Evolutionary Support Vector Regression Machines

Ruxandra Stoean
University of Craiova
Department of Computer Science
A. I. Cuza, No 13, 200585, Craiova, Romania
ruxandra.stoean@inf.ucv.ro

D. Dumitrescu
Babes-Bolyai" University of Cluj-Napoca
Department of Computer Science
M. Kogalniceanu, No 1B, 400084, Cluj, Romania
ddumitr@cs.ubbcluj.ro

Mike Preuss
University of Dortmund
Department of Computer Science
Otto-Hahn 14, 44221, Dortmund, Germany
mike.preuss@cs.uni-dortmund.de

Catalin Stoean
University of Craiova
Department of Computer Science
A. I. Cuza, No 13, 200585, Craiova, Romania
catalin.stoean@inf.ucv.ro

Abstract

Evolutionary support vector machines (ESVMs) are a novel technique that assimilates the learning engine of the state-of-the-art support vector machines (SVMs) but evolves the coefficients of the decision function by means of evolutionary algorithms (EAs). The new method has accomplished the purpose for which it has been initially developed, that of a simpler alternative to the canonical SVM approach for solving the optimization component of training. ESVMs, as SVMs, are natural tools for primary application to classification. However, since the latter had been further on extended to also handle regression, it is the scope of this paper to present the corresponding evolutionary paradigm. In particular, we consider the hybridization with the classical ϵ-support vector regression (ϵ-SVR) introduced by Vapnik and the subsequent evolution of the coefficients of the regression hyperplane. ϵ-evolutionary support regression (ϵ-ESVR) is validated on the Boston housing benchmark problem and the obtained results demonstrate the promise of ESVMs also as concerns regression.

1. Introduction

This paper represents the first attempt of the novel evolutionary support vector machines (ESVMs) learning technique to tackle regression.

ESVMs [11], [12], [13] represent the paradigm that emerged from the hybridization between support vector machines (SVMs) and evolutionary algorithms (EAs). The reason for their development has come to meet the idea of an easier approach for the optimization problem within the canonical counterpart. While the latter then employs the quite difficult generalization of the Lagrange multipliers method, the new ESVMs evolve the coefficients of the decision function and, in addition, obtain their values in a direct and interactive means.

As SVMs had been initially designed for classification purposes and later conversed for regression tasks, the evolutionary alias technique has followed the same route.

The aim of this paper is therefore to also extend the application of ESVMs to the regression field and to demonstrate that they remain as competitive as they have a priori proved to be for the classification domain [11], [12], [13].

The novel ESVMs for regression will incorporate the classical ϵ-support vector regression (ϵ-SVR) learning engine [14] while regression coefficients will be consequently evolved by an EA.

The paper is organized as follows. Section 2 introduces the concepts specific to canonical SVMs for regression. The choice and construction of either a linear or a nonlinear SVM regression model is explained in the different subsections. The novel ESVMs for regression are presented in Section 3: the components of the EA are described and the way to compute the prediction capacity of the obtained regression model is explained. In addition, a reconsidered version of the EA, which confers a simpler and more effective representation from an evolutionary point of view, is undertaken. Section 4 contains the experiment of the new technique against the Boston housing task. Experimental setup and parameter setting are outlined and the obtained results are illustrated. Also, the comparison to those of the canonical approach and a simple linear regression model on the same problem is undertaken. In the final section conclu-

sions are reached and ideas for future work are discussed.

2. An Overview of Support Vector Machines for Regression

Let it be given a training set $\{(x_i, y_i)\}_{i=1,2,...,m}$, where every $x_i \in R^n$ represents a data sample and each $y_i \in R$ a target. Such a data set could represent exchange rates of a currency measured in subsequent days together with econometric attributes [9] or a medical indicator registered in multiple patients along with personal and medical information [1].

The task of ϵ-SVR [14] is to find a function $f(x)$ that has at most ϵ deviation from the actual targets of data samples and, simultaneously, is as flat as possible [9]. In other words, the aim is to estimate the regression coefficients of $f(x)$ with these requirements.

While the former condition for ϵ-SVR is straightforward, i.e. errors are allowed as long as they are less than ϵ, the latter one needs some further explanation [8]. Resulting values of the regression coefficients may affect the model in the sense that it fits current training data but has low generalization ability, which would contradict the principle of Structural Risk Minimization for SVMs [15]. In order to avoid this situation, it is required to choose the flattest function in the definition space.

Another way to interpret ϵ-SVR is that training data are constrained to lie on a hyperplane that allows for some error and, at the same time, has high generalization ability.

2.1. Linear Support Vector Machines for Regression

Suppose a linear regression model can fit the training data. Consequently, function f has the form:

$$f(x) = \langle w, x \rangle - b, \tag{1}$$

where $w \in R^n$ is the slope of the regression hyperplane and $b \in R$ is the intercept, i.e. the point at which the surface intersects the y-axis.

The task of ϵ-SVR is then mathematically translated as follows. On the one hand, the condition that f approximates training data with ϵ precision is written as:

$$|y_i - (\langle w, x_i \rangle - b)| \leq \epsilon, i = 1, 2, ..., m. \tag{2}$$

or, alternatively, as:

$$\begin{cases} y_i - \langle w, x_i \rangle + b \leq \epsilon \\ \langle w, x_i \rangle - b - y_i \leq \epsilon \end{cases}, i = 1, 2, ..., m \tag{3}$$

On the other hand, flattest function means smallest slope, i.e. w, which leads to condition:

$$\text{minimize } \|w\|^2 \tag{4}$$

Summing up, the optimization problem that is reached in the case of linear ϵ-SVR is stated as:

$$\begin{cases} \text{find } w \text{ and } b \text{ as to minimize } \|w\|^2 \\ \text{subject to } \begin{cases} y_i - \langle w, x_i \rangle + b \leq \epsilon \\ \langle w, x_i \rangle - b - y_i \leq \epsilon. \end{cases} \end{cases}, i = 1, 2, ..., m \tag{5}$$

2.2. Linear Support Vector Machines for Regression with Indicators for Errors

It may happen that the linear function f is not able to fit all training data and consequently ϵ-SVR will also allow for some errors, analogously to the corresponding situation in SVMs for classification [3], [6].

Therefore, the positive slack variables ξ_i and ξ_i^*, both attached to each sample, are introduced into the condition for approximation of training data:

$$\begin{cases} y_i - \langle w, x_i \rangle + b \leq \epsilon + \xi_i, \\ \langle w, x_i \rangle - b - y_i \leq \epsilon + \xi_i^*. \end{cases}, i = 1, 2, ..., m \tag{6}$$

Simultaneously, the sum of these indicators for errors is minimized:

$$C \sum_{i=1}^{m} (\xi_i + \xi_i^*), \tag{7}$$

where C is a parameter which denotes the penalty for errors.

Adding up, the optimization problem in the case of linear ϵ-SVR with indicators for errors is written as:

$$\begin{cases} \text{find } w \text{ and } b \text{ as to minimize } \|w\|^2 + C \sum_{i=1}^{m} (\xi_i + \xi_i^*) \\ \text{subject to } \begin{cases} y_i - \langle w, x_i \rangle + b \leq \epsilon + \xi_i \\ \langle w, x_i \rangle - b - y_i \leq \epsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0 \end{cases} \end{cases}, i = 1, 2, ..., m \tag{8}$$

2.3. Nonlinear Support Vector Machines for Regression

If a linear function is not at all able to fit training data, a nonlinear function has to be chosen. The procedure follows the same steps as in SVMs for classification [4]. Data is mapped via a nonlinear function into a high enough dimensional space and linearly modelled there as in the previous subsection. This corresponds to a nonlinear regression hyperplane in the initial space.

Hence, data samples are mapped into some Euclidean space, *H*, through a mapping $\Phi : R^n \mapsto H$. Therefore, the equation of the regression hyperplane in H is stated as:

$$\langle \Phi(w), \Phi(x_i) \rangle - b = 0 \tag{9}$$

where $\Phi(w)$ is the slope of the hyperplane.

Also, the squared norm:

$$\|w\|^2 = \langle w, w \rangle \tag{10}$$

changes to:

$$\langle \Phi(w), \Phi(w) \rangle. \tag{11}$$

The appointment of a function Φ with the required properties is nevertheless not a straightforward task to perform. However, as in the training algorithm vectors appear only as part of dot products, if there were a kernel function K such that:

$$K(x, y) = \langle \Phi(x), \Phi(y) \rangle \tag{12}$$

where $x, y \in R^n$, one would use K in the training algorithm and would never need to explicitly even know what Φ is.

The kernel functions that meet (12) are given by Mercer's theorem from functional analysis [2]. Still, it may not be easy to check whether the condition is satisfied for every new kernel. There are, however, a couple of classical kernels that had been demonstrated to meet Mercer's condition [2]:

- Polynomial classifier of degree p: $K(x, y) = \langle x, y \rangle^p$
- Radial basis function classifier of parameter σ: $K(x, y) = e^{\frac{\|x-y\|^2}{\sigma}}$

To conclude, the linear regression in H (which corresponds to the nonlinear regression in the initial space) leads to the optimization problem in:

$$\begin{cases} \text{find } w \text{ and } b \text{ as to minimize } K(w, w) + C \sum_{i=1}^{m} (\xi_i + \xi_i^*) \\ \text{subject to } \begin{cases} y_i - K(w, x_i) + b \leq \epsilon + \xi_i \\ K(w, x_i) - b - y_i \leq \epsilon + \xi_i^* \quad , i = 1, 2, ..., m \\ \xi_i, \xi_i^* \geq 0 \end{cases} \end{cases} \tag{13}$$

3. Evolutionary Support Vector Machines for Regression

Canonical ϵ-SVR solves either optimization problem that we earlier arrived at through the generalized form of the method of Lagrange multipliers that is typical to any SVM. As previously in ESVMs for classification [13], where the solving of the optimization problem within SVMs was conducted by means of a canonical EA [5], ϵ-evolutionary support vector regression (ϵ-ESVR) makes use of an EA as well, this time with the aim of finding the optimal estimated regression coefficients [10].

Again, contrarily to ϵ-SVR, where the mathematics of the method is both complicated and not always able to state the values of w and b in a straight way (and various mechanisms to appoint the target of test data samples are used instead), in ϵ-ESVR the coefficients are determined in a simple and direct fashion.

3.1. The Evolutionary Algorithm

Training follows the same steps as in canonical ϵ-SVR. For the sake of generality, the employed EA solves the last optimization problem that was reached, because previously defined cases (5) and (8) are particular situations of equation (13). The components of the EA to solve the inherent optimization problem were experimentally chosen as in the following subsections.

3.1.1 Representation of Individuals

An individual c encodes the regression coefficients together with the indicators for errors of regression (included for reasons of reference in the EA formulation of the optimization problem), i.e. w, b, ξ and ξ^*:

$$c = (w_1, ..., w_n, b, \xi_1, ..., \xi_m, \xi_1^*, ..., \xi_m^*) \tag{14}$$

The best individual from all generations will give the optimal estimated values for w and b.

3.1.2 Initial Population

Individuals are randomly generated following a uniform distribution, such that $w_i \in [-1, 1], i = 1, 2, ..., n$, $b \in [-1, 1]$ and ξ_j and $\xi_j^* \in [0, 1], j = 1, 2, ..., m$.

3.1.3 Fitness Evaluation

The expression of the fitness function is considered as follows:

$$\begin{aligned} f(w_1, ..., w_n, b, \xi_1, ..., \xi_m, \xi_1^*, ..., \xi_m^*) = K(w, w) + \\ C \sum_{i=1}^{m} (\xi_i + \xi_i^*) + \sum_{i=1}^{m} [t(\epsilon + \xi_i - y_i + K(w, x_i) - b)]^2 + \\ \sum_{i=1}^{m} [t(\epsilon + \xi_i^* + y_i - K(w, x_i) + b)]^2, \end{aligned} \tag{15}$$

where

$$t(a) = \begin{cases} a, & a < 0, \\ 0, & \text{otherwise.} \end{cases} \quad (16)$$

The fitness function embodies the objective function of equation (13) while the three constraints within are handled by penalizing infeasible individuals; this is done by introducing the penalty function (16) in the fitness evaluation (15). Finally one is led to:

$$\text{minimize } (f(c), c). \quad (17)$$

3.1.4 Genetic Operators

Tournament selection is used. Intermediate crossover and mutation with normal perturbation are considered. Mutation is restricted only for ξ and ξ^*, preventing the indicators for errors from taking negative values.

3.2. A Reconsideration of the Evolutionary Algorithm

Although the current proposed approach is very competitive as compared to the canonical technique (as to be seen in the experimental results section), it may still be improved concerning simplicity and efficiency. The current optimization problem requires to treat the error values, which in the a priori proposed EA variant are included in the representation. These can be expected to severely complicate the problem by increasing the genome length (variable count) by the number of training samples. We propose to tackle this issue by a radical reconsideration of the elements of the EA as follows.

Since ESVMs directly and interactively provide regression hyperplane coefficients at all times, we propose to drop the indicators for errors from the EA representation and, instead, calculate their values in a simple and natural fashion.

3.2.1 Representation of Individuals

Consequently, this time, an individual c encodes solely the regression coefficients, i.e. w, b, as in (18):

$$c = (w_1, ..., w_n, b). \quad (18)$$

3.2.2 Initial Population

Individuals are again randomly generated following a uniform distribution, such that $w_i \in [-1, 1], i = 1, 2, ..., n$ and $b \in [-1, 1]$.

3.2.3 Fitness Evaluation

Currently all indicators for errors will have to be computed in order to be referred in the expression of the fitness function. The method we propose for acquiring the errors is subsequently described.

For every training sample, one firstly calculates the difference between the actual target and the predicted value that is obtained following the coefficients of the current individual (regression hyperplane), as in (19):

$$\text{difference}_i = |K(w, x_i) - b - y_i|, i = 1, 2, ..., m \quad (19)$$

Subsequently, one tests the difference against the ϵ threshold, following (20):

$$\begin{cases} \text{if difference}_i < \epsilon & \text{then } \xi_i = 0, \\ \text{else} & \xi_i = \text{difference}_i - \epsilon. \end{cases} \quad i = 1, 2, ..., m \quad (20)$$

The newly obtained indicators for errors can now be employed in the fitness evaluation of the corresponding individual, which changes from (15) to (21):

$$f(w_1, ..., w_n, b) = K(w, w) + C \sum_{i=1}^{m} \xi_i \quad (21)$$

The function to be fitted to the data is thus still required to be as flat as possible and to minimize the errors of regression that are higher than the permitted ϵ.

All the other evolutionary elements remain the same.

3.3. Test Accuracy of Evolutionary Support Vector Machines for Regression

Either algorithm stops after a predefined number of generations and, in the end, one obtains the optimal estimated regression coefficients, i.e. w and b, which are subsequently applied to the test data.

Given a test data sample x, its predicted target is computed following:

$$f(x) = K(w, x) - b \quad (22)$$

Suppose the test set $\{(x_i, y_i)\}_{i=1,2,...,p}$ is given, where y_i is the actual target and $y_i^{(pred)}$ the prediction, where $y_i^{(pred)} = f(x_i)$. In order to verify the accuracy of the technique, the value of the root mean square error (RMSE) is computed as in:

$$RMSE = \sqrt{\frac{1}{p} \sum_{i=1}^{p} (y_i^{(pred)} - y_i)^2} \quad (23)$$

4. Application to the Boston Housing Regression Task

The proposed technique is validated against the Boston housing data from the UCI repository of machine learning databases.

This regression task deals with the prediction of the median price of housing in the Boston area based on socio-economic and environmental factors, such as crime rate, nitric oxide concentration, distance to employment centers and age of a property.

There are 506 samples, thirteen continuous attributes (including the target attribute) and one binary-valued attribute. There are no missing values.

4.1 Experimental Setup

The Boston housing data set was split into 380 cases for training and remaining 126 for test. The cases for the two sets were each time chosen in a random fashion. Normalization was not performed to the Boston housing data, as preliminary tests proved it to be unnecessary in order to reach optimal performance.

4.2 Parameter Setting

A linear kernel was experimentally chosen. Both parameters of the SVM and of either of the EAs were manually picked and are depicted in Table 1.

Table 1. Values for parameters of ϵ-ESVR for the Boston housing regression problem (left for evolution / right for computation of indicators for errors)

Parameter	Manually picked value (err. ev./comp.)
C	1/1
ϵ	0/5
Population size	200/200
Number of generations	2000/2000
Crossover prob.	0.5/0.5
Mutation prob.	0.5/0.5
Mutation prob. for ξ	0.5/-
Mutation strength	0.1/0.1
Mutation strength for ξ	0.1/-

4.3 Experimental Results

The mean square error in 10 runs was computed and the obtained values are depicted in Table 2. The results of the reconsidered EA are comparable to those of the initial algorithm, but slightly worse. This is rather surprising as it is the outcome of a technique that was introduced to reduce the problem complexity and thus bolster EA performance. Obviously, accurate estimation of the error terms is a requirement of major concern for the overall ESVM success on regression tasks. However, we expect that the results of the reconsidered EA can be improved by a fine tuning of the evolutionary parameters or a different approach to error computation.

Table 2. Root mean square error of ϵ-ESVR after 10 runs with a random split of training/test

Descriptor	Evolution of errors	Computation of errors
Average RMSE	4.64	5.03
Worst RMSE	5.88	5.76
Best RMSE	4.06	4.51
St.D.	0.52	0.43

4.4 Comparison to Other Approaches

Comparison to the results obtained by canonical ϵ-SVR and a linear regression model was next performed.

For reasons of achieving an objective comparison between the three techniques, the two other methods were also personally implemented with the same configurations. In this respect, the R language and environment was employed.

For the canonical ϵ-SVR implementation the R *e1071* package for SVMs was used. The kernel type was set to linear and 0 was experimentally appointed as the value for ϵ. The penalty for errors C was by default equal to 1 in specified package (same as in the ϵ-ESVR case).

The Boston housing data was taken from the R *mlbench* package. The canonical ϵ-SVR was run 10 times and each time 380 random cases constituted the training set while the remaining 126 made the test set, respectively. Consequently, the experimental setup and the corresponding parameter setting are identical to that of ϵ-ESVR.

Additionally, a linear regression model was implemented with the same training/test set sizes and random manner of appointment. The obtained results are given in Table 3.

Table 3. Obtained root mean square error of ϵ-ESVR versus canonical ϵ-SVR and a linear regression model after 10 runs with a random split of training/test

	ESVR (err ev./comp.)	SVR	Linear model
Average RMSE	**4.64/5.03**	5.3	4.76
Worst RMSE	5.88/5.76	6.53	5.49
Best RMSE	4.06/4.51	3.76	4.26
St.D.	0.52/0.43	0.93	0.33

Comparison to canonical ϵ-SVR and the linear regression model shows ϵ-ESVR as a competitive alternative.

However, automatic tuning of parameters may yield even better results.

5. Conclusions and Future Work

Following the course of application of their parent paradigm, the present paper extends the novel technique of ESVMs to the regression case. In this respect, we achieved the hybridization between the ϵ-SVR engine and EAs. Validation of ϵ-ESVR is conducted through the test against a real-world regression case, i.e. the benchmark problem of Boston housing. The obtained prediction error is compared to corresponding results of the canonical counterpart and a linear regression model. The expanded version of ESVMs once again proves to be competitive to SVMs while, in addition, the former have a simpler nature and a direct handling of the learning function.

A reconsideration of the underlying EA is performed through the removal of indicators for errors from an individual's representation and their converse computation by a straightforward method. In this way, we significantly reduce the high dimensionality of the genome. The second approach performs in a similar fashion to the initial one while is yet more elegant and natural.

However, there is still some potential for improving the EA. We could alternatively use a shrinking procedure, e.g. adapt the chunking method in SVMs [7] to fit the hybridized technique. This would certainly boost runtime which is currently very high due to data dimensionality.

Moreover, we could adopt other selection, crossover and mutation schemes as it is not clear how well-adapted to the problem the used EA is.

In addition, the way of treating the two criteria (i.e. reduce errors and obtain a flat function) through proposed fitness evaluation may not be the best choice. In this respect, we could alternatively try a multicriterial approach in future work.

Finally, we could perhaps imagine an enhanced manner to approach the problem of the computation of indicators for errors.

References

[1] D. G. Altman. *Practical Statistics for Medical Research*. Chapman and Hall, 1991.

[2] B. E. Boser, I. M. Guyon, and V. Vapnik. A training algorithm for optimal margin classifiers. *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*, pages 11–152, 1992.

[3] C. Cortes and V. Vapnik. Support vector networks. *Machine Learning*, pages 273–297, 1995.

[4] T. M. Cover. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE Transactions on Electronic Computers, vol. EC-14*, pages 326–334, 1965.

[5] A. E. Eiben and J. E. Smith. *Introduction to Evolutionary Computing*. Springer-Verlag, 2003.

[6] S. Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice Hall, 1999.

[7] F. Perez-Cruz, A. R. Figueiras-Vidal, and A. Artes-Rodriguez. Double chunking for solving svms for very large datasets. *Proceedings of Learning 2004, Elche, Spain*, 2004. eprints.pascal-network.org/archive/00001184/01/learn04.pdf.

[8] R. Rosipal. *Kernel-based Regression and Objective Nonlinear Measures to Access Brain Functioning*. PhD thesis, Applied Computational Intelligence Research Unit School of Information and Communications Technology University of Paisley, Scotland, September 2001.

[9] A. J. Smola and B. Scholkopf. A tutorial on support vector regression. Technical Report NC2-TR-1998-030, NeuroCOLT2 Technical Report Series, October 1998.

[10] R. Stoean, M. Preuss, D. Dumitrescu, and C. Stoean. ϵ - evolutionary support vector regression. *Symbolic and Numeric Algorithms for Scientific Computing, SYNASC 2006*, pages 21–27, 2006.

[11] R. Stoean, M. Preuss, C. Stoean, and D. Dumitrescu. Evolutionary support vector machines and their application for classification. Technical Report CI-212/06, Collaborative Research Center on Computational Intelligence, University of Dortmund, June 2006.

[12] R. Stoean, C. Stoean, M. Preuss, and D. Dumitrescu. Evolutionary multi-class support vector machines for classification. *Proceedings of International Conference on Computers and Communications - ICCC 2006, Baile Felix Spa - Oradea, Romania*, pages 423–428, 2006.

[13] R. Stoean, C. Stoean, M. Preuss, E. El-Darzi, and D. Dumitrescu. Evolutionary support vector machines for diabetes mellitus diagnosis. *Proceedings of IEEE Intelligent Systems 2006, London, UK*, pages 182–187, 2006.

[14] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer, 1995.

[15] V. Vapnik. *Statistical Learning Theory*. Wiley, 1998.