Regression Analysis Using Core Vector Machine Technique Based on Kernel Function Optimization

Babak Afshin[1], MohammadEbrahim Shiri[2]*, Kamran Layeghi[3], Hamid HajSeyyedJavadi[4]

Abstract–Core vector regression (CVR) is an extension of the core vector machine algorithm for regression estimation by generalizing the minimum bounding ball (MEB) problem. As an estimator, both the kernel function and its parameters can significantly affect the prediction precision of CVR. In this paper, a method to improve CVR performance using pre-processing based on data feature extraction and Grid algorithm is proposed to obtain appropriate parameters values of the main formulation and its kernel function. The CVR estimated mean absolute error rate here is the evaluation criterion of the proposed method that should be minimized. In addition, some benchmark datasets out of different databases were used to evaluate the proposed parameter optimization approach. The obtained numerical results show that the proposed method can reduce the CVR error with an acceptable time and space complexity. Therefore, it is able to deal with very large data and real world regression problems.

Keywords: Core vector regression, Kernel function, Grid algorithm, Parameter selection

1. Introduction

Kernel function-based methods have always been used in a wide variety of machine learning and pattern recognition problems. These methods, such as support vector (SVM) in classification problems [1] or support vector regression (SVR) in regression analysis [2], are mainly developed as quadratic programming problems (QP) and based on the size of the training sample. A typical implementation of QP for kernel function-based methods can take the $O(N^3)$ time complexity and the $O(N^2)$ space complexity.

Core vector machine (CVM), as one of the common methods, uses an approximation algorithm to deal with the minimal bounding ball (MEB) problems in computational geometry. According to the size of the training patterns, the CVM algorithm has the advantage of O(N) time complexity and space complexity independent of the size of the training set. The core vector regression (CVR) algorithm also inherits the simplicity of CVM and has small asymptotic time and space complexities. Its generalized form can be used with linear/nonlinear kernel functions or applied to kernel based methods and other normalized kernel function. Empirically, they are supposed to be as accurate as existing SVM implementations but much faster and produce much fewer support vectors for large data[3].

In methods based on the kernel function, the regression performance depends on the kernel function and the value of its parameters, which can be applied in the CVR theory. Common kernel functions used in regression analysis using CVR are radial basis function (RBF), polynomial and other normalized kernel function [4].

A number of studies utilized experiential technique to determine parameters such as C parameter, ε along with those parameters that are usually found in standard kernel functions. They also used them to analyze CVR regression estimation. The parameter C makes the trade-off between the training error and the generalization ability, whereas the parameter ε controls the growing rate of the MEB radius until it encloses the uncovered points.

In this article, a method to automatically determine the kernel function parameters using the Grid algorithm, which has been successfully used in data classification, will be presented to minimize the CVR regression error estimation. Therefore, there is no need to determine the parameter settings in advance and trial and error can be avoided to determine the appropriate parameter settings. In addition, to the measurement criterion, the mean absolute error (MAE), was used to check the performance of the proposed method dealing with the benchmark data. Based on the experimental results, the proposed method has a low regression error rate that can be used for mathematical problems and real-world applications.

The other sections of this article are prepared as follow: Section 2 provides a summary of former studies for parameter optimization that tries to reduce CVR regression error. Section 3 gives a brief introduction to the methods based on kernel function for regression. Section 4 contains

1,3 Department ofComputer Engineering, North Tehran Branch, Islamic Azad University, Tehran, Iran.Email:b.afshin@iau-tnb.ac.ir, k_layeghi@iau-tnb.ac.ir

2* Corresponding Author: Department of Mathematics and Computer Sciences, Amirkabir University of Technology, Tehran, Iran. Email:shiri@aut.ac.ir.

4 Department of Mathematics and Computer Sciences, Shahed University, Tehran, Iran. Email:h.s.javadi@shahed.ac.ir

Received: 2023.10.09; Accepted: 2023.12.17

our proposed method. Section 5 includes the numerical results and finally, the conclusion is explained in detail.

2. Former Researches

In this section, the kernel methods of previous studies are examined, the details are as follows. Lorenzi et al. [5] have proposed a new reconstruction method for missing data in the image, which integrates radiometric information and the position of the missing value in the reconstruction process. For this purpose, the information including the position of the pixels in the image has been added to the radiometric information and a special kernel function has been selected and adapted for each type of information adopted in the regression. They have proposed a new combination of kernel functions, which can be used to achieve better reconstruction. By adopting this new kernel function combination in SVR, it turns out that a small number of support vectors are needed to reconstruct the missing region. However, the cost of its superior effectiveness is the larger number of free parameters for estimation compared to the reference reconstruction methods, which require higher calculation time. Shafizadeh Moghadam and others [6] to model land cover change from SVR with different linear (LN) and non-linear single kernel functions such as RBF, sigmoid, polynomial to identify non-linear patterns in land use data in urban growth has been used. With such an approach, choosing the appropriate kernel function in SVR has a direct impact on the accuracy of the Land Cover Change (LCC) model. Therefore, different kernel functions are used in two general steps in combination with GA and without GA, and then the resulting simulated maps of each combination using a recently modified version of receiver operating characteristics (ROC) called Total Operating Characteristic (TOC) has been studied. To show the importance of feature selection in the performance of the model, when all the selected explanatory factors were related to the selected kernel functions, GA was used to perform an optimal feature selection process by removing the features which are relevant in the binary bit coding. It is used to set them to zero at the end of the algorithm operation. The results show that the synergy between GA and SVR can effectively optimize the process of selecting the variables used in the preparation of the LCC model, and as a result, strengthen the accuracy of the SVR prediction, which has the highest performance related to the SVR-RBF-GA model and the lowest one, corresponds to the SVR-LN-GA model. Wu et al. [7] proposed a multi-kernel regression algorithm based on gradient boosting for short-term electric charge forecasting. The proposed BMKR solves the MKL problem using a reinforcement learning approach with less computational cost. They first adopt reinforcement to learn a set of multiple kernel regressions and then extend this framework to the field of homogeneous and heterogeneous transfer learning. Experimental results on the residential data show that the prediction error can be reduced by a large margin based on the knowledge which has been learned from other houses.

The mentioned methods are limited to specific applications and suffer from the limitations of SVM in large-scale regression problems. Besides, they are not planned to work with very large data. Even the methods that have used the kernel combination or MKL, except for a few cases, have often not taken any action to pre-process the data and reduce its redundancy. Therefore, in this research, we tried to solve this weakness with the proposed method.

Since CVM regression in kernel function parameter optimization has not been noticed by researchers, previous studies are reviewed as follows. In [8], Tsang et al. have extended the CVM algorithm to regression settings by generalizing the MEB problem. Their core vector regression algorithm can be used with both linear and non-linear kernel functions and is able to obtain the possible approximate optimal solution. Their proposed method has also been successfully applied to large 3D point sets in computer graphics for implicit surface modeling. Gu et al. in [9] have presented a new algorithm called sequential corevector machine, which, like the original CVM, has approximately linear time complexity and space complexity independent of the number of training data. Experiments on five different preliminary regression algorithms have shown that their proposed algorithm has a generalization capability comparable to existing SVM implementations when the data size increases, and additional training for OCVM is also possible, so it can be used in an online scenario.

Gu et al. [10] proposed an extreme vector machine (EVM) for fast training of SVM with different but common loss functions in a large dataset. For the regression work, they used the least square loss function and named the resulting machine LS-EVMR. Their experimental results show that EVM performs better or at least comparable to CVR algorithm, L2 Support Vector Machine - Fast Regression Vector Kernel Function Density Estimation Algorithm (L2SVR-FastKDE) in terms of accuracy, training time, number of support vectors and robustness. Approximate extreme point support vector machine (AESVM).

3. Kernel Based Methods for Regression

In this section, the basis of the CVR theory is explained and the condition of the proposed kernel function is discussed to construct the hybrid kernel CVR to improve its regression performance.

3.1 Kernel Based Methods as MEB Problems

Consider the Support Vector Data Description (SVDD) [11] with a hard margin:

$$arg \min_{c,R} J_1 = \min_{c,R} R^2 : \|c - \varphi(x_i)\|^2 \leq R^2 . \mathrm{i} = 1, \dots, \mathrm{m} \tag{1}$$

Where φ is the feature mapping associated with a given kernel function k and B(c, R) is the desired minimal enclosing ball in the feature space induced by the kernel function. The dual of that QP is the following:

$$arg\max_{\alpha} J_2 = arg\max_{\alpha} \alpha^T diag\,(K) - \alpha^T K\alpha\,, \quad \alpha \geq 0, \alpha^T 1 = 1 \tag{2}$$

where α = $[\alpha_i,\ldots, \alpha_m]^T$ are Lagrange multipliers, 0=$[0, \ldots, 0]^T$, 1=$[1, \ldots, 1]^T$ and $K_{m\times m} = [k(x_i, x_j)] = [\varphi(x_i)^T\varphi(x_j)]$ is the kernel matrix. When k satisfies condition (3).

$$k\ (x,\ x) = \kappa \tag{3}$$

Using α^T=1 we have α^Tdiag (K) = κ. By removing this constant term from QP, we obtain a simpler optimization problem:

$$arg\max_{\alpha} J_3 = arg\max_{\alpha} \quad -\alpha^T K\alpha\,.\,\alpha^T 1 = 1, \alpha \geq 0, \mathrm{i} = 1, \ldots, \mathrm{m} \tag{4}$$

On the contrary, whenever the kernel function k satisfies the condition (3), any QP of the form (4) can be considered as a MEB problem.

3.2 TheCenter Constrained MEB Problem

MEB in equation (1) finds the smallest ball containing all φ (x_i) inside S. Now each φ (x_i) with an additional $\Delta_i \in$ R augmented to form [φ (x_i)', Δ_i]', and then find the MEB for these augmented points while the last coordinates of the center of the ball limited to zero (for example, as [c', 0]'). [See Figure (1)]

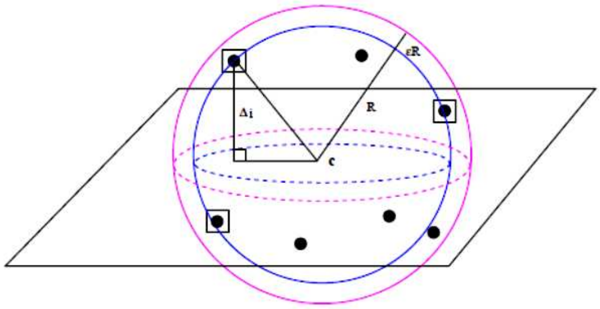

Fig.1. The center constrained MEB problem

Subsequently, the objective function of equation (1) will change as follows:

$$arg\min_{c,R} J_4 = \arg\min_{c,R} \quad R^2 \quad \text{s.t.} \|c - \varphi(x_i)\|^2 + \Delta_i{}^2 \leq R^2 \ , \mathrm{i} = 1, \ldots, \mathrm{m} \tag{5}$$

Where $\Delta = [\Delta_1{}^2 \ldots\ldots \Delta_m{}^2]' \geq 0$. The new duo will be as follows:

$$arg\max_{\alpha} J_5 = \arg\max_{\alpha} \quad \alpha^T (diag(K) + \Delta) - \alpha^T K\alpha \quad \text{s.t.} \boldsymbol{\alpha^T 1 = 1}\ , \alpha \geq 0 \tag{6}$$

Using the optimal α we can recover the values of c and R as follows:

$$c = \sum_{i=1}^{n} \alpha_i \varphi(x_i)\,, \qquad R = \sqrt{\alpha^T diag(K + \Delta) - \alpha^T K\alpha} \tag{7}$$

So, the square of the distance between c and any other point will be as follows:

$$\|c - \varphi(x_l)\|^2 + \Delta_l{}^2 \leq \|c\|^2 - 2(K\alpha)_l + k_{ll} + \Delta_l^2 \tag{8}$$

Due to the restriction of α'1=1, several arbitrary α'1 can be added to the objective function without affecting the solution of α. In other words, for aη∈ R, we can transform the dual as follows:

$$arg\max_{\alpha} J_6 = \arg\max_{\alpha} \quad \alpha^T (diag(K) + \Delta) - \alpha^T K\alpha \quad \text{s.t.} \alpha^T 1 = 1\ , \alpha \geq 0 \tag{9}$$

Any quadratic programming problem in the form of equation (9), with Δ ≥ 0 can now be considered as a MEB problem with the definition of equation (5).

3-3- CVR algorithm

Assuming that the set of core, center and radius of a ball in repetition t is represented by S_t, c_t and R_t respectively, the CVR algorithm considering the parameter ε> 0 will be as follows:

The steps of the CVR algorithm **Error! Reference source not found.**[12] are as follows:

a) Initialize s_0, c_0 and R_0.

b) Terminate if there is no training point z_i such that φ (z) falls outside the (1 + ε) ball distance B(c_t , (1+ε) r_t); For example, $\sqrt{\|c_t - \varphi(z_i)\|^2 + \Delta_i^2} > (1 + \varepsilon)R_t$

c) Find z_i such that $\widetilde{\varphi}(z_i)$ has the farthest distance from ct. Set S_{t+1} to $S_t \cup$ {z}.

d) Find the new MEB(S_{t+1}). Set c_{t+1} = c $_{\mathrm{MEB(St+1)}}$ and R_{t+1} = R $_{\mathrm{MEB(St+1)}}$.

e) Set t = t + 1 and go back to step (b).

The most common kernel functions are[17], [18]:

RBF:	$K(x,y) = e^{(-\gamma*\|x-y\|^2)}$	(10)
Polynomial:	$K(x,y) = (\gamma\langle x,y\rangle + r)^d$	(11)
Sigmoid:	$K(x,y) = tanh(\gamma\langle x,y\rangle + r)$	(12)
Laplacian:	$K(x,y) = e^{-\sqrt{\gamma}*\|x-y\|}$	(13)
Inverse Distance:	$K(x,y) = \dfrac{1}{\gamma * \|x-y\|^2 + 1}$	(14)

4. Proposed Method

In this section, our parameter optimization method for CVR is described as follows.

4.1 Parameter Optimization for CVR

In this method, we try to form a CVR -based regression analysis system using the conventional kernelfunctions and find the optimal values of the common kernelfunctions and parameters in the main CVR formulas such as C and ε the resulting system by the Grid algorithm [14]. It should be noted that the Grid algorithm is used in the data classification by the RBF kernel function, and in this study we will also apply this algorithm to select the parameter of other kernel functions to improve its effectiveness in improving system regression analysis. In order to reduce data dimensions, computational complexity and facilitate system performance on large investors, a preprocessing based on the feature extraction of system input data has also been applied. Figure (2) shows the diagram of proposed regressor using Grid algorithm and its components.

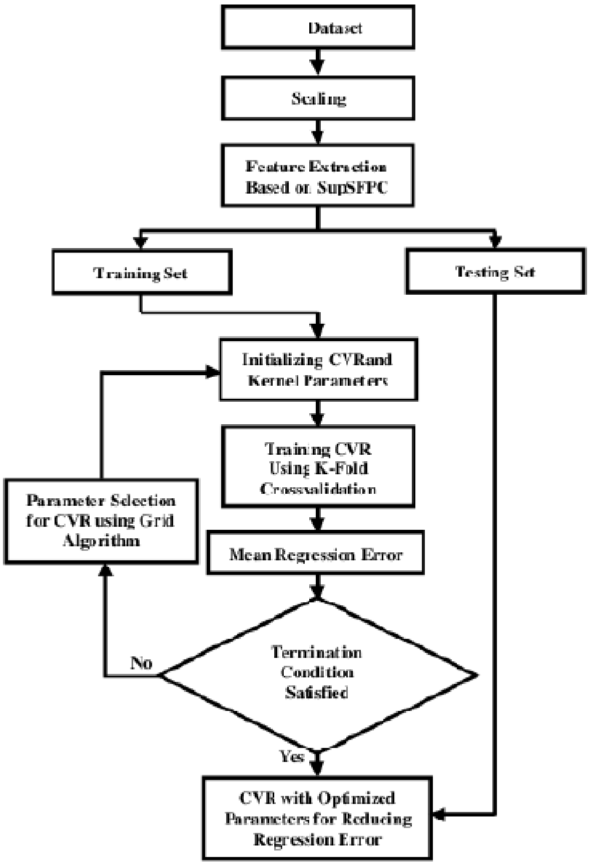
Dataset
Scaling
Feature Extraction Based on SupSFPC
Training Set
Testing Set
Initializing CVRand Kernel Parameters
Training CVR Using K-Fold Crossvalidation
Parameter Selection for CVR using Grid Algorithm
Mean Regression Error
Termination Condition Satisfied
No
Yes
CVR with Optimized Parameters for Reducing Regression Error

Fig.2. The architecture of the proposed system to reduce the CVR analysis error based on the parameter optimization using Grid algorithm

4.1.1 Scaling of Input Dataset

At the beginning of this procedure, the input data is read and the attribute values are scaled to the range [1, 0] or [-1, 1] depending on the type of data and its attribute values, in order to avoid the influence of rates. Different data in the values of features and numerical problems during calculations must be avoided.

4.1.2DataPre-processing Based on Distributed Principal Component Analysis and Performance with Observer to the Proposed Method in Regression Analysis

Finding patterns between data in high dimensions can be difficult. The Supervised Sparse and Functional Principal Component Analysis (SupSFPC) framework can combine monitoring information to recover underlying data structures that are more interpretable. This framework unifies and generalizes several existing methods in which it develops an efficient modified expectation-maximization algorithm for parameter estimation [15]. Also, it has fast data-driven procedures for adjusting parameter selection. In this way, we can perform the regression analysis of the desired data better and more efficiently.

4.1.3 Parameter Selection for CVR Parameters and its Kernel Function Based on Grid Algorithm

The description of kernel function parameters and their improvement method using Grid algorithm in the proposed method is given below.

In this method, the list of parameters in the CVR formulation and kernel functions that we have considered for this problem in order to find suitable values for them is shown in Figure (3):

RBF:	C	ε	γ_1		
Laplacian:	C	ε	γ_2		
Inverse Distance:	C	ε	γ_3		
Polynomial:	C	ε	γ_4	r	D

Fig.3. List of real value parameters for the CVR formulation and considered kernel functions based on the Grid algorithm

Parameter C is related to the term of penalty in the main formula of CVR and determines the margin of the decision making boundaries with the interval of allowed changes [0 1000], parameter ε is the growth rate of the radius of MEB in order to include new samples with the interval of changes [0.000001 0.1], the γ parameter is related to RBF, laplacian, inverse distance and polynomial functions with the allowed variation interval [0.1 10]. Also, the bias parameter with the range of changes [1 10] and the degree parameter with the range of changes [2 5] are also considered in the polynomial kernel function.

4.1.3.1 Evaluation criterion

Since the purpose of this method is to reduce the regression analysis error of the core vector machine, we have considered the smallness of the regression analysis error on the test set of the target data using the mean absolute error method that will be introduced.

This method of error calculation in regression analysis is done using equation (15).

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|f(x_i) - y_i| \qquad (15)$$

5- Experimentation and results

We have used various examples of artificial and real world datasets to conduct experiments in different stages of our research. Synthetic data are production data used for a specific situation that are not obtained by direct measurement, but are data that are produced, stored and used by researchers to perform scientific processes. Real world data is defined as data derived from a number of sources that relate to outcomes in a heterogeneous population of patients, biological, economic, statistical phenomena, etc. in real world settings. The most popular source for real-world data, the UCI [16], which we will use in our experiments.

In experimental research, artificial data after statistical distributions as well as data from real world programs are used **as** test data sets. The use of artificial data set allows testing the algorithm or data structure under certain exact conditions or in extreme conditions.

In order to examine the abilities and capabilities of the proposed model in regression analysis, we have used the change and update of the toolbox implemented by Frank Michael Schleif[19] in the classification of data for CVM in the MATLAB software environment.

To measure the effectiveness of our proposed methods, we used the datasets of UCI databases, DELVE repository, MLnet Archive and StatLib. Table 1 shows the specifications of the benchmark datasets. In the case of the Forest cover type dataset, according to[20], we intend to separate class 2 from other classes. As described in[21], we transform this classification problem into a regression by predicting +1 for class 2 samples and −1 for other classes.

Table 1. Characteristics of benchmark datasets used in regression analysis experiments

Dataset Name	Number of Features	Number of Samples
Pyrimidines	27	74
Traizine	60	176
Machine CPU	6	209
Housing	13	506
Stock	9	949
Abalone	8	4118
Computer Activity	21	8192
Kinematics	8	8192
Pole Telecommunication	48	15000
Elevators	18	16599
Census House	137	22884
Friedman	10	40768
Forest Cover Type	54	581012

In Table 2, we have shown the test results with the proposed method, which contains the error rate of the regression analysis obtained for each of the investigated dataset in the average state of ten times of running the Grid algorithm. The result of the regression analysis by implementing the proposed method for each data is displayed in the form of mean ± standard deviation. The results indicate that other kernel functions often have comparable regression analysis error than the RBF kernel function. In case of small and medium datasets, laplacian kernel function has comparable performance rather than RBF kernel but in case of large datasets RBF kernel function has a lower error rate.

In Table 3, we have reported the amount of core vector stored by the proposed method for each of the benchmark dataset. As can be seen from the results, in most cases, the polynomial kernel function has stored less number of core vectors than other kernel functions, particularly the RBF kernel function. Therefore, the regression analysis system based on this kernel function in the proposed method will be less complicated. Figure 4 shows the number of storedcore vectors of the proposed parameter optimization method on forest cover type dataset for different sizes of the training set. Thus, the number of core vectors increases with the size of the training set.

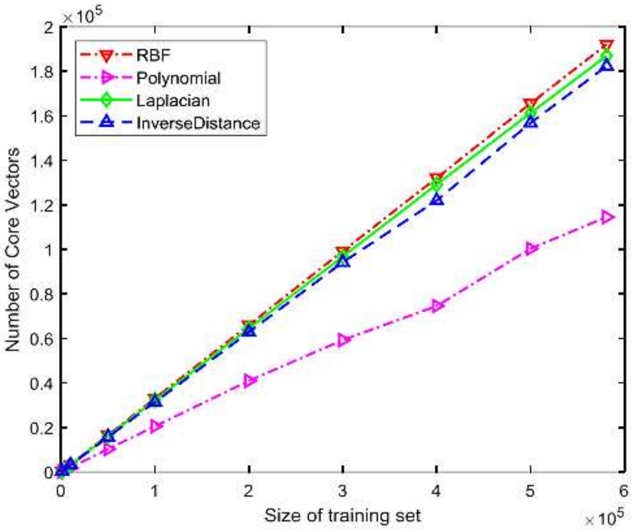

Fig.4. The number of core vectors stored in regression analysis of the proposed method for the Forest Cover Type dataset

In Table 4, we have reported the average execution time in each experiment for training and testing each of the benchmark datasets in the proposed method. All the columns related to the training times follow the pattern of minutes: seconds, which is related to the entire training set. All the columns related to the training and testing time of the dataset with the value of less than one minute, are calculated in seconds.

As results showed, the inverse distance kernel function had a better training time in most cases and had a faster testing time for large datasets.

Figure 5 shows the performance of the proposed method based on MAE criterion for different values of the parameter ε on the Friedman dataset. It can be seen that the proposed method has low MAE rate for different kernel functions. Therefore, setting a value of $\varepsilon = 10^{-4}$ is acceptable as a compromise between training speed and regression error for most kernel functions.

5. Conclusion

In the current paper, an approach to reduce CVR regression error was proposed. Based on the experimental results, the proposed method has a low error rate, which can be related to the use of pre-processing based on distributed and functional principal component analysis with the data observer and the selection of the kernel function parameter by the Grid algorithm.

The reasons for the success of the proposed method can be pointed to the use of the linear time complexity of CVR, the ability to estimate a better regression of the kernel function with the real modeling of the parameters based on the Grid algorithm, which enables us to better search and more effectively adjust the values of the CVR parameters, leads to a reduction in the regression error rate with an acceptable time and memory complexity.

Table 2. Regression errors of the proposed method based on the MAE criterion

Dataset Name	RBF	Laplacian	Inverse Distance	Polynomial
Pyrimidines	0.0106±0.0050	0.0108±0.0031	0.0116±0.0044	**0.0105±0.0035**
Traizine	0.0049±0.0023	**0.0041±0.0018**	0.0045±0.0019	0.0048±0.0021
Machine CPU	**0.5317±0.3246**	5731±0.3319	0.6035±0.3034	0.5593±0.3333
Housing	0.1074±0.0481	**0.0975±0.0406**	0.1097±0.0410	0.1018±0.0396
Stock	0.1227±0.0185	**0.1193±0.0159**	0.1205±0.0164	0.1221±0.0183
Abalone	0.0054±0.0019	**0.0052±0.0015**	0.0055±0.0023	0.0053±0.0021
Computer Activity	**0.0244±0.0066**	0.0252±0.0058	0.0250±0.0069	0.0248±0.0067
Kinematics	9.4914e-05±1.0203e-05	0.0011±1.4895e-05	0.0096±8.7152e-05	**9.4034e-05±5.3494e-06**
Pole Telecomm	**0.0099±0.0057**	0.0117±0.0069	0.0132±0.0058	0.0116±0.0070
Census House	4.1915e-04±5.0221e-05	1.1878e-04±3.6017e-05	**9.7574e-05±2.0287e-05**	1.0624e-04±1.3708e-05
Friedman	**7.4558e-04±1.4096e-04**	8.0420e-04±5.0855e-05	8.2576e-04±1.1848e-05	8.3975e-04±1.3133e-04
Forest Cover Type	**2.7092e-06±1.3801e-06**	3.5060e-06±1.3902e-06	3.1076e-06±1.6903e-06	3.7054e-06±1.1953e-06

Table 3. The number of the core vectors produced in the proposed method for each of the benchmark dataset

Dataset Name	RBF	Laplacian	Inverse Distance	Polynomial
Pyrimidines	36	38	34	**33**
Traizine	89	93	87	**85**
Machine CPU	103	101	104	**51**
Housing	247	245	244	**196**
Stock	471	474	453	**336**
Abalone	2006	2038	2031	**1947**
Computer Activity	**3347**	4062	4064	3889
Kinematics	**3934**	4053	3947	3986
Pole Telecomm	4921	4948	4942	**1456**
Census House	9743	9754	9729	**2948**
Friedman	18997	20183	19583	**18627**
Forest Cover Type	192071	191604	191436	**123104**

Table 4. The caption must be followed by the table

Dataset Name	RBF Train	Laplacian Train	Inverse Distance Train	Polynomial Train	RBF Test	Laplacian Test	Inverse Distance Test	Polynomial Test
Pyrimidines	0.0238	0.0201	**0.0168**	0.0232	**0.0050**	0.0087	0.0055	0.0061
Traizine	0.0489	0.0747	**0.0393**	0.0441	0.0039	0.0046	**0.0011**	0.0029

Machine CPU	0.0383	**0.0375**	0.0379	0.0391	0.0031	0.0038	0.0031	**0.0028**
Housing	0.1401	0.1332	**0.1324**	0.1379	0.0059	0.0062	**0.0051**	0.0054
Stock	0.3552	0.3478	**0.3467**	0.3515	0.0125	**0.108**	0.0136	0.0152
Abalone	**0.4789**	0.4983	0.5031	0.4904	**0.1070**	0.1219	0.1203	0.1153
Computer Activity	1.4968	**1.4693**	1.5273	1.5473	0.2989	**0.2954**	0.3024	0.3198
Kinematics	1.8532	1.7471	**1.7258**	1.7552	0.3601	0.3464	**0.3282**	0.3555
Pole Telecommunication	**6.3002**	6.3799	6.4002	6.5244	1.2301	0.1415	0.1423	**0.1332**
Census House	1.5712	1.7467	1.6155	**1.5644**	0.2498	0.3326	0.2565	**0.2496**
Friedman	2.9953	**2.7269**	2.8238	2.8608	0.6242	0.5852	**0.5468**	0.5546
Forest Cover Type	20.5658	20.5254	**20.3267**	20.6564	0.0299	0.0315	**0.0294**	0.0306

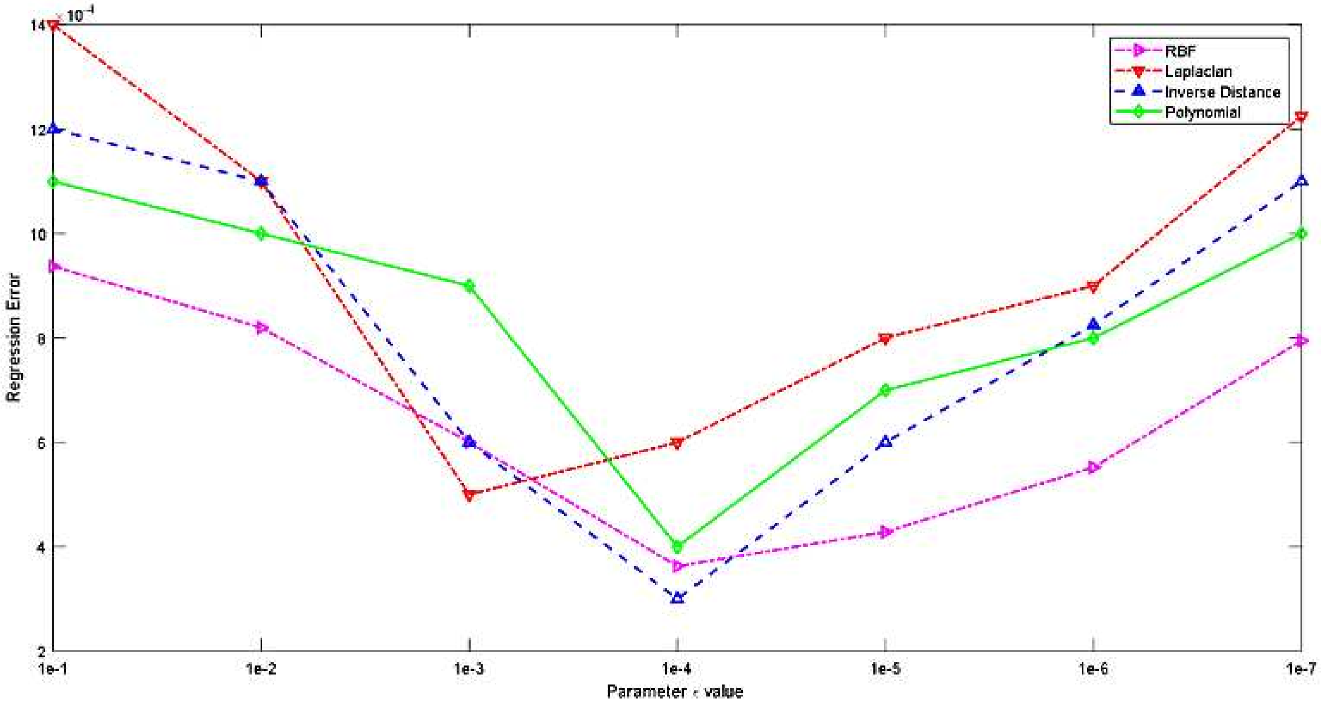

Fig.5. Performance studyof CVR mean absolute error for different types of kernel functions with different values of ε parameter on Friedman dataset.

References

[1] V. Vapnik, *The Nature of Statistical Learning Theory, New York*: Springer Verlag, 1995.

[2] B. Scholkopf and A.J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*: MIT Press, 2002.

[3] IvorW. Tsang James T. Kwok and Pak-Ming Cheung, "Very Large SVM Training using Core Vector Machines", *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics (AISTATS)*, Barbados, 2005.

[4] E. Alpaydin, *Introduction to Machine Learning, 2nd edition*: MIT Press, 2010.

[5] Luca Lorenzi, Grégoire Mercier and FaridMelgani," Support vector regression with kernel combination for missing data reconstruction", *IEEE Geoscience and Remote Sensing Letters*, Vol.10, No.2:367 – 371, 2013.

[6] HosseinShafizadeh-Moghadam, Amin Tayyebi, Mohammad Ahmadlou, Mahmoud Reza Delavar and Mahdi Hasanlou, "Integration of genetic algorithm and multiple kernel support vector regression for modeling urban growth", *Journal of Computers, Environment and Urban Systems*, Vol. 65: 28–40, 2017.

[7] D. Wu, B. Wang, D. Precup and B. Boulet, "Multiple Kernel Learning-Based Transfer Regression for Electric Load Forecasting," in *IEEE Trans on Smart Grid*, Vol.11, No.2: 1183-1192, March. 2020.

[8] Ivor W. Tsang, James T. Kwok and Kimo T. Lai, "Core vector regression for very large regression problems,"

Proceedings of the Twenty-Second International Conference on Machine Learning, p. 913–920, 2005.

[9] B. Gu, J.D. Wang and T. Li, "Ordinal-class core vector machine", *Journal Of Computer Science And* Technology, Vol. 25, No.4: 699–708, 2010.

[10] X. Gu, F.L. Chung and S. Wang, "Extreme vector machine for fast training on large data,"*International Journal of Machine Learning and Cybernetics*,Vol.11:33–53,2019.

[11] D.M.J. Tax and R.P.W. Duin, "Support vector data description," *Journal of Machine Learning,* Vol.54: 45-66, 2004.

[12] I. Tsang, J. Kwok and J. Zurada, "Generalized core vector machines," *IEEE Trans on Neural Networks*, Vol. 17, No. 5: 1126-1140, 2006.

[13] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*: Cambridge University Press, 2004.

[14] C.W. Hsu, C.C. Chang and C.J. Lin, *A Practical Guide to Support Vector Classification*,Department of Computer Science and Information Engineering, National Taiwan University,2003, Available at: http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf.

[15] G. Li, H. Shen and J.Z. Huang, "Supervised Sparse and Functional Principal Component Analysis," *Journal of Computational and Graphical Statistics*, Vol. 25, No. 3: 859-878, 2016.

[16] S. Hettich, C.L. Blake and C.J. Merz, *UCI Repository of Machine Learning Databases, Department of Information and Computer Science*,University of California, Irvine, 1998.

[17] A. J. Smola and B. Scholkophf, "A tutorial on Support Vector Regression", *NeuroCOLT2 Technica Report Series, NC2-TR-1998-03*, Oct, 2003.

[18] M.G. Genton, "Classes of Kernels for Machine Learning: A Statistics Perspective,"*Journal of Machine Learning Research*, Vol. 2: 299-312, 2001.

[19] F.M. Schleif, Matlab implementation of the Core Vector Machine of I. Tsang et al., "Core Vector Machines: Fast SVM Training on Very Large Data Sets," *Published in Journal of Machine Learning Research*,http://www.techfak.unibielefeld.de/~fschleif/software.xhtml.

[20] R. CollobertandS. Bengio, "SVMTorch: Support vector machines for large-scale regression problems.,"*Journal of Machine Learning Research*, Vol. 1: 143–160, 2001.

[21] R. Collobert, S. BengioandY.Bengio, "A parallelmixture of SVMs for very large scale problems," *Journal ofNeural Computation*, Vol. 14: 1105–1114, 2002.