Applied Intelligence (2020) 50:4117–4142
https://doi.org/10.1007/s10489-020-01687-3

An interpretable regression approach based on bi-sparse optimization

Zhiwang Zhang [1,2] · Guangxia Gao [3] · Tao Yao [1,2] · Jing He [4] · Yingjie Tian [5]

Published online: 11 July 2020

Abstract
Given the increasing amounts of data and high feature dimensionalities in forecasting problems, it is challenging to build regression models that are both computationally efficient and highly accurate. Moreover, regression models commonly suffer from low interpretability when using a single kernel function or a composite of multi-kernel functions to address nonlinear fitting problems. In this paper, we propose a bi-sparse optimization-based regression (BSOR) model and corresponding algorithm with reconstructed row and column kernel matrices in the framework of support vector regression (SVR). The BSOR model can predict continuous output values for given input points while using the zero-norm regularization method to achieve sparse instance and feature sets. Experiments were run on 16 datasets to compare BSOR to SVR, linear programming SVR (LPSVR), least squares SVR (LSSVR), multi-kernel learning SVR (MKLSVR), least absolute shrinkage and selection operator regression (LASSOR), and relevance vector regression (RVR). BSOR significantly outperformed the other six regression models in predictive accuracy, identification of the fewest representative instances, selection of the fewest important features, and interpretability of results, apart from its slightly high runtime.

Keywords Data mining · Multi-kernel learning · Sparse learning · Zero-norm regularization · Support vector regression

1 Introduction

Regression is an important data mining technique that is known to fit input points from training data with high accuracy. Value prediction is a common application of regression that can be found in many domains, such as finance, telecommunication, economics and management, power and energy, web customer management, industrial production, and scientific computing [40, 59]. Regression predicts values by constructing functions that can estimate the relationship between independent and dependent variables. Many regression methods have been proposed for value prediction [3, 6]. These include linear regression [22], nonlinear regression [59], polynomial regression [13], ridge regression [28], stepwise regression [25], quantile regression [44], least angle regression [27], lasso regression [77], elastic net regression [89], neural networks, and support vector regression (SVR) [5, 68].

SVR is considered an effective prediction method for value forecasting because of its higher generalization on small and medium datasets than that of some traditional methods. SVR can identify a small number of support vectors from the training set and use them to define the regression function. Particularly, in SVR, a tube hyperplane with a smaller radius is first constructed so that most training data are within the tube and the minority are outside the tube with errors. Then SVR can be defined as a quadratic optimization problem that employs a regularization parameter to trade-off between model complexity and total fitting error [1, 5, 19, 36, 68, 73]. The former can be used to minimize the ℓ_2-norm of the weight vector, and the latter to minimize the sum of errors of input points with constraint violations. Although SVR can find support vectors from the training set, it can't distinguish important features from other features. That is, SVR has no ability of feature selection for value prediction, hence it has a lack of result explainability.

✉ Zhiwang Zhang
zzwmis@163.com

[1] School of Information and Electrical Engineering, Ludong University, Yantai 264025, China

[2] Yantai Research Institute of New Generation Information Technology, Southwest Jiaotong University, Yantai 264000, China

[3] Shandong Technology and Business University, Yantai 264005, China

[4] Swinburne University of Technology, Melbourne, VIC 3122, Australia

[5] Research Center on Fictitious Economy and Data Science, Chinese Academy of Sciences, Beijing 100190, China

Some variations of SVR with advanced features have been developed in the last two decades. According to the dual optimization problem of SVR, the weight vector is expressed as a linear combination of input points and Lagrange multipliers. If the ℓ_2-normof the weight vector in SVR is replaced with the ℓ_1-normof Lagrange multipliers, then we have the ℓ_1-norm SVR (L1SVR) model [39, 85, 88]. Because L1SVR is essentially a linear programming (LP) problem, it is also called LPSVR [39, 82, 85]. LPSVR can predict output values and it can find fewer support vectors than SVR. However, for LPSVR more variables are required to remove absolute value function, which leads to highly computational complexity.

Different from SVR, least squares SVR (LSSVR) minimizes the sum of squared errors instead of the sum of errors [53, 63, 72]. Due to simple constraints, the computational complexity is remarkably reduced when solving the dual problem of LSSVR. The dual LSSVR model can be formulated as a system of linear equations or an unconstrained convex quadratic optimization problem. In the dual LSSVR model the inverse of the regularization constant is added to the diagonal elements of the Hessian matrix. But LSSVR can hardly identify important instances and features for value prediction.

Multi-kernel learning (MKL) methods that substitute the single kernel in the SVR model (MKLSVR) for the convex combination of multi-kernel functions of different features have recently been proposed [34, 54, 55]. MKL methods have been applied to many practical problems [35, 49, 52]. In light of the way of combining various kernels, MKL is classified the types of MKL as linear and nonlinear. Considering the role of data in the model, MKL has both data-dependent (data-driven) and data-independent methods [29, 38, 54, 69, 70, 87]. These methods generally combine MKL with SVR or LSSVR [65], and they can achieve better predictive accuracy than some single kernel methods. However, these methods have likewise no ability of feature selection in addition to high computational cost.

The above regression methods have a notable drawback: they are unable to simultaneously identify important instances and features from the training set to obtain interpretable results by using instance sparsity and feature selection, especially for large-scale and high-dimensional data. To address this problem, one line of research has focused on feature transformation or feature selection, also called dimensionality reduction [11, 16, 21, 45]. These methods mainly adopt a ranking filter [26], relief algorithm [43], decision tree [84], principle component analysis (PCA) [14, 47, 64], discriminant analysis (DA) [2, 47, 48], singular value decomposition [80], rough set [45, 76], function data analysis [7, 9, 10, 56, 57]. However, feature selection and regression are often conducted in different feature spaces. Due to the potential inconsistency and loss of information, it is difficult to achieve high accuracy and sufficient interpretability with the feature selection approach.

Other researchers have integrated sparse learning methods into regression or classification models to implement prototype discovery or feature selection [3, 8, 15, 18, 60, 71, 75]. These sparse learning methods mainly contain concave minimization,ℓ_0-norm regularization, and ℓ_1-norm regularizations [12, 39, 46, 50, 82]. The concave minimization approach for feature selection uses an exponential function to approximate the number of nonzero elements of the weight vector. The ℓ_0-norm regularization can achieve the sparse solution in theory, however, due to its discontinuity, it is difficult to directly solve the corresponding mathematical problem. Thus, a proper approximation function is first defined to replace the ℓ_0-norm and obtain the sparse instance and feature sets. Although the ℓ_1-normis a non-smooth function, it is convex and can be used for feature sparsification. Some hybrid methods of different norms have been proposed to obtain instance- or feature-reduction. These methods include sparsity-based gradient descent [32], least absolute shrinkage and selection operator (LASSO) [51, 58, 74, 77, 83, 90], least angle regression (LAR) [27], relevance vector machine [20, 78], sure independence screening (SIS) [30], and elastic net [89]. However, their application is limited to either feature selection or instance reduction. The instance-sparsity methods showed that prototype instances can be used as a benchmark for predicting output values of unseen input points [17, 37, 41, 42, 61, 85]. Specifically, the relevance vector regression (RVR) introduces a probabilistic Bayesian learning framework for obtaining sparse solutions to value prediction and it uses fewer support vectors to achieve the better predictive performance than SVR [20, 78]. However, it has no ability to select a feature subset from a given training set. For the LASSO regression (LASSOR), by using the ℓ_1-norm regularization [62, 67], some important features are selected from the initial feature set. But it is unable to obtain sparse instance set and it is very difficult to introduce the kernel trick to the primal LASSO regression model to solve nonlinearly predictive problems. Under the framework of LSSVR, a least squares regression model based on bi-sparse optimization is proposed to address the problem unable to obtain sparse instance and feature solutions for LSSVR [86]. By solving two systems of equations or two unconstrained quadratic programming problems sparse instance and feature solutions are obtained. Besides, the regression prediction method verifies that combining both instance and feature sparsity increased the predictive performance and model interpretability.

This study's main motivation is to construct a bi-sparse optimization-based regression (BSOR) to enhance accuracy, generalization, and interpretability in value prediction. In the framework of SVR, the proposed regression model can select both relevant instances and features with sparsification methods based on the reconstructed row and column kernel matrices. These representative instances and important features are taken as a benchmark for predicting output values of unseen input points.

The weighted distance with respect to a coefficient vector from an unseen input point to prototype instances is a benchmark for prediction and explanation, while another weighted distance regarding a weight vector between the features of an unseen input point and important features is the basis for forecasting and interpretation. Obviously, sparse coefficient or weight vector indicates whether an input point or attribute is a prototype instance or an important feature or not respectively. By using the instance- and feature-sparsity methods, the predictive accuracy, generalization, and interpretability can be enhanced, which has great practical implications. Therefore, this is our main contribution to predictive regression.

The rest of this paper is organized as follows: We first review the basic theories of SVR. Then we present the BSOR model, corresponding algorithm, and simulation. Next, we describe the experimental results of predictive evaluation on real datasets, the predictive results and comparison analysis, importance analysis of instances and features extracted by BSOR, and analysis of experimental results. We conclude after discussing the study's implications.

2 Support vector regression

In this section, we briefly review the basic principles of SVR, the dual representation theory, and kernel tricks [19, 24, 36] for value prediction. For a regression problem, given training data $T = \{(x_i, y_i)\}_{i=1}^{n}$ $(i \in \mathbb{N})$ with an attribute or feature set $F = \{f_m\}_{m=1}^{d}$ $(m \in \mathbb{N}, f_m \in \mathbb{R}^n)$, each input point or instance x_i $(x_i \in \mathbb{R}^d)$ corresponds to a continuous output value y_i $(y_i \in \mathbb{R})$, where d is the dimensional size of the input space and n is the sample size.

Given the training set T, and a basis function $\phi(x)$ that maps any input point x $(x \in \mathbb{R}^d)$ from the input space to a high-dimensional feature space, the regression function is defined as $y = w^T\phi(x) + b$, where w $(w \in \mathbb{R}^p, p \geq d)$ is a weight vector and b $(b \in \mathbb{R})$ is a scalar. At the same time, two $\varepsilon-$band hyperplanes $y - (w^T\phi(x) + b) = \varepsilon$ and $(w^T\phi(x) + b) - y = \varepsilon$ are constructed so that as many input points as possible are inside a tube with diameter 2ε between two $\varepsilon-$band hyperplanes, where ε is a user-specified parameter. For those input points that are outside the tube, the slack variable $\xi = \left(\xi_1, \cdots, \xi_n, \xi_1^*, \cdots, \xi_n^*\right)^T$ $(\xi_i, \xi_i^* \geq 0, i = 1, \cdots, n)$ is used to measure the errors of input points deviating from two parallel hyperplanes. Thus the primal optimization problem of the SVR model is expressed as

$$\min_{w,b,\xi} \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{n}\left(\xi_i + \xi_i^*\right) s.t. \quad \left(w^T\phi(x_i) + b\right) - y_i \leq \varepsilon + \xi_i, \qquad y_i - \left(w^T\phi(x_i) + b\right) \leq \varepsilon + \xi_i^*, \qquad \xi_i, \xi_i^* \geq 0, i = 1, \cdots, n, \tag{1}$$

where C $(C > 0)$ is a user-defined penalty factor that trades off between the model complexity and the total fitting errors and ε $(\varepsilon > 0)$ is a sufficiently small constant as an input parameter.

The dot product $\phi(x_i)^T\phi(x_j)$ of any two input points x_i and x_j $(i, j = 1, \cdots, n)$ in the high-dimensional feature space can be replaced by a kernel function $K(x_i, x_j)$. Furthermore, by constructing the Lagrange function of the SVR model (1), we can use the KKT optimality and complementary conditions to obtain the dual optimization problem of the SVR model, which has the form

$$\min_{\alpha} \frac{1}{2}\alpha^T\begin{pmatrix} K & -K \\ -K & K \end{pmatrix}\alpha + \begin{pmatrix} \varepsilon e + y \\ \varepsilon e - y \end{pmatrix}^T \alpha \, s.t. \quad \begin{pmatrix} e \\ -e \end{pmatrix}^T \alpha = 0, 0 \leq \alpha \leq Ce, \tag{2}$$

where $\alpha = \left(\alpha_1, \cdots, \alpha_n, \alpha_1^*, \cdots, \alpha_n^*\right)^T$ $(\alpha_i, \alpha_i^* \geq 0, i = 1, \cdots, n)$ is the Lagrange multiplier vector, K $(K \in \mathbb{R}^{n \times n})$ is the kernel matrix derived from the training set T, and $e = (1, \cdots, 1)^T$ $(e \in \mathbb{R}^n)$.

After solving the quadratic programming (QP) problem (2) with the sequential minimal optimization (SMO) algorithm, we can obtain the optimal solution $\overline{\alpha} = \left(\overline{\alpha}_1, \cdots, \overline{\alpha}_n, \overline{\alpha}_1^*, \cdots, \overline{\alpha}_n^*\right)^T$. According to the KKT complementarity conditions, those input points x_i on two $\varepsilon-$band hyperplanes with $0 < \overline{\alpha}_i < C$ or $0 < \overline{\alpha}_i^* < C$ are called support vectors (SVs). If $\overline{\alpha}_i = 0$ or $\overline{\alpha}_i^* = 0$, then the input point x_i lies inside two $\varepsilon-$band hyperplanes, while if $\overline{\alpha}_i = C$ or $\overline{\alpha}_i^* = C$, then the input point x_i is outside two $\varepsilon-$bandhyperplanes. The weight vector $w$is

$$w = \sum_{i=1}^{n}\left(\overline{\alpha}_i^* - \overline{\alpha}_i\right)\phi(x_i) \tag{3}$$

For any input point x_j with $0 < \overline{\alpha}_j < C$ or $0 < \overline{\alpha}_j^* < C$, the intercept b can be computed by

$$b = \frac{1}{\left|\left\{j | 0 < \overline{\alpha}_j, \overline{\alpha}_j^* < C\right\}\right|}\sum_{0 < \overline{\alpha}_j, \overline{\alpha}_j^* < C}^{*}\left(y_j - \sum_{0 < \overline{\alpha}_i, \overline{\alpha}_i < C}^{*}\left(\overline{\alpha}_i^* - \overline{\alpha}_i\right)K\left(x_i, x_j\right)\right) \tag{4}$$

where $|\cdot|$ is the cardinality of a set.

Thus the output value y of a new input point x from a test set is predicted by the regression function

$$g(x) = \sum_{0 < \overline{\alpha}_i, \overline{\alpha}_i < C}^{*}\left(\overline{\alpha}_i^* - \overline{\alpha}_i\right)K(x_i, x) + b \tag{5}$$

In this paper, the radial basis function (RBF) is used as the kernel function, which for any two input points x_i and x_j is defined as.

$$K(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|_2^2}{2\sigma^2}\right)(\sigma > 0) \tag{6}$$

where the bandwidth σ is an input parameter.

3 Bi-sparse optimization-based regression approach

We propose bi-sparse optimization-based regression (BSOR) under the framework of SVR to solve value-forecasting problems. Two reconstructed kernels based on various features are first demonstrated. Then the mathematical model and corresponding algorithm of BSOR with simultaneous instance- and feature-sparsification is described. Finally, we report the results of a simulation based on a real dataset to evaluate the proposed regression model and algorithm.

3.1 BSOR model

The BSOR model is based on the ideas of the ℓ_0-norm regularization support vector classifier model [39, 82], dual representations, and multi-kernel learning methods [34, 85, 86]. Apart from value prediction, the BSOR model alternates between selecting relevant input points and selecting relevant features until convergence. It has two interrelated parts, which are described as follows.

In the first-stage of the BSOR model, for any two input points x_i and x_j ($i, j = 1, \cdots, n$) from the training set T, given the basis function $\psi(x_{im})$ of the feature value x_{im} ($m = 1, \cdots, d$), which maps different feature values from the original input space to a new feature space, their kernel vector h_{ij} ($h_{ij} \in \mathbb{R}^d$) with respect to d features is

$$\begin{aligned} h_{ij} &= \left(\psi(x_{i1})\psi(x_{j1}), \cdots, \psi(x_{id})\psi(x_{jd})\right)^T \\ &= \left(k(x_{i1}, x_{j1}), \cdots, k\left(x_{id}, x_{jd}\right)\right)^T, \end{aligned} \tag{7}$$

where $k(x_{im}, x_{jm})$ is a kernel function value of any two input points x_i and x_j with respect to the mth feature.

If the weight vector $\mu^{(t)}(\mu^{(t)} \in \mathbb{R}^d)$ of d features is provided in the tth iteration, then the row kernel vector $a_i(a_i \in \mathbb{R}^{n+1})$ with respect to the input point x_i is

$$a_i = \left[\left(\mu^{(t)}\right)^T h_{i1}, \cdots, \left(\mu^{(t)}\right)^T h_{in}, 1\right]^T \tag{8}$$

Thus, the row kernel matrix A ($A \in \mathbb{R}^{(n+1)\times n}$) for all input points in the training set T has the form

$$A = (a_1, \cdots, a_n) \tag{9}$$

where the initial value of $\mu^{(t)}$ is set to $\mu^{(0)} = (1/d, \cdots, 1/d)^T$. An iterative counter t is initialized to 1.

For the purpose of instance-sparsification, the ℓ_0-norm with respect to the augmented coefficient vector $\overline{\lambda}^{(t)}$ ($\overline{\lambda}^{(t)} = \left(\lambda_1^{(t)}, \cdots, \lambda_n^{(t)}, \lambda_{n+1}^{(t)}\right)^T$, and $\overline{\lambda}^{(t)} = \left(\lambda^{(t)}, \lambda_{n+1}^{(t)}\right)$ if $\lambda^{(t)} = \left(\lambda_1^{(t)}, \cdots, \lambda_n^{(t)}\right)^T$, $\lambda_i^{(t)} \in \mathbb{R}$, $i = 1, \cdots, n+1$) is approximated by $\left\|\overline{\lambda}^{(t)}\right\|_0 \propto \left(\overline{\lambda}^{(t)}\right)^T \mathrm{diag}\left(\theta^{(t)}\right)\overline{\lambda}^{(t)}$. The ith element $\theta_i^{(t)}$ of the diagonal matrix $\mathrm{diag}(\theta^{(t)})$ ($\mathrm{diag}(\theta^{(t)}) \in \mathbb{R}^{(n+1)\times(n+1)}$) in the tth iteration is computed by

$$\theta_i^{(t)} = \begin{cases} \dfrac{1}{\left(\lambda_i^{(t-1)}\right)^2}, \text{if } \left|\lambda_i^{(t-1)}\right| > \rho, \\ \dfrac{1}{\rho^2}, \text{otherwise}, \end{cases} \tag{10}$$

where $\rho(\rho > 0)$ is a sufficiently small constant and the initial value of $\overline{\lambda}^{(t)}$ is set to $\overline{\lambda}^{(0)} = (1, \cdots, 1)^T$. When $t \to +\infty$, we have $\left\|\lambda^{(t)}\right\|_0 = \left|\left\{i | \lambda_i^* \neq 0, i = 1, \cdots, n\right\}\right|$ for all n instances, where λ_i^* is the optimal coefficient value regarding input point x_i. That is, the ℓ_0-norm of the coefficient vector $\lambda^{(t)}$ amounts to the number of those important instances with $\lambda_i^* \neq 0$, so it should be minimized. Therefore, to effectively identify important or representative input points from the training set T, the first-stage model of BSOR for instance-sparsification can be defined by the following primal optimization problem:

$$\begin{aligned} \min_{\overline{\lambda}^{(t)}, \eta} \frac{1}{2}\left(\overline{\lambda}^{(t)}\right)^T \mathrm{diag}\left(\boldsymbol{\theta}^{(t)}\right)\overline{\lambda}^{(t)} + C_1 \sum_{i=1}^{n}\left(\eta_i + \eta_i^*\right) \quad s.t. \left(\overline{\lambda}^{(t)}\right)^T \boldsymbol{a}_i - y_i \leq \varepsilon + \eta_i, \quad y_i - \left(\overline{\lambda}^{(t)}\right)^T \boldsymbol{a}_i \leq \varepsilon + \eta_i^*, \\ \overline{\lambda}^{(t)} \in \mathbb{R}^{n+1}, \eta_i, \eta_i^* \geq 0, i = 1, \cdots, n, \end{aligned} \tag{11}$$

where $\eta = \left(\eta_1, \cdots, \eta_n, \eta_1^*, \cdots, \eta_n^*\right)^T$ is the slack variable corresponding to those input points with constraint violations, λ_{n+1} can be regarded as the intercept of the regression function $A^T \overline{\lambda}^{(t)} = y$ ($y \in \mathbb{R}^n$), and C_1 ($C_1 > 0$) is a user-defined parameter.

The regression model (11) can be transformed to the corresponding dual optimization problem by constructing a Lagrange function. For the primal model of the first-stage BSOR (11), we can construct the corresponding Lagrange optimization problem as

$$\begin{aligned}\max_{\widehat{\gamma}^{(t)},\widehat{\omega}^{(t)}}\inf_{\overline{\lambda}^{(t)},\eta} L(\Lambda)=&\frac{1}{2}\left(\overline{\lambda}^{(t)}\right)^T\mathrm{diag}\left(\theta^{(t)}\right)\overline{\lambda}^{(t)}+C_1\sum_{i=1}^{n}\left(\eta_i+\eta_i^*\right)+\\&\sum_{i=1}^{n}\gamma_i^{(t)}\left[\left(\overline{\lambda}^{(t)}\right)^T a_i-y_i-\varepsilon-\eta_i\right]-\sum_{i=1}^{n}\omega_i\eta_i+\\&\sum_{i=1}^{n}\gamma_i^{*(t)}\left[y_i-\left(\overline{\lambda}^{(t)}\right)^T a_i-\varepsilon-\eta_i^*\right]-\sum_{i=1}^{n}\omega_i^*\eta_i^* s.t.\widehat{\gamma}^{(t)},\widehat{\omega}^{(t)}\geq\mathbf{0},\end{aligned}\tag{12}$$

where $\widehat{\gamma}^{(t)}=\left(\gamma^{(t)},\gamma^{(t)*}\right)=\left(\gamma_1^{(t)},\cdots,\gamma_n^{(t)},\gamma_1^{(t)*},\cdots,\gamma_n^{(t)*}\right)^T$ ($\widehat{\gamma}^{(t)}\in\mathbb{R}^{2n}$) and $\widehat{\omega}^{(t)}=\left(\omega^{(t)},\omega^{(t)*}\right)=(\omega_1^{(t)},\cdots,\omega_n^{(t)},\omega_1^{(t)*},\cdots,\omega_n^{(t)*})^T$ ($\widehat{\omega}^{(t)}\in\mathbb{R}^{2n}$) are two Lagrange multiplier vectors and $\Lambda=\left[\widehat{\gamma}^{(t)},\widehat{\omega}^{(t)},\overline{\lambda}^{(t)},\eta\right]$.

Suppose that $\Lambda'=\left[\widehat{\gamma}',\widehat{\omega}',\overline{\lambda}',\eta'\right]$ is a solution of the Lagrange optimization problem (12). Since the primal objective function in (11) is convex, the Lagrange function $L(\Lambda)$ has a unique optimal solution Λ'. Thus, the solution Λ' satisfies the Karush-Kuhn-Tucker (KKT) optimality conditions

$$\frac{\partial L(\Lambda)}{\partial\overline{\lambda}^{(t)}}=\mathrm{diag}\left(\theta^{(t)}\right)\overline{\lambda}^{(t)}+\sum_{i=1}^{n}\gamma_i^{(t)}a_i-\sum_{i=1}^{n}\gamma_i^{(t)*}a_i=0\tag{13}$$

$$\frac{\partial L(\Lambda)}{\partial\eta_i}=C_1-\gamma_i-\omega_i=0\tag{14}$$

$$\frac{\partial L(\Lambda)}{\partial\eta_i^*}=C_1-\gamma_i^*-\omega_i^*=0\tag{15}$$

Then from the KKT optimality condition in (13) we get

$$\overline{\lambda}^{(t)}=\left[\mathrm{diag}\left(\theta^{(t)}\right)\right]^{-1}A\left(\gamma^{(t)*}-\gamma^{(t)}\right)\tag{16}$$

Combining the constraint in the optimization problem (12) and the KKT optimality conditions in eq. (14) and eq. (15), we have

$$0\leq\gamma^{(t)},\gamma^{(t)*}\leq C_1e\tag{17}$$

If the Hessian matrix M ($M\in\mathbb{R}^{n\times n}$) is defined as

$$M=A^T\left[diag\left(\theta^{(t)}\right)\right]^{-1}A\tag{18}$$

and we integrate the results in eq. (14), eq. (15), eq. (16), and eq. (18) in the Lagrange optimization problem (12), then we have

$$\begin{aligned}L(\Lambda)=&\frac{1}{2}\left\{\left[\mathrm{diag}\left(\boldsymbol{\theta}^{(t)}\right)\right]^{-1}\boldsymbol{A}\left(\boldsymbol{\gamma}^{*(t)}-\boldsymbol{\gamma}^{(t)}\right)\right\}^T\mathrm{diag}\left(\boldsymbol{\theta}^{(t)}\right)\left[\mathrm{diag}\left(\boldsymbol{\theta}^{(t)}\right)\right]^{-1}\times\\&\boldsymbol{A}\left(\boldsymbol{\gamma}^{*(t)}-\boldsymbol{\gamma}^{(t)}\right)+C_1\boldsymbol{\eta}^*+C_1\boldsymbol{e}^T\boldsymbol{\eta}^*+\left\{\left[\mathrm{diag}\left(\boldsymbol{\theta}^{(t)}\right)\right]^{-1}\boldsymbol{A}\left(\boldsymbol{\gamma}^{*(t)}-\boldsymbol{\gamma}^{(t)}\right)\right\}^T\boldsymbol{A}\boldsymbol{\gamma}^{(t)}-\\&\boldsymbol{y}^T\boldsymbol{\gamma}^{(t)}-\varepsilon\boldsymbol{e}^T\boldsymbol{\gamma}^{(t)}-\left(\boldsymbol{\gamma}^{(t)}\right)^T\boldsymbol{\eta}+\boldsymbol{y}^T\boldsymbol{\gamma}^{*(t)}-\left\{\left[\mathrm{diag}\left(\boldsymbol{\theta}^{(t)}\right)\right]^{-1}\boldsymbol{A}\left(\boldsymbol{\gamma}^{*(t)}-\boldsymbol{\gamma}^{(t)}\right)\right\}^T\boldsymbol{A}\boldsymbol{\gamma}^{(t)}-\\&\varepsilon\boldsymbol{e}^T\boldsymbol{\gamma}^{*(t)}-\left(\boldsymbol{\gamma}^{*(t)}\right)^T\boldsymbol{\eta}^*-\Psi^*\boldsymbol{\eta}-\left(\Psi^*\right)^T\boldsymbol{\eta}^*\\=&-\frac{1}{2}\left(\boldsymbol{\gamma}^{*(t)}-\boldsymbol{\gamma}^{(t)}\right)^T\boldsymbol{M}\left(\boldsymbol{\gamma}^{*(t)}-\boldsymbol{\gamma}^{(t)}\right)+\boldsymbol{y}^t\left(\boldsymbol{\gamma}^{*(t)}-\boldsymbol{\gamma}^{(t)}\right)-\boldsymbol{e}^T\left(\boldsymbol{\gamma}^{*(t)}-\boldsymbol{\gamma}^{(t)}\right)\\=&-\frac{1}{2}\left(\widehat{\gamma}^t\right)^T\begin{pmatrix}\boldsymbol{M}&-\boldsymbol{M}\\-\boldsymbol{M}&\boldsymbol{M}\end{pmatrix}\widehat{\gamma}^{(t)}-\begin{pmatrix}\varepsilon\boldsymbol{e}+\boldsymbol{y}\\\varepsilon\boldsymbol{e}-\boldsymbol{y}\end{pmatrix}^T\widehat{\gamma}^{(t)}\end{aligned}\tag{19}$$

Thus, according to eq. (19) and the range in eq. (17) of Lagrange multipliers, we obtain the dual optimization problem of the first-stage BSOR model (11).

$$\min_{\widehat{\gamma}^{(t)} \in \mathbb{R}^n} \frac{1}{2}\left(\widehat{\gamma}^{t}\right)^{T}\begin{pmatrix} \boldsymbol{M} & -\boldsymbol{M} \\ -\boldsymbol{M} & \boldsymbol{M} \end{pmatrix}\widehat{\gamma}^{t} + \begin{pmatrix} \varepsilon \boldsymbol{e}+\boldsymbol{y} \\ \varepsilon \boldsymbol{e}-\boldsymbol{y} \end{pmatrix}^{T}\widehat{\gamma}^{(t)} \; s.t. \; 0 \le \widehat{\gamma}^{t} \le C_1 \boldsymbol{e} \tag{20}$$

where $\widehat{\gamma}^{(t)} = \left(\gamma^{(t)}, \gamma^{*(t)}\right) = \left(\gamma_1^{(t)}, \cdots, \gamma_n^{(t)}, \gamma_1^{*(t)}, \cdots, \gamma_n^{*(t)}\right)^T$ ($\gamma^{(t)} \in \mathbb{R}^n$, $\gamma^{*(t)} \in \mathbb{R}^n$, $\widehat{\gamma}^{(t)} \in \mathbb{R}^{2n}$) is a Lagrange multiplier vector and its initial value $\widehat{\gamma}^{(0)}$ is set to $(0, \cdots, 0, 0, \cdots, 0)^T$.

By solving the box- or bound-constrained QP problem (20) with the modified SMO algorithm, we can obtain the optimal solution $\widehat{\gamma}^{(t)}$ for the tth iteration and $\widehat{\gamma}^{(t)} = \left(\gamma^{(t)}, \gamma^{*(t)}\right)$, so the coefficient vector $\overline{\lambda}^{(t)}$ is directly computed by

$$\overline{\lambda}^{(t)} = \left[\mathrm{diag}\left(\theta^{(t)}\right)\right]^{-1} A\left(\gamma^{*(t)} - \gamma^{(t)}\right) \tag{21}$$

without solving the primal optimization problem (11).

Then the new diagonal matrix $\mathrm{diag}(\theta^{(t+1)})$ with $\overline{\lambda}^{(t)}$ is updated by eq. (10), the optimization problem (20) is solved again and the optimal solution $\widehat{\gamma}^{(t+1)}$ for the $(t+1)$th iteration is obtained. Thus, the new coefficient vector $\overline{\lambda}^{(t+1)}$ is calculated by eq. (21). Accordingly, with a given weight vector $\mu^{(t)}$, the above five steps, which are composed of constructing the row kernel matrix A in eq. (9), updating the diagonal matrix $\mathrm{diag}(\theta^{(t)})$ with eq. (10), computing the Hessian matrix M in eq. (13), solving the optimization problem (20), and computing the new coefficient vector $\overline{\lambda}^{(t+1)}$ by eq. (21) in order of priority, form an iterative process until the termination condition with respect to the two adjacent multiplier vectors $\widehat{\gamma}^{(t-1)}$ and $\widehat{\gamma}^{(t)}$,

$$\max\left|\widehat{\gamma}^{(t)} - \widehat{\gamma}^{(t-1)}\right| < \tau \tag{22}$$

is satisfied, where τ ($\tau > 0$) is a sufficiently small constant specified by the user, or the maximum number of iterations is reached. That is, when the maximum of the difference of $\widehat{\gamma}^{(t-1)}$ and $\widehat{\gamma}^{(t)}$ remains the same, they are considered to be approximately equal.

When the optimal augmented coefficient vector $\overline{\lambda}^{*} = \left(\lambda^{*}, \lambda_{n+1}^{*}\right)$ is obtained, for the optimal coefficient vector $\lambda^{*} = \left(\lambda_1^{*}, \cdots, \lambda_n^{*}\right)^T$, if $\left|\lambda_i^{*}\right| > \rho$ ($i = 1, \cdots, n$), the corresponding input point x_i is regarded as a representative point that is important for regression. Otherwise, the input point x_i with $\lambda_i^{*} = 0$ (let $\lambda_i^{*} = 0$ if $\left|\lambda_i^{*}\right| \le \rho$) is noisy or unimportant and can be removed from the instance set. Since the first-stage BSOR model generates the sparse instance set, it can use those representative instances as a benchmark for prediction and result explanation.

Regarding the second-stage BSOR model, for any feature f_m ($f_m \in F, f_m \in \mathbb{R}^n$, $m = 1, \cdots, d$) from the training set T, given the mapping function $\psi(x_{im})$ of the feature value x_{im} ($i = 1, \cdots, n$), the outer product matrix D ($D \in \mathbb{R}^{n \times n}$) of f_m is computed by

$$\begin{aligned} D &= f_m f_m^T \\ &= \left[\psi[\psi(x_{1m}), \cdots, \psi(x_{nm})\right] \\ &= \begin{pmatrix} \psi(x_{1m})\psi(x_{1m}) & \cdots & \psi(x_{1m})\psi(x_{nm}) \\ \vdots & \ddots & \vdots \\ \psi(x_{nm})\psi(x_{1m}) & \cdots & \psi(x_{nm})\psi(x_{nm}) \end{pmatrix} \\ &= \begin{pmatrix} k(x_{1m}, x_{1m}) & \cdots & k(x_{1m}, x_{nm}) \\ \vdots & \ddots & \vdots \\ k(x_{nm}, x_{1m}) & \cdots & k(x_{nm}, x_{nm}) \end{pmatrix}. \end{aligned} \tag{23}$$

Because the coefficient vector $\lambda^{(t)}$ ($\lambda^{(t)} = \left(\lambda_1^{(t)}, \cdots, \lambda_n^{(t)}\right)^T$, $\lambda_j^{(t)} \in \mathbb{R}$) can be obtained from eq. (21), the weighted kernel vector v_m ($v_m = (v_{1m}, \cdots, v_{nm})^T$, $v_{im} \in \mathbb{R}$) with respect to the mth feature f_m and the coefficient vector $\lambda^{(t)}$ is defined as

$$\begin{aligned} v_m &= D\lambda^{(t)} \\ &= \begin{pmatrix} k(x_{1m}, x_{1m}) & \cdots & k(x_{1m}, x_{nm}) \\ \vdots & \ddots & \vdots \\ k(x_{nm}, x_{1m}) & \cdots & k(x_{nm}, x_{nm}) \end{pmatrix} \begin{bmatrix} \lambda_1^{(t)} \\ \vdots \\ \lambda_n^{(t)} \end{bmatrix} \\ &= \left[\sum_{j=1}^{n} \lambda_j^{(t)} k\left(x_{1m}, x_{jm}\right), \cdots, \sum_{j=1}^{n} \lambda_j^{(t)} k\left(x_{nm}, x_{jm}\right)\right]^T. \end{aligned} \tag{24}$$

From the result of (24), for any input point x_i ($i = 1, \cdots, n$) the column kernel value v_{im} ($v_{im} \in \mathbb{R}$, $m = 1, \cdots, d$) with respect to the mth feature has the form

$$v_{im} = \sum_{j=1}^{n} \lambda_j^{(t)} k\left(x_{im}, x_{jm}\right). \tag{25}$$

The column kernel vector b_i ($b_i \in \mathbb{R}^{d+1}$) of the input point x_i can be written as

$$b_i = \left(v_{i1}, \cdots, v_{id}, 1\right)^T \tag{26}$$

Algorithm 1. The BSOR algorithm

Inputs: a training set, a test set, and input parameters C_1, C_2, and σ.

Outputs: regression functions and predictive results.

Initialization:

− Let ρ and τ be the sufficiently small constants.

− Assign the maximum number of iterations to M.

− Set an iterative counter, the weight vector, the coefficient vector, and two Lagrange multiplier vectors as t=0, $\overline{\boldsymbol{\mu}}^{(0)} = (1/d, \cdots, 1/d)^T$, $\overline{\boldsymbol{\lambda}}^{(0)} = (1, \cdots, 1)^T$, $\hat{\boldsymbol{\gamma}}^{(0)} = (0, \cdots, 0)^T$, and $\hat{\boldsymbol{\varphi}}^{(0)} = (0, \cdots, 0)^T$ respectively.

Repeat

− $t \leftarrow t+1$

− Construct the row kernel matrix $\boldsymbol{A}$ in eq. (9) and the diagonal matrix $\mathrm{diag}(\boldsymbol{\theta}^{(t)})$ with eq. (10).

− Compute the Hessian matrix $\boldsymbol{M} \leftarrow \boldsymbol{A}^T[\mathrm{diag}(\boldsymbol{\theta}^{(t)})]^{-1}\boldsymbol{A}$ in eq. (18).

− Solve the optimization problem (20) of the BSOR model to obtain the Lagrange multiplier vectors $\boldsymbol{\gamma}^{*(t)}$ and $\boldsymbol{\gamma}^{(t)}$.

− Update the coefficient vector $\overline{\boldsymbol{\lambda}}^{(t)} \leftarrow [\mathrm{diag}(\boldsymbol{\theta}^{(t)})]^{-1}\boldsymbol{A}(\boldsymbol{\gamma}^{*(t)} - \boldsymbol{\gamma}^{(t)})$ in eq. (21).

− Construct the column kernel matrix $\boldsymbol{B}$ in eq. (27) and the diagonal matrix $\mathrm{diag}(\boldsymbol{\pi}^{(t)})$ with eq. (28).

− Calculate the Hessian matrix $\boldsymbol{N} \leftarrow \boldsymbol{B}^T[\mathrm{diag}(\boldsymbol{\pi}^{(t)})]^{-1}\boldsymbol{B}$ in eq. (31).

− Solve the optimization problem (30) of the BSOR model to gain the Lagrange multiplier vectors $\boldsymbol{\varphi}^{*(t)}$ and $\boldsymbol{\varphi}^{(t)}$.

− Update the weight vector $\overline{\boldsymbol{\mu}}^{(t)} \leftarrow [\mathrm{diag}(\boldsymbol{\pi}^{(t)})]^{-1}\boldsymbol{B}(\boldsymbol{\varphi}^{*(t)} - \boldsymbol{\varphi}^{(t)})$ in eq. (32).

Until ($\max\left|\hat{\boldsymbol{\gamma}}^{(t)} - \hat{\boldsymbol{\gamma}}^{(t-1)}\right| < \tau$ and $\max\left|\hat{\boldsymbol{\varphi}}^{(t)} - \hat{\boldsymbol{\varphi}}^{(t-1)}\right| < \tau$) or ($t > M$) (See (22) and (33)).

− Set $\lambda_i^* = 0$ if $\left|\lambda_i^*\right| \le \rho$ ($i = 1, \cdots, n$) and $\mu_j^* = 0$ if $\left|\mu_j^*\right| \le \rho$ ($j = 1, \cdots, d$) for the optimal coefficient and weight vectors $\boldsymbol{\lambda}^*$ and $\boldsymbol{\mu}^*$.

− Construct the regression functions in eq. (34) or eq. (35) and predict output values of input points from a test set.

and the column kernel matrix B ($B \in \mathbb{R}^{(d+1)\times n}$) for all input points in the training set T is

$$B = (b_1, \cdots, b_n) \tag{27}$$

Similarly, for the purpose of feature-sparsification, the ℓ_0- normwith respect to the augmented weight vector $\overline{\mu}^{(t)}$ ($\overline{\mu}^{(t)} = \left(\mu_1^{(t)}, \cdots, \mu_d^{(t)}, \mu_{d+1}^{(t)}\right)^T$, and $\overline{\mu}^{(t)} = \left(\mu^{(t)}, \mu_{d+1}^{(t)}\right)$ if $\mu^{(t)} = \left(\mu_1^{(t)}, \cdots, \mu_d^{(t)}\right)^T$, $\mu_j^{(t)} \in \mathbb{R}$, $j = 1, \cdots, d+1$) is estimated by $\left\|\overline{\mu}^{(t)}\right\|_0 \propto \left(\overline{\mu}^{(t)}\right)^T \mathrm{diag}\left(\pi^{(t)}\right)\overline{\mu}^{(t)}$. Then, the $j$thelement $\pi_j^{(t)}$ of the diagonal matrix $\mathrm{diag}(\pi^{(t)})$ ($\mathrm{diag}(\pi^{(t)}) \in \mathbb{R}^{(d+1)\times(d+1)}$) in the tth iteration is defined as

$$\pi_j^{(t)} = \begin{cases} \dfrac{1}{\left(\mu_j^{(t-1)}\right)^2}, \text{if } \left|\mu_j^{(t-1)}\right| > \rho, \\ \dfrac{1}{\rho^2}, \text{otherwise}, \end{cases} \tag{28}$$

where the initial value $\overline{\mu}^{(0)}$ for the weight vector $\overline{\mu}^{(t)}$ is set to $(\mu^{(0)}, 1/d)$. Likewise, when$t \to +\infty$, we have $\left\|\mu^{(t)}\right\|_0 = \left|\left\{j | \mu_j^* \neq 0, j = 1, \cdots, d\right\}\right|$ for all d features, where μ_j^* is the optimal weight value regarding the feature f_j. That is, the ℓ_0 $-$norm of weight vector $\mu^{(t)}$ equals the number of those important features with $\mu_j^{(t)} \neq 0$, so it should be minimized.

Hence, in the interest of identifying important features for regression, the second-stage model of BSOR for feature-sparsification can be defined as the primal optimization problem:

$$\begin{aligned}
&\min_{\overline{\boldsymbol{\mu}}^{(t)},\varsigma} \frac{1}{2}\left(\overline{\boldsymbol{\mu}}^{(t)}\right)^T \operatorname{diag}\left(\boldsymbol{\pi}^{(t)}\right)\overline{\boldsymbol{\mu}}^{(t)} + C_2\sum_{i=1}^{n}\left(\varsigma_i+\varsigma_i^*\right) \\
&s.t.\left(\overline{\boldsymbol{\mu}}^{(t)}\right)^T \boldsymbol{b}_i - y_i \le \varepsilon+\varsigma_i, \\
&y_i - \left(\overline{\boldsymbol{\mu}}^{(t)}\right)^T \boldsymbol{b}_i \le \varepsilon+\varsigma_i^*, \\
&\overline{\boldsymbol{\mu}}^{(t)} \in \mathbb{R}^{d+1}, \varsigma_i, \varsigma_i^* \ge 0, i=1,\cdots,n,
\end{aligned} \tag{29}$$

where $\varsigma=(\varsigma,\cdots,\varsigma,\varsigma^*,\cdots,\varsigma^*)^T$ is the slack vector of input points with fitting errors, μ_{d+1} can be considered the intercept of the regression function $B^T\overline{\mu}^{(t)}=y$, and C_2 ($C_2>0$) is a user-specified parameter.

For the second-stage BSOR model (29), we can construct the corresponding Lagrange function to obtain its dual optimization problem. Similarly, the dual optimization model can be written as

$$\begin{aligned}
&\min_{\widehat{\varphi}^{(t)}\in\mathbb{R}^n} \frac{1}{2}\left(\widehat{\varphi}^{(t)}\right)^{(T)}\begin{pmatrix}\boldsymbol{N} & -\boldsymbol{N}\\ -\boldsymbol{N} & \boldsymbol{N}\end{pmatrix}\widehat{\varphi}^{(t)} + \begin{pmatrix}\varepsilon\boldsymbol{e}+\boldsymbol{y}\\ \varepsilon\boldsymbol{e}-\boldsymbol{y}\end{pmatrix}^T\widehat{\varphi}^{(t)} \\
&s.t.\quad 0\le\widehat{\varphi}^{(t)}\le C_2\boldsymbol{e},
\end{aligned} \tag{30}$$

where $\widehat{\varphi}^{(t)}=\left(\varphi^{(t)},\varphi^{*(t)}\right)=\left(\varphi_1^{(t)},\cdots,\varphi_n^{(t)},\varphi_1^{*(t)},\cdots,\varphi_n^{*(t)}\right)^T$ ($\varphi^{(t)}\in\mathbb{R}^n$, $\varphi^{*(t)}\in\mathbb{R}^n$, $\widehat{\varphi}^{(t)}\in\mathbb{R}^{2n}$) is a Lagrange multiplier vector whose initial value $\widehat{\varphi}^{(0)}$ is assigned to $(0,\cdots,0,0,\cdots,0)^T$. The Hessian matrix N ($N\in\mathbb{R}^{n\times n}$) is calculated by

$$N=B^T\left[diag\left(\pi^{(t)}\right)\right]^{-1}B \tag{31}$$

When solving the box- or bound-constrained QP problem (30) with the modified SMO algorithm, the optimal solution $\widehat{\varphi}^{(t)}$ for the tth iteration is obtained, and $\widehat{\varphi}^{(t)}=\left(\varphi^{(t)},\varphi^{*(t)}\right)$. There is no need to solve the optimization problem (29), the weight vector $\overline{\mu}^{(t)}$ is directly computed by

$$\overline{\mu}^{(t)}=\left[\operatorname{diag}\left(\pi^{(t)}\right)\right]^{-1}B\left(\varphi^{*(t)}-\varphi^{(t)}\right) \tag{32}$$

Then the new diagonal matrix diag($\pi^{(t+1)}$) with $\overline{\mu}^{(t)}$ can be updated by formulation (28), the optimization problem (30) is solved again and the optimal solution $\widehat{\varphi}^{(t+1)}$ for the $(t+1)$th iteration is found. We can calculate the new weight vector $\overline{\mu}^{(t+1)}$ by eq. (32). So, with a given coefficient vector $\lambda^{(t)}$, the above five steps, which consist of constructing the column kernel matrix B in eq. (27), updating the diagonal matrix diag($\pi^{(t)}$) with eq. (28), calculating the Hessian matrix N in eq. (31), solving the optimization problem (30), and computing the new weight vector $\overline{\mu}^{(t+1)}$ by eq. (32), is virtually an iterative process until the stopping condition with respect to the two adjacent multiplier vectors $\widehat{\varphi}^{(t-1)}$ and $\widehat{\varphi}^{(t)}$ is satisfied by

$$\max\left|\widehat{\varphi}^{(t)}-\widehat{\varphi}^{(t-1)}\right|<\tau \tag{33}$$

Similarly, when the maximum difference of $\widehat{\varphi}^{(t-1)}$ and $\widehat{\varphi}^{(t)}$ does not change, they are considered to be approximately equal.

Once we gain the optimal augmented weight vector $\overline{\mu}^*=\left(\mu^*,\mu_{d+1}^*\right)$, for the optimal weight vector $\mu^*=\left(\mu_1^*,\cdots,\mu_d^*\right)^T$, if $\left|\mu_j^*\right|>\rho$ ($j=1,\cdots,d$), then the corresponding feature f_j is considered an important variable that contributes to regression. Otherwise, the feature f_j with $\mu_j^*=0$ (let $\mu_j^*=0$ if $\left|\mu_j^*\right|\le\rho$) is redundant and can be removed from the feature set. Because the second-stage BSOR model produces the sparse feature set, it can employ those important features for causal analysis in decision-making.

From the global perspective of associating the first and second stages of the BSOR model, after assigning the initial value $\overline{\mu}^{(0)}$, by solving the quadratic optimization problem (20) with the weight vector $\overline{\mu}^{(t-1)}$ in the $(t-1)$th iteration, we can use eq. (21) to compute the new coefficient vector $\overline{\lambda}^{(t)}$ based on the solution $\widehat{\gamma}^{(t)}$. By solving the quadratic optimization problem (30) with the obtained coefficient vector $\overline{\lambda}^{(t)}$, eq. (32) is used to calculate the new weight vector $\overline{\mu}^{(t)}$ based on the solution $\widehat{\varphi}^{(t)}$ for the tth iteration. Then the weight vector $\overline{\mu}^{(t)}$ is further used to obtain the new coefficient vector $\overline{\lambda}^{(t+1)}$ and new weight vector $\overline{\mu}^{(t+1)}$. Because of the convergence of the Lagrange multipliers $\widehat{\gamma}^{(t)}$ in eq. (22) and $\widehat{\varphi}^{(t)}$ in eq. (33), the optimal augmented coefficient vector $\overline{\lambda}^*$ and optimal augmented weight vector$\overline{\mu}^*$ are obtained by eq. (21) and eq. (32), respectively. Otherwise, the iterative counter t is incremented by 1, and the iterative process of searching the optimal augmented

coefficient vector $\overline{\lambda}^*$ and weight vector $\overline{\mu}^*$ continues until the maximum number of iterations is reached.

The output value y for a new input point x from an independent test set can be predicted by the regression functions below. According to the optimal augmented coefficient vector $\overline{\lambda}^*$ (let $\lambda_i^* = 0$ if $|\lambda_i^*| \le \rho$, $i = 1, \cdots, n$) and the row kernel vector a_x ($a_x \in \mathbb{R}^{n+1}$) with respect to input point x, the regression function is written as

$$g(x) = \left(\overline{\lambda}^*\right)^T a_x \tag{34}$$

Moreover, given the optimal augmented weight vector $\overline{\mu}^*$ (let $\mu_j^* = 0$ if $|\mu_j^*| \le \rho$, $j = 1, \cdots, d$) and the column kernel vector b_x ($b_x \in \mathbb{R}^{d+1}$) with respect to input point x, the regression function is defined as

$$g(x) = \left(\overline{\mu}^*\right)^T b_x \tag{35}$$

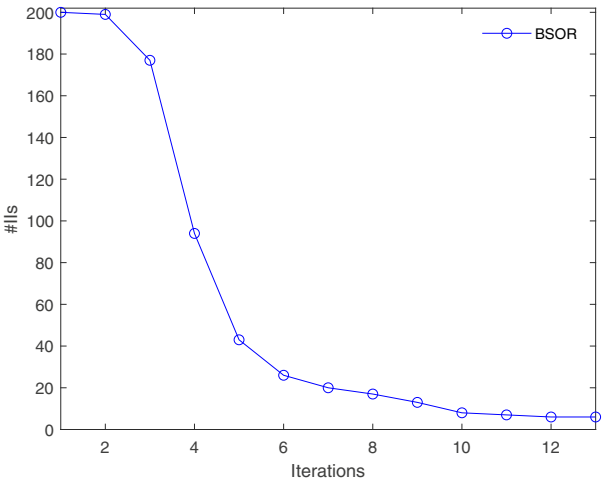
where the vectors a_x and b_x can be computed by eq. (8) and eq. (26), respectively. Thus one of two regression functions can be used to predict the output values of any input point. We can use the mean of two predictive output values to gain the final forecasting value of any input point.

For computing the row and column kernel matrices, the RBF kernel of two input points x_i and x_j with respect to the *mth* feature is defined as.

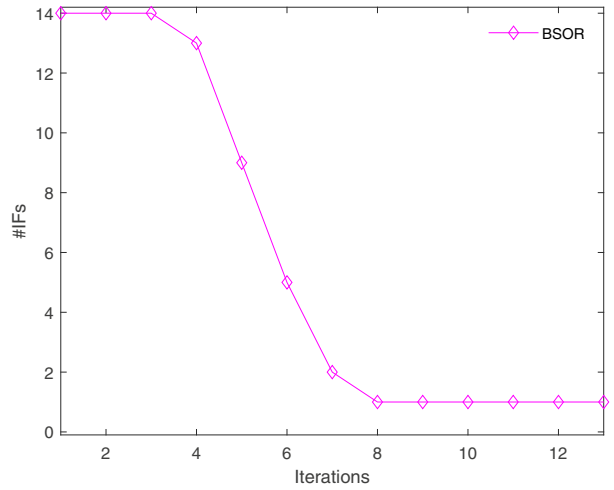
$$k(x_{i,m}, x_{j,m}) = \exp\left(-\frac{(x_{i,m} - x_{j,m})^2}{2\sigma^2}\right) (\sigma > 0) \tag{36}$$

Overall, when the two-stage BSOR model is applied to solve regression problems, apart from value prediction, it can simultaneously select relevant instances and features. Specifically, in the first-stage, the BSOR model can effectively identify a comparatively small number of representative or prototype instances, while in the second-stage, it can efficiently extract a comparatively small number of important features. The former generates the sparse coefficient vector λ^*, where the coefficient value λ_i^* ($\lambda_i^* \neq 0$) indicates the degree of importance for prediction, and the latter produces the sparse weight vector μ^*, where the weight value μ_j^* ($\mu_j^* \neq 0$) indicates the degree of importance of each feature for forecasting.

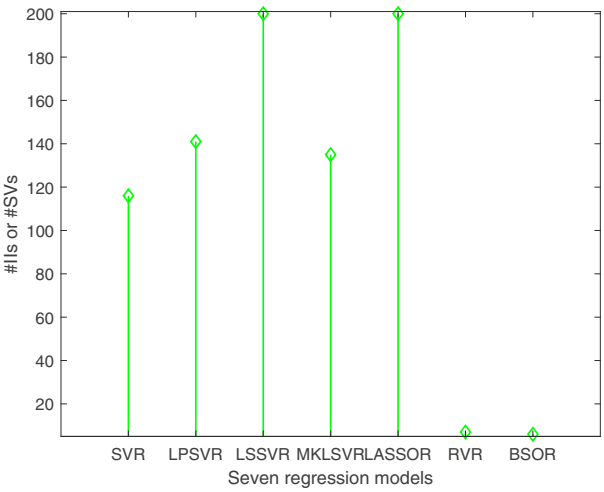
From the above regression functions in eq. (34) and eq. (35), for a new input points its predictive output value can



(a) #IIs identified by BSOR decreases with the increasing iterations



(b) #IFs extracted by BSOR decreases with the increasing iterations



(c) #IIs or #SVs found by the seven regression models

Fig. 1 Evaluating the seven regression methods on the bodyfat dataset. (**a**) #IIs identified by BSOR decreases with the increasing iterations. (**b**) #IFs extracted by BSOR decreases with the increasing iterations. (**c**) #IIs or #SVs found by the seven regression models. (**d**) Curve fitting of predictive results for the seven regression models on the bodyfat test set

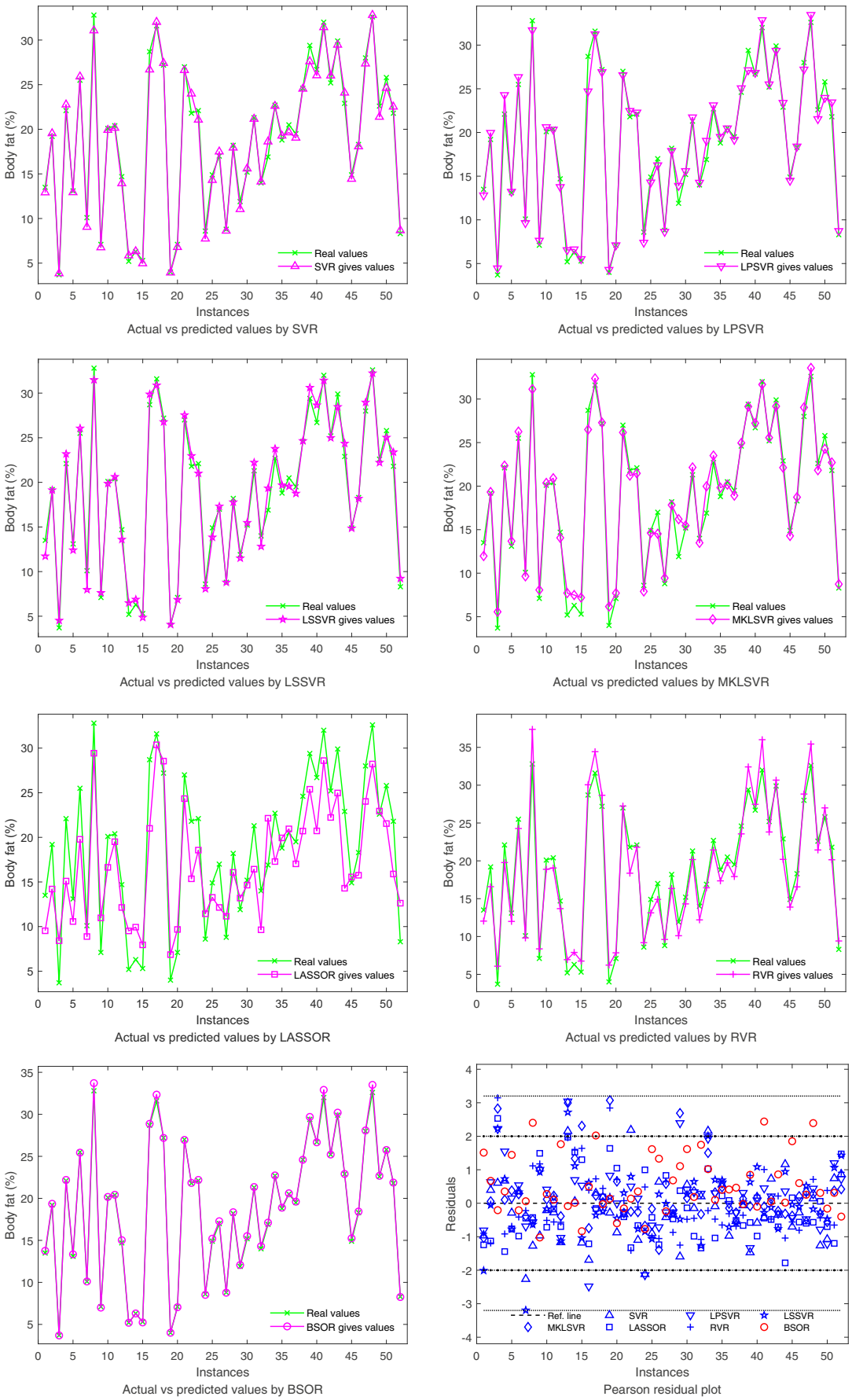


Fig. 1 (continued)

Table 1 Predictive evaluation of the seven regression models on the bodyfat test set

Regression models	MSE	MAE	MAPE
SVR	0.0016	0.0320	1.1543
LPSVR	0.0031	0.0404	1.5239
LSSVR	0.0037	0.0451	1.7102
MKLSVR	0.0100	0.0631	2.6530
LASSOR	0.0567	0.2020	7.6536
RVR	0.0123	0.0891	3.4800
BSOR	**0.0001**	**0.0077**	**0.2624**

Note: The bold statistics show that the regression model has the better predictive performance than others (the same below)

be obtained by the weighted distance with respect to the coefficient vector of prototype instances and the corresponding row kernel vector which is composed of the distances from the new input point to prototype instances, and the weighted distance with respect to the weight vector of important features and the corresponding column kernel vector which is constituted of the distances between the features of the new input point and important features. In other words, representative instances with important features are good enough for the similarity based on distance without using all instances and features in the training set. Consequently, the accuracy, generalization, and interpretability of BSOR on the reduced dataset are all enhanced, which are critical for the regression application in large-scale and high-dimensional datasets.

3.2 BSOR algorithm

The BSOR algorithm corresponding to the above model, can be summarized Algorithm 1, including inputs, outputs, initialization, iterative processing steps, and regression functions.

3.2.1 Simulation

To intuitively evaluate the new BSOR method, we ran a simulation on the real bodyfat dataset from the StatLib repository (http://lib.stat.cmu.edu/datasets/). For each of 252 men, 14 numeric features were used to estimate the percentage of body fat determined by underwater weighing and various body-condition measures. These conditional features consist of density (gm/cm^3), age (years), weight (lbs), height (inches), neck circumference (cm), chest circumference (cm), abdomen circumference (cm), hip circumference (cm), thigh circumference (cm), knee circumference (cm), ankle circumference (cm), biceps circumference (cm), forearm circumference (cm), and wrist circumference (cm). The dataset was partitioned into a training set with 200 instances and an independent test set with 52 instances. The SVR, LPSVR, LSSVR, MKLSVR, LASSOR, RVR, and BSOR methods were trained on the training set and tested on the independent test set. Then, the number of important instances (#IIs) identified by BSOR, the number of features (#IFs) extracted by BSOR, #IIs or the number of SVs (#SVs) found by the seven regression models, curve fitting of actual and predicted values, and Pearson residuals (also called standardized residuals) are demonstrated in Fig. 1.

Three common measures to evaluate the predictive performance of various regression models, are the mean square error (MSE), mean absolute error (MAE), and mean absolute percentage error (MAPE), defined as

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \widehat{y}_i\right)^2 \tag{37}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \widehat{y}_i\right| \tag{38}$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \widehat{y}_i}{y_i}\right| \times 100\% \tag{39}$$

where y_i is the actual output value and $\widehat{y}_i$ is the predictive output value with respect to the input point x_i ($i = 1, \cdots, n$). The evaluation results for the bodyfat dataset are listed in Table 1.

As the iterative process demonstrates in Fig. 1(a), BSOR converges and identifies six important instances with $x_{53}(\lambda^*_{53} = -2.28)$, x_{93} $(\lambda^*_{93} = 1.65)$, x_{111} $(\lambda^*_{111} = 6.02)$, x_{128} $(\lambda^*_{128} = 3.60)$, x_{186} $(\lambda^*_{186} = -13.53)$, and x_{198} $(\lambda^*_{198} = -2.34)$ ($\lambda^*_i = 0$ for $i = 1, \cdots, 200$, and $i \neq 20, 42,$

Table 2 Fifteen datasets for experiments

Datasets	#Features	#Instances
Abalone	10	4177
Autoprice	15	159
Carbolenes	1142	37
Cpuact	21	8192
Housing	14	506
Lowbwt	10	189
Parkinsonsupdrs	22	5875
Pdgfr	320	79
Phenetyl1	628	22
Sensory	11	576
Strupcz	1142	34
Topo21	266	8885
Triazines	60	186
Winequalityred	11	1599
Wisconsin	32	194

121, 139, 170) after 13 iterations, while SVR, LPSVR, LSSVR, MKLSVR, LASSOR, and RVR, respectively select 116, 141, 200, 135, 200, and 7 (x_{80}, x_{115}, x_{128}, x_{137}, x_{174}, x_{186}, and x_{190} with the coefficients 1.56, −1.69, −0.11, 4.69, −1.93, −3.79, and − 0.38 respectively) support vectors from 200 instances, as shown in Fig. 1(c). As shown in Fig. 1(b), BSOR extracts one important feature after the same number of iterations, i.e., the body density with $f_1(\mu_1^* = 0.15\ )$ ($\mu_j^* = 0$ for $j = 2, \cdots, 14$) is considered the most important feature for body fat prediction, while the other five regression models employ the entire feature set except for LASSOR with 6 selected features (f_1, f_2, f_4, f_7, f_9, andf_{13} with the weights −1.10, 0.31, 1.13, 1.00, 0.06, and 0.23 respectively). For BSOR, in order to predict the percentage of body fat of a new unseen man, we just use six representative men (x_{53}, x_{93}, x_{111}, x_{128}, x_{186}, and x_{198}) with an important feature (f_1) that have been selected by the BSOR algorithm to compute its output value based on their dot product in eq. (34) and eq. (35). From the curve fitting of predictive results shown in Fig. 1(d), we find that BSOR generally has better predictive accuracy than the other regression models, especially for those points marked with different symbols in regression curves. For the seven regression models, all residuals lie on the interval [−3.2, 3.2]. For BSOR, the majority of Pearson residuals are normally distributed between −2 and 2 except for two residuals, so it achieves the best predictive accuracy. As the predictive performance shows in Table 1, we find that the predictive accuracy of BSOR generally is better than that of the other regression methods. The definitions of parametric sets for the seven regression models can be found in the experiment design section in Section 4.2. For the above predictive results, the best parameters are $C = 50$and $\sigma = 1$ for SVR; $C = 100$and $\sigma = 0.5$ for LPSVR; $C = 20$and $\sigma = 1$ for LSSVR; $C = 100$ for MKLSVR; $C = 2$ for LASSOR; and $C_1 = 50$, $C_2 = 5$, and $\sigma = 0.1$ for BSOR. Compared with SVR, LPSVR, LSSVR, MKLSVR, and BSOR, the LASSOR model provided the poor predictive accuracy (see Fig. 1(d) and Table 1), although it employed the second smallest number of features. Similarly, the RVR model used the second smallest number of support vectors, but it also gave the poor predictive performance (see Fig. 1(d) and Table 1). Overall, the above experimental results show that the predictive accuracy and interpretability are obviously enhanced after instances and features in the bodyfat dataset are simultaneously reduced by BSOR. For CPU time (in seconds), the training time and test time are 0.0450 and 0.0033 for SVR, 0.4785 and 0.0332 for LPSVR, 0.0135 and 0.0061 for LSSVR, 0.7389 and 0.0183 for MKLSVR, 0.1363 and 0.0067, 0.1417 and 0.0049, and 1.2803 and 0.0082 for BSOR, respectively.

4 Experiments

BSOR and the other six regression models were applied to 15 real datasets to further evaluate the predictive performance. This section includes datasets, experiment design, experimental results and comparison analysis, importance analysis of extracted instances and features by the BSOR model, and analysis of experimental results.

4.1 Datasets

In this experiment, 15 datasets were used to evaluate SVR, LPSVR, LSSVR, MKLSVR, LASSOR, RVR, and BSOR. Among them, the abalone, autoprice, cpuact, housing, lowbwt, parkinsonsupdrs, sensory, triazines, winequalityred, and Wisconsin datasets were sourced from the online StatLib (http://lib.stat.cmu.edu/datasets) and UCI Machine Learning Repository [4], while carbolenes, pdgfr, phenetyl1, strupcz, and topo21 were selected from drug-design datasets (www.molecular-networks.com/software/adrianacode). The number of features (#Features) and number of instances (#Instances) in the 15 datasets are shown in Table 2.

As the characteristics of the 15 datasets show in Table 2, we see that the abalone, cpuact, parkinsonsupdrs, and winequalityred datasets are relatively large; the carbolenes, pdgfr, phenetyl1, strupcz, and triazines are high-dimensional; topo21 is both large-scale and high-dimensional; and the others are small.

The abalone data are used to predict the age of abalone from physical measurements. The age of an abalone is determined by cutting the shell through the cone, staining it, and counting the number of rings through a microscope. The autoprice data include the specification of an auto, its assigned insurance risk rating, and its normalized losses in use compared to other cars. Its 14 numeric attributes and one nominal attribute are employed to forecast the price of an auto. The cpuact dataset was collected from a Sun Sparcstation running in a multi-user university department, and was used to predict the portion of time that CPUs run in user mode from different attributes. The housing dataset mainly concerns housing values in suburbs of Boston. The lowbwt dataset was generated by the Baystate Medical Center, Springfield, Massachusetts, in 1986. It was used to identify risk factors associated with giving birth to a low-birth-weight (less than 2, 500 g) baby. Data were collected on 189 women, 59 with low-birth-weight babies and 130 with normal birth weight babies. The parkinsonsupdrs dataset was created by Athanasios Tsanas and Max Little of the University of Oxford, who developed a telemonitoring device to record speech signals. The original study used a range of linear and nonlinear regression methods to predict a clinician's Parkinson's disease symptom score on the UPDRS scale. The sensory dataset was used for the sensory evaluation

experiment, which involved two phases of a viticultural experiment and a produce evaluation. The winequalityred dataset, which is related to red variants of Portuguese Vinho Verde wine, was created in 2009, using red wine samples. The inputs included objective tests (e.g. pH values), and the output was based on sensory data (median of at least three evaluations by wine experts). Each expert graded the wine quality between 0 (very bad) and 10 (very excellent). The triazines dataset is a pyrimidine QSAR dataset. The goal is to predict the inhibition of dihydrofolate reductase by pyrimidines. In the wisconsin dataset, each record represents follow-up data for one breast cancer case. Thirty features were computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. They describe characteristics of the cell nuclei present in the image. The output feature recorded the recurrent time of a breast cancer, so the dataset is used to predict recurrent time.

The drug-design datasets, the carbolenes dataset was sourced from a study of comparative molecular moment analysis by Silverman and Platt [66]. The pdgfr dataset is for the prediction and interpretation of the biological activity of a set of PDGFR inhibitors. The phenetyl1 dataset comes from a study of finding a new remedy for a certain disease using the QSAR tool to aid scientists in drug design. The strupcz dataset is used for concept validation of the neighbourhood behavior of molecular diversity descriptors. Finally, the topo21 dataset is applied to the toxicological prediction in drug design.

4.2 Experiment design

In our experiment, we randomly selected 3000, 100, 25, 5000, 400, 150, 4000, 70, 15, 400, 25, 5000, 100, 1000, and 150 instances from the abalone, autoprice, carbolenes, cpuact, housing, lowbwt, parkinsonsupdrs, pdgfr, phenetyl1, sensory, strupcz, topo21, triazines, winequalityred, and wisconsin datasets, respectively, to form the training sets, and the remainders were used as the independent test sets. First, the grid search method based on predefined parametric sets was employed. Then the 5-fold cross-validation (CV) method was used to train SVR, LPSVR, LSSVR, MKLSVR, LASSOR, RVR, and BSOR on different training sets. They were validated on validation subsets and the optimal regression models with the best parameters corresponding to 5-fold CV were selected. Finally, the best predictive performance using the optimal regression functions on independent test sets was determined.

For each of the 15 datasets, values of different features were normalized to the interval [0, 1] by min-max standardization, i.e., a new input point x' for the original input point x was obtained by

$$x' = \frac{x - x_{\max}}{x_{\max} - x_{\min}} \tag{40}$$

where the vectors $x_{\max}$ and $x_{\min}$ are calculated from the union of a training set and the corresponding test set. For a large and positive output value y in a training set and the corresponding test set, the logarithmic function was employed to obtain its stationary output value. At the same time, for non-positive output value y, a proper offset term Δy ($\Delta y > 0$) was added to the original output value y so as to avoid dividing by zero when computing the performance measure of MAPE in eq. (39). Thus the new output value y' was computed by

$$y' = log(y + \Delta y) \tag{41}$$

Some datasets, including abalone, housing, lowbwt, sensory, and bodyfat in the simulation section, used the transformation method in eq. (41) for their output values.

The grid search method was used in this experiment to find the best parameters of the seven regression models, i.e., the penalty factor C for SVR, LPSVR, LSSVR, MKLSVR, and LASSOR, and the penalty constants C_1 and C_2 for BSOR were defined as the discrete set {1, 2, 5, 10, 20, 50, 100}. The bandwidth σ for the RBF kernel was set to the finite set {0.01, 0.1, 0.2, 0.5, 1, 2, 5, 10}. The small constants ε, ρ, τ, and the maximum iterative times were respectively set to 0.02, 1e-6, 1e-2, and 50. The initial weight vector $\overline{\mu}^{(0)}$ was assigned to $(1/d, \cdots, 1/d)^T$. In this paper, the RBF kernel in eq. (6) and the polynomial kernel were used in MKLSVR. The polynomial kernel of any two input points x_i and x_j is defined as

$$K(x_i, x_j) = \left(x_i^T x_j + 1\right)^{\delta} \tag{42}$$

where the degree δ ($\delta \geq 1$) is a user-defined parameter from the set {1, 2, 3}. For MKLSVR, the parametric sets of the bandwidth σ and degree δ were used in the experiment. For RVR, the expectation-maximization algorithm is used to estimate unknown parameters.

For BSOR the number of important instances (#IIs) and the number of important features (#IFs) were used to evaluate the efficiencies of regression models and selected instances and features that could be utilized to obtain interpretable results for forecasting. The important instances are composed of representative or prototype instances in observation data, while the important features consist of important and relevant variables in feature set. But, for SVR, LPSVR, LSSVR, MKLSVR, and RVR, important instances are called SVs and they are unable to extract important features. LASSOR can select important features, whereas it is unable to identify important instances.

To evaluate a given regression model's ability to select important instances and features and remove noisy or redundant instances and features, the instance reduction rate

Table 3 Predictive evaluation for MSE on 15 independent test sets

DATASETS	SEVEN REGRESSION MODELS						
	SVR	LPSVR	LSSVR	MKLSVR	LASSOR	RVR	BSOR
Abalone	0.0299	0.0286	0.0287	0.0798	0.0291	0.0300	0.0271
Autoprice	0.0262	0.0319	0.0291	0.1900	0.2105	**0.0215**	0.0232
Carbolenes	0.0346	0.0284	0.0150	0.0842	0.0415	0.0437	0.0125
Cpuact	0.0035	0.0319	0.0035	0.4629	0.1634	0.1503	0.0026
Housing	0.0213	0.0184	0.0206	0.1406	0.0766	0.0500	0.0307
Lowbwt	0.0134	0.0101	0.0226	0.0709	0.8969	0.0261	0.0042
Parkinsonsupdrs	0.0013	0.0014	0.0015	0.0089	0.0023	0.0019	0.0015
Pdgfr	0.0699	0.0188	0.0189	0.0520	0.0246	0.0265	0.0079
Phenetyl1	0.0404	0.0194	0.0109	0.0332	0.0240	0.0075	0.0050
Sensory	0.0031	0.0028	0.0028	0.0031	0.0368	0.0028	0.0025
Strupcz	0.0023	0.0070	0.0032	0.0012	**0.0004**	0.0026	0.0007
Topo21	0.0008	0.0008	0.0007	0.0008	0.0010	0.0010	0.0008
Triazines	0.0227	0.0224	0.0220	0.0297	0.0269	0.0231	0.0159
Winequalityred	0.0171	0.0172	0.0166	0.0246	0.0191	0.0188	0.0159
Wisconsin	0.0658	0.0604	0.0644	0.0738	0.0490	**0.0371**	0.0429

(IRR) and feature reduction rate (FRR) with #IIs and #IFs are respectively defined as

$$IRR = \left(1 - \frac{\#IIs}{|T|}\right) \times 100\% \tag{43}$$

$$FRR = \left(1 - \frac{\#\mathrm{IF}s}{|F|}\right) \times 100\% \tag{44}$$

For BSOR, based on the optimal coefficient vector λ^* and weight vector μ^*, the relative importance of instances (II) and relative importance of features (FI) for value prediction can be respectively computed by

$$II\ (x_i) = \frac{\lambda_i^*}{\sum_{i=1}^{n} |\lambda_i^*|} \times 100\%, i = 1, \cdots, n \tag{45}$$

Table 4 Predictive evaluation for MAE on 15 independent test sets

DATASETS	SEVEN REGRESSION MODELS						
	SVR	LPSVR	LSSVR	MKLSVR	LASSOR	RVR	BSOR
Abalone	0.1338	0.1312	0.1298	0.2140	0.1296	0.1290	0.1264
Autoprice	0.1263	0.1319	0.1370	0.3298	0.3971	**0.1068**	0.1224
Carbolenes	0.1598	0.1407	0.0956	0.2392	0.1801	0.1798	0.0771
Cpuact	0.0420	0.0655	0.0400	0.2662	0.2680	0.2448	0.0363
Housing	0.1017	0.0944	0.0995	0.2776	0.1798	0.1504	0.1241
Lowbwt	0.0970	0.0784	0.1182	0.2505	0.7251	0.1264	0.0578
Parkinsonsupdrs	0.0267	0.0265	0.0261	0.0681	0.0362	0.0331	0.0282
Pdgfr	0.1880	0.0872	0.1032	0.1696	0.1389	0.1322	0.0766
Phenetyl1	0.1607	0.1163	0.0897	0.1468	0.1304	0.0780	0.0624
Sensory	0.0476	0.0433	0.0435	0.0447	0.1561	0.0427	0.0406
Strupcz	0.0341	0.0393	0.0459	0.0292	**0.0132**	0.0451	0.0213
Topo21	0.0203	0.0198	0.0197	0.0188	0.0199	0.0200	0.0186
Triazines	0.1077	0.1041	0.1059	0.1278	0.1302	0.1153	0.0933
Winequalityred	0.1007	0.1010	0.1000	0.1326	0.1079	0.1057	0.0968
Wisconsin	0.2181	0.2088	0.2138	0.2310	0.1896	**0.1562**	0.1773

Table 5 Predictive evaluation for MAPE (%) on 15 independent test sets

DATASETS	SEVEN REGRESSION MODELS						
	SVR	LPSVR	LSSVR	MKLSVR	LASSOR	RVR	BSOR
Abalone	5.6773	5.5171	5.5615	9.0456	5.2674	5.2547	5.0950
Autoprice	1.3487	1.3897	1.4730	3.4669	4.3166	**1.1437**	1.3191
Carbolenes	38.1121	26.7350	18.9236	71.7956	34.3588	32.8202	16.8096
Cpuact	1.2634	3.8168	1.3523	20.4029	12.5387	11.5267	1.0550
Housing	3.6501	3.5961	3.5646	8.7288	6.1830	5.1818	4.2671
Lowbwt	1.2597	1.0135	1.5013	3.2371	9.0120	1.6026	0.7458
Parkinsonsupdrs	14.5976	13.1840	13.1101	33.2094	19.0936	17.3491	15.1888
Pdgfr	100.9529	16.8198	22.0317	60.8667	23.0945	20.3880	13.0025
Phenetyl1	92.8691	27.0226	23.6310	22.4161	22.6113	24.1058	23.6935
Sensory	1.6900	1.5988	1.5964	1.6571	5.7341	1.5716	1.5001
Strupcz	28.0019	18.1159	39.6220	23.4518	**12.1454**	37.4012	15.4822
Topo21	1.9608	1.9146	1.9161	1.8156	2.1393	2.1553	1.7900
Triazines	30.0280	30.3953	30.5569	35.2671	31.2660	28.1462	21.8780
Winequalityred	20.3013	20.3495	20.3644	26.4903	22.2907	21.8146	19.7983
Wisconsin	65.4162	68.3766	92.0629	71.2009	81.4570	**58.6001**	63.3174

$$FI\left(f_j\right) = \frac{\mu_j^*}{\sum_{j=1}^{d}\left|\mu_j^*\right|} \times 100\%, j = 1, \cdots, d \tag{46}$$

where $x_i$is the ith instance in a training set and f_j is the j-thfeature in an original feature set.

According to the importance measure $II\ (x_i)$ ($\lambda_i^* \neq 0$) of instances in eq. (38), if $\lambda_i^* > 0$ ($\left|\lambda_i^*\right| > \rho$) then the II value of the input point x_i is greater than 0 and it has a positive contribution to regression. Otherwise, it has a negative contribution. The larger the absolute value of II, the more important the input point x_i, and it is considered a more representative or prototype instance. Obviously, if the II percentage of the input point x_i is zero (let $\lambda_i^* = 0$ if $\left|\lambda_i^*\right| \leq \rho$), then it may be noise or

Table 6 Predictive evaluation for #IIs (IRR %) on 15 training sets

DATASETS	SEVEN REGRESSION MODELS						
	SVR	LPSVR	LSSVR	MKLSVR	LASSOR	RVR	BSOR
Abalone	2054 (14.42)	215 (91.04)	2400 (0.00)	2124 (11.50)	2400 (0.00)	**6 (99.75)**	85 (96.46)
Autoprice	64 (20.00)	38 (52.50)	80 (0.00)	76 (5.00)	80 (0.00)	7 (91.25)	2 (97.50)
Carbolenes	18 (10.00)	18 (10.00)	20 (0.00)	19 (5.00)	20 (0.00)	5 (60.00)	1 (95.00)
Cpuact	813 (79.68)	130 (96.75)	4000 (0.00)	1912 (52.20)	4000 (0.00)	11 (99.73)	6 (99.85)
Housing	222 (30.63)	223 (30.31)	320 (0.00)	297 (7.19)	320 (0.00)	**6 (98.13)**	10 (96.88)
Lowbwt	110 (8.33)	63 (47.50)	120 (0.00)	106 (11.67)	120 (0.00)	6 (95.00)	1 (99.17)
Parkinsonsupdrs	1273 (60.22)	242 (92.44)	3200 (0.00)	2455 (23.28)	3200 (0.00)	**14 (99.56)**	34 (98.94)
Pdgfr	50 (10.71)	49 (12.50)	56 (0.00)	51 (8.93)	56 (0.00)	**4 (92.86)**	4 (92.86)
Phenetyl1	11 (8.33)	6 (50.00)	12 (0.00)	10 (16.67)	12 (0.00)	9 (25.00)	1 (91.67)
Sensory	220 (31.25)	77 (75.94)	319 (0.31)	240 (25.00)	320 (0.00)	5 (98.44)	2 (99.38)
Strupcz	13 (35.00)	11 (45.00)	20 (0.00)	16 (20.00)	20 (0.00)	15 (25.00)	2 (90.00)
Topo21	1945 (51.38)	110 (97.25)	4000 (0.00)	1363 (65.93)	4000 (0.00)	10 (99.75)	1 (99.98)
Triazines	64 (20.00)	17 (78.75)	80 (0.00)	70 (12.50)	80 (0.00)	7 (91.25)	1 (98.75)
Winequalityred	670 (16.25)	44 (94.50)	800 (0.00)	734 (8.25)	800 (0.00)	5 (99.38)	2 (99.75)
Wisconsin	115 (4.17)	51 (57.50)	120 (0.00)	118 (1.67)	120 (0.00)	4 (96.67)	1 (99.17)

Table 7 Predictive evaluation for #IFs (FRR %) on 15 training sets

DATASETS	SEVEN REGRESSION MODELS						
	SVR	LPSVR	LSSVR	MKLSVR	LASSOR	RVR	BSOR
Abalone	10 (0.00)	10 (0.00)	10 (0.00)	10 (0.00)	9 (10.00)	10 (0.00)	7 (30.00)
Autoprice	15 (0.00)	15 (0.00)	15 (0.00)	15 (0.00)	8 (46.67)	15 (0.00)	2 (86.67)
Carbolenes	1142 (0.00)	1142 (0.00)	1142 (0.00)	1142 (0.00)	4 (99.65)	1142 (0.00)	1 (99.91)
Cpuact	21 (0.00)	21 (0.00)	21 (0.00)	21 (0.00)	8 (61.90)	21 (0.00)	7 (66.67)
Housing	14 (0.00)	14 (0.00)	14 (0.00)	14 (0.00)	**5 (57.14)**	14 (0.00)	8 (42.86)
Lowbwt	10 (0.00)	10 (0.00)	10 (0.00)	10 (0.00)	5 (50.00)	10 (0.00)	2 (80.00)
Parkinsonsupdrs	22 (0.00)	22 (0.00)	22 (0.00)	22 (0.00)	10 (54.55)	22 (0.00)	8 (63.64)
Pdgfr	320 (0.00)	320 (0.00)	320 (0.00)	320 (0.00)	**6 (98.13)**	320 (0.00)	13 (95.94)
Phenetyl1	628 (0.00)	628 (0.00)	628 (0.00)	628 (0.00)	**11 (98.25)**	628 (0.00)	11 (98.25)
Sensory	11 (0.00)	11 (0.00)	11 (0.00)	11 (0.00)	11 (0.00)	11 (0.00)	4 (63.64)
Strupcz	1142 (0.00)	1142 (0.00)	1142 (0.00)	1142 (0.00)	26 (97.72)	1142 (0.00)	1 (99.91)
Topo21	266 (0.00)	266 (0.00)	266 (0.00)	266 (0.00)	**5 (98.12)**	266 (0.00)	9 (96.62)
Triazines	60 (0.00)	60 (0.00)	60 (0.00)	60 (0.00)	**2 (96.67)**	60 (0.00)	3 (95.00)
Winequalityred	11 (0.00)	11 (0.00)	11 (0.00)	11 (0.00)	**5 (54.55)**	11 (0.00)	5 (54.55)
Wisconsin	32 (0.00)	32 (0.00)	32 (0.00)	32 (0.00)	4 (87.50)	32 (0.00)	3 (90.63)

redundancy, and it can be removed from the training set. Similarity, for the importance measure *FI* (f_j) ($\mu_j^* \neq 0$) of features in eq. (39), if $\mu_j^* > 0$ ($\left|\mu_j^*\right| > \rho$), then the FI value of the feature f_j is greater than zero and is positively correlated with regression, and it is negatively correlated if the converse is true. The larger the absolute value of FI, the more important the feature f_j, which is considered an important feature. If the FI percentage of the feature f_j is zero (let $\mu_j^* = 0$ if $\left|\mu_j^*\right| \leq \rho$), then it is unimportant and redundant and can be removed from the feature set. With sparse instance and feature sets, the computational efficiency and interpretability of BSOR are enhanced in real-world applications.

All of experiments using SVR, LPSVR, LSSVR, MKLSVR, LASSOR, RVR, and BSOR were implemented on the MATLAB 8.1 (http://www.mathworks.com). To be

Table 8 Predictive evaluation for the training time (seconds) on 15 training sets

DATASETS	SEVEN REGRESSION MODELS						
	SVR	LPSVR	LSSVR	MKLSVR	LASSOR	RVR	BSOR
Abalone	1.4284	13.0092	0.3006	124.6196	4.3902	71.1834	162.7950
Autoprice	0.0065	0.2193	0.0037	0.1009	0.0593	0.0630	0.2727
Carbolenes	0.0064	0.2123	0.0037	0.1218	0.9996	0.0280	0.1113
Cpuact	3.2304	44.1243	0.9884	502.9674	24.9945	253.9874	689.8515
Housing	0.0402	0.7085	0.0080	0.4304	0.1001	0.3270	5.1335
Lowbwt	0.0071	0.2292	0.0038	0.3380	0.1136	0.0693	0.6204
Parkinsonsupdrs	2.4173	39.0570	0.6119	188.5887	11.8648	191.1649	539.2717
Pdgfr	0.0067	0.2202	0.0041	0.0835	0.1393	0.0412	1.4577
Phenetyl1	0.0067	0.2112	0.0034	0.0674	0.3435	0.0204	0.0664
Sensory	0.0171	0.5274	0.0078	0.3251	0.0946	0.2205	3.4673
Strupcz	0.0009	0.2106	0.0037	0.0978	0.6823	0.0245	0.0527
Topo21	3.6367	132.1501	2.9286	263.6716	27.9960	266.2104	919.1724
Triazines	0.0068	0.2367	0.0038	0.1582	0.0655	0.0623	0.1934
Winequalityred	0.5928	1.2169	0.2933	44.3602	7.7791	2.1055	16.4682
Wisconsin	0.0079	0.2260	0.0041	0.7373	0.0640	0.0792	0.9691

Table 9 Predictive evaluation for the test time (seconds) on 15 independent test sets

DATASETS	SEVEN REGRESSION MODELS						
	SVR	LPSVR	LSSVR	MKLSVR	LASSOR	RVR	BSOR
Abalone	0.0766	0.0244	0.0960	0.7537	0.0082	0.0066	0.6320
Autoprice	0.0016	0.0041	0.0039	0.0104	0.0038	0.0035	0.0066
Carbolenes	0.0018	0.0043	0.0040	0.0556	0.0059	0.0036	0.0099
Cpuact	0.1108	0.0249	0.5347	2.1016	0.0064	0.0046	4.8403
Housing	0.0024	0.0052	0.0051	0.0205	0.0039	0.0037	0.0138
Lowbwt	0.0015	0.0041	0.0035	0.0084	0.0058	0.0035	0.0058
Parkinsonsupdrs	0.1029	0.0275	0.2521	1.3861	0.0054	0.0042	2.7063
Pdgfr	0.0016	0.0040	0.0041	0.0110	0.0041	0.0033	0.0083
Phenetyl1	0.0016	0.0039	0.0038	0.0250	0.0047	0.0035	0.0067
Sensory	0.0016	0.0046	0.0055	0.0218	0.0040	0.0035	0.0177
Strupcz	0.0005	0.0042	0.0039	0.0535	0.0048	0.0035	0.0083
Topo21	2.1521	0.1367	4.4906	1.9596	0.0120	0.0040	69.0944
Triazines	0.0018	0.0042	0.0042	0.0126	0.0040	0.0054	0.0098
Winequalityred	0.1443	0.0066	0.1468	1.0931	0.0052	0.0036	0.1063
Wisconsin	0.0017	0.0040	0.0041	0.0108	0.0042	0.0035	0.0081

specific, the convex QP problems of SVR were solved by the SMO algorithm programmed in C++ MEX functions. The linear programming problem of LPSVR was solved by MATLAB with the ellipsoid or interior point algorithm. The system of linear equations of LSSVR was solved with MATLAB, and MKLSVR was sourced from the SimpleMKL toolbox [55]. The QP problem of the LASSOR model was solved with the modified SpaSM [67], which is a MATLAB toolbox for sparse statistical modeling. The RVR model based on expectation-maximization algorithm was solved with the pattern regression MATLAB toolbox [20]. In this study, the SMO algorithm was modified by us as a fast solver for the bound-constrained QP problems, and the computation of the row and column kernel matrices and the revised SMO algorithm were implemented in C++ MEX functions that we defined. Finally, the bound-constrained QP problems of BSOR were solved by using these C++ MEX functions from the MATLAB platform.

4.3 Predictive results and comparative analysis

On the 15 training sets, SVR, LPSVR, LSSVR, MKLSVR, RVR, and BSOR with the RBF kernels along with LASSOR were trained and the optimal regression models with the best performance were found by using the grid search and 5-fold CV methods. Their regression functions were applied to the independent test sets and the corresponding predictive output values were obtained. Thus, for each of 15 test sets, based on the actual and predictive output values, MSE, MAE, MAPE, #IIs, #IFs, IRR, FRR, the training time (seconds), and the test time (seconds) were computed, and their values are shown in Tables 3, 4, 5, 6, 7, 8 and 9. It should be pointed out that for the sake of further analysis of the significant prototype instances and the most important features the best predictive performance on test sets regarding the 5-fold CV method is selected and reported.

The statistics in Table 3 show that BSOR obtained a better MSE than the other regression models on the 9 of the 15 independent test sets. SVR achieved the best MSE on the parkinsonsupdrs dataset, while LPSVR obtained the best MSE on the housing dataset. On the topo21 dataset, the MSE of LSSVR was better than that of the other six regression models. At the same time, SVR, LPSVR, MKLSVR, and BSOR had nearly the same MSE on the topo21 dataset. LASSOR had the best MSE on the strupcz dataset, whereas RVR obtained the best MSE on the autoprice and wisconsin datasets.

The statistics in Table 4 show that the MAE values of BSOR were generally lower than those of the other regression models on 10 of the 15 independent test sets. However, LPSVR had the best MAE on the housing dataset, and LSSVR had the best MAE on the parkinsonsupdrs dataset. Similar to MSE, the best MAE was also obtained by LASSOR on the strupcz dataset, and RVR on the autoprice and wisconsin datasets, respectively.

The results in Table 5 show that the MAPE statistics of BSOR were better than those of the others on 9 of the 15 independent test sets. However, LSSVR had lower MAPE statistics on the housing and parkinsonsupdrs datasets than

Table 10 The best parameters for seven regression models on 15 datasets

DATASETS	SVR		LPSVR		LSSVR		MKLSVR		LASSOR	RVR	BSOR		
	C	σ	C	σ	C	σ	C	σ, δ	C	–	C_1	C_2	σ
Abalone	1	0.2	50	0.5	50	1	5	–	1	–	5	10	0.01
Autoprice	1	2	50	5	50	5	1	–	5	–	2	10	0.5
Carbolenes	1	2	2	0.01	5	10	10	–	1	–	50	50	2
Cpuact	20	1	1	0.5	100	1	20	–	10	–	1	20	0.2
Housing	100	2	100	0.5	50	1	2	–	5	–	20	50	0.1
Lowbwt	1	10	20	10	1	0.2	10	–	5	–	20	2	0.5
Parkinsonsupdrs	1	1	50	1	100	1	1	–	1	–	100	100	0.01
Pdgfr	1	1	100	2	100	2	2	–	1	–	5	50	0.1
Phenetyl1	2	10	5	10	100	10	1	–	1	–	100	20	0.1
Sensory	5	5	5	1	1	0.5	2	–	2	–	20	20	2
Strupcz	2	10	2	2	1	5	2	–	1	–	50	100	1
Topo21	100	0.01	10	2	1	2	1	–	1	–	2	5	0.01
Triazines	1	10	20	5	2	2	20	–	1	–	2	20	0.5
Winequalityred	1	1	20	1	2	0.5	2	–	1	–	5	10	0.2
Wisconsin	1	2	50	10	50	10	100	–	1	–	10	20	0.2

the other regression models, and MKLSVR had the best MAPE on the phenetyl1 dataset. Similarly, LASSOR achieved the best MAPE on the strupcz dataset, while RVR obtained the best MAPE on the autoprice and wisconsin datasets.

Looking at the results of #IIs and IRR in Table 6, BSOR identified the fewest important instances or prototypes in 12 of the 15 training sets than the other six regression models. Similarly, BSOR generally obtained better IRR results than the other six regression models. For #IIs and corresponding IRR, RVR was in second place and LPSVR was in third place. Specifically, RVR achieved the best IRR on the abalone, housing, and parkinsonsupdrs datasets, and it had the same IRR as BSOR on the pdgfr dataset. Apparently, we found that LSSVR and LASSOR can hardly obtain the sparse instance set. After using instance-sparsification, BSOR improved the predictive performance with fewer instances (important or representative input points) in different datasets (see Tables 3, 4, and 5). At the same time, the interpretability of BSOR was evidently enhanced because only a small number of important or representative instances are extracted from the training sets. In other words, these instances can be used as a benchmark or prototype for value predictions of unseen data based on a measure of similarity or distance between them.

The results of #IFs and FRR in Table 7 show that apart from instance-sparsification, BSOR extracted the fewest important features from 10 of the 15 training sets, while the other five regression models employed all the features, except for LASSOR. Concretely, LASSOR selected the minimum number of features from the housing, pdgfr, topo21, and triazines datasets, and it and BSOR selected the equal number of features on the phenetyl1 and winequalityred datasets. FRR showed that BSOR outperforms the other five regression models. Obviously, after using feature-sparsification, BSOR increased predictive accuracy and interpretability for forecasting by regression (see Tables 3, 4, and 5). That is to say, after removing redundant or irrelevant features from dataset, BSOR only used a small quantity of important features to produce a better predictive generalization. At the same time, BSOR can provide interpretable results for users by using the analysis of important factors based on the statistical correlation between selected relevant features and output values.

From the perspective of the comparative CPU time, for the seven regression models of SVR, LPSVR, LSSVR, MKLSVR, RVR, and BSOR with the RBF kernels along with LASSOR, the averages of their training and test time (seconds) on 15 datasets by using the 5-fold CV method are collected and reported in Tables 8 and 9 respectively.

As the training time shown in Table 8, LSSVR spent the less training time than the other six regression models on 14 of the 15 training sets. At the same time, SVR occupied the least training time on the strupcz dataset, and it is in second place for the training time. And then there are LASSOR, LPSVR, and RVR with the low CPU usage. MKLSVR took more training time than others due to the simultaneous usage of polynomial and RBF kernels. Similarly, BSOR iteratively solved two bound-constrained QP problems so that more training time is spent by it than that of others.

As the test time shown in Table 9, SVR utilized the less test time than the other six regression models on 10 of the 15

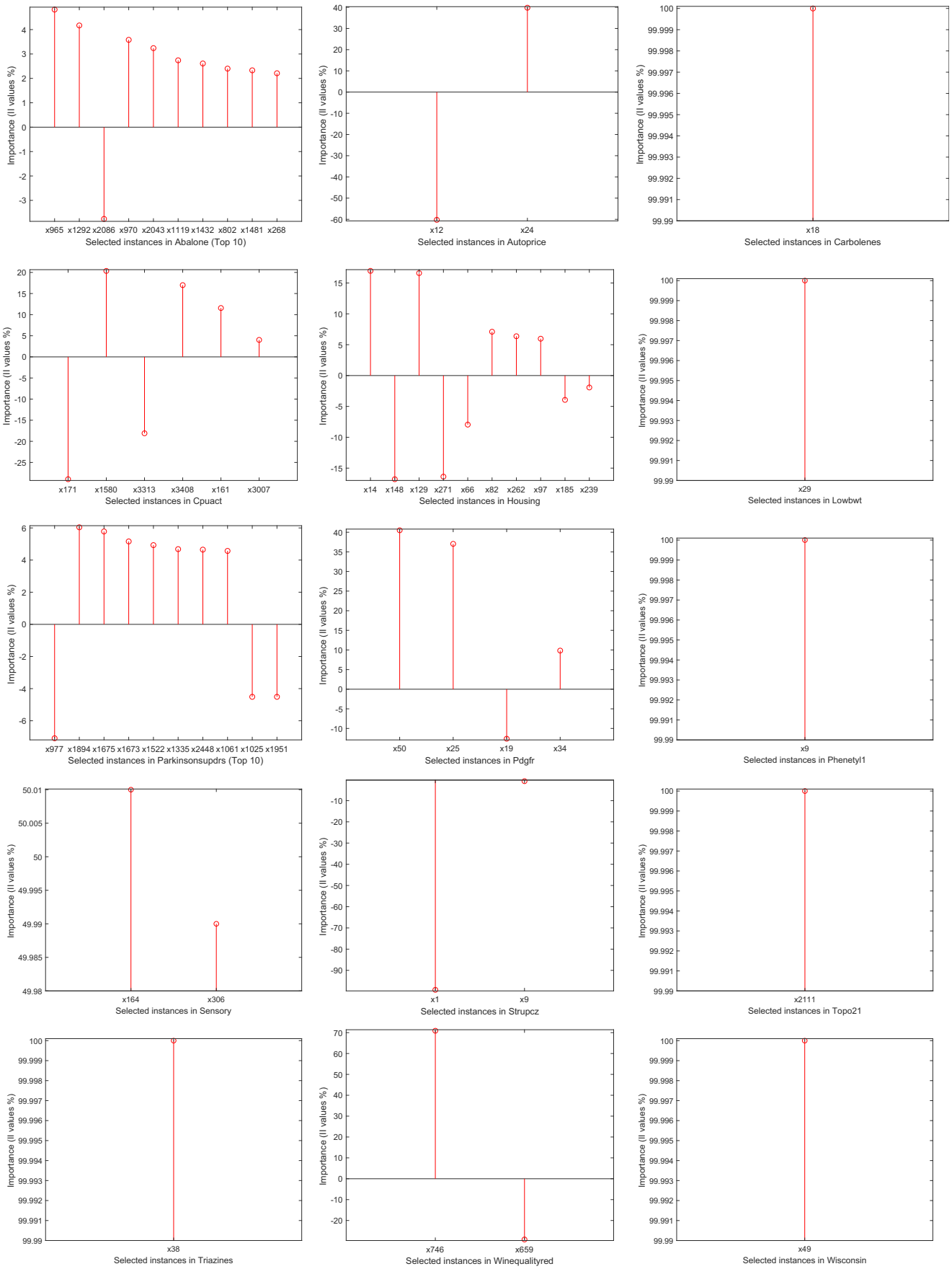


Fig. 2 The II analysis for BSOR on the 15 datasets

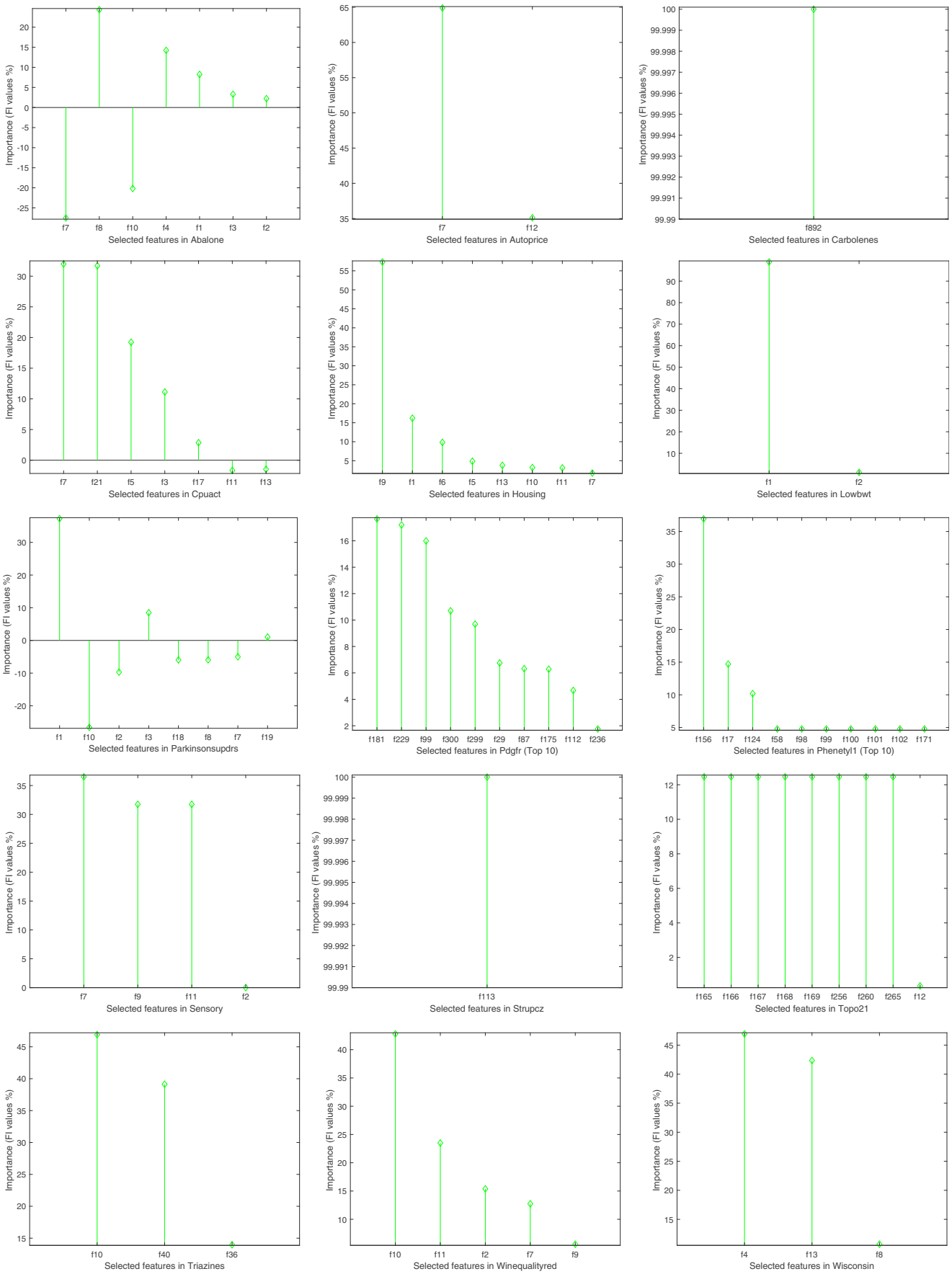


Fig. 3 The FI analysis for BSOR on the 15 datasets

Table 11 The MIIs and MIFs with their HII and HFI values on 15 datasets

Datasets	MIIs	HII (%)	MIFs	HFI (%)
Abalone	x_{965}	4.82	f_7	−27.51
Autoprice	x_{12}	−60.23	f_7	64.89
Carbolenes	x_{18}	100.00	f_{892}	100.00
Cpuact	x_{171}	−28.97	f_7	36.92
Housing	x_{14}	16.96	f_9	36.52
Lowbwt	x_{29}	100.00	f_1	98.89
Parkinsonsupdrs	x_{977}	−7.09	f_1	37.26
Pdgfr	x_{50}	40.50	f_{181}	17.65
Phenetyl1	x_9	100.00	f_{156}	100.00
Sensory	x_{164}	50.01	f_7	16.88
Strupcz	x_1	−99.18	f_{113}	100.00
Topo21	x_{2111}	100.00	f_{165}	12.46
Triazines	x_{38}	100.00	f_{10}	46.90
Winequalityred	x_{746}	70.90	f_{10}	42.79
Wisconsin	x_{49}	100.00	f_4	46.93

independent test sets. RVR took the least training time on 5 of the 15 independent test sets, and it is therefore second place. For BSOR with the similar time complexity to MKLSVR, owing to the minimum number of instances and features selected by BSOR, it spent the less test time than the former.

Besides, for the penalty factors C, C_1, and C_2, the bandwidth $\sigma$for the RBF kernel, and the degree δ for the polynomial kernel, the best parametric values found by the grid search method of seven regression models on 15 datasets are collected and reported in Table 10.

Finally, it should be pointed out that MKLSVR has employed the whole parametric sets of the bandwidth σ for the RBF kernel and the degree δ for the polynomial kernel by using multiple kernel learning, whereas RVR utilized the expectation-maximization algorithm to estimate unknown parameters.

4.4 Importance analysis of extracted instances and features

Compared with SVR, LPSVR, LSSVR, MKLSVR, LASSOR, and RVR, BSOR generally identified the minimum number of important instances from 15 training sets (see Table 6). Based on the optimal coefficient vector λ^*, these extracted instances with $|\lambda_i^*|\neq 0$ ($i = 1, \cdots, n$) were considered as prototypes or representatives, which are important evidence for decision-making. Similarly, BSOR extracted the minimum number of important features from 15 training sets (see Table 7) while the other five regression models used all the features except for LASSOR. According to the optimal weight vector μ^*, those selected features with $|\mu_j^*|\neq 0$ ($j = 1, \cdots, d$) were regarded as critical factors, which are important

Table 12 Performance comparison for p values (TT / WSRT / FT) of nine measures

PERFORMANCE MEASURES	BSOR vs Others					
	SVR	LPSVR	LSSVR	MKLSVR	LASSOR	RVR
MSE	0.0397 / 0.0043 / 0.0045	0.0117 / 0.0043 / 0.0045	0.0350 / 0.0051 / 0.0045	0.0341 / 0.0001 / 0.0001	0.1180 / 0.0006 / 0.0005	0.0931 / 0.0044 / 0.0027
MAE	0.0146 / 0.0038 / 0.0027	0.0111 / 0.0061 / 0.0027	0.0082 / 0.0041 / 0.0027	0.0003 / 0.0004 / 0.0001	0.0169 / 0.0009 / 0.0005	0.0220 / 0.0057 / 0.0027
MAPE	0.0746 / 0.0084 / 0.0027	0.0147 / 0.0072 / 0.0027	0.0633 / 0.0131 / 0.0124	0.0118 / 0.0007 / 0.0005	0.0022 / 0.0027 / 0.0027	0.0230 / 0.0052 / 0.0027
#IIs (IRR)	0.0129 / 0.0004 / 0.0001	0.0002 / 0.0004 / 0.0001	0.0191 / 0.0004 / 0.0001	0.0120 / 0.0004 / 0.0001	0.0191 / 0.0004 / 0.0001	0.6672 / 0.1548 / 0.0201
#IFs (FRR)	0.0429 / 0.0004 / 0.0001	0.0429 / 0.0004 / 0.0001	0.0429 / 0.0004 / 0.0001	0.0429 / 0.0004 / 0.0001	0.1744 / 0.1788 / 0.1088	0.0429 / 0.0004 / 0.0001
trTime	0.0651 / 0.0004 / 0.0001	0.0628 / 0.0045 / 0.0455	0.0650 / 0.0004 / 0.0001	0.1165 / 0.0113 / 0.0455	0.0661 / 0.0052 / 0.0124	0.0698 / 0.0004 / 0.0001
tsTime	0.2787 / 0.0038 / 0.0005	0.2783 / 0.0032 / 0.0005	0.2809 / 0.0038 / 0.0005	0.3132 / 0.2342 / 0.0124	0.2779 / 0.0001 / 0.0001	0.2778 / 0.0004 / 0.0001

references for reason tracking and explanation. Therefore, for each of the 15 datasets, corresponding with the best predictive performance of the optimal BSOR model found by the combing grid search and 5-fold CV method, the optimal coefficient vector λ^* and optimal weight vector μ^* were selected, and their II and FI values, defined by eq. (45) and eq. (46), respectively, were computed and are reported in Figs. 2 and 3.

As shown in Figs. 2 and 3, some representative instances and important features were identified and extracted from each dataset. At the same time, other noisy or redundant instances and unimportant or irrelevant features were automatically removed by BSOR. Because the number of representative instances and important features in some datasets was greater than 10, for the II values, the top 10 instances of the abalone and parkinsonsupdrs datasets are listed, and for the FI values, the top 10 features of the pdgfr and phenetyl1 datasets are given. The IIs and FIs are shown in decreasing order for all the datasets. Obviously, for each of 15 datasets, the most important instances (MIIs), their highest II (HII) values, the most important features (MIFs), and their highest FI (HFI) values are respectively reported in Table 11.

For the above HII or HFI values of MIIs or MFIs in Table 11, if a value is greater than zero, then the corresponding instance or feature had a positive correlation with forecasting. Otherwise, it has a negative correlation with value prediction. Finally, for each dataset in practical applications, we can provide the prototype and factor analysis for forecasting based on selected important instances and features. Hence the prediction by BSOR is traceable and interpretable.

4.5 Analysis of experimental results

The simulation, experimental results, comparative analysis, and importance analysis of instances and features show that BSOR generally performs better than SVR, LPSVR, LSSVR, MKLSVR, LASSOR, and RVR on 16 real datasets (a simulation and 15 experimental datasets), i.e., BSOR extracts and employs the minimal number of instances and features without sacrificing predictive performance (see Tables 3, 4, and 5). At the same time, the BSOR model enhances the interpretability of forecasting by selecting relevant instances and features. Thus we say that the proposed model has multiple functions of instance identification, feature selection, and value prediction. Generally, we also find that BSOR has better instance- and feature-reduction than the other six regression models, especially for high-dimensional datasets (see Tables 6 and 7). It should be noted that since BSOR iteratively solves two quadratic optimization problems until convergence, its model training sometimes may consume more system time than the other regression models. At the same time, the bottom-up computation of the reconstructed row and column kernel matrices may affect the training and test time of BSOR (see Tables 8 and 9). For instance-sparsity, the best regression model is BSOR, and then we have RVR in second place, LPSVR in third place, SVR and MKLSVR in fourth place. LSSVR and LASSOR utilize almost the entire sets of instances and features, i.e., they are unable to generate sparse instance sets. For feature-sparsity, the best regression model is BSOR and LASSOR is in second place. We also find that SVR, LPSVR, MKLSVR, and RVR cannot identify the important features, hence these methods are not helpful in providing interpretable results for value prediction in many practical applications.

5 Discussion

To objectively and fairly evaluate the differences between BSOR and the other six regression models, we employed the statistical tests for the above seven measures of MSE, MAE, MAPE, #IIs (IRR), #IFs (FRR), the training time (trTime), and the test time (tsTime). Specifically, parametric and non-parametric statistical comparisons with the two-sample t-test (TT), two-sample Wilcoxon signed-ranks test (WSRT), and Friedman's test (FT) were conducted on 16 datasets (1 simulation +15 experimental datasets) [23, 31, 33, 79, 81, 86]. The p values of the TT, WSRT, and FT statistics for nine measures were calculated and are reported in Table 12.

We chose 0.05 as the threshold value for p. To put it another way, we should reject the null hypothesis that there is no significant difference between BSOR and another regression model if the p-value from TT, WSRT, or FT is less than 0.05. Otherwise, we should accept it. As the p-values in Table 12 demonstrate, for the TT statistics, in general, there was a statistically significant difference between BSOR and the other six regression models. At the same time, from the p-values in Table 12, at the 0.05 level of significance there was no significant difference between BSOR and the other four models of SVR with 0.0746 for MAPE, LSSVR with 0.0633 for MAPE, LASSOR with 0.1180 for MSE and 0.1744 for #IFs (FRR), and RVR with 0.0931 for MSE and 0.6672 for #IIs (IRR). Similarly, for the WSRT statistics, there was a statistically significant difference between BSOR and the other six regression models. For MSE, MAE, MAPE, #IIs (IRR), and #IFs (FRR), the p-values show that the predictive performance of BSOR is generally better than those of SVR, LPSVR, LSSVR, MKLSVR, LASSOR, and RVR on the 16 test sets. Except that at the 0.05 level of significance there was no significant difference between BSOR and the other two models of LASSOR with 0.1788 for #IFs (FRR) and RVR with 0.1548 for #IIs (IRR). Besides, for the FT statistics, the p-values showed that the predictive performance of BSOR was generally better than that of SVR, LPSVR, LSSVR, MKLSVR, LASSOR, and RVR on the 16 test sets except for LASSOR with 0.1088 for #IFs (FRR). Although LASSOR and BSOR had no significant difference in #IFs

(FRR) on some datasets, MSE, MAE, MAPE, and #IIs (IRR) showed that the predictive performance of LASSOR was significantly degraded. At the same time, RVR and BSOR had the similar #IIs (IRR) on some datasets, but MSE, MAE, MAPE, and #IFs (FRR) showed that RVR for instance-sparsity causes the degraded accuracy. Obviously, simultaneous instance- and feature-sparsity does not cause information loss, instead BSOR provides the best predictive performance. Under equal predictive performance, BSOR extracts and employs the minimal number of instances and features for value forecasting.

For the TT statistics of trTime and tsTime, at the 0.05 level of significance their p-values show that there was no statistically significant difference between BSOR and the other six regression models. However, for the WSRT and FT statistics of trTime and tsTime, at the 0.05 level of significance their p-values show that there was a statistically significant difference between BSOR and the other six regression models, except for MKLSVR with the WSRT statistic equalling 0.2342 for tsTime.

Finally, computational complexity measures how many basic steps an algorithm uses to solve a problem as a function of its size. Here, we only consider time complexity. SVR and LASSOR employ the SMO algorithm to solve the QP problem (2) with time complexity $O(n^3)$, where n is the sample size. LPSVR employs the ellipsoid or interior point algorithm to solve the LP problem with polynomial time complexities $O(n^6d^2)$ and $O(n^{3.5}d^2)$, respectively, where d is the dimensional size. LSSVR solves the unconstrained QP problem or the system of linear equations with the same time complexity $O(n^3)$. RVR needs to compute the posterior weight covariance matrix, which requires a Cholesky decomposition with the order $O(N^3)$ complexity, where N is the number of basis functions. MKLSVR combines the multi-kernel learning method and the SMO algorithm to solve a QP problem with time complexity $O(Kn^3)$, where K ($K \in \mathbb{N}$) is determined by the type of kernel function and the number of kernel parameters. For Algorithm 1, BSOR solves the bound-constrained QP problems (20) and (30) using the modified SMO algorithm with time complexity $O(Mn^3)$, where M ($M \in \mathbb{N}$) is the maximum number of iterations.

6 Conclusion

We have proposed a novel regression (BSOR) approach based on the reconstructed row and column kernel matrices and the iterative bi-sparse optimization. In addition to value prediction, BSOR can simultaneously identify prototype instances with the most important features. Then they are used as a benchmark to obtain interpretable predictions of unseen input points. On the 16 real datasets, BSOR generally achieved better predictive performance than SVR, LPSVR, LSSVR, MKLSVR, LASSOR, and RVR. The parametric and nonparametric statistics and their p-values of a two-sample t-test, two-sample Wilcoxon signed-ranks test, and Friedman's test show that there is a statistically significant difference between BSOR and the other six regression models, except for the inconsistency between parametric and nonparametric tests in the CPU time. Simulations based on the practical dataset, experimental results, comparison analysis, and importance analysis of selected instances and features have shown that BSOR is an effective regression method for predicting continuous values, discovering representative instances, and identifying important features, and it can give the best reduction for high-dimensional data and the interpretability for forecasting. So, it has great potential as a forecasting approach for other real-world applications. In this study, because BSOR needs to iteratively solve two convex QP problems, its training time sometimes exceeds those of the other six regression models. Thus we plan to construct an online learning-based bi-sparse regression model with simultaneous value prediction and selection of relevant instances and features that can be used to efficiently solve large-scale and high-dimensional forecasting problems in real-world applications.

Acknowledgements The authors would like to thank the anonymous reviewers for their valuable comments and suggestions. This research has been partially supported by the Natural Science Foundation of Shandong, China (ZR2016FM15), and the National Natural Science Foundation of China (#61877061, #61872170).

Compliance with Ethical Standards

Conflict of Interest The authors declare that they have no conflict of interest.

Ethical Approval The article does not contain any studies with human participants or animals performed by any of the authors.

Informed Consent Informed consent was obtained from all individual participants included in the study.

References

1. Abe S (2010) Support vector Machines for Pattern Classification, 2nd edn. Springer, London, UK
2. Ahdesmaki M, Strimmer K (2010) Feature selection in omics prediction problems using cat scores and false nondiscovery rate control. Ann Appl Stat 4(1):503–519
3. Bach F, Jenatton R, Mairal J, Obozinski G (2012) Optimization with sparsity-inducing penalties. Found Trends Mach Learn 4(1): 1–106
4. Bache K, Lichman M (2013) UCI machine learning repository. University of California, School of Information and Computer Science, Irvine, CA http://archive.ics.uci.edu/ml
5. Basak D, Pal S, Patranabis DC (2007) Support vector regression. Neural Inf Process-Lett Rev 11(10):203–224
6. Berk RA (2008) Statistical learning from a regression perspective. Springer, New York

7. Berrendero JR, Cuevas A, Torrecilla JL (2016) Variable selectionin functional data classification: a maxima-hunting proposal. Stat Sin 26:619–638
8. Bi J, Bennett K, Embrechts M, Breneman C, Song M (2003) Dimensionality reduction via sparse support vector machines. J Mach Learn Res 3:1229–1243
9. Blanquero R, Carrizosa E, Jimenez-Cordero A, Martin-Barragan B (2018) Variable selection with support vector regression for multivariate functional data. In: Technical report. Edinburgh - Universidad de Sevilla, University of
10. Blanquero R, Carrizosa E, Jimenez-Cordero A, Martin-Barragan B (2019) Variable selection in classification for multivariate functional data. Inf Sci 481:445–462
11. Bolón-Canedo V, Sánchez-Maroño N, Alonso-Betanzos A (2015) Feature selection for high-dimensional data. Artificial Intelligence: Foundations, Theory, and Algorithms 10:978–973
12. Bradley PS, Mangasarian OL (1998) Feature selection via concave minimization and support vector machines. In ICML 98:82–90
13. Broniatowski M, Celant, Giorgio (2016) Interpolation and extrapolation optimal designs. 1, polynomial regression and approximation theory 1st Ed. Wiley-ISTE
14. Carrizosa E, Guerrero V (2014) Rs-sparse principal component analysis: A mixed integer nonlinear programming approach with vns. Comput Oper Res 52:349–354
15. Carrizosa E, Ramirez-Cobo P, Olivares-Nadal AV (2016) A sparsity-controlled vector autoregressive model. Biostatistics 18(2):244–259
16. Chandrashekar G, Sahin F (2014) A survey on feature selection methods. Comput Electr Eng 40(1):16–28
17. Cheng L, Ramchandran S, Vatanen T, Lietzén N, Lahesmaa R, Vehtari A, Lähdesmäki H (2019) An additive Gaussian process regression model for interpretable non-parametric analysis of longitudinal data. Nat Commun 10(1):1798
18. Cotter A, Shalev-Shwartz S, Srebro N (2013) Learning optimally sparse support vector machines. In ICML, pp:266–274
19. Cristianini N, Shawe-Taylor J (2000) An introduction to support vector machines and other kernel-based learning methods. Cambridge University Press, Cambridge
20. Cui Z, Gong G (2018) The effect of machine learning regression algorithms and sample size on individualized behavioral prediction with functional connectivity features. NeuroImage 178:622–637
21. Cunningham JP, Ghahramani Z (2015) Linear dimensionality reduction: survey, insights, and generalizations. J Mach Learn Res 16:2859–2900
22. David JO (2017) Linear regression. Springer
23. Demsar J (2006) Statistical comparison of classifiers over multiple data sets. J Mach Learn Res 7:1–30
24. Deng N, Tian Y, Zhang C (2013) Support vector machines: optimization based theory. Algorithms and Extensions, Chapman & Hall/CRC
25. Draper NR, Smith H (1998) Applied regression analysis, vol 326. John Wiley & Sons
26. Duch W, Winiarski T, Biesiada J, Kachel A (2003) Feature selection and ranking filters. In: International conference on artificial neural networks (ICANN) and International conference on neural information processing (ICONIP), vol 251, p 254
27. Efron B, Hastie T, Johnstone I, Tibshirani R (2004) Least angle regression. Ann Stat 32(2):407–499
28. Ehsanes Saleh AKM, Arashi M, Golam Kibria BM (2019) Theory of ridge regression estimation with applications. Wiley
29. Fabio A, Donini M (2015) EasyMKL: a scalable multiple kernel learning algorithm. Neurocomputing 169:215–224
30. Fan J, Lv J (2008) Sure independence screening for ultrahigh dimensional feature space. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 70(5):849–911
31. García S, Fernández A, Luengo J, Herrera F (2010) Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: experimental analysis of power. Inf Sci 180:2044–2064
32. Garg R, Khandekar R (2009) Gradient descent with sparsification: an iterative algorithm for sparse recovery with restricted isometry property. In: Proceedings of the 26th annual international conference on machine learning. ACM, pp 337–344
33. Gibbons JD, Chakraborti S (2011) Nonparametric statistical inference, 5th edn. Chapman & Hall/CRC Press, Taylor & Francis Group, Boca Raton
34. Gönen M, Alpaydin E (2011) Multiple kernel learning algorithms. J Mach Learn Res 12:2211–2268
35. Gu Y, Liu T, Jia X, Benediktsson JA, Chanussot J (2016) Nonlinear multiple kernel learning with multiple-structure-element extended morphological profiles for hyperspectral image classification. IEEE Trans Geosci Remote Sens 54(6):3235–3247
36. Gunn SR (1998) Support vector machines for classification and regression, vol 14. ISIS technical report, pp 85–86
37. Hammer B, Villmann T (2002) Generalized relevance learning vector quantization. Neural Netw 15(8):1059–1068
38. Hu M, Chen Y, Tin-Yau KJ (2009) Building sparse multiple-kernel SVM classifiers. IEEE Trans Neural Netw 20(5):827–839
39. Huang K, Zheng D, Sun J, Hotta Y, Fujimoto K, Naoi S (2010) Sparse learning for support vector classification. Pattern Recogn Lett 31(13):1944–1951
40. Jacek W, Rodriguez PJ, Esquerdo (2018) Applied regression analysis for business: tools, Traps and Applications. Springer
41. James GM, Wang J, Zhu J (2009) Functional linear regression that's interpretable. Ann Stat 37(5A):2083–2108
42. Johansson U, Linusson H, Löfström T, Boström H (2018) Interpretable regression trees using conformal prediction. Expert Syst Appl 97:394–404
43. Kira K, Rendell LA (1992) A practical approach to feature selection. In: Proceedings of the ninth international workshop on machine learning, pp 249–256
44. Koenker R, Hallock KF (2001) Quantile regression. J Econ Perspect 15(4):143–156
45. Liu H, Motoda H (2007) Computational methods of feature selection. CRC Press
46. López J, Maldonado S, Carrasco M (2018) Double regularization methods for robust feature selection and SVM classification via DC programming. Inf Sci 429:377–389
47. Martínez AM, Kak AC (2001) PCA versus LDA. IEEE Trans Pattern Anal Mach Intell 23(2):228–233
48. McLachlan GJ (2004) Discriminant analysis and statistical pattern recognition. Wiley Interscience
49. Micchelli CA, Pontil M (2005) Learning the kernel function via regularization. J Mach Learn Res 6:1099–1125
50. Neumann J, Schnörr C, Steidl G (2005) Combined SVM-based feature selection and classification. Mach Learn 61(1–3):129–150
51. O'Brien CM (2016) Statistical learning with Sparsity: the lasso and generalizations. CRC press
52. Orabona F, Jie L, Caputo B (2012) Multi kernel learning with online-batch optimization. J Mach Learn Res 13:227–253
53. Pelckmans K., Goethals I., Brabanter J. De, Suykens J. A., Moor B. De (2005). Componentwise Least Squares Support Vector Machines. in Support Vector Machines: Theory and Applications, (Wang L., ed.), Springer, Berlin 77–98
54. Qiu S, Lane T (2005) Multiple kernel learning for support vector regression. In: Computer science department, the University of new Mexico, Albuquerque, NM, USA, tech. Rep, 1
55. Rakotomamonjy A, Bach FR, Canu S, Grandvalet Y (2008) SimpleMKL. J Mach Learn Res 9:2491–2521

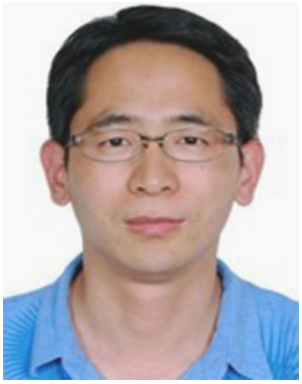
56. Ramsay JO, Silverman BW (2002) Applied functional data analysis: methods and case studies, volume 77 of springer series in statistics. Springer-Verlag
57. Ramsay JO, Silverman BW (2005) Functional data analysis, 2nd edn. Springer-Verlag, Springer Series in Statistics
58. Rao N, Nowak R, Cox C, Rogers T (2016) Classification with the sparse group lasso. IEEE Trans Signal Process 64(2):448–463
59. Rhinehart RR (2016) Nonlinear regression modeling for engineering applications: modeling, model validation, and enabling design of experiments. John Wiley & Sons
60. Rish I, Grabarnik G (2014) Sparse modeling: theory, algorithms, and applications. CRC press
61. Sato A, Yamada K (1996) Generalized learning vector quantization. In: Advances in neural information processing systems, pp 423–429
62. Schmidt M (2005) Least squares optimization with L1-norm regularization, vol 504. CS542B project report, pp 195–221
63. Shim J, Hwang C (2015) Varying coefficient modeling via least squares support vector regression. Neurocomputing 161:254–259
64. Shlens J (2014) A tutorial on principal component analysis. arXiv preprint arXiv:1404.1100
65. Shrivastava A, Patel VM, Chellappa R (2014) Multiple kernel learning for sparse representation-based classification. IEEE Trans Image Process 23(7):3013–3024
66. Silverman BD, Platt DE (1996) Comparative molecular moment analysis (CoMMA): 3D-QSAR without molecular superposition. J Med Chem 39(11):2129–2140
67. Sjöstrand K, Clemmensen LH, Larsen R, Einarsson G, Ersbøll BK (2018) Spasm: A matlab toolbox for sparse statistical modeling. J Stat Softw 84(10):1–37
68. Smola AJ, Schölkopf B (2004) A tutorial on support vector regression. Stat Comput 14(3):199–222
69. Sonnenburg S, Ratsch G, Schafer C, Scholkopf B (2006) Large scale multiple kernel learning. J Mach Learn Res 7:1531–1565
70. Subrahmanya N, Shin YC (2010) Sparse multiple kernel learning for signal processing applications. IEEE Trans Pattern Anal Mach Intell 32(5):788–798
71. Suykens JA, Lukas L, Vandewalle J (2000) Sparse least squares support vector machine classifiers. In ESANN, pp:37–42
72. Suykens JA, Van Gestel T, De Brabanter J (2002) Least squares support vector machines. World Scientific
73. Suykens JA, Signoretto M, Argyriou A (2014) Regularization, optimization, kernels, and support vector machines. Chapman and Hall/CRC
74. Suykens JA (2017) Efficient Sparse Approximation of Support Vector Machines Solving a Kernel Lasso. Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications: 21st Iberoamerican Congress, CIARP 2016, Lima, Peru, Nov. 8–11, 2016, Proceedings. vol. 10125. Springer
75. Tan M, Wang L, Tsang IW (2010) Learning sparse svm for feature selection on very high dimensional datasets. In: Proceedings of the 27th international conference on machine learning (ICML-10), pp 1047–1054
76. Thangavel K, Pethalakshmi A (2009) Dimensionality reduction based on rough set theory: A review. Appl Soft Comput 9(1):1–12
77. Tibshirani R (1996) Regression shrinkage and selection via the lasso. J R Stat Soc, Series B (Methodological), 267–288
78. Tipping ME (2001) Sparse Bayesian learning and the relevance vector machine. J Mach Learn Res 1:211–244
79. Trawiński B, Smętek M, Telec Z, Lasota T (2012) Nonparametric statistical analysis for multiple comparison of machine learning regression algorithms. Int J Appl Math Comput Sci 22(4):867–881
80. Wall ME, Rechtsteiner A, Rocha LM (2003) Singular value decomposition and principal component analysis. In: A practical approach to microarray data analysis. Springer US, pp 91–109
81. Wasserstein RL, Lazar NA (2016) The ASA statement on p-values: context, process, and purpose. Am Stat 70(2):129–133
82. Weston J, Elisseeff A, Schölkopf B, Tipping M (2003) Use of the zero-norm with linear models and kernel methods. J Mach Learn Res 3:1439–1461
83. Yamada M, Jitkrittum W, Sigal L, Xing EP, Sugiyama M (2014) High-dimensional feature selection by feature-wise kernelized lasso. Neural Comput 26(1):185–207
84. Zhang Y, Wang S, Phillips P (2014) Binary PSO with mutation operator for feature selection using decision tree applied to spam detection. Knowl-Based Syst 64:22–31
85. Zhang Z, Gao G, Tian Y, Yue J (2016) Two-phase multi-kernel LP-SVR for feature sparsification and forecasting. Neurocomputing 214:594–606
86. Zhang Z, He J, Gao G, Tian Y (2019) Bi-sparse optimization-based least squares regression. Appl Soft Comput 77:300–315
87. Zhao YP, Sun JG (2011) Multikernel semiparametric linear programming support vector regression. Expert Syst Appl 38:1611–1618
88. Zhou W, Zhang L, Jiao L (2002) Linear programming support vector machines. Pattern Recogn 35(12):2927–2936
89. Zou H, Hastie T (2005) Regularization and variable selection via the elastic net. J R Stat Soc Ser B 67:301–320
90. Zou H (2006) The adaptive lasso and its oracle properties. J Am Stat Assoc 101(476):1418–1429

Zhiwang Zhang received the Ph.D. degree in computer science from Chinese Academy of Sciences in 2009. He is currently a Professor in the School of Information and Electrical Engineering at Ludong University, China. His research interests are in the areas of data mining and knowledge discovery, forecasting, machine learning, optimization, artificial intelligence, and management and decision support.

Guangxia Gao received the MS degree from Dalian Maritime University major in applied linguistics and computational linguistics. She is currently a Lecturer in Shandong Technology and Business University. Her research interests are in the areas of computational linguistics, corpus linguistics, text mining, and natural language processing.

Tao Yao received the Ph.D. degree from the Dalian University of Technology, China, in 2017. He is currently an Associate Professor in the School of Information and Electrical Engineering, Ludong University, China. His research interests include multimedia retrieval, computer vision, and machine learning.

Jing He is currently a Professor in the school of software and electrical engineering, Swinburne University of Technology, Australia. She was awarded a PhD degree from the Academy of Mathematics and System Science, Chinese Academy of Sciences in 2006. She has been active in areas of Data Mining, Web service/Web search, Spatial and Temporal Database, Multiple Criteria Decision Making and has published over 120 research papers in refereed international journals and conference proceedings.



Yingjie Tian is a Doctor, Professor of Research Center on Fictitious Economy and Data Science, Chinese Academy of Sciences. He has done his first degree in mathematics (1994), Master in applied mathematics (1997), and Ph.D. in Management Science and Engineering. He has published 4 books about SVMs, one of which has been cited over 1000 times. His research interests include support vector machines, optimization theory and applications, data mining, intelligent knowledge management, risk management.