Biostatistics (2021) **00**, 00, *pp.* 1–14
doi:10.1093/biostatistics/kxab022

Feature selection for support vector regression using a genetic algorithm

SHANNON B. MCKEARNAN*, DAVID M. VOCK

Division of Biostatistics, University of Minnesota, A460 Mayo Building, MMC 303, 420 Delaware St. SE, Minneapolis, MN 55414, USA

mckea018@umn.edu

G. ELISABETA MARAI

Department of Computer Science, The University of Illinois at Chicago, Chicago, IL 60612, USA

GUADALUPE CANAHUATE

Department of Electrical and Computer Engineering, The University of Iowa, Iowa City, IA 52242, USA

CLIFTON D. FULLER

Department of Radiation Oncology, The University of Texas MD Anderson Cancer Center, Houston, TX 77030, USA

JULIAN WOLFSON

Division of Biostatistics, University of Minnesota, Minneapolis, MN 55414, USA

Summary

Support vector regression (SVR) is particularly beneficial when the outcome and predictors are nonlinearly related. However, when many covariates are available, the method's flexibility can lead to overfitting and an overall loss in predictive accuracy. To overcome this drawback, we develop a feature selection method for SVR based on a genetic algorithm that iteratively searches across potential subsets of covariates to find those that yield the best performance according to a user-defined fitness function. We evaluate the performance of our feature selection method for SVR, comparing it to alternate methods including LASSO and random forest, in a simulation study. We find that our method yields higher predictive accuracy than SVR without feature selection. Our method outperforms LASSO when the relationship between covariates and outcome is nonlinear. Random forest performs equivalently to our method in some scenarios, but more poorly when covariates are correlated. We apply our method to predict donor kidney function 1 year after transplant using data from the United Network for Organ Sharing national registry.

Keywords: Genetic algorithm; Support vector regression; Variable selection.

*To whom correspondence should be addressed.

1. Introduction

Machine learning methods including popular methods such as neural networks, support vector machines, and ensemble methods are powerful and flexible tools for prediction. A âŁœflexibleâŁž model can more closely adapt to the shape of the data it is supplied with, yielding accurate predictions; however, allowing for too much flexibility in a model can lead to overfitting, in which the model fits too closely to the exact data it is trained on and thus loses accuracy in its application to outside contexts. This problem is particularly likely to occur when many patient characteristics or features are used, for example, in clinical risk prediction settings. To counteract overfitting, improve predictive performance, and increase the interpretability of results, the number of patient features used in the model can be narrowed down via feature selection. However, traditional feature selection techniques designed for regression problems, such as LASSO, do not always generalize to more complex machine learning methods. In this article, we propose a feature selection technique for support vector regression (SVR) (Drucker *and others*, 1997), a machine learning technique for predicting continuous outcomes that derives from the popular support vector machine (SVM) method for classification. In a clinical setting, it is important to predict outcomes accurately, while not requiring a number of predictors that would be unreasonable to collect. In a retrospective study, researchers may also be interested in using feature selection to identify key relationships between predictors and the outcome of interest. We first develop our method in a general context to demonstrate its viability and accuracy, then we apply it to a national registry of kidney transplant donors to predict kidney function following transplant.

SVR utilizes a kernel for flexibility and computational efficiency. The use of a nonlinear kernel makes SVR most advantageous when the functional relationship between predictors and outcome is nonlinear, especially because the nonlinear relationship does not need to be prespecified for accurate predictions as they would be for linear regression. Penalized methods for feature selection limit the choice in kernel to finite dimensional transformations and are thus insufficient. While most feature selection methods for SVM have primarily focused on classification without adaptation to SVR, some are able to be easily adapted and others have been proposed for SVR. For example, kernel iterative feature extraction (known as KNIFE), which includes a penalty term on weighted features within a kernel, and kernel-penalized SVM (KP-SVM) have been shown to be successful feature selection methods in the classification setting (Allen, 2013; Maldonado *and others*, 2011). Recursive feature elimination (RFE) has been demonstrated to correctly select features for various kernel problems, including SVR specifically (Dasgupta *and others*, 2019; Dasgupta and Huang, 2020). Other proposed methods for conducting feature selection in the regression case require prespecification of the number of features to select for the model (Yang and Ong, 2011) or increase the tuning required with added SVR hyperparameters (Maldonado and Weber, 2010).

Our approach to SVR feature selection involves a genetic algorithm, an optimization technique modeled after evolutionary processes. With the genetic algorithm, candidate sets of selected variables are “bred" together and passed on over subsequent generations, with “strongest" (i.e., most relevant) set of potential covariates retained for each generation. Genetic recombination operators such as crossover and mutation are applied to maintain some diversity in each generation (Goldberg and Holland, 1988; Leardi *and others*, 1992; Yang and Honavar, 1998). Genetic algorithms have been used in combination with support vector machines for classification problems in areas such as brain MRI classification, blood cell recognition, and microarray-based tumor classification (Kharrat *and others*, 2010; Osowski *and others*, 2009; Peng *and others*, 2003; Li *and others*, 2005). Various methods have jointly used both SVR and the genetic algorithm, most commonly to optimize the parameters of the SVR model (Liu *and others*, 2013; Saravanan and Sailakshmi, 2015; Wu *and others*, 2009). We are not aware of any prior work that tackles the problem of SVR feature selection using the genetic algorithm.

In this article, we briefly review the basics of SVR and genetic algorithms and describe our proposed method for using the genetic algorithm to perform feature selection for SVR. We compare it to other methods for feature selection in a simulation study and illustrate the proposed method in an application to a national registry of kidney transplant donors.

2. Methods

2.1. *Support vector regression*

SVR is an extension of the support vector machine method for classification which constructs an optimal separating hyperplane between classes (Cortes and Vapnik, 1995). We will briefly review the SVR methodology; Smola and Scholkopf (2004) provide a more detailed explanation.

Consider data features $\mathbf{X}_i$ in the space $\mathcal{X} \in \mathbb{R}^L$ along with continuous outcome Y_i for individuals $i = 1, \ldots, n$. One of the advantages of the SVR algorithm is its ability to capture nonlinear relationships. This is implemented in the algorithm by preprocessing the input data $\mathbf{X}_i$ with a transformation to some new feature space. We can write this transformation as the function $\Phi(\boldsymbol{x}) : \mathcal{X} \mapsto \mathcal{H}$, where $\mathcal{H}$ is a high-dimensional, possibly infinite-dimensional, reproducing kernel Hilbert space. Define the function to approximate the relationship between Y_i and $\boldsymbol{X}_i$ in the feature space as an inner product between a weight vector $\boldsymbol{w}$ and the transformed predictors $\Phi(\boldsymbol{x})$ as

$$f(\boldsymbol{x}) = \langle \boldsymbol{w}, \Phi(\boldsymbol{x}) \rangle_{\mathcal{H}} + b. \tag{2.1}$$

Then, the SVR problem consists of optimizing

$$\min_{\boldsymbol{w},b} \left[\frac{1}{2} \|\boldsymbol{w}\|^2_{\mathcal{H}} + C \sum_{i=1}^{n} (\xi_i + \xi_i^*) \right] \tag{2.2}$$

where C is a positive-valued constant regularization parameter and ξ_i, ξ_i^* are slack variables constraining the margin of error. Instead of applying the same loss function as for a support vector machine, we use an ϵ-insensitive loss function Drucker *and others* (1997) defined as

$$L(y, f(x))_\epsilon = \begin{cases} 0 & \text{if } |y_i - f(x_i))| < \epsilon \\ |y_i - f(x_i)| - \epsilon & \text{otherwise} \end{cases}. \tag{2.3}$$

This loss function allows for some errors by treating errors that are small enough, as defined by ϵ, simply as zero, while measuring the loss of larger errors by their magnitude. Minimizing Equation 2.2 under this loss function corresponds to the following constraints on the optimization:

$$\begin{aligned} f(x_i) - y_i &\leq \epsilon + \xi_i \\ y_i - f(x_i) &\leq \epsilon + \xi_i^* \\ \xi_i, \xi_i^* &\geq 0, \forall i = 1, \ldots, n. \end{aligned} \tag{2.4}$$

Note that we can avoid direct computation and specification of the transformation $\Phi(\mathbf{X})$ by applying a kernel function, defined as the inner product $k(\mathbf{x}, \mathbf{x}') = \langle \Phi(\mathbf{x}), \Phi(\mathbf{x}') \rangle_{\mathcal{H}}$. Because we have defined $\mathcal{H}$ as a reproducing kernel Hilbert space, we must select a kernel that meets the Mercer conditions, which includes commonly used kernels such as the Gaussian radial basis function. In an alternate formulation of the SVR optimization problem known as the “Lagrangian dual formulation," the problem can be rewritten such that the transformation $\Phi(\boldsymbol{x})$ is only involved via dot products. This formulation allows for the computational use of a kernel function instead; because the transformation $\Phi(\boldsymbol{x})$ is typically high-dimensional or infinite-dimensional in order to capture the nonlinearity of the function, it would be computationally intensive to directly compute, and a kernel function is more feasible. A quadratic programming algorithm is then applied to solve the dual formulation optimization problem. In our simulations and data application, we apply a radial basis function kernel to account for lack of prior knowledge about the relationships between the data and outcome, rather than assuming a linear or quadratic relationship (two other commonly used kernels). This kernel is defined as $k(\mathbf{x}, \mathbf{x}') = \exp(-\gamma||\mathbf{x} - \mathbf{x}'||^2)$. γ is an additional parameter introduced in the selected kernel function and is jointly tuned via grid search across potential values with the cost parameter C.

Many common approaches to feature selection involve adding a L_1 penalty term to the loss function and inducing sparsity (Tibshirani, 1996; Zou, 2006; Han *and others*, 2015). While this could be applied to the SVR model with a linear kernel, it is not applicable in general. Many of SVR's strengths are derived from its handling of nonlinear data using alternative kernels; thus, a feature selection technique that is applicable for any kernel choice is preferred. Descent-based methods are not easily applicable for this question, as we cannot define the changes in the SVR model based on the inclusion or exclusion of a certain covariate when it is optimized in the dual formulation, which we seek to use for its kernel properties.

2.2. *Genetic algorithms and feature selection*

The genetic algorithm is a general-purpose optimization technique designed to model biological evolutionary practices in which only the fittest individuals reproduce and pass on their genetic information to the following generation, leading to stronger individuals as generations pass. Genetic operators such as mutation and crossover are used to increase diversity of genetic information across generations. Genetic algorithms are well-suited to problems where the objective function implies a complex solution space that cannot be easily traversed by descent-based methods. The genetic algorithm is a flexible technique and has been utilized in many fields of study for a wide range of problems, including but not limited to clustering analyses, multiobjective optimization, and multiprocessor scheduling (Maulik and Bandyopadhyay, 2000; Horn *and others*, 1994; Hou *and others*, 1994).

The first step of the genetic algorithm consists of defining a procedure for encoding the objects to be optimized as binary strings, for example, 011010100. Each string represents a “chromosome," that is, a distinct object, and each binary component is referred to as a “gene." The population is initialized with a number of such chromosomes via random generation. Individuals are then “bred" over multiple generations using various genetic operators such as recombination and mutation. The potential solutions as represented by the chromosomes are evaluated at each generation according to the objective (fitness) function; the “fittest" offspring are retained at each generation. The algorithm ends after a given number of generations or when there is no more diversity in the population of potential solutions. The final solution has the best performance, as determined by the fitness function, among the last generation of chromosomes.

Though not originally designed for feature selection, when the genetic algorithm is adapted to feature selection, the encoding process is straightforward; the binary components of each chromosome are indicators of whether each potential covariate is included or excluded in the statistical model.

The various genetic operators applied are elitism, selection, crossover, and mutation. The generations of chromosomes following the initialization are created via elitism, in which the top-performing portion is automatically advanced to the next generation, and selection, in which some chromosomes are selected to form the “reproducing population." Different selection schemes can be applied. In linear-rank selection, chromosomes are selected with replacement for the reproducing population with probability linearly assigned based on rank of performance according to the designated fitness function. In roulette wheel selection, chromosomes are selected with probability proportionate to the value of their fitness function. Blickle and Thiele (1995) describe several other selection schemes.

Crossover is applied to the reproducing population of chromosomes, with the intent to create diversity in the possible solutions for the next generation. The reproducing population chromosomes are randomly

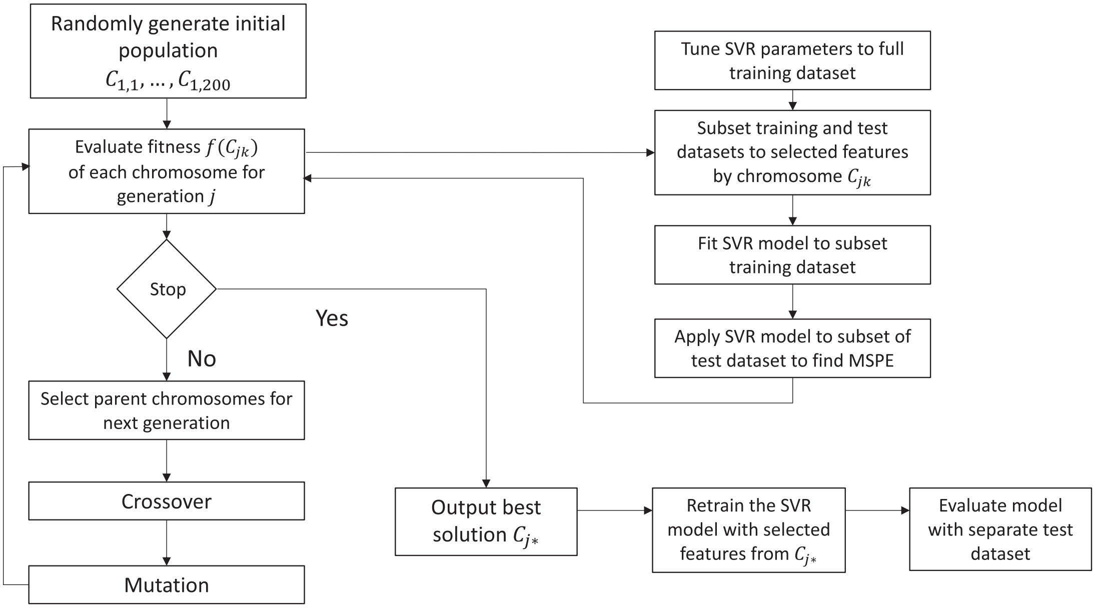
Randomly generate initial population $C_{1,1}, \ldots, C_{1,200}$
Evaluate fitness $f(C_{jk})$ of each chromosome for generation j
Stop
Yes
No
Select parent chromosomes for next generation
Crossover
Mutation
Tune SVR parameters to full training dataset
Subset training and test datasets to selected features by chromosome C_{jk}
Fit SVR model to subset training dataset
Apply SVR model to subset of test dataset to find MSPE
Output best solution C_{j*}
Retrain the SVR model with selected features from C_{j*}
Evaluate model with separate test dataset

Fig. 1. Flowchart detailing the feature selection method for SVR using the genetic algorithm.

matched to form pairs referred to as “parents". Information is swapped, that is, crossover is applied, between the two individuals with a fixed crossover probability. For pairs undergoing crossover, information is swapped between the two parents to form two new chromosomes referred to as “offspring," which replace the parents in the new population. For pairs not undergoing crossover, parent chromosomes are treated as the offspring. A variety of crossover methods can be applied. In single-point crossover, modeled after traditional biological crossover between chromosomes, a randomly selected gene is chosen within the chromosome as the changeover point; then, each offspring receives the genetic material from one parent prior to that point and the other parent after that point. In uniform crossover, information is exchanged at the gene level rather than dividing the chromosome into two segments (Syswerda, 1989). Each gene is exchanged between the two parents with a fixed probability. Additional genetic diversity is then induced via mutation. Each gene within each chromosome is randomly modified by simply changing the indicator of whether or not the particular variable is included in analysis; mutation of a particular gene is performed with a fixed probability that is typically small. Following the iterative application of these genetic operators and the assessment of chromosomes under the fitness function, the final solution indicates a set of covariates to be included for analysis in the prediction method.

2.3. *Feature selection for SVR model*

Unlike other methods such as generalized linear models, SVR does not have an obvious way to organize and explore the model space. Applying the genetic algorithm can therefore add functionality not otherwise available. Our feature selection technique for the SVR model based on the genetic algorithm is displayed graphically in Figure 1 and can be described by the following steps:

(1) For features $\boldsymbol{X} \in \mathbb{R}^L$, define a chromosome C_{jk} composed of L genes $g_{jkl} \in \{0, 1\}$, where each entry g_{jkl} indicates whether or not covariate l is selected for inclusion in the model. For $k = 1, \ldots, K$ chromosomes, initialize the first generation, $j = 1$, of K chromosomes via random generation.

(2) Tune the SVR hyper-parameters cost and γ by grid search, using the training data set with all features.

(3) Evaluate each chromosome C_{jk} under a fitness function $f(C_{jk})$ designed to evaluate the model given the selected features indicated by the g_{jkl} entries of the chromosome.

   i. Subset training and test data sets to selected features by C_{jk}.

   ii. Fit SVR model to subsetted training data using optimal hyper-parameters as found in Step 2.

   iii. Apply SVR model to test dataset and calculate $f(C_{jk})$.

(4) Generate the reproducing population of chromosomes for the following generation.

   i. Elitism: Automatically include the top-performing chromosomes as evaluated by the fitness function in the following generation. Include the top $K * p_E$ chromosomes, where p_E is the proportion of elitism.

   ii. Selection: Select $K * (1 - p_E)$ high-performing chromosomes for inclusion in the reproducing population, using linear-rank selection. Exclude the $K * p_E$ chromosomes from Step 4ii from the selection process.

(5) Apply genetic operators to the $K * (1 - p_E)$ chromosomes generated in step 4ii to form the next generation.

   i. Crossover: Randomly pair the reproducing population chromosomes to form $K * (1 - p_E)/2$ pairs referred to as "parents." Apply uniform crossover between the two individuals with probability p_C.

   ii. Mutation: Randomly modify a small number of g_{jkl} to induce genetic variation. Change the indicator of each 0/1 g_{jkl} in each chromosome in the new population with probability p_M.

(6) Repeat steps 3–5 for $j = 1, \ldots, N$ iterations, leading to a final generation of chromosomes. Select covariates for inclusion in the model per the chromosome with the best performance according to the fitness function f, denoted C_{N*}.

Elitism proportion for use in selection was set to $p_E = 0.05$, a commonly used value. Linear-rank selection was chosen as the selection schematic, in order to maintain diversity in potential solutions in the case that fitness function values vary in magnitude. We applied uniform crossover with a crossover probability $p_C = 0.5$, to eliminate possible dependence on the order in which the variables are coded in the chromosome stemming from single-point crossover. Mutation probability was set to $p_M = 0.1$. Mutation probability is often kept small; the goal is to induce some additional genetic diversity, not completely change the potential solutions.

A population size of $K = 200$ was chosen and the algorithm was repeated for $N = 200$ generations. The genetic algorithm utilizes a fitness function at each iteration to evaluate the performance of each individual. We compared versions of the genetic algorithm implementing mean squared prediction error (MSPE) and Bayesian information criterion (BIC) as fitness functions. Separate training and test data sets were used to calculate the value of the fitness function within the iteration of the genetic algorithm.

3. Simulation study

3.1. *Simulation methods*

We assessed the performance of our feature selection method for SVR with a set of simulations. Since our proposed method combines feature selection (via the genetic algorithm) with flexible regression modeling, we designed our simulation study to investigate the impact of both these elements on prediction accuracy under a wide variety of scenarios. To isolate the effects of feature selection, we compared our method to an SVR without selection. To isolate the effects of the regression modeling, we combined genetic algorithm feature selection with a linear regression model, denoted LR. We compared our method to the RFE method for SVR, using a change-point analysis to determine the stopping point for feature elimination.

We also considered alternative prediction approaches such as LASSO-based variable selection with a linear regression model, as well as a random forest. Average percentage of simulations in which at least one related covariate is selected, average number of related covariates included, average percentage of simulations in which at least one unrelated covariate is selected, average number of additional unrelated covariates included, and average mean squared prediction error (MSPE) are reported to assess the selection of variables and model performance. Performance of the selected set of predictors was evaluated using a separately generated test data set. Five hundred simulations were performed for each scenario. The following different data generating scenarios were considered.

For scenario 1, uncorrelated normal random covariates were generated with underlying associated quadratic terms associated with the outcome. Covariates $\boldsymbol{X}_C = (X_1, \ldots, X_p)$ were generated independently from a normal distribution with mean zero, variance one. Quadratic terms were added to the design matrix $\boldsymbol{X} = [X_1^2\ X_2^2\ \ldots X_{\pi p}^2\ X_1\ X_2\ \ldots X_p]$, where $\pi\ \in [0, 1]$ was varied across scenarios to change the proportion of covariates related to the outcome. We let z_i be the i^{th} row of $\boldsymbol{X}$; the outcome $\boldsymbol{Y}$ was generated as $Y_i = \boldsymbol{\beta}' z_i + \epsilon_i$ with $\epsilon_i \sim \text{Normal}(0, 1)$. We set $\boldsymbol{\beta} = (\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \boldsymbol{\beta}_3)$, where $\boldsymbol{\beta}_1$ and $\boldsymbol{\beta}_2$ are vectors each of length $p * \pi$ corresponding to the quadratic and linear predictors associated with the outcome and $\boldsymbol{\beta}_3$ is a vector of length $p * (1 - \pi)$ set to be zero in all scenarios. The values of $\boldsymbol{\beta}_1$ and $\boldsymbol{\beta}_2$ were varied across simulations to adjust the amount of relative weight of the quadratic term and its related linear term on the outcome respectively, for the following scenarios: positive values of $\boldsymbol{\beta}_1$ only, indicating that the relationship between covariates and outcome is purely quadratic; positive values of both $\boldsymbol{\beta}_1$ and $\boldsymbol{\beta}_2$, but with greater values for one of the two, shifting from a more quadratic to a more linear relationship; positive values of $\boldsymbol{\beta}_2$ only, indicating that the relationship is only linear. Following the data generation in this manner, only covariates $\boldsymbol{X}_C$ were supplied to the prediction methods in order to assess how well the methods would capture the quadratic effects in their predictions and how often the associated main effect terms would be selected via the feature selection processes.

For scenario 2, uncorrelated normal random covariates were generated with underlying associated interaction terms associated with the outcome. Covariates $\boldsymbol{X}_C = (X_1, \ldots, X_p)$ were generated independently from a normal distribution with mean zero, variance one. As in scenario 1, we let z_i be the ith row of $\boldsymbol{X}$ and the outcome $\boldsymbol{Y}$ was generated as $Y_i = \boldsymbol{\beta}' z_i + \epsilon_i$ with $\epsilon_i \sim \text{Normal}(0, 1)$. Pairwise interaction terms were added to the design matrix for all covariates with corresponding main effect terms, such that $\boldsymbol{X} = [X_1X_2\ X_1X_3\ \ldots X_{\pi p-1}X_{\pi p}\ X_1\ X_2\ \ldots X_p]$. We set $\boldsymbol{\beta} = (\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \boldsymbol{\beta}_3)$, where $\boldsymbol{\beta}_1$ is a vector of length $\binom{\pi p}{2}$ corresponding to the interaction terms, $\boldsymbol{\beta}_2$ is a vector of length $p * \pi$ corresponding to the linear terms, and $\boldsymbol{\beta}_3$ is a vector of length $p * (1 - \pi)$ set to zero. As with the quadratic data generation scenario, interaction terms were not supplied to the methods for use in feature selection or prediction.

For scenario 3, correlated multivariate normal data was generated with varying levels of association with the outcome. The $N \times p$ design matrix $\boldsymbol{X}$ was formed by partitioning into equal-sized blocks of correlated covariates. Each block was generated from a multivariate normal distribution with mean zero and first order autoregressive correlation structure with $\rho = 0.5$. Given z_i the ith row of $\boldsymbol{X}$, the outcome $\boldsymbol{Y}$ was generated as $Y_i = \boldsymbol{\beta}' z_i + \epsilon_i$ with $\epsilon_i \sim \text{Normal}(0, 1)$. The value of $\boldsymbol{\beta}$ was defined to match the block design of the covariates in two different ways: one such that the main effect terms were correlated with irrelevant terms but uncorrelated with each other, and one such that the main effect terms were correlated with each other but not with irrelevant terms. Technical details of the settings for each scenario, including values of $\boldsymbol{\beta}$, are presented in Section 1 of the Supplementary Material available at *Biostatistics* online.

3.2. *Simulation results*

In scenario 1, we assessed the performance of our method in the presence of underlying quadratic relationships between the outcome and covariates, while only providing the corresponding linear covariates to the model. Results are displayed in Figures 2 and 3. Tables displaying the full results are presented in Section 2

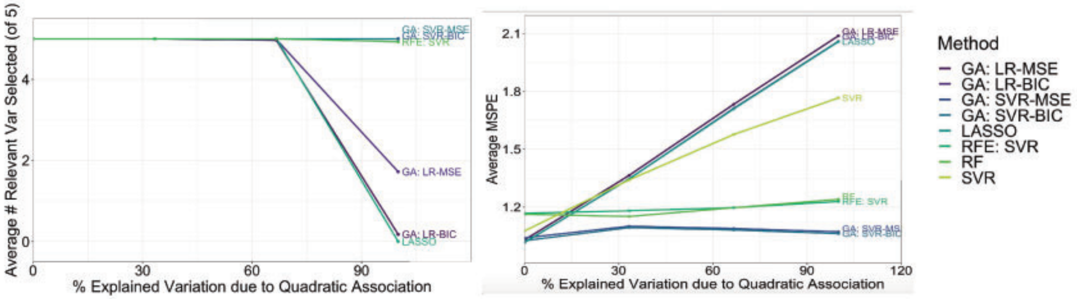

Fig. 2. Results of simulation 1 setting with 50 total covariates **X**, 5 of them truly associated with the outcome **Y**. Trend in average MSPE and number of variables selected by various prediction methods as the amount of true variation in **Y** due to **X** stemming from the quadratic covariates is increased. Random forest method is not displayed for number of variables due to lack of feature selection.

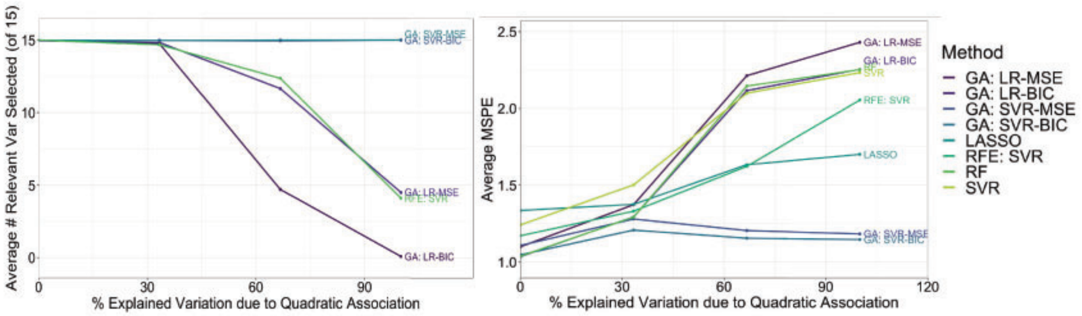

Fig. 3. Results of simulation 1 setting with 300 total covariates **X**, 15 of them truly associated with the outcome **Y**. Trend in average MSPE and number of variables selected by various prediction methods as the amount of true variation in **Y** due to **X** stemming from the quadratic covariates is increased. Random forest method is not displayed for number of variables due to lack of feature selection.

of the Supplementary Material available at *Biostatistics* online. Our feature selection method for SVR selected all of the relevant covariates in both the 50 and 300 covariate scenarios, where the proportion of relevant covariates π is equal to 0.1 and 0.05, respectively, regardless of the level of quadratic relationship. On average, the BIC fitness function in the genetic algorithm yielded the selection of fewer extraneous, unrelated variables than the use of the MSE fitness function, in all scenarios. In addition, the accuracy of predictions from our method, as measured by MSPE, was lower in all scenarios than that of the SVR model implemented without any feature selection. While the LASSO method selected the correct variables when the relationship was primarily linear, it did not do so when the relationship shifted to quadratic. The same pattern was seen for the genetic algorithm feature selection method applied to linear regression models. The random forest method maintained roughly the same accuracy of predictions no matter the level of quadratic relationship between outcome and predictors; additionally, no feature selection was implemented with the random forest. The RFE method performed similarly to random forest in the 50 covariate scenario, however, it does include feature selection and it correctly selected predictors. In the 300 covariate scenario, random forest outperformed our method in terms of MSPE when the relationship between outcome and predictors was mostly quadratic but still maintained a linear component. In all other cases, our method performed better. While RFE and the GA applied to linear regression selected some of

Table 1. *Results of simulation 2 setting with 50 total covariates* **X***, 5 of them truly associated with the outcome* **Y***, and 10 pairwise underlying interaction effects not supplied to the model for selection or prediction. Columns represent the proportion of simulations in which at least one of the relevant covariates is selected, the number of relevant covariates selected on average (out of 5), the proportion of simulations in which at least one unassociated covariate is selected, the number of unassociated covariates selected on average, and a scaled MSPE described in the footnote.*

Feature selection	Model fit	% Selection relevant (1+)	Avg # relevant Var	% Other Var (1+)	Avg # additional Var	Avg % variation unexplained by Pred.[†]
GA: MSE	LR	100	5	100	16.12	55.36
GA: BIC	LR	100	5	6.0	0.07	53.26
GA: MSE	**SVR**	100	5	95.2	4.30	24.31
GA: BIC	**SVR**	100	5	27.4	0.30	22.06
RFE	SVR	100	5	100	3	22.76
LASSO	LR	100	5	0.6	0.12	53.26
None	RF	100	5	100	45.0	27.47
None	SVR	100	5	100	45.0	51.22

Bold values refer to our method developed in the paper and are highlighted for reference. [†]Column displays average across simulation iterations of the proportion of true variation in **Y** due to **X** that is left unexplained by the prediction. For example, in the case where the total variance in **Y** is 2, where half is due to the error term and half is due to the covariates **X**, a MSPE of 1.05 would be reported as 5% in this column. Of the explainable variance 1, the model misses 5%.

the correct predictors in the 300 covariate scenario, the proportion declined as the relationship between outcome and predictors became more quadratic.

In scenario 2, we assessed the performance of our method when there are true interactive effects between covariates, while only providing the corresponding linear covariates to the model. Results are displayed in Table 1. In this scenario with 50 covariates, when 10 underlying interactive terms were not supplied to the prediction method but 5 related linear terms were, our method for SVR with either the MSE or BIC fitness function used in the genetic algorithm and RFE with SVR yielded the lowest MSPEs. While the random forest method predictions were nearly as accurate, the linear model (using either LASSO or the genetic algorithm for feature selection) and the SVR model without any feature selection left over twice as much of the variance due to X unexplained on average. In addition, our method with the BIC fitness function used in the genetic algorithm selected the fewest extraneous variables for inclusion in the model. The genetic algorithm using BIC in combination with linear regression and LASSO methods also picked up very minimal extraneous variables. All methods correctly included the covariates that were truly associated with the outcome.

In scenario 3, we compared the performance of our method to competing methods when the covariates are correlated. Results are displayed in Tables 2 and 3. Fifty covariates were used in this scenario, comprised of equal-sized correlated blocks of 10 covariates. We applied two different implementations of correlated data: one in which truly related covariates were correlated with extraneous variables and one in which truly related covariates were correlated with each other. In both cases, all methods correctly selected the relevant variables for inclusion in the model. Similar trends in both the accuracy of the predictions and the variables selected were seen in both cases. The implementation of our feature selection method for SVR yielded a sizable decrease in MSPE compared to SVR without any feature selection. However, in this scenario, LASSO and the linear regression with a genetic algorithm and BIC fitness function methods yielded both the lowest MSPEs and the fewest amounts of extraneous variables selected for inclusion in

Table 2. *Results of simulation 3 setting with 50 total covariates* **X**, *5 of them truly associated with the outcome* **Y**, *and each correlated to 9 variables unrelated to the outcome with AR-1 correlation structure.*

Feature selection	Model fit	% Selection relevant (1+)	Avg # relevant Var	% Other Var (1+)	Avg # additional Var	Avg % variation unexplained by Pred.[†]
GA: MSE	LR	100	5	100	15.2	2.96
GA: BIC	LR	100	5	7.8	0.08	1.62
GA: MSE	**SVR**	100	5	100	12.39	3.87
GA: BIC	**SVR**	100	5	15.0	0.16	2.53
RFE	SVR	100	5	100	3	15.63
LASSO	LR	100	5	100	0.12	1.6
None	RF	100	5	100	45.0	14.55
None	SVR	100	5	100	45.0	7.17

Bold values refer to our method developed in the paper and are highlighted for reference. [†]Column displays average across simulation iterations of the proportion of true variation in **Y** due to **X** that is left unexplained by the prediction. For example, in the case where the total variance in **Y** is 2, where half is due to the error term and half is due to the covariates **X**, a MSPE of 1.05 would be reported as 5% in this column. Of the explainable variance 1, the model misses 5%.

Table 3. *Results of simulation 3 setting with 50 total covariates* **X**, *10 of them truly associated with the outcome* **Y** *and correlated with each other with AR-1 block correlation structure.*

Feature selection	Model fit	% Selection relevant (1+)	Avg # relevant Var	% Other Var (1+)	Avg # additional Var	Avg % variation unexplained by Pred.[†]
GA: MSE	LR	100	10	100	13.47	3.23
GA: BIC	LR	100	10	0	0	2.07
GA: MSE	**SVR**	100	10	100	11.30	3.57
GA: BIC	**SVR**	100	10	0.6	0.01	2.39
RFE	SVR	100	7.99	0	0	31.38
LASSO	LR	100	10	0.02	2.0	1.96
None	RF	100	10	100	40.0	21.97
None	SVR	100	10	100	40.0	7.31

Bold values refer to our method developed in the paper and are highlighted for reference. [†]Column displays average across simulation iterations of the proportion of true variation in **Y** due to **X** that is left unexplained by the prediction. For example, in the case where the total variance in **Y** is 2, where half is due to the error term and half is due to the covariates **X**, a MSPE of 1.05 would be reported as 5% in this column. Of the explainable variance 1, the model misses 5%.

the model. As seen in the previous scenarios, the use of the BIC fitness function in the genetic algorithm as opposed to the MSE fitness function led to far fewer additional variables being selected. In addition, the random forest and RFE methods yielded predictions less accurate than any other method.

In the Supplementary Material available at *Biostatistics* online, we present additional simulation results for alternative signal-to-noise ratios and an alternate choice in kernel for SVR. The overall trends in the results presented here are similar to those seen in the lower signal-to-noise ratio scenario presented in the Supplementary Material available at *Biostatistics* online.

Table 4. *Performance results of GA-SVR method and comparison methods applied to predict eGFR for living transplant kidney donors at 1 year post-transplant.*

Feature Selection	Model Fit	No. Var. Selected	MSPE
GA: MSE	LR	15	112.6
GA: BIC	LR	1	118.5
GA: MSE	**SVR**	13	110.3
GA: BIC	**SVR**	1	114.2
RFE	SVR	5	120.1
LASSO	LR	2	114.5
None	RF	32	104.1
None	SVR	32	122.6

Bold values refer to our method developed in the paper and are highlighted for reference.

4. Application to data

Patients in need of a kidney transplant can benefit greatly from a living donor transplant. While some evidence has shown minimal long-term consequences of kidney donation, other evidence has shown an increased risk of end-stage renal disease among donors (Ibrahim *and others*, 2009). We aim to further investigate the impact of kidney donation on the donor's renal function post-transplant and identify factors that could indicate a donor is at additional risk for future complications. Data for this analysis were collected from January 2015 to December 2019 from the United Network for Organ Sharing national registry. We focus our analysis on patients aged 18 or older who donated kidneys for living donor transplant.

A total of 21 121 patients are included in the analysis, and 32 covariates are available regarding donor demographics, quality of the transplanted organ, and donor health. Missing covariate data were imputed using Multivariate Imputation by Chained Equations (MICE). Descriptive characteristics and imputation details are presented in Section 4 of the Supplementary Material available at *Biostatistics* online. We consider the outcome estimated glomerular filtration rate (eGFR) 1 year post-transplant; eGFR is a measure of kidney function based on the patient's creatinine, age, and gender. We evaluate performance of the methods by splitting the data into separate training, evaluation (for use in evaluating the genetic algorithm fitness function), and test data sets. We use 60% of the data for training data, 20% for evaluation data, and 20% for testing data.

Results of the application of our method and comparative methods to predict eGFR at 1 year following kidney transplant are displayed in Table 4. We find that using our feature selection method for SVR with an MSE fitness function yields a lower mean squared prediction error (MSPE) than all tested alternatives except random forest, most significantly improving performance over SVR without any feature selection.

The genetic algorithm with a BIC fitness function heavily penalizes the number of variables selected in this application, selecting only one variable, pre-transplant eGFR, to be included in the model for both linear regression or SVR. The genetic algorithm with a MSE fitness function selects 15 variables for inclusion in the linear regression and 13 variables for inclusion in the SVR model, though there is some variation in the variables selected. In addition to the donor's pre-transplant eGFR, both methods select donor blood type, education history, race, and BMI. RFE used with SVR selects similar variables: pre-transplant eGFR, race, blood type, and BMI.

These results are not inconsistent with the results of the simulation studies we performed, in which our methods had lower MSPE than SVR without feature selection in all scenarios examined. Some improvement in MSPE was seen in our method when using the MSE fitness function, which allowed for more variables to be included in a richer model. In methods with fewer variables selected, the same few were seen in common, indicating consistency across the methods in identifying predictive variables of post-transplant eGFR. In addition, these results illustrate the impact of different fitness criterion. In this example, because of the strong relationship between one specific predictor and the outcome, the BIC fitness function selects only the one key predictor and misses additional predictors that are beneficial to the model. This contrasts to our simulation results, in which relevant predictors had equal impact and the results generally showed a lower MSPE for the BIC fitness criterion than the MSE fitness criterion. Analysts applying our method should be mindful of the structure of their own data when choosing the fitness criterion.

5. Discussion

The biggest gain offered by SVR compared to more traditional statistical methods is its well-handling of nonlinear data without the need to prespecify transformed or interaction terms; however, the same structure of SVR that yields this benefit impedes the development of typical feature selection methods. To overcome this disadvantage, we developed a genetic algorithm based approach that allows for ease of continued use of SVR's various kernel functions. The computation of nonlinear SVR does not directly depend on the dimensionality of the data, but rather the number of samples; while some may take this to mean that SVR is inherently resistant to typical problems of dimensionality, we have not found that to be the case. Our GA-SVR feature selection method yields better predictive accuracy than SVR without feature selection in all simulation experiments examined as well as in our data application, indicating otherwise. Random forest, an alternative machine learning technique also popular for its predictive accuracy, performs comparably to our method in many settings of our simulation setting, though not out-performing it. Notably, our method out-performs random forest most significantly in the case of correlated predictors. This is also a scenario in which the LASSO method is known to have poor performance and indicates a situation in which use of our method may be most advantageous. In addition, we found that our method yielded considerable improvement in predictive accuracy over LASSO when covariates had quadratic or interactive effects.

As with all uses of SVR, proper tuning of parameters and choice in kernel function are necessary steps for good performance. Choice in kernel is discussed extensively in existing literature. We conducted a sensitivity analysis for our simulation study using a linear kernel instead of a radial basis kernel for SVR; results are presented in Section 5 of the Supplementary Material available at *Biostatistics* online. As expected, our method with a linear kernel performed well when the data generating mechanism was linear, and poorly when it was not. Given that the use of the radial basis kernel when the data generation was linear was comparable to alternate methods, we suggest using the radial basis kernel in cases when the data generation is either unknown or known to be nonlinear. A further examination of additional alternate kernels may be of interest. Alternative methods for tuning of parameters have been developed; it may be of interest in the future to examine their impact in conjunction with our feature selection method. Computational effort for our method is not insurmountable especially with the implementation of stopping criteria; however, due to the iterative nature of the genetic algorithm, if computational speed is a major concern an alternative such as random forest may be preferable. A further examination of alternative fitness functions used in the genetic algorithm and their impact on feature selection is also warranted.

6. Software

R code is available on GitHub at https://github.com/smckearnan/ga-svr.

Supplementary material

Supplementary material is available online at http://biostatistics.oxfordjournals.org.

Acknowledgments

Conflict of Interest: None declared.

Funding

NCI R01CA214825 and NCI R01CA225190.

References

Allen, G. I. (2013). Automatic feature selection via weighted kernels and regularization. *Journal of Computational and Graphical Statistics* **22**, 284–299.

Blickle, T. and Thiele, L. (1995). A comparison of selection schemes used in genetic algorithms. *Technical Report* 11. Computer Engineering and Communication Networks Lab TIK.

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine Learning* **20**, 273–297.

Dasgupta, S., Goldberg, Y. and Kosorok, M. R. (2019). Feature elimination in kernel machines in moderately high dimensions. *Annals of Statistics* **47**, 497–526.

Dasgupta, S. and Huang, Y. (2020). Selecting biomarkers for building optimal treatment selection rules by using kernel machines. *Journal of the Royal Statistical Society: Series C (Applied Statistics)* **69**, 69–88.

Drucker, H., Burges, C. J. C., Kaufman, L., Smola, A. and Vapnik, V. (1997). Support vector regression machines. In: Jordan, M., Kearns, M. and Solla, S. (editors), *Advances in Neural Information Processing Systems*. Denver, CO, USA. pp. 155–161.

Goldberg, D. E. and Holland, J. H. (1988). Genetic algorithms and machine learning. *Machine Learning* **3**, 95–99.

Han, S., Pool, J., Tran, J. and Dally, W. (2015). Learning both weights and connections for efficient neural network. In: Cortes, C., Lawrence, N., Lee, D., Sugiyama, M. and Garnett, R. (editors), *Advances in Neural Information Processing Systems*, Montreal, Canada: Volume 28. Curran Associates, Inc.

Horn, J., Nafpliotis, N. and Goldberg, D. E. (1994). A niched Pareto genetic algorithm for multiobjective optimization. In: Michalewicz, Z. (editor), *Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence*, Volume 1. Piscataway, New Jersey: IEEE Service Center. pp. 82–87.

Hou, E. S. H., Ansari, N. and Ren, H. (1994). A genetic algorithm for multiprocessor scheduling. *IEEE Transactions on Parallel and Distributed Systems* **5**, 113–120.

Ibrahim, H. N., Foley, R., Tan, L. P., Rogers, T., Bailey, R. F., Guo, H., Gross, C. R. and Matas, A. J. (2009). Long-term consequences of kidney donation. *New England Journal of Medicine* **360**, 459–469.

Kharrat, A., Gasmi, K. and Messaoud, M. B. E. N. (2010). A hybrid approach for automatic classification of brain MRI using genetic algorithm and support vector machine. *Leonardo Journal of Sciences* **9**, 71–82.

Leardi, R., Boggia, R. and Terrile, M. (1992). Genetic algorithms as a strategy for feature selection. *Journal of Chemometrics* **6** 267–281.

Li, L., Jiang, W., Li, Xia, M., Kathy L., Guo, Z., Du, L., Wang, Q., Topol, E. J., Wang, Q. and Rao, S. (2005). A robust hybrid between genetic algorithm and support vector machine for extracting an optimal feature gene subset. *Genomics* **85**, 16–23.

Liu, S., Tai, H., Ding, Q., Li, D., Xu, L. and Wei, Y. (2013). A hybrid approach of support vector regression with genetic algorithm optimization for aquaculture water quality prediction. *Mathematical and Computer Modelling* **58**, 458–465.

Maldonado, S. and Weber, R. (2010). Feature selection for support vector regression via Kernel penalization. In: *Proceedings of the International Joint Conference on Neural Networks*. Barcelona, Spain: IEEE, pp. 1–7.

Maldonado, S., Weber, R. and Basak, J. (2011). Simultaneous feature selection and classification using kernel-penalized support vector machines. *Information Sciences* **181**, 115–128.

Maulik, U. and Bandyopadhyay, S. (2000). Genetic algorithm-based clustering technique. *Pattern Recognition* **33**, 1455–1465.

Osowski, S., Siroi, R., Markiewicz, T. and Siwek, K. (2009). Application of support vector machine and genetic algorithm for improved blood cell recognition. *IEEE Transactions on Instrumentation and Measurement* **58**, 2159–2168.

Peng, S, Xu, Q., Ling, X. B., Peng, X., Du, We. and Chen, L. (2003). Molecular classification of cancer types from microarray data using the combination of genetic algorithms and support vector machines. *FEBS Letters* **555**, 358–362.

Saravanan, P. and Sailakshmi, P. (2015). Missing value imputation using fuzzy possibilistic c means optimized with support vector regression and genetic algorithm. *Journal of Theoretical and Applied Information Technology* **72**, 34–39.

Smola, A. J. and Scholkopf, B. (2004). A tutorial on support vector regression. *Statistics and Computing* **14**, 199–222.

Syswerda, G. (1989). Uniform crossover in genetic algorithms. In: David Schaffer, J. (editor), *Proceedings of the Third International Conference on Genetic Algorithms*. San Francisco, CA, USA: Morgan Kaufmann Publishers, pp. 2–9.

Tibshirani, R. (1996). Regression shrinkage and selection via the LASSO. *Journal of the Royal Statistical Society: Series B (Methodological)* **58**, 267–288.

Wu, C. H., Tzeng, G. H. and Lin, R. H. (2009). A Novel hybrid genetic algorithm for kernel function and parameter optimization in support vector regression. *Expert Systems with Applications* **36**, 4725–4735.

Yang, J. and Honavar, V. (1998). Feature subset selection using a genetic algorithm. *IEEE Intelligent Systems* **13**, 44–49.

Yang, J. B. and Ong, C. J. (2011). Feature selection using probabilistic prediction of support vector regression. *IEEE Transactions on Neural Networks* **22**, 954–962.

Zou, H. (2006). The adaptive lasso and its oracle properties. *Journal of the American Statistical Association* **101**, 1418–1429.

[*Received June 16, 2020; revised February 12, 2021; accepted for publication March 21, 2021*]