the Average Marginal Effects (AMEs). Out-of-sample, however, the performance gap is much smaller. As a result, we encourage users to employ cross-validation or a similar out-of-sample predictive strategy if precise estimates of effect heterogeneity are desirable.

D.2 Comparison to Other Approaches

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

Table 5: Overview of Crossvalidation Results

	In Sample	Out of Sample
RMSE	3.002	3.076
	(0.056)	(0.199)
RMSE_{AMEs}	3.336	3.339
	(0.184)	(0.271)
Pseudo R^2	0.673	0.659
	(0.012)	(0.038)
Pseudo R^2_{AME}	0.104	0.641
	(0.011)	(0.046)

Results of 100 five-fold cross-validation replicates (N_{train} = 80% = 2,485 observations; N_{test} = 20% = 621 observations). Average measures of fit provided along with margins of error. *AME* subscript indicates that only the Average Marginal Effects were used to obtain fitted values in sample or predicted values out of sample. For convenience, *crossvalidate.bigKRLS*, which also performs K folds cross validation, computes these measures of fit. The low R^2_{AME} on the training data is best interpreted as an artifact of regularization and the discrepancy between y and y^*.

for details). As before, we use a 5-fold cross-validation procedure, with performance results generated by averaging root-mean squared error (RMSE) and root-mean squared predictive error (RMSPE) across each fold and replicated 100 times to generate uncertainty estimates. To generate uncertainty estimates, we replicated our cross-validation process 100 times, and presented mean and margin of error for each performance statistic.

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

Specifications for all additional models are as follows. For our random forest models, we trained our models using 500 trees, with the parameter denoting the number of candidate

variables to consider at each split selected by optimizing out-of-bag error (Breiman 2001). For our causal forest models, we used 2000 trees, with our county-level age-adjusted mortality variable acting as the "treatment" variable.[30] For our penalized regression approaches, we trained two sets of models. In the first set, we only allowed the models to estimate linear and additive effects for each variable. In the second set, we additionally allowed the model to estimate coefficients corresponding to all two-way interactions between each variable included in our dataset. In both cases, we trained one model using a LASSO penalty, one model using a ridge penalty, and one using an elastic net penalty (with mixing parameter set to $\alpha = 0.5$). For all penalized approaches, we selected the regularization parameter λ by optimizing cross-validated error within the training set.

The results of this comparison are given in Figure 6. Compared with most other approaches, KRLS is prone to overfitting, with the largest gap between RMSE and RMSPE of the models we examine. However, KRLS's out-of-sample performance remains competitive. By RMSPE, KRLS is the second-best performer, producing approximately a 1% higher RMSPE than a simple random forest. In our view, this result is encouraging. In most predictive modeling contexts, random forests represent the best-performing general-purpose approach. As a result, KRLS's ability to essentially match a random forest's predictive performance is reassuring.

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

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

Table 6: In- and out-of-sample prediction results, comparing KRLS to related approaches

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

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

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

for the treatment variable (Wager and Athey 2017). However, since our application in this paper is more exploratory, KRLS's superior out-of-sample predictive performance and ability to estimate effects for each variable directly offer strong reasons to prefer KRLS in this context.

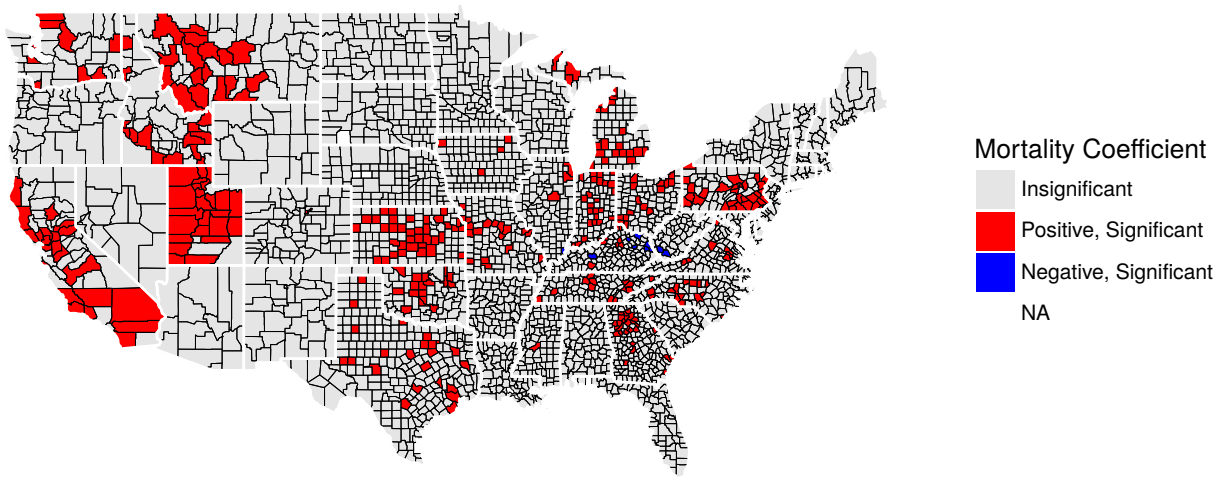
E First Differences

E.1 Significance Testing

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

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

Figure 11: Significance results for individual county-level predictions



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

E.2 Effect Sizes

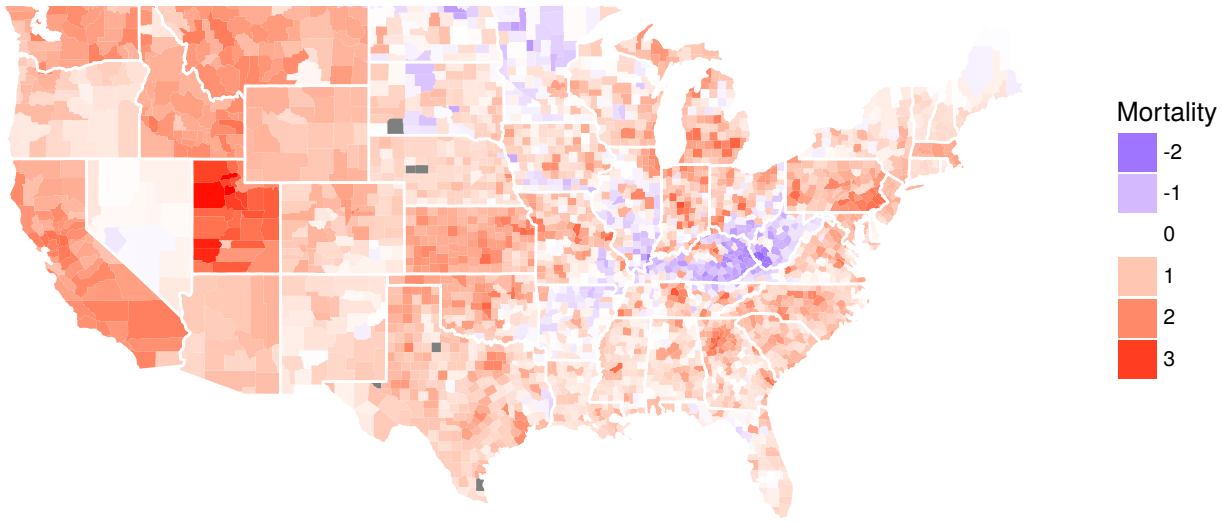


Figure 12: Effect sizes for county-level predictions