Feasible Generalized Least Squares Using Machine Learning

Steve Miller*[a], Richard Startz[b]

[a]*Department of Applied Economics, University of Minnesota*
[b]*Department of Economics, University of California, Santa Barbara*

Abstract

In the presence of heteroskedastic errors, regression using Feasible Generalized Least Squares (FGLS) offers potential efficiency gains over Ordinary Least Squares (OLS). However, FGLS adoption remains limited, in part because the form of heteroskedasticity may be misspecified. We investigate machine learning methods to address this concern, focusing on Support Vector Regression. Monte Carlo results indicate the resulting estimator and an accompanying standard error correction offer substantially improved precision, nominal coverage rates, and shorter confidence intervals than OLS with heteroskedasticity-consistent (HC3) standard errors. Reductions in root mean squared error are 87-98% of those achievable when the form of heteroskedasticity is known.

Keywords: Heteroskedasticity, Feasible Generalized Least Squares, Machine Learning
JEL Classification: C13; C21

*Corresponding author. Email: s-miller@umn.edu; Phone: 612-625-3212; Address: 316E Ruttan Hall, 1994 Buford Avenue, St. Paul, MN 55108

1. Introduction

Heteroskedastic errors render Ordinary Least Squares (OLS) estimators inefficient and induce bias in the corresponding standard errors. Two broad methods exist for dealing with heteroskedasticity: Generalized Least Squares (GLS), which accounts for heteroskedasticity, or OLS with an attempt to correct standard errors. The most common forms of alternate estimators are Weighted Least Squares (WLS) if the form of heteroskedasticity is known and Feasible Generalized Least Squares (FGLS) if the form of heteroskedasticity must be estimated. By accounting for heteroskedasticity, these estimators hold the promise of greater efficiency. If the decision is made to stick with OLS coefficient estimates, the most popular corrections to standard errors are the heteroskedasticity-consistent variants (Eicker, 1963; Huber, 1967; White, 1980; MacKinnon and White, 1985; Cribari-Neto, 2004).

More recent empirical work tends to favor simply correcting OLS standard errors, largely because doing so imposes less stringent assumptions. WLS requires strong assumptions about the exact knowledge of the heteroskedasticity. Similarly, the first stage of FGLS, in which the relationship between error variance and covariates is estimated, may entail specifying a particular functional form. Non-parametric estimation of the form of heteroskedasticity is possible, but may require other assumptions, such as kernel bandwidths or a type of kernel. If these assumptions are incorrect, the erroneous weights derived from this first stage may cause FGLS to be *less* efficient than OLS. Further, if the transformed model produced by FGLS does not eliminate heteroskedasticity, standard errors and the resulting inference (if using standard errors estimated under the assumption of conditional homoskedasticity) will be incorrect. While the latter problem can in principle be dealt with by estimating heteroskedasticity-consistent standard errors even when estimating parameters via FGLS (Wooldridge, 2010; Romano and Wolf, 2017), the potential for misspecification of the form of heteroskedasticity has kept FGLS and WLS largely on the sidelines of applied research.

In this paper, we consider whether approaches from the field of machine learning, which impose relatively few assumptions on the data, may improve the performance of FGLS. Those tools have excellent out-of-sample predictive performance, which may prove useful in modeling the form of heteroskedasticity. Since, for the purposes of FGLS, any causal relationship between regressors and the error variance is not of direct interest, the purely predictive nature of these algorithms is suitable and their performance attractive. Some machine learning algorithms can be seen to build on the use of nonparametric estimators proposed before (Carroll, 1982; Robinson, 1987; Newey, 1994).[2] For example, Lin and Jeon (2006) draw connections between tree-based methods and adaptive nearest-neighbor approaches, suggesting

[2]For an example application of a kernel estimator see, e.g., O'Hara and Parmeter (2013).

the use of some machine learning methods can be cast as a generalization of prior nonparametric methods. Analogously, in this context, support vector regression[3] can be seen to estimate a regression model with three key changes: a basis expansion (offering more modeling flexibility), as well as a different loss function and a penalty for coefficient size akin to ridge regression, both of which dampen the variance of the estimator. While those theoretical connections offer promise for the use of machine learning tools in the context of FGLS, to our knowledge the performance of these tools has yet to be investigated.

Our investigations offer two potential contributions. First, we present Monte Carlo evidence suggesting that the use of machine learning tools to estimate the skedastic function (the relationship between the error variance and regressors) can offer substantial gains in precision without requiring strong functional form assumptions. As with other FGLS methods, these gains do come at the cost of precision under homoskedasticity. However, when heteroskedasticity is present, the estimator provides dramatic reductions in root mean squared error (RMSE) compared to OLS with heteroskedasticity-consistent standard errors. On simulated datasets we construct, the reductions in RMSE are 87-98% of those achieved if the form of heteroskedasticity is known. Moreover, these benefits accrue not only for purely simulated data, but also for more realistic data generated via wild bootstrap from the well-known Boston housing dataset (Harrison and Rubinfeld, 1978; Drucker et al., 1997; Wooldridge, 2015). Second, when using machine learning tools to estimate the skedastic function, heteroskedasticity-robust standard errors[4] tend to result in coverage rates that are too low. In response, we propose a correction to standard errors that brings coverage rates near nominal levels. The correction accounts for the higher degrees of freedom inherent in more flexible modeling of the skedastic function, which may also prove helpful for correcting standard errors of nonparametric methods for which suitable degrees of freedom estimates are available.

The remainder of the paper is organized as follows. Section 2 sets out the problem of estimation under heteroskedasticity, the FGLS approach, and how machine learning might help. Section 3 details options for estimating the skedastic function, with a focus on our proposal to use support vector regression (SVR) for that purpose. Section 4 describes the Monte Carlo studies we use to investigate the performance of the proposed estimator. Section 5 presents results, and the final section concludes. An Appendix provides supplementary findings concerning alternate machine learning approaches.

[3]We provide more detail on support vector regression in section 3.

[4]Specifically, HC3 standard errors.

2. Problem setup

We are interested in estimation of and inference about parameters in linear models where error terms may be conditionally heteroskedastic. We restrict our attention to linear data generating processes of the form

$$Y = X\beta + u, \tag{1}$$

where Y is an n by 1 vector of outcomes of interest, X is an n by k matrix of predictors, β is a k by 1 parameter vector, and u is an n by 1 vector of unobserved error terms. We make several standard, simplifying assumptions to focus our attention on heteroskedasticity. In particular, we assume regressors are exogenous ($E[u|X] = 0$), the observed data represents a random sample, and there is variability in but no perfect collinearity among among regressors, such that $X'X$ is invertible. We assume throughout that errors are uncorrelated across observations and u exhibits heteroskedasticity of unknown form, where $v(X) = Var(u|X)$ is non-constant and we may not have theoretical or intuitive guidance as to the process giving rise to $Var(u|X)$.

A well-known approach to estimating (1) is Feasible Generalized Least Squares (FGLS), which, as its name indicates, is a feasible form of Weighted Least Squares (WLS) in which weights are not given but instead estimated. FGLS consists of three primary steps:

1. Estimate (1) via ordinary least squares. Denote the residuals for observation i by $\hat{u}_i$.

2. Estimate a model $\hat{u_i}^2 = g(z_i)$ explaining the squared residuals, where z_i may include some or all elements of x_i or transformations of them. Denote the predicted values from this model by $\tilde{u}_i^2$.

3. Estimate (1) again, this time minimizing a weighted sum of squared residuals, with weight for squared residual i given by $\frac{1}{\tilde{u}_i^2}$.

The benefit of this procedure is that the estimator remains consistent but is asymptotically more efficient than OLS because it attaches less weight to observations considered to be noisier. The resulting estimator can be written

$$\hat{\beta}_{FGLS} = (X'\hat{W}^{-1}X)^{-1}X'\hat{W}^{-1}Y \tag{2}$$

where $\hat{W}$ is a diagonal matrix with entries $\hat{w}_{ii} = \tilde{u}_i^2$ from step 3 in the procedure above. Denote the residuals from this estimator by $\hat{u}_{i,FGLS}$.

Several different approaches have been proposed for estimation of the skedastic function in the second step. The simplest approach is to estimate a linear model explaining $\hat{u_i}^2$. To

ensure positive predictions, often $\log(\hat{u_i}^2)$ is treated as the response variable and predictions are exponentiated. Suggestions differ in terms of what form the right hand side should take in such models: Wooldridge (2010) suggests using regressors as they appear in the main model, while Romano and Wolf (2017) suggest the log of the absolute value of regressors. Other transformations or specifications are of course possible while restricting attention to linear models. Past proposals include nonparametric approaches to estimating $g(x)$, including kernel regression (Carroll, 1982), nearest neighbors (Robinson, 1987), and series estimation (Newey, 1994).

Despite the potential efficiency gains from FGLS, researchers frequently choose to estimate such models via OLS and simply correct standard errors using robust approaches. One concern which has limited FGLS use is that a poor choice of model for $g(x)$ can render the FGLS estimator more imprecise than OLS, since reweighting could give more weight to noisier observations. Flexible nonparametric specifications impose few assumptions on the functional form of $g(x)$ and should mitigate those concerns, but adoption of nonparametric FGLS remains limited.

To address this first concern, we investigate estimation of $g(x)$ using machine learning methods, with a focus on support vector regression (SVR). The set of methods we consider offer good predictive performance in the absence of information about functional form. Some of these machine learning tools are related to nonparametric methods such as nearest-neighbors, kernel regression, and series estimators; we identify some of these links later. Still, the machine learning methods we consider offer additional benefits in comparison to standard nonparametric approaches, often through regularization which penalizes coefficients or introduces zeros in the coefficient vector (i.e. sparsity). In this context, regularization acts to lower the variance of the predicted weights used in the final step of FGLS. Thus, machine learning tools offer some of the benefits of modeling flexibility shared by nonparametric methods, but also avoid overfitting squared residuals and thus may improve efficiency over FGLS using previously suggested nonparametric methods.

A second argument which has limited FGLS adoption is that estimation of $g(x)$ in step 2 will rarely eliminate heteroskedasticity entirely, so that inference using conventional OLS standard errors in step 3 will be incorrect. Recent work by Romano and Wolf (2017) addresses this concern, illustrating that heteroskedasticity-robust standard errors will provide asymptotically valid inference even if the estimated weights used in step 3 do not correspond to the inverse of the conditional variance $1/v(x_i)$. In particular, Romano and Wolf (2017) suggest estimators of the form

$$\widehat{Var}(\hat{\beta}_{FGLS}|X) = (X'\hat{W}^{-1}X)^{-1}\hat{\Omega}(X'\hat{W}^{-1}X)^{-1} \tag{3}$$

where $\hat{\Omega}$ is an estimator of $E\left[\frac{u_i^2}{(\hat{v}_{lim}(x_i))^2}x_i x_i'\right]$, where $\hat{v}_{lim}(x_i)$ is the probability limit of the estimator of $v(x_i)$. Their proposal takes the form of well-known sandwich estimators used for robust inference after OLS. Most common heteroskedasticity-robust standard errors use a form of $\hat{\Omega}$ which can be written $\hat{\Omega} = X'\hat{\Sigma}X$ where $\hat{\Sigma}$ is a diagonal matrix. For example, we consider the HC3 variant suggested by MacKinnon and White (1985), which, adapted to the weighted least squares estimator, has ith diagonal entry

$$\hat{\Sigma}_{ii} = \frac{\hat{u}_{i,FGLS}^2}{(\tilde{u}_i^2)^2(1 - h_{i,FGLS})^2}$$

where

$$h_{i,FGLS} = [X(X'\hat{W}^{-1}X)^{-1}X'\hat{W}^{-1}]_{ii}.$$

Note that the squared residuals ($\hat{u}_i^2$) and hat matrix values (h_i) from the first stage OLS regression could also be used, but efficiency gains in FGLS suggest that using the second stage residuals and hat matrix values should provide a lower variance estimate of $E[u_i^2]$.

These two issues are not the only two limiting adoption of FGLS. In addition, OLS remains unbiased in finite samples, while FGLS does not. Further, OLS remains consistent under slightly weaker exogeneity assumptions, while FGLS does not. The approach examined here does nothing to mitigate either of those concerns.

While the use of machine learning tools to estimate the skedastic function relaxes functional form assumptions, it also affects inference. In response, we propose a standard error correction to address two finite sample concerns. First, the proposed methods are likely to "burn" many more degrees of freedom in estimating the skedastic function than a conventional approach using a linear model with k parameters. As a result, the final FGLS estimation step after reweighting may have far fewer residual degrees of freedom to work with, and any standard error calculations should account for that. Second, and also because of the flexible modeling of the skedastic function, high leverage observations are likely to have even more influence in FGLS than in OLS. Intuitively, observations with high "hat" values h_i pull the first stage OLS regression line closer, thereby decreasing the first stage residuals on those influential observations. The HC3 standard errors given above can be seen to address exactly that influence when employing OLS. However, in the context of FGLS, those same high leverage points are also likely to have lower estimated conditional variance in step 2 of FGLS, resulting in higher weights and even more influence over the final FGLS estimates. This second effect is not normally considered in the derivation of heteroskedasticity-robust standard errors, since they have largely been designed for and applied to OLS estimates.

In light of these effects, we propose the following revised estimator $\hat{\Sigma}_{ii}^{FGLS}$ as a part of $\widehat{Var}(\hat{\beta}_{FGLS}|X)$:

$$\hat{\Sigma}_{ii}^{FGLS} = \frac{n}{n-\hat{df}-k} \frac{\hat{u}_{i,FGLS}^2}{(\tilde{u}_i^2)^2(1-h_{i,FGLS})^{1+\frac{\hat{df}}{k-1}}} = \frac{n}{n-\hat{df}-k} \frac{1}{(1-h_{i,FGLS})^{\frac{\hat{df}}{k-1}-1}} \hat{\Sigma}_{ii},$$

where $\hat{df}$ is an estimate of the degrees of freedom used by the second stage weight estimation. We refer to the resulting estimator $\widehat{Var}(\hat{\beta}_{FGLS}|X)$ which uses $\hat{\Sigma}_{ii}^{FGLS}$ as $HCFGLS$. The factor $\frac{n}{n-\hat{df}-k}$ is a degrees of freedom correction akin to that in HC1 standard errors (MacKinnon and White, 1985; Hinkley, 1977). Unlike HC1, however, it accounts for use of degrees of freedom in both the first and second stages of FGLS; we discuss estimation of $\hat{df}$ for the machine learning method we investigate. The power on $(1-h_{i,FGLS})$ is based on the intuition that influential observations have greater effect on weighting when the function used to approximate the skedastic function is flexible (high $\hat{df}$) in comparison to the number of variables with which the true skedastic function may vary $(k-1)$. Note that when $\hat{df} = k-1$, the second fraction corresponds to the HC3 estimator. In sum, the inflation of squared residuals should depend upon the flexibility of the skedastic function estimation. Note that neither change influences the consistency of the estimator, since the total multiplier we introduce, $\frac{n}{(n-\hat{df}-k)(1-h_{i,FGLS})^{\frac{\hat{df}}{k-1}-1}}$, converges in probability to one.

3. Estimating the skedastic function

Before describing our simulations, we briefly explain the methods we use to estimate the skedastic function $g(x_i)$ as part of FGLS. The discussion below primarily emphasizes intuition; for a clear and more formal introduction to machine learning methods, see Friedman et al. (2001). We begin with an explanation of previously proposed parametric and nonparametric estimators to which we compare the machine learning methods we investigate.

3.1. Previously proposed parametric estimators

Textbook explanations of FGLS emphasize the use of OLS to estimate the skedastic function. In keeping with this convention, we estimate three models of the conditional variance function using OLS. In all cases, we regress the log of squared residuals on a vector of explanatory variables and exponentiate the resulting predictions.

First, we estimate a model using the correct functional form of the conditional variance function. That is, if $log(u_i^2) = Z\alpha$, where entries in Z are some transformations of X, we estimate $log(\hat{u}_i^2) = Z\alpha$ via OLS. We refer to this specification as parametric FGLS with the correct functional form. The second model uses the same regressors as the main model. That

is, we estimate $log(\hat{u}_i^2) = X\alpha$ via OLS, which ignores any differences between Z and X. We refer to this specification as parametric FGLS with the main model functional form; it also corresponds to specification WLS-S2 in Romano and Wolf (2017). Finally, for comparison with prior work, we estimate the proposed WLS-S1 specification from Romano and Wolf (2017), which entails a regression:

$$log(\max(\hat{u}_i^2, \delta^2)) = log(|X|)\alpha. \tag{4}$$

Here, the constant δ bounds squared residuals away from zero to limit the possibility of extreme weights.

3.2. Previously proposed nonparametric estimators

Prior researchers have suggested the use of nonparametric estimation of the form of heteroskedasticity as part of FGLS. We implement two such estimators to compare the performance of machine learning methods to more established nonparametric approaches. Carroll (1982) suggested an FGLS estimator based on kernel estimation of the form of heteroskedasticity, showing consistency for the simple linear regression case and using Monte-Carlo evidence to evaluate performance. The kernel estimator uses a weighted average of squared residuals to estimate the form of heteroskedasticity, with weights assigned by a kernel function, which is typically chosen to decay with distance between covariates of two observations. The k nearest neighbor approach suggested by Robinson (1987) can actually be seen as a variant of the same idea. Rather than specifying a fixed kernel function, k nearest neighbor estimation applies a kernel function only to the closest k observations (based on distance between covariates). Frequently the kernel chosen for k nearest neighbor estimation gives equal weight to the k neighbors in the weighted average.

3.3. Machine learning estimators

We focus our investigation of machine learning estimation of the skedastic function for FGLS on Support Vector Regression (SVR). SVR, like OLS, seeks to minimize a function of residuals, but it penalizes residuals in a different way, penalizes coefficient vectors that are large in magnitude, and allows for nonlinearities through the use of kernels (Vapnik, 1995; Drucker et al., 1997). In the form of SVR we consider, the loss function for residuals is the ϵ-insensitive loss function, which imposes no penalty on residuals smaller than $|\epsilon|$ and penalizes residuals linearly in the degree to which they exceed $|\epsilon|$. Coefficient size is penalized using the squared l^2 norm (as in ridge regression), which reduces the variance of the model and hence overfitting.

Formally, if we seek to model a variable z_i using SVR (in the context of FGLS, $z_i = log(\hat{u}_i^2)$), we solve

$$\min_{\gamma} \sum_{i=1}^{n} L(z_i - g(x_i)) + \lambda \sum_{m=1}^{M} \gamma_m^2 \tag{5}$$

where

$$g(x_i) = \sum_{m=1}^{M} \gamma_m h_m(x)$$
$$L_\epsilon(r) = \begin{cases} 0 & \text{if } |r| < \epsilon, \\ |r| - \epsilon & \text{otherwise.} \end{cases}$$

Here the $h_m(x)$ functions are basis functions, γ_m are parameters to be estimated, λ controls the relative importance placed on the parameter size (regularization) penalty, and ϵ is the tolerance defining the minimum size a residual must have before it affects the loss function.

It turns out that the solution to this problem depends only on the inner product of the basis functions $h_m(x)$, referred to as the kernel $K(x_i, x_j)$. For certain basis functions, the inner product may be faster to compute than the basis functions themselves, so SVR is frequently specified and implemented in terms of the kernel function. In this study, we use the commonly applied radial basis function (RBF) kernel:

$$K(x_i, x_j) = e^{\phi ||x_i - x_j||^2}. \tag{6}$$

Because the number of basis functions can grow quite large for SVR,[5] the method intuitively has many of the potential benefits of series estimators, including the ability to approximate functions of arbitrary form. In addition, the penalization of the parameter vector helps control potential overfitting, while the ϵ-insensitive loss function guarantees sparsity of the solution. Viewed another way, the predictions from an SVR can also be written as a weighted sum of Lagrange multipliers (from the dual formulation of (5)) with weights determined by the kernel. Thus SVR has some connections to kernel regression, though the contribution of each observation to a new prediction depends not only on the kernel but also on the degree to which the observation contributes to the SVR solution.[6]

Before proceeding, it is worth noting when our choice of kernel is likely to be appropriate,

[5]For some common kernels, including the RBF kernel, the kernel corresponds to an infinite series of basis functions.

[6]The linear penalization of residuals beyond ϵ also caps the contribution an individual observation can contribute to this weighted sum.

and when it is not. The RBF kernel can, in principle, approximate any continuous, bounded function with arbitrary precision (Micchelli et al., 2006). As a result, if the skedastic function is believed to be both continuous and bounded, the RBF kernel is an attractive choice. Still, its flexibility is best leveraged with large amounts of data; with smaller numbers of observations a less flexible modeling approach (e.g. linear or polynomial) may be more appropriate. Similarly, if the skedastic function is known to be discontinuous, an estimation technique producing discontinous approximations, such as a regression tree, may be more appropriate.

In light of our proposed modified standard error estimator, we also require an estimate of the degrees of freedom used in estimation of the skedastic function. An unbiased estimator for the degrees of freedom used in SVR is (Gunter and Zhu, 2007):

$$\hat{df} = |\varepsilon_+| + |\varepsilon_-|, \tag{7}$$

where ε_+ and ε_- are the sets of observations with residuals equal to ϵ and $-\epsilon$. Note that $\hat{df}$ may be much larger than k for SVR.

In the Appendix, we also explain and investigate the performance of three additional machine learning methods as candidates for estimating the skedastic function: gradient boosting, regression trees, and random forests. Aside from performance considerations, one reason we focus on SVR is that a closed form, reliable estimate of the degrees of freedom is readily available, enabling the standard error correction proposed earlier. It is possible to estimate degrees of freedom for the other methods, but to our knowledge, doing so would require a Monte Carlo approach (see, e.g. Ye (1998)).

4. Monte Carlo simulations

4.1. Simple data generating process

To examine the potential for estimation of the error variance using tools from machine learning, we evaluate the performance of several data generating processes and conduct Monte Carlo simulations using various estimators. Because all estimators are consistent, we focus evaluation on precision and inference. In particular, for each FGLS variant we compute three key statistics to evaluate performance: empirical root mean squared error (RMSE), 95% confidence interval coverage probability (Cov. Prob.), and ratio of confidence interval length to OLS confidence interval length (CI ratio). RMSE is the square root of the average squared difference between the point estimate and true parameter value across the B Monte

Carlo runs:

$$RMSE = \sqrt{\frac{1}{B}\sum_{b=1}^{B}(\hat{\beta}_{FGLS,b} - \beta)^2}. \tag{8}$$

The coverage probability is the fraction of Monte Carlo runs for which the estimated confidence interval included the true parameter value. The CI ratio is the confidence interval length for the given estimator divided by that for the OLS estimator.

We compare eight estimators: (1) a baseline of OLS, as well as (2-8) FGLS with the relationship between $Var(u|x)$ and x estimated seven different ways. The FGLS variants estimate the skedastic function in the following ways: (2) OLS on the correct functional form relating $Var(u)$ and x, (3) OLS assuming $Var(u|x)$ has the same functional form as $y(x)$ (the main model, or WLS-S2), (4) WLS-S1 from Romano and Wolf (2017) (5) kernel estimation, (6) k nearest neighbor estimation, (7) support vector regression estimation with fixed tuning parameters (SVR), and (8) support vector regression estimation with tuning parameters chosen via cross-validation (SVR-CV).

For each Monte Carlo run, we generate a random sample of covariates x, draw a sample of error variables u from a distribution with variance that may depend on x, compute the resulting y variables according to a known data generating process, and estimate all models using that dataset. In all cases, the main model is correctly specified, so that the focus is solely on the effects of heteroskedasticity. We repeat the Monte Carlo exercise 50,000 times for each form of heteroskedasticity. In all cases, we construct confidence intervals using heteroskedasticity-robust standard errors and a critical value, based on a normal approximation, of 1.96. In particular, we employ the HC3 variant of heteroskedasticity-consistent standard errors suggested by MacKinnon and White (1985), adding our proposed HCFGLS estimate only for the SVR estimator.

We repeat this process for three data generating processes, which differ only in the error variance. In all cases, the data generating process takes the following form:

$$\begin{aligned} y =& \beta_0 + \beta_1 x_1 + u, \\ u =& \sqrt{h(x_1)}\theta, \\ x_1 \sim& \mathcal{N}(0, \sigma_x^2), \\ \theta \sim& \mathcal{N}(0, \sigma^2). \end{aligned} \tag{9}$$

Note that for this model $Var(u|x_1) = \sigma^2 h(x_1)$. In the different scenarios, we simply choose different forms for $h(x_1)$. The scenarios for the form of heteroskedasticity are summarized in Table 1.

In all scenarios, OLS and all FGLS variants remain consistent, but we expect OLS to be inefficient and for FGLS with correctly specified error variance function to offer efficiency gains over OLS. The focus of the exercise is on the remaining FGLS variants, since misspecification or poor variance estimation could result in weights that actually reduce the efficiency of the estimator.

All simulations are performed using the R programming language. Standard packages with default settings are used for parametric, nonparametric (`npreg, kknn`), and machine learning tools (`e1071, rpart, randomForest, gbm`), except as detailed next (Racine and Li, 2004; Li and Racine, 2004).[7] The kernel estimator uses a gaussian kernel with a bandwidth of $b = n^{-\frac{1}{5}}$, while nearest neighbor estimation uses four neighbors. For our focal SVR estimator, we estimate one version using default package parameters and another using 5-fold cross-validation to select the values of λ, ϵ, and ϕ.[8]

4.2. Boston housing data

To evaluate the performance of FGLS using SVR on a more realistic dataset, we turn to the commonly-used Boston housing dataset from Wooldridge (2015). In particular, we model the log of community median housing prices in Boston in 1970 as a function of several characteristics:

$$log(price) = \beta_0 + \beta_1 log(nox) + \beta_2 log(dist) + \beta_3 rooms + \beta_4 stratio + u. \tag{10}$$

Here *nox* is nitrogen oxide concentration (parts per million), *dist* is weighted distance (miles) from employment centers, *rooms* is the mean rooms per house, and *stratio* is the mean student-to-teacher ratio.

We choose this dataset both because it is well known and for comparison with recent work by Romano and Wolf (2017). We employ the same wild bootstrap approach as Romano and Wolf (2017), generating 50,000 replicates and evaluating RMSE and size of candidate estimators. Note that in the wild bootstrap approach, the "true" parameter values are those generated via OLS estimation on the main dataset, so that RMSE is calculated based on deviation of the FGLS estimates from the OLS estimates in the original data. For this exercise, we limit our attention to the SVR-based FGLS estimators (with tuning parameters first fixed and then chosen via cross-validation), a kernel estimator, and the WLS-S1 and

[7] Default settings for the SVR estimator use a radial basis function kernel, rescale data to have mean zero and standard deviation 1, and use $\epsilon = 0.1$.

[8] Due to computational complexity and the number of models we estimate, we do a grid search over a limited set of 27 $(\lambda, \epsilon, \phi)$ tuples. Further improvements are likely with a more thorough optimization over those tuning parameters, which would not be as burdensome for estimating a single dataset.

WLS-S2 estimators in order to emphasize comparison of the SVR estimator to proposals in Romano and Wolf (2017).

5. Results

5.1. Simple data generating process

The primary results from our simulations are presented in Tables 2-3. Each table presents relative RMSE along with coverage probability and average CI length for both HC3 standard errors and our new proposed correction to standard errors. These metrics are presented for each combination of heteroskedasticity scenario (row groups) and the estimator used for the skedastic function (columns). In all cases, metrics are calculated across 50,000 Monte Carlo replications, and the data generating process is as given earlier, with $\beta_0 = \beta_1 = 1$, $\sigma^2 = 4$, and $\sigma_x^2 = 5$. Table 2 presents results for $n = 100$, while Table 3 increases n to 500.

The first group of rows in Table 2, which come from a homoskedastic data generating process, match expectations based on classical econometric theory. OLS and FGLS with the correct functional form (which will be equivalent to OLS in this case) provide the most precise estimates and have coverage probabilities near the nominal levels. All other FGLS variants necessarily have higher RMSE due to the increased variance introduced by estimating an incorrect skedastic model. In light of our focus on machine learning, we note that SVR has lower RMSE than the nonparametric variants and only slightly higher RMSE than the parametric estimators. Our proposed corrected standard errors offer coverage probabilities at nominal levels, successfully addressing under-coverage of HC3 standard errors. We note that under homoskedasticity and $n = 100$, the average CI length for the SVR estimator is approximately 30% larger than that under OLS. The data-hungry machine learning approach we propose clearly has costs in small samples when there is no heteroskedasticity. However, using cross-validation reduces these costs substantially, such that the average CI length is only 6% larger than that under OLS.

When the data generating processes is characterized by heteroskedasticity, we begin to see the advantage of the machine learning approach. The second group of results in Table 2 correspond to moderate heteroskedasticity. The SVR-based estimator and the nonparametric estimators outperform OLS in terms of RMSE but do not achieve the efficiency of the FGLS estimator that uses the correct functional form. Further, SVR achieves nearly 87% of the reduction in RMSE achieved by FGLS with the correctly assumed functional form. That reduction is larger than all other approaches except for WLS-S1, which is nearly identical to the correct functional form. The principal advantage is that the efficiency gains under SVR come without functional form assumptions. The test size issues remain when using HC3 standard errors, but our proposed correction again yields the proper coverage probability.

As the form of heteroskedasticity gets more severe (last group of results in Table 2), we see that SVR continues to outperform OLS, this time achieving 98% of the reduction in RMSE that would result from FGLS with correctly specified functional form. Our proposed standard error correction again leads to correctly sized tests.

Increasing the sample size to $n = 500$ yields similar results (Table 3). SVR and kernel regression continue to offer favorable RMSE performance. More importantly, the improvements in RMSE over OLS grow larger. Further, coverage based on HC3 standard errors moves toward the nominal rate in all cases, while our proposed correction continues to offer correct size.

5.2. Boston housing data

Estimation results for the wild bootstrapped Boston housing data are presented in Table 4. The SVR-based FGLS estimator again has small empirical RMSE and confidence intervals in comparison to OLS with HC3 standard errors. Similarly, the coverage probabilities for the corrected standard errors we proposed are near their nominal levels. For comparison, Table 4 also includes the same metrics for the WLS-S1 and WLS-S2 estimators proposed by Romano and Wolf (2017). The proposed SVR estimator has favorable RMSE and confidence interval length and similar, though slightly inferior, coverage properties. Thus, FGLS using support vector regression to estimate the skedastic function, together with our proposed correction to standard errors, appears to have desirable properties on a realistic dataset.

6. Conclusion

In keeping with the trend in econometrics to relax assumptions, the use of feasible generalized least squares (FGLS) has declined in favor of the use of ordinary least squares (OLS) with robust standard errors. As is well known, that approach comes at the cost of some efficiency. In this paper, we have proposed and investigated the use of machine learning tools to estimate the form of heteroskedasticity as part of FGLS estimation. These tools allow for more flexible function estimation than parametric approaches, but may not be as susceptible to outliers as nonparametric alternatives. As a result, machine learning tools, and in particular support vector regression (SVR), may offer a useful compromise in accounting for heteroskedasticity. Further, to account for the flexibility afforded by machine learning tools, we also proposed corrections to robust standard errors computed after FGLS.

Our simulation results suggest that using machine learning tools as a part of FGLS imposes minor efficiency costs under homoskedasticity, but may offer substantial efficiency improvements in the presence of heteroskedasticity. The SVR estimator we propose offers comparable performance to FGLS using kernel estimation of the skedastic function, but tends

to outperform the kernel estimator as the heteroskedasticity becomes more severe. This accords with intuition: SVR and other machine learning tools penalize model complexity, which makes them less susceptible to extreme observations. Together with recent work by Romano and Wolf (2017) providing guidance on inference after FGLS, we believe our approach can contribute to making FGLS a viable, modern tool in researchers' econometrics toolkits.

Appendix

Summary of other machine learning methods
Gradient Boosting

Gradient boosting constructs an overall model as a weighted sum of several simpler models. The intuition is to start with a simple model that is fast to fit (e.g. a regression tree with only one split), then iteratively boost its performance by introducing new additive components to deal with the residuals. Each additive term comes from fitting another simple model to the negative gradient of the loss function chosen for the residuals (e.g. squared loss, absolute loss, Huber loss). That strategy is derived from gradient descent of the loss function; hence each term helps the function approximation get closer to minimizing the selected loss function.

Regression Trees

Regression trees build a multi-dimensional step function approximation to an unknown function $f(x)$. The approximation is constructed by repeatedly splitting a data sample into two sub-samples and making a constant prediction within each sub-sample equal to the mean outcome in that sub-sample. Since there are many ways to split a sample in two, the regression tree algorithm requires further specification. First, the algorithm only splits the sample based on a single variable at a time, i.e. given predictors x_1 and x_2, it might consider a split based on the criterion $x_1 \geq 5$, or $x_1 \geq 10$, or $x_2 \geq 3$, but it will never consider a criterion involving both x_1 and x_2. The best split is defined as that which minimizes the sample variance of outcomes within each sub-sample. The regression tree algorithm searches for this best split using a brute force approach, but methods exist to make that search computationally tractable. Once a split has been identified, the same algorithm is applied to each sub-sample in turn. The algorithm stops once the improvement in fit (reduction of variance within sub-samples) is sufficiently small.[9]

[9]Frequently, the improvement in fit is traded off with a penalty for the number of sub-samples, with the cutoff for "sufficiently small" chosen via cross-validation.

Random Forests

A random forest is a collection of regression trees with randomization injected into the algorithm. The randomization is introduced to overcome the propensity for regression trees to overfit a sample. As a result, regression trees tend to provide variable predictions; intuitively, since regression trees produce step functions, small changes in the sample can produce different breaks for the steps and jumps in predictions. To smooth the predictions offered by a single tree, random forests build many trees and average their predictions. For that averaging process to be useful, the trees must of course offer different predictions. Those different predictions come from two sources of randomization. First, each tree is built using a bootstrapped sample constructed from the original data sample. Second, each time the algorithm considers potential splits of the data, it only considers a random subset of predictor variables. These two sources of randomization result in a collection of different trees, and predictions for any single covariate value are constructed using an average of the predictions of each individual tree.

Results for other machine learning methods

Results of Monte Carlo simulations when using these alternate machine learning methods to estimate the skedastic function are reported in Tables A1 and A2, which are analogous to Tables 2 and 3 in the main text. Results for OLS with HC3 standard errors and FGLS using the correct functional form are reported for reference. Standard errors for machine learning variants are HC3 variants except for SVR, which reports both HCFGLS and HC3 variants. Both gradient boosting and random forest implementations offer RMSE gains that approach those of SVR. However, coverage probabilities for boosting and forests under HC3 are too low to be usable, and we were unable to find a suitable, closed-form estimator for the degrees of freedom of those methods which would enable use of our proposed HCFGLS SE variant.

Acknowledgements

We thank Marc Bellemare, Clément de Chaisemartin, and Doug Steigerwald for helpful comments. We are also grateful for computational resources provided by the Minnesota Supercomputing Institute.

References

Carroll, R. J., 1982. Adapting for heteroscedasticity in linear models. The Annals of Statistics, 1224–1233.

Cribari-Neto, F., 2004. Asymptotic inference under heteroskedasticity of unknown form. Computational Statistics & Data Analysis 45 (2), 215–233.

Drucker, H., Burges, C. J., Kaufman, L., Smola, A., Vapnik, V., et al., 1997. Support vector regression machines. Advances in neural information processing systems 9, 155–161.

Eicker, F., 1963. Asymptotic normality and consistency of the least squares estimators for families of linear regressions. The Annals of Mathematical Statistics, 447–456.

Friedman, J., Hastie, T., Tibshirani, R., 2001. The elements of statistical learning. Vol. 1. Springer series in statistics Springer, Berlin.

Gunter, L., Zhu, J., 2007. Efficient computation and model selection for the support vector regression. Neural Computation 19 (6), 1633–1655.

Harrison, D., Rubinfeld, D. L., 1978. Hedonic housing prices and the demand for clean air. Journal of environmental economics and management 5 (1), 81–102.

Hinkley, D. V., 1977. Jackknifing in unbalanced situations. Technometrics 19 (3), 285–292.

Huber, P. J., 1967. The behavior of maximum likelihood estimates under nonstandard conditions. In: Proceedings of the fifth Berkeley symposium on mathematical statistics and probability. Vol. 1. pp. 221–233.

Li, Q., Racine, J., 2004. Cross-validated local linear nonparametric regression. Statistica Sinica, 485–512.

Lin, Y., Jeon, Y., 2006. Random forests and adaptive nearest neighbors. Journal of the American Statistical Association 101 (474), 578–590.

MacKinnon, J. G., White, H., 1985. Some heteroskedasticity-consistent covariance matrix estimators with improved finite sample properties. Journal of econometrics 29 (3), 305–325.

Micchelli, C. A., Xu, Y., Zhang, H., 2006. Universal kernels. Journal of Machine Learning Research 7 (Dec), 2651–2667.

Newey, W. K., 1994. Series estimation of regression functionals. Econometric Theory 10 (01), 1–28.

O'Hara, M., Parmeter, C. F., 2013. Nonparametric generalized least squares in applied regression analysis. Pacific Economic Review 18 (4), 456–474.

Racine, J., Li, Q., 2004. Nonparametric estimation of regression functions with both categorical and continuous data. Journal of Econometrics 119 (1), 99–130.

Robinson, P. M., 1987. Asymptotically efficient estimation in the presence of heteroskedasticity of unknown form. Econometrica: Journal of the Econometric Society, 875–891.

Romano, J. P., Wolf, M., 2017. Resurrecting weighted least squares. Journal of Econometrics 197 (1), 1–19.

Vapnik, V. N., 1995. The nature of statistical learning theory. Springer.

White, H., 1980. A heteroskedasticity-consistent covariance matrix estimator and a direct test for heteroskedasticity. Econometrica: Journal of the Econometric Society, 817–838.

Wooldridge, J. M., 2010. Econometric analysis of cross section and panel data. MIT press.

Wooldridge, J. M., 2015. Introductory econometrics: A modern approach. Nelson Education.

Ye, J., 1998. On measuring and correcting the effects of data mining and model selection. Journal of the American Statistical Association 93 (441), 120–131.

Tables

Scenario	True $Var(u\|x_1)$
Homoskedasticity	σ^2
Moderate heteroskedasticity	$\sigma^2(\alpha_0 + \alpha_1 x_1^2)$
Severe heteroskedasticity	$\sigma^2 exp(\alpha_0 + \alpha_1 x_1)$

Table 1: Scenarios: true model for $Var(u|x)$, along with implicitly assumed form if, in the second step of FGLS, a researcher simply estimates a model of the log of squared residuals as a function of the main (structural) model regressors.

			Parametric FGLS				Nonparametric FGLS		Machine Learning FGLS	
Heterosk.	Quantity	OLS	Correct	Main	WLS-S1	WLS-S2	KNN	Kernel	SVR	SVR-CV
		(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
None	Rel. RMSE	1	1	1.079	1.008	1.067	1.783	1.248	1.174	1.07
	Coverage prob.	0.945	0.945	0.935	0.944	0.936	0.868	0.824	0.946(0.886)	0.944(0.917)
	Rel. CI length	1	1	1.002	1	0.997	1.062	0.814	1.3(0.945)	1.059(0.969)
Moderate	Rel. RMSE	1	0.428	1.157	0.437	1.132	0.894	0.528	0.502	0.478
	Coverage prob.	0.94	0.945	0.931	0.945	0.931	0.831	0.869	0.947(0.91)	0.944(0.915)
	Rel. CI length	1	0.655	1.035	0.662	1.026	0.692	0.578	0.882(0.659)	0.796(0.652)
Severe	Rel. RMSE	1	0.032	0.032	0.373	0.032	0.179	0.065	0.048	0.047
	Coverage prob.	0.95	0.958	0.958	0.952	0.958	0.758	0.897	0.954(0.924)	0.951(0.929)
	Rel. CI length	1	0.148	0.148	0.753	0.15	0.252	0.201	0.239(0.205)	0.21(0.199)

Table 2: FGLS estimates of slope parameter β_1 with various weight estimation methods using 100 observations per sample and 50,000 Monte Carlo samples. Row groups correspond to error covariance scenarios: homoskedastic, moderate heteroskedasticity, severe heteroskedasticity. Coverage probabilities and CI lengths for all but SVR estimator are for HC3 standard errors; SVR results are for HCFGLS estimator with HC3 results in parentheses.

			Parametric FGLS				Nonparametric FGLS		Machine Learning FGLS	
Heterosk.	Quantity	OLS	Correct	Main	WLS-S1	WLS-S2	KNN	Kernel	SVR	SVR-CV
		(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
None	Rel. RMSE	1	1	1.018	1.001	1.016	2.53	1.222	1.12	1.045
	Coverage prob.	0.947	0.947	0.946	0.947	0.946	0.882	0.866	0.943(0.924)	0.943(0.936)
	Rel. CI length	1	1	1.002	1	1	1.276	0.888	1.048(0.984)	1.007(0.987)
Moderate	Rel. RMSE	1	0.368	1.049	0.384	1.042	1.218	0.425	0.424	0.409
	Coverage prob.	0.948	0.949	0.947	0.949	0.947	0.854	0.921	0.95(0.943)	0.95(0.943)
	Rel. CI length	1	0.608	1.019	0.622	1.016	0.82	0.587	0.66(0.64)	0.645(0.632)
Severe	Rel. RMSE	1	0.009	0.009	0.473	0.009	0.183	0.02	0.016	0.016
	Coverage prob.	0.956	0.953	0.953	0.955	0.954	0.676	0.931	0.953(0.944)	0.951(0.945)
	Rel. CI length	1	0.074	0.074	0.816	0.076	0.202	0.13	0.128(0.125)	0.126(0.124)

Table 3: FGLS estimates of slope parameter β_1 with various weight estimation methods using 500 observations per sample and 50,000 Monte Carlo samples. Row groups correspond to error covariance scenarios: homoskedastic, moderate heteroskedasticity, severe heteroskedasticity. Coverage probabilities and CI lengths for all but SVR estimator are for HC3 standard errors; SVR results are for HCFGLS estimator with HC3 results in parentheses.

		SVR	SVR-CV	WLS-S1	WLS-S2	kernel
(Intercept)	Rel. RMSE	0.515	0.459	0.615	0.661	0.503
	Coverage prob.	0.943	0.936	0.947	0.947	0.775
	Rel. CI length	0.72	0.658	0.779	0.807	0.442
log(nox)	Rel. RMSE	0.562	0.511	0.674	0.7	0.506
	Coverage prob.	0.95	0.942	0.949	0.948	0.828
	Rel. CI length	0.761	0.702	0.813	0.83	0.496
log(dist)	Rel. RMSE	0.446	0.403	0.51	0.597	0.416
	Coverage prob.	0.946	0.938	0.949	0.949	0.777
	Rel. CI length	0.664	0.612	0.713	0.771	0.404
rooms	Rel. RMSE	0.388	0.374	0.504	0.553	0.447
	Coverage prob.	0.932	0.923	0.949	0.949	0.673
	Rel. CI length	0.612	0.583	0.709	0.741	0.35
stratio	Rel. RMSE	0.804	0.715	0.932	0.976	0.69
	Coverage prob.	0.95	0.943	0.949	0.949	0.821
	Rel. CI length	0.911	0.828	0.953	0.975	0.57

Table 4: FGLS estimates of Boston housing data model with 50,000 replicates. RMSE and CI length are reported relative to the RMSE and CI length of OLS with HC3 standard erors. Coverage and CI length for SVR based on proposed corrected standard errors.

			Parametric FGLS	Machine Learning FGLS			
Heterosk.	Quantity	OLS	Correct	SVR	Gradient Boosting	Regression Tree	Random Forest
		(1)	(2)	(3)	(4)	(5)	(6)
None	Rel. RMSE	1	1	1.174	1.152	1.453	1.194
	Coverage prob.	0.945	0.945	0.946(0.886)	0.891	0.899	0.846
	Rel. CI length	1	1	1.3(0.945)	0.9	1.031	0.831
Moderate	Rel. RMSE	1	0.428	0.502	0.534	0.691	0.562
	Coverage prob.	0.94	0.945	0.947(0.91)	0.871	0.858	0.868
	Rel. CI length	1	0.655	0.882(0.659)	0.573	0.638	0.59
Severe	Rel. RMSE	1	0.032	0.048	0.065	0.144	0.062
	Coverage prob.	0.95	0.958	0.954(0.924)	0.858	0.8	0.853
	Rel. CI length	1	0.148	0.239(0.205)	0.168	0.222	0.169

Table A1: FGLS estimates of slope parameter β_1 with various weight estimation methods using 100 observations per sample and 50,000 Monte Carlo samples. Row groups correspond to error covariance scenarios: homoskedastic, moderate heteroskedasticity, severe heteroskedasticity.

			Parametric FGLS	Machine Learning FGLS			
Heterosk.	Quantity	OLS	Correct	SVR	Gradient Boosting	Regression Tree	Random Forest
		(1)	(2)	(3)	(4)	(5)	(6)
None	Rel. RMSE	1	1	1.12	1.138	1.402	1.209
	Coverage prob.	0.947	0.947	0.943(0.924)	0.92	0.929	0.897
	Rel. CI length	1	1	1.048(0.984)	0.963	1.086	0.936
Moderate	Rel. RMSE	1	0.368	0.424	0.446	0.597	0.527
	Coverage prob.	0.948	0.949	0.95(0.943)	0.928	0.913	0.928
	Rel. CI length	1	0.608	0.66(0.64)	0.615	0.664	0.67
Severe	Rel. RMSE	1	0.009	0.016	0.016	0.039	0.016
	Coverage prob.	0.956	0.953	0.953(0.944)	0.923	0.891	0.923
	Rel. CI length	1	0.074	0.128(0.125)	0.111	0.137	0.11

Table A2: FGLS estimates of slope parameter β_1 with various weight estimation methods using 500 observations per sample and 50,000 Monte Carlo samples. Row groups correspond to error covariance scenarios: homoskedastic, moderate heteroskedasticity, severe heteroskedasticity.