SVM Regression Parameters Optimization Using Parallel Global Search Algorithm

Konstantin Barkalov, Alexey Polovinkin, Iosif Meyerov, Sergey Sidorov, and Nikolai Zolotykh

N.I. Lobachevsky State University of Nizhni Novgorod, Nizhni Novgorod, Russia
barkalov@fup.unn.ru, alexey.polovinkin@itlab.unn.ru, {meerov,zolotykh}@vmk.unn.ru, sidorov.sergey@gmail.com

Abstract. The problem of optimal parameters selection for the regression construction method using Support Vector Machine is stated. Cross validation error function is taken as the criterion. Arising bound constrained nonlinear optimization problem is solved using parallel global search algorithm by R. Strongin with a number of modifications. Efficiency of the proposed approach is demonstrated using model problems. A possibility of the algorithm usage on large-scale cluster systems is evaluated. Linear speed-up of combined parallel global search algorithm is demonstrated.

Keywords: Machine learning, global optimization, support vector machine, global search algorithm.

1 Introduction

The problem of reconstructing a functional dependency of a given value on measurement results (regression reconstruction problem) is often met in applied studies. In practice the nature of the given dependence is usually unknown, therefore the unknown function is chosen from a certain pre-determined class which depends on the selected model. Parameters of this function are calculated based on the available experimental data. Among such models [1] we could note polynomial regression, multivariate adaptive regression splines (MARS), radial basis function, decision trees and their ensembles, various boosting algorithms, neural network, etc. An experimental comparison of a number of models and algorithms is given, for instance, in [8].

One of the widely used solution methods for the regression reconstruction problem is the Support Vector Machine (SVM) algorithm [14]. The possibility of efficient nonlinear dependencies modeling and independence of generalization capability from the feature space dimension could be marked out among this algorithm's advantages. Though, in some cases practical application of the algorithm is limited due to the fact that accuracy of the method strongly depends on its parameters selection [3]. Most frequently used approaches to optimal parameter selection are eventually reduced to global optimization problem solution. Thus, for instance, [3] offers to use a genetic algorithm and particle swarm optimization, [9] uses chaos optimization algorithm [10], [11] describes application of a modification of the Efficient Global Optimization (EGO) algorithm [12], [13] uses Pattern Search approach, etc.

As is known, complexity of a global optimization problem solution is significantly influenced by its dimension. For example, for the class of multi-extremal functions satisfying Lipschitz condition the so called curse of dimensionality takes place which represents exponential growth of computing time as a function of dimensionality. Thus, if p calculations of function value are required to reach ε accuracy of solution in a one-dimensional problem, αp^N trials are required to reach the same accuracy in an N-dimensional problem, where α depends on the objective function and on the optimization algorithm in use. Virtually the only way of solving problems of the mentioned class in a reasonable time is the development of parallel algorithms and their implementation on high performance computing systems.

This paper introduces a novel method of SVM regression parameters selection based on optimization of the cross validation error function using parallel global search algorithm. This algorithm is based on the informational statistical approach [5] and as experimentally proven in [5], [7] outperforms many known methods of similar purpose. The paper is organized as follows: an SVM regression parameters optimization problem is stated in the second section; section three describes the basic parallel global search algorithm [5] and its modifications which increase parallel computing efficiency; in section four successfulness of the described approach is demonstrated based on model problems, the possibility of the approach usage on large-scale cluster systems is discussed.

2 Optimization of SVM Regression Algorithm Parameters

This paper considers the regression reconstruction problem in the following statement. Let a training set $D = \{(x_i, y_i), i = 1,\ldots,N\}$ be given, where $x_i \in R^d$ is the feature vector, $y_i \in R$ – the response. It is required to find a function $f(x)$ from some specific class K which minimizes the value of empirical risk (prediction error on the training set). For the SVM-regression construction algorithm the function $f(x)$ can be written in general form as

$$f(x) = w^T \phi(x) + b,$$

where $\phi(x)$ is a nonlinear (in general case) mapping $R^d \to R^m$, $w \in R^m$ – a vector of linear function coefficients in the new feature space R^m. As a loss function a piecewise linear function of ε-sensitivity is used

$$L_\varepsilon(y, f(x)) = \max(0, |y - f(x)| - \varepsilon),$$

where ε – a predetermined threshold (if the predicted value differs from the actual value less than given threshold the error is considered equal zero). The function of empirical risk is written as:

$$R_{emp}(w) = \frac{1}{N}\sum_{i=1}^{N} L_\varepsilon(y_i, f(x_i)).$$

Considered problem of empirical risk minimization is reduced to a quadratic optimization problem

$$\min_{w} \frac{\|w\|^2}{2} + C\sum_{i=1}^{N}\left(\xi_i + \xi_i^*\right)$$
$$f(x_i) - y_i \le \varepsilon + \xi_i,\ y_i - f(x_i) \le \varepsilon + \xi_i^*,\ \xi_i, \xi_i^* \ge 0 \qquad (1)$$

where C is a regularization parameter which represents the trade-off between the model complexity and the empirical error in the minimized function. Using the Lagrange multiplier method the problem (1) can be reduced to a dual form:

$$\max \sum_{i=1}^{N} y_i\left(\alpha_i^* - \alpha_i\right) - \varepsilon\sum_{i=1}^{N}\left(\alpha_i^* + \alpha_i\right) - \frac{1}{2}\sum_{i,j=1}^{N}\left(\alpha_i^* - \alpha_i\right)\left(\alpha_j^* - \alpha_j\right)K(x_i, x_j)$$
$$\sum_{i=1}^{N}\left(\alpha_i^* - \alpha_i\right) = 0,\ 0 \le \alpha_i, \alpha_i^* \le C \qquad , \qquad (2)$$

where α_i, α_i^* – Lagrange multipliers, $K(x_i, x_j)$ – kernel function representing inner product in the new feature space R^m. Some of the most often used kernels are radial basis functions:

$$K\left(x_i, x_j\right) = \exp\left(-\left\|x_i - x_j\right\|^2 / 2\sigma^2\right).$$

As shown in [3] generalization capability of SVM-regression algorithm significantly depends on the choice of parameters C, ε and σ. In paper [2] a method based on cross-validation error minimization is offered. The idea of the method is to split the training set randomly into S subsets $\{G_s, s = 1,...,S\}$, train the model on $(S-1)$ subsets and use the remaining subset to calculate the test error. The error averaged over all training subsets is used as an estimate of algorithm's generalization capability

$$MSE_{CV} = \frac{1}{N}\sum_{s=1}^{S}\sum_{i \in G_s}(y_i - f(x_i \mid \theta_s))^2,$$

where θ_s is the solution of the problem (2) achieved using the set $D \backslash G_s$ as a training set. In case the number of objects in the training set is not too large LOO (leave-one-out) error can be used

$$MSE_{LOO} = \frac{1}{N}\sum_{i=1}^{N}(y_i - f(x_i \mid \theta_i))^2,$$

where θ_i is the solution of the problem (2) achieved using the set $D \backslash \{(x_i, y_i)\}$ as a training set. Due to the fact that a solution of a quadratic programming problem (2) for each parameter set (C, ε, σ) exists and is unique we can consider the leave-one-out error for a given training set as a function of C, ε and σ:

$$MSE_{LOO} = \frac{1}{N}\sum_{i=1}^{N}(y_i - f(x_i \mid \theta_i(C, \varepsilon, \sigma)))^2 = F(C, \varepsilon, \sigma),$$

Thus the problem of optimal parameter selection for the SVM-regression construction algorithm has been reduced to a problem of function $F(C,\varepsilon,\sigma)$ minimization. In a general case the function is multi-extremal so global optimization algorithms should be applied to find its optimum.

Let us note that the usage of cross-validation method for finding optimal parameter values is traditional in machine learning [15]. Nevertheless, with a large number of determined parameters it can lead to overfitting [16]. In our case only 3 parameters are used which allows us to hope that overfitting won't happen.

3 Parallel Global Search Algorithm

We will find the optimal value of the parameters C, ε and σ in the hypercube $D=[C_{\min};C_{\max}]\times[\varepsilon_{\min};\varepsilon_{\max}]\times[\sigma_{\min};\sigma_{\max}]$. Let us define $\varphi(y)=F(C,\varepsilon,\sigma)$, where $y=(C,\varepsilon,\sigma)$.

Without loss of generality we can consider an unconstrained global optimization problem having the form

$$\begin{gathered}\varphi^*=\varphi(y^*)=\min\{\varphi(y):y\in D\}\\ D=\{y\in R^N:-2^{-1}\le y_i\le 2^{-1},1\le i\le N\}'\end{gathered}\tag{3}$$

where the objective function $\varphi(y)$ satisfies the Lipschitz condition

$$|\varphi(y_1)-\varphi(y_2)|\le K\|y_1-y_2\|,\ \ y_1,y_2\in D$$

with constant K which in a general case in unknown. This statement covers a wide class of problems as any hyper interval S

$$S=\{y\in R^N:a_i\le y_i\le b_i,1\le i\le N\}$$

can be reduced to a hyper cube D using a linear coordinate transformation.

In the discussed approach [5] solution of multidimensional problems is reduced to solution of equivalent one-dimensional problems (*dimension reduction*). Thus, the usage of a continuous single-valued mapping such as Peano curve

$$\{y\in R^N:-2^{-1}\le y_i\le 2^{-1},1\le i\le N\}=\{y(x):0\le x\le 1\}$$

allows to reduce the minimization problem in domain D to a minimization problem on the segment [0,1].

$$\varphi^*=\varphi(y^*)=\varphi(y(x^*))=\min\{\varphi(y(x)):x\in[0,1]\}.$$

Numerical methods which allow efficient construction of such mappings with any given accuracy are considered in details in [5]. According to those methods this dimension reduction scheme maps a multidimensional problem with Lipschitz minimized function to one-dimensional problem in which the function satisfies the uniform Hölder condition

$$|\varphi(y(x_1))-\varphi(y(x_2))|\le G|x_1-x_2|^{1/N},\ \ x_1,x_2\in[0,1]$$

where N is the dimension of the original multidimensional problem and Holder coefficient G is connected with Lipschitz constant K of the original problem by the relation $G\le 4K\sqrt{N}$.

To solve the arising one-dimensional problem it is suggested to use an efficient information-statistical global search algorithm [5]. But when solving the reduced problem with a single scanning part of proximity information for the points in multidimensional space is lost. It is explained by the fact that the point $x\in[0,1]$ has only left and right neighbors while the corresponding point $y(x)\in R^N$ has neighbors along $2N$ directions. Part of points' proximity information can be preserved by using a set of mappings

$$Y_L(x)=\{y^1(x),...,y^L(x)\} \tag{4}$$

Instead of using a single Peano curve $y(x)$. Each Peano curve $y^i(x)$ from $Y_L(x)$ can be achieved as a result of some transformation of the original curve (shift along the main diagonal of the hypercube [4] or rotation about the origin of the coordinates [7]). The set of Peano curves constructed in this fashion allows to achieve close inverse images x', x'' for any close images y', y'' differing along only one coordinate for some mapping $y^i(x)$.

Usage of the mappings set (4) leads to forming a corresponding set of one-dimensional problems

$$\min\{\varphi(y^l(x)):x\in[0,1]\},\ \ 1\le l\le L. \tag{5}$$

Each problem from this set can be solved independently on a separate processor using the scanning $y^s, 1\le s\le L$. The result of the minimized function $\varphi(y(x))$ value calculation at point x^k received by a particular processor for its solved problem is interpreted as the results of calculations for all remaining problems (in corresponding points $x^{k1},...,x^{kL}$) and is sent to other processors. Within such approach a *trial* at point $x^k\in[0,1]$ conducted in problem l consists of the following steps:

1. Determine the image $y^k=y^l(x^k)$ with scanning $y^l(x)$;
2. Inform the rest of the processors of the trial start at point y^k (*blocking* of point y^k);
3. Calculate the value $\varphi(y^k)$. The pair $\{y^l(x^k), z^k=\varphi(y^l(x^k))\}$ is *the result of the trial* at point x^k;

4. Determine inverse images $x^{kl} \in [0,1], 1 \le l \le L$ of the point y^k.
5. Interpret the trial at point $y^k \in D$ as trials at L points $x^{k1},...,x^{kL}$ with same results $\varphi(y^1(x^{k1}) = ... = \varphi(y^L(x^{kL})) = z^k$, i.e. inform the rest of the processors of the trial results at point y^k having sent them pairs (y^k, z^k).

Such informational unity allows to solve the original problem (3) by parallel solving of L problems of form (5) on a set of segments [0,1]. Each processor has an own copy of the software which implements calculation of problem's functions and the decision rule of the algorithm. To organize the communication a queue is created on each processor where processors store information about the performed iterations as pairs: the point of an iteration and minimized function value at that point. Computing scheme of the algorithm is given below.

Algorithm. Starting iteration is performed at an arbitrary point $x^1 \in (0,1)$ (starting points are different for all processors). The choice of the point x^{q+1}, $q \ge 1$, for any subsequent trial on the processor l is defined by the following rules.

Rule 1. For the queue assigned to a given processor remove stored for the processor results including the set Y_q of iteration points in domain D and function values computed for those points. Determine the set X_q of inverse images for the points of set Y_q with scanning $y^l(x)$. Capacity of sets Y_q and X_q is the value s_q such that $s_q \ge q$ as these sets contain the points computed on this processor and received from other processors.

Rule 2. Enumerate the points of iterations set $\{x^1\} \cup X_q$ using subscripts in order of increasing coordinate values

$$0 = x_0 < x_1 < ... < x_i < ...x_k < x_{k+1} = 1, \tag{6}$$

where $k = s_q + 1$, and match them with values $z_i = \varphi(y^l(x_i))$ calculated in these points and integer values $\nu(x_i)$ – the *index* of a point. The index of a non-blocked point x_i (i.e. the point for which results of the trial have been already received) is taken to be equal to 1 while the index of a blocked point x_i (i.e. the point for which the trial has been started by another processor) is taken to be equal to 0, the value z_i is undefined in this case. The points x_0, x_{k+1} are additionally introduced (they do not participate in the trial), indices of these points are taken to be equal to 0 and the values z_0, z_{k+1} are taken as undefined.

Rule 3. Calculate the current lower bound

$$\mu = \max\left\{ \frac{| z_i - z_{i-1} |}{(x_i - x_{i-1})^{1/N}} : 2 \le i \le k, \nu(x_i) = \nu(x_{i-1}) = 1 \right\} \tag{7}$$

for relative differences of function φ. If the value μ turns out undefined (due to unsatisfiability of indices equality conditions from (7)), or if $\mu = 0$ take $\mu = 1$.

Rule 4. For each interval (x_{i-1}, x_i), $1 \le i \le k+1$ calculate the characteristic $R(i)$, where

$$R(i) = \Delta_i + \frac{(z_i - z_{i-1})^2}{r^2 \mu^2 \Delta_i} - 2\frac{(z_i + z_{i-1})}{r\,\mu},\ \nu(x_{i-1}) = \nu(x_i) = 1,$$

$$R(i) = 2\Delta_i - 4\frac{z_i}{r\,\mu},\quad \nu(x_{i-1}) < \nu(x_i),$$

$$R(i) = 2\Delta_i - 4\frac{z_{i-1}}{r\,\mu},\quad \nu(x_{i-1}) > \nu(x_i),$$

$$\Delta_i = \left(x_i - x_{i-1}\right)^{1/N}$$

The value $r > 1$ is a parameter of the algorithm.

Rule 5. Determine the interval (x_{t-1}, x_t) which has the maximum characteristic

$$R(t) = \max\{R(i) : 1 \le i \le k+1\} \tag{8}$$

Rule 6. Conduct the next trial at the middle point of the interval (x_{t-1}, x_t) if indices of its end points are not equal, i.e.

$$x^{q+1} = \frac{x_t + x_{t-1}}{2},\ \nu(x_{t-1}) \ne \nu(x_t).$$

Otherwise conduct a trial at point

$$x^{q+1} = \frac{x_t + x_{t-1}}{2} - \mathrm{sign}(z_t - z_{t-1})\frac{1}{2r}\left[\frac{|z_t - z_{t-1}|}{\mu}\right]^N,$$

$$\nu(x_{t-1}) = \nu(x_t).$$

Store the results of the trial in the queue assigned to the given processor. Increment q by 1 and move to the next iteration.

Described rules can be augmented with a stopping condition which stops trials if $\Delta_t \le \varepsilon$ where t is from (8) and $\varepsilon > 0$ has the order of the desired per-coordinate accuracy in problem (3)

The following convergence condition is satisfied for this algorithm (as a special case of more general theorem about convergence of a parallel global search algorithm using a set of scannings from [5]).

Convergence Conditions. Let y^* be the point of absolute minimum of a Lipschitz with a constant *K* function $\varphi(y)$, $y \in D$, and $\{y^k\}$ is a sequence of trials generated by a parallel global search algorithm during this function minimization. Then

1. If y', y'' are two limit points of the sequence $\{y^k\}$, then $\varphi(y') = \varphi(y'')$.
2. If y' is a limit point of the sequence $\{y^k\}$, then $\varphi(y') \le \varphi(y^k)$.
3. If at some $k > 0$ the following condition is satisfied for the value μ from (7)

$$r\mu > 8K\sqrt{N},$$

where $r > 1$ is a parameter of the algorithm, then y^* is the limit point of the sequence $\{y^k\}$, and $\lim_{k\to\infty} y^k = y^*$ if y^* is the only point of absolute minimum.

More general variants of parallel global search algorithms (for solution of conditional and multicriteria problems) and corresponding convergence theory are presented in [5-7].

Perspectives of Usage in Parallel Computing. The analysis shows that the described basic algorithm has limitations on the number of used computing devices while using shift scannings [4] (shift along the main diagonal of the hyper cube) as well as while using rotated scannings [7] (rotation about the origin of the coordinates).

In the first case the set of scannings comes out as a result of a shift along the main diagonal of the hyper cube *D* and the step of this shift decreases by 2 times at the construction of each next mapping. The scanning itself is constructed with the accuracy $\delta=2^{-m}$ along the coordinate; *m* here is a parameter of the scanning construction. Construction of a very accurate scanning will require significant resources and the value δ will eventually be limited by the machine error. To solve the problem with accuracy from 10^{-3} to 10^{-4} it is sufficient to choose *m* in the range from 10 to 15 (if necessary, further refinement of the found solution can be performed using one of the known local methods).

As follows from the above the number of scanning shifts during construction of mappings set (4) will be limited by the accuracy of the original scanning construction. If the shift is made by the value smaller than $\delta=2^{-m}$ the next scanning will match the previous one. This way the limitation $L \le m$ is put on the number of used scannings *L* and, as a consequence, on the number of problems solved in parallel.

In the second case the number of processors is limited by the number of possible rotations of an scanning about the origin of the coordinates. In total there can be up to 2^N of such rotations (where *N* is the dimension of the solved problem). But to apply them all would be redundant, rotations in each of coordinate planes are sufficient and these make up $N(N-1)+1$ mappings. This relation gives potential for parallelism at large *N* and limits parallelization possibilities of in problems of smaller dimensions.

A promising in terms of parallelization computing scheme can be based on the approach described in [17]. The essence of the approach consists in conducting simultaneous p trials on intervals which have p greatest characteristics rather than conducting a single trial on interval having the maximum characteristic (8). This approach can be applied together with the described above mappings method: conduct p trials in parallel for each problems from the set (5) solution of which is also performed in parallel [18].

Mentioned combined algorithm can be used on modern cluster systems with multi-core processors. An own sub-problem (5) is assigned to each node and the number of simultaneous trials for one problem corresponds to the number of cores on the node.

4 Computational Experiment

4.1 Optimization of a Function of Two Variables: Comparison with the Exhaustive Search

In this section we will demonstrate the efficiency of the basic algorithm for SMV-regression parameters optimization on a two-dimensional problem example. We will also compare the algorithm with the exhaustive search. Consider the following problem [2]. Let the function $y(x) = 0.5 + 0.4\sin(2\pi x)\cos(6\pi x)$ be given. $x_i = 0.05 \cdot (i-1), i = 1,...,21$, $y_i = y(x_i) + N(0,0.05)$, where $N(0,0.05)$ – a Gaussian distribution with the mean equal to 0 and standard deviation equal to 0.05. Figure 1 presents the plot of the cross-validation error MSE_{LOO} dependency on ε and σ with fixed value of parameter C = 1. As seen from the plot the function describing this dependency has several local minima in the search domain.

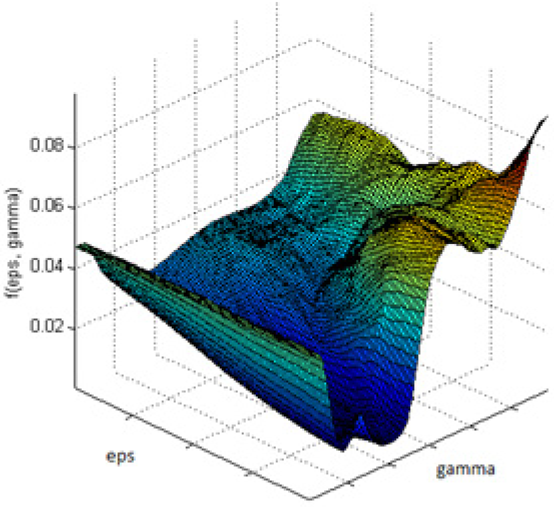

Fig. 1. Dependency of cross-validation error on SVM-regression parameters

The algorithm was run with the following parameters: the accuracy of the search $\rho = 0.01$, reliability parameter $r = 2.5$, the accuracy of the scanning construction along each coordinate was $2.5 \cdot 10^{-4}$, the number of scannings $L = 2$. The total number of iterations performed by the parallel version of the algorithm before reaching stopping criteria was 1550 (to reach the same accuracy with the exhaustive search 10000 iterations would be required), found optimum value was 0.006728. The experiment showed significant advantage of the algorithm over the exhaustive search. With increasing dimension of the problem the gap in the number of performed iterations and thus the gap in solution time will only be growing.

4.2 Optimization of a Function of Two Variables: Efficiency of the Algorithm and Its Modifications When Using Parallel Computing

In this section we consider a possibility of using different modifications of parallel global search algorithm when optimizing parameters of SVN-regression. We will use the following described above implementations of the algorithm:

1. Basic parallel global search algorithm with shift scannings.
2. Parallel global search algorithm with rotated scannings.
3. Combined parallel global search algorithm.

Taking into account considerations on usage perspectives of implementations 1-3 in parallel computing let us ascertain the applicability of the most promising implementation 3, also let us compare the problem solution times using same number of scannings *L* in all implementations. In addition accounting for the fact that the number of scannings in the second implementation is limited by the value $N(N-1)+1$ where N is the problem dimension we will take $L = 3$. Having this setting implementations 1 and 2 will allow to use 3 cores each and implementation 3 will allow to use $3p$ cores (3 rotated scannings and p cores for each of the scannings).

Consider the following problem: let a real function of n real variables be given having the form: $y(x_1, x_2, ..., x_n) = \sum_{i=1}^{n} w_i \sin(\alpha_i \pi x_i)\cos(\beta_i \pi x_i)$, where parameters $w_i \sim Uniform(0,1)$, $\alpha_i \sim Uniform(2,8)$, $\beta_i \sim Uniform(2,8)$, $i = 1,...,n$, $Uniform(a,b)$ – a uniform distribution on segment $[a;b]$. Consider the case $n = 3$, $y_{ijk} = y(x_1^i, x_2^j, x_3^k) + N(0,0.3)$, where $x_1^i = 0.1\cdot(i-1)$, $i = 1,...,11$, $x_2^j = 0.1\cdot(j-1)$, $j = 1,...,11$, $x_3^k = 0.1\cdot(k-1)$, $k = 1,...,11$. This way the training set $(y_{ijk}, x_1^i, x_2^j, x_3^k)$ contains $11^3 = 1331$ objects.

Consider an optimization problem of the cross-validation error MSE_{LOO} of ε and σ with fixed value of parameter $C = 5$.

The cluster of the University of Nizhni Novgorod was used to conduct experiments. A node of the cluster contains 2x Intel Xeon 5150 (total 8 cores), 24GB RAM, Microsoft Windows HPC Server 2008.

Let us study the results of the experiments. In all runs the following parameters were used: search accuracy $\rho = 0.01$, reliability parameter $r = 2$. Let us note that solving the given problem with the specified accuracy using full search over a uniform grid would require 10^4 iterations which is significantly more than the number of iterations spent on optimum search using the considered global search algorithm. All its implementations 1-3 have reached the required accuracy after 400 iterations, obtained values of the optimum were 0.090543, 0.090564 and 0.09054. Working times of the algorithms are given in the table below.

Table 1. Running time of the algorithms (optimization of a function of two variables)

Implementation	Scannings	Cores on scanning	Time (hours)
1	3	1	2,4
2	3	1	2,4
3	3	2	1,2
3	3	4	0,6

Implementation 3 showed 2x performance advantage compared to the other implementations having used 2 times more cores. Taking into account that implementations 1 and 2 are substantially limited in ability of parallel usage of computing devices, implementation 3 (having same computing time per core) allows usage of bigger number of cluster nodes which leads to computing time decrease.

4.3 Optimization of a Function of Three Variables: Efficiency of the Algorithm and Its Modifications When Using Parallel Computing

This section presents scalability of implementation 3 measured on the cluster of the University of Nizhni Novgorod. We solved the optimization problem on three parameters stated in the previous section, the value of parameter C is no longer fixed. Accuracy of the optimum search (along the coordinate) in this series of experiments was $\rho = 0.02$. It was sufficient to make 1000 iterations to achieve the desired search accuracy. Taking into account that the number of scannings is limited by $N(N-1)+1$, where N=3 (the problem dimension), we ran the algorithms to test scalability in the following configurations: using L=3 scannings on three nodes and using L=6 scannings on six cluster nodes (implementation 2 of the algorithm), using L=6 scannings on six cluster nodes and 2 or 4 cores for each of the scannings (implementation 3 of the algorithm). All algorithms found the optimal value 0.09054 of the objective function, working time of the algorithms is given in the table below.

Table 2. Running time of the algorithms (optimization of a function of three variables)

Implementation	Scannings	Cores on scanning	Time (hours)
2	3	1	5.5
2	6	1	2.8
3	6	2	1.4
3	6	4	0.8

The results demonstrate a linear speedup in the number of used nodes and cores. More cluster nodes could be used in case of a larger number of SVM parameters (which depends on used kernel).

5 Conclusions

This paper considers the problem of optimal parameter selection for the regression construction method using support vector machines. Cross-validation error function was chosen as the optimized criteria in the given problem. Arising problem belongs to the class of bound constrained nonlinear optimization, objective function is often multi-extremal which stipulates the necessity of global optimization methods application.

For the solution of the problem the paper suggests to use global search methods presented in studies of R. Strongin, V. Gergel, V. Grishagin, Ya. Sergeev, et al., with a number of modifications (rotated scannings, combined scheme of parallel computing). The questions of parallel computing usage are considered within this approach. The combined scheme was proved to be the most suitable for cluster systems with large number of multicore processors due to the possibility of all computing resources utilization. Computational experiments were conducted to support the stated propositions. Three schemes of parallel implementation of the global search algorithm are analyzed. Scalability of the first two schemes is limited by the number of scannings used by the algorithm. The combined scheme of the parallel global search algorithm demonstrates linear speedup in SVM-regression parameters optimization, lets use multicore processors effectively and therefore scales much better.

Acknowledgements. This study was conducted with the support of the Ministry of Education and Science of Russian Federation, the program "Research and development in priority directions of Russia science and technology complex", contract 11.519.11.4015; the program "Scientific and academic human resources of innovation Russia", contract 14.B37.21.0393.

The authors thank Alexander Sysoyev, Alexey Sidnev, Vasily Ryabov and Nina Kurina for help and useful discussions.

References

1. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning: Data Mining, Inference and Prediction. Springer, Heidelberg (2009)
2. Ito, K., Nakano, R.: Optimization Support Vector Regression Hyperparameters Based on Cross-Validation. In: Proceedings of the International Joint Conference on Neural Networks, vol. 3, pp. 2077–2083 (2003)
3. Ren, Y., Bai, G.: Determination of Optimal SVM Parameters by Using GA/PSO. Journal of Computers 5(8), 1160–1168 (2010)
4. Strongin, R.G.: Algorithms for multi-extremal mathematical programming problems employing the set of joint space-filling curves. J. of Global Optimization 2, 357–378 (1992)

5. Strongin, R.G., Sergeyev, Y.D.: Global optimization with non-convex constraints. Sequential and parallel algorithms. Kluwer Academic Publishers, Dordrecht (2000)
6. Gergel, V.P., Strongin, R.G.: Parallel computing for globally optimal decision making on cluster systems. Future Generation Computer Systems 21(5), 673–678 (2000)
7. Barkalov, K., Ryabov, V., Sidorov, S.: Parallel Scalable Algorithms with Mixed Local-Global Strategy for Global Optimization Problems. In: Hsu, C.-H., Malyshkin, V. (eds.) MTPP 2010. LNCS, vol. 6083, pp. 232–240. Springer, Heidelberg (2010)
8. Jin, R., Chen, W., Simpson, T.W.: Comparative Studies of Meta-modeling Techniques under Multiple Modeling Criteria. Struct. Multidiscip. Optim. 23(1), 1–13 (2001)
9. Wang, Y., Liu, Y., Ye, N., Yao, G.: The Parameters Selection for SVM Based on Improved Chaos Optimization Algorithm. In: Zhang, J. (ed.) ICAIC 2011, Part V. CCIS, vol. 228, pp. 376–383. Springer, Heidelberg (2011)
10. Zhang, H., He, Y.: Comparative study of chaotic neural networks with different models of chaotic noise. In: Wang, L., Chen, K., S. Ong, Y. (eds.) ICNC 2005. LNCS, vol. 3610, pp. 273–282. Springer, Heidelberg (2005)
11. Fröhlich, H., Zell, A.: Efficient parameter selection for support vector machines in classification and regression via model-based global optimization. In: IEEE International Joint Conference on Neural Networks, IJCNN 2005, vol. 3, pp. 1431–1436 (2005)
12. Jones, D., Schonlau, M., Welch, W.: Efficient global optimization of expensive black-box functions. J. Global Optimization 13, 455–492 (1998)
13. Momma, M., Bennett, K.P.: A pattern search method for model selection of support vector regression. In: Proceedings of SIAM Conference on Data Mining, pp. 261–274. SIAM, Philadelphia (2002)
14. Vapnik, V.: The Nature of Statistical Learning Theory. Springer, New York (1996)
15. Kearns, M.: A bound on the error of cross validation using the approximation and estimation rates, with consequences for the training-test split. In: Adv. In Neural Information Processing Systems, vol. 8, pp. 183–189. MIT Press (1996)
16. Ng, A.Y.: Preventing of overfitting of cross-validation data. In: Proc. 14th Int. Conf. on Machine Leaning, pp. 245–253. Morgan Kaufmann (1997)
17. Grishagin, V.A., Sergeyev, Y.D., Strongin, R.G.: Parallel characteristical global optimization algorithms. Journal of Global Optimization 10(2), 185–206 (1997)
18. Sidorov, S.V.: Two-level parallel index algorithm for global optimization. Vestnik of Lobachevsky State University of Nizhni Novogorod 5(2), 208–213 (2012) (in Russian)