Constrained Motion Particle Swarm Optimization and Support Vector Regression for Non-Linear Time Series Regression and Prediction Applications

Nicholas I. Sapankevych
Raytheon Company
St. Petersburg, FL, USA
nicholas_i_sapankevych@raytheon.com

Prof. Ravi Sankar
Department of Electrical Engineering
University of South Florida
Tampa, FL, USA
sankar@eng.usf.edu

***Abstract*— Support Vector Regression (SVR) has been applied to many non-linear time series prediction applications [1]. There are many challenges associated with the use of SVR for non-linear time series prediction, including the selection of free parameters associated with SVR training. To optimize SVR free parameters, many different approaches have been investigated, including Particle Swarm Optimization (PSO). This paper proposes a new approach, termed Constrained Motion Particle Swarm Optimization (CMPSO), which selects SVR free parameters and solves the SVR quadratic programming (QP) problem simultaneously. To benchmark the performance of CMPSO, Mackey-Glass non-linear time series data is used for validation. Results show CMPSO performance is consistent with other time series prediction methodologies, and in some cases superior.**

Keywords—Support Vector Regression, Particle Swarm Optimization, Time Series Regression and Prediction

I. INTRODUCTION

As outlined in [1], numerous time series regression/prediction applications have used SVR successfully; however, one of the significant challenges associated with using SVR is the selection of the free parameters associated with this algorithm. The motivation of the proposed approach is to 1) take advantage of the “model-less” SVR properties, and 2) simultaneously solve the SVR QP problem and find optimal SVR free parameters.

Constrained Motion Particle Swarm Optimization (CMPSO) described in this paper is the application of a modified PSO algorithm designed specifically to aid in the SVR free parameter selection process. It is based on Kennedy and Eberhart's [7] original PSO algorithm, but modified to accomplish several optimization tasks simultaneously.

II. BACKGROUND

A. Support Vector Regression

One method to predict non-linear time series is Support Vector Regression (SVR) originally developed by Vapnik et. al. [2] and has been used successfully applied for many time series prediction applications [1] (the reader is encouraged to see [3] for technical background and [4] for a general tutorial). It has advantages for problems that are non-linear in nature and do not have any underlying process identified. The fundamental principal behind the SVR approach is to cast the time series data into "feature" space by use of kernels and solve for a linear best fit using quadratic programming with Lagrange multipliers.

With the use of Lagrange multipliers and some mathematical manipulation [5], the dual of the quadratic programming (QP) problem for SVR can be formulated in equations (1) and (2):

$$\text{Maximize} \sum_{i=1}^{N} \alpha_i y_i - \varepsilon \sum_{i=1}^{N} |\alpha_i| - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_i K(x_i, x_j) \tag{1}$$

$$\text{Subject to:} \quad -C \le \alpha_i \le C\,, \sum_{i=1}^{N} \alpha_i = 0\,, C > 0 \tag{2}$$

Where:
x,y: Independent, Dependent Variables
α: Lagrange Multiplier (to be solved using QP or other technique)
K: Kernel Function
C: Capacity Term
ε: Epsilon Tube Size

There are many different kernel functions that can be used - for this study, the exponential kernel given in equation (3) is used:

$$K(A,B) = exp\left(-\frac{1}{2\sigma^2}\|A-B\|^2\right) \tag{3}$$

Where:
σ: "Kernel free parameter"
A,B: "Input vectors"

The resulting function approximation is given in equation (4):

978-0-7695-5144-9/13 $31.00 © 2013 IEEE
DOI 10.1109/ICMLA.2013.164

$$\hat{y} = \sum_{i=1}^{N} a_i^* \, K(x, x_i) + b^* \qquad (4)$$

Note the star notation denotes the Lagrange multipliers that lie on or outside the ε tube. The solution for the Lagrange multipliers and the bias term b can be found using standard QP programs as well as other techniques such as Sequential Minimization Optimization (SMO) [6]. From this formulation, a minimum of three free parameters have been identified: capacity term C, tube size ε, and Kernel function parameter σ. These are user defined parameters that must be tuned to minimize time series prediction error and produce the best fitting curve.

B. Particle Swarm Optimization

Particle Swarm Optimization (PSO) was first developed by Kennedy and Eberhart [7] in 1995 and is one technique that can be used to solve for free parameters associated with the QP problem in SVR. The reader is encouraged to read [8] and [9] for general technical tutorials of PSO and [10] for a detailed survey of the many different PSO applications to real world problems.

The PSO formulation starts with the definition of the free variables and the boundaries of the solution space. Equations (5a, 5b) are generalizations of a particle for the SVR example given in the previous section:

Particle i Position:	$p_i(C, \varepsilon, \sigma, \alpha_1 \ldots \alpha_N, b)$	(5a)
Particle i Velocity:	$v_i(C, \varepsilon, \sigma, \alpha_1 \ldots \alpha_N, b)$	(5b)

Each particle p has a position in solution space that includes the three SVR free parameters, the Lagrange multipliers, and the bias term. The boundaries of each of the free variables are determined by the user before the PSO algorithm is executed. A particle moves through the solution space using the simple kinematic equation (6):

Particle Position Update at k+1: $$p_{i,k+1} = p_{i,k} + \Delta_t v_{i,k} \qquad (6)$$

The position of each particle is updated every epoch (indexed by k) by adding the velocity term with each position and setting Δt equal to one. The velocity of each particle is given as (7):

Particle Velocity Calculation:
$$\begin{aligned} v_{i,k} &= w_I v_{i,k} + \\ &rand \cdot w_c\left(p_{i,best} - p_{i,k}\right) + \\ &rand \cdot w_s\left(g_{best} - p_{i,k}\right) \end{aligned} \qquad (7)$$

Where:
wI, wc, ws: Inertial, Cognitive, and Social Weights
rand: Random numbers over closed interval [0,1] - note the two rand functions in (7) are independent
p(i,best): Particle i's best result over all epochs k
gbest: Swarm's best result over all epochs k ("global best")

The selection of pbest and gbest is determined by the evaluation of a "fitness" function that represents the "goodness" of the solution.

There are several techniques for handling cases where the updated particles' position is outside the solution area boundaries, and for the approach described here both absorbing walls and invisible walls are used.

III. Constrained Motion Particle swarm optimization

One of the challenges associated with implementing PSO for optimizing SVR, specifically the SVR free parameters, is the requirement to re-compute the Lagrange multipliers and bias term via a QP solver for every particle position update. This can lead to computational inefficiency and longer optimization times. Also, the free variable capacity term C constrains the solution space boundaries of the Lagrange multipliers (see constraint in equation 2). Since C is a variable, the Lagrange multiplier boundaries for any given particle will always be moving, making the PSO implementation complicated. Constrained Motion Particle Swarm Optimization is an approach that optimizes the SVR free parameters while minimizing the need for a QP program to solve for the Lagrange multipliers and bias term each time a free parameter is changed.

The first step in the development of the CMPSO process is to factor the capacity term C in equations (8) and (9):

$$\text{Maximize } \; C \cdot \left(\sum_{i=1}^{N} \alpha_i y_i - \varepsilon \sum_{i=1}^{N} |\alpha_i| - \frac{C}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_i \, K(x_i, x_j) \right) \qquad (8)$$

$$\text{Subject to: } \quad -1 \le \alpha_i \le 1 \, , \; \sum_{i=1}^{N} \alpha_i = 0, C > 0 \qquad (9)$$

Now the Lagrange multiplier boundary is no longer variable and the capacity term can be varied without concern about changing PSO variable boundary limits.

The next step is to manipulate the motion of the particles such that every Lagrange multiplier set (α1...αN) for any particle will offer a feasible solution (i.e. satisfy the summation to zero constraint of equation 9). For CMPSO, the same random selection is used; however, the Lagrange multiplier initial positions and velocities are randomly selected such that the constraint in equation (9) is satisfied.

Note that the PSO equations of motion in equations (6) and (7) are weighted sums of the previous position and also note that the sum of each particle's Lagrange multipliers' position and velocity sum to zero. For each epoch, the updated Lagrange multiplier particle position will also always sums to zero. By "constraining" the motion such that every particle Lagrange multiplier position update sums to zero, the constraints of equation (9) are always satisfied, thus providing a feasible solution for the dual SVR objective function in equation (8) for every particle throughout the entire optimization process.

The last part of CMPSO is the implementation of the Particle Swarm objective function. Recall the ultimate goal is to concurrently estimate a non-linear time series accurately and to solve the SVR QP. For CMPSO, the fitness function is defined in equation (10) ([4] and [5]):

$$\frac{\textit{Primal Objective} - \textit{Dual Objective}}{\textit{Primal Objective} + 1} \tag{10}$$

This fitness function is also the stopping criteria for the SMO algorithm. For CMPSO (and as shown in the results section), a value of 0.25 is sufficient to complete the CMPSO process for selecting SVR free parameters and seed the SMO algorithm which uses a value of 0.001 as a stopping criteria ([4] and [5]).

Figure 1 illustrates the difference between the CMPSO process and other published formulations.

IV. Results

To measure the utility of CMPSO, a "standard" non-linear time series is used: Mackey-Glass data:

$$\frac{dx(t)}{dt} = -bx(t) + \frac{ax(t-\tau)}{1+x(t-\tau)^{10}} \tag{11}$$

Where:
$a = 0.2$
$b = 0.1$
$\tau = 17$
$x(t0) = 1.2$
$\Delta t = 0.1$ (used in integration)

The performance results using CMPSO are based on specific measures. For the results stated, the Error Mean, Normalized Mean Square Error, Root Mean Square Error and Directional Symmetry were used as measures of effectiveness.

Two experiments using Mackey-Glass data were used to evaluate CMPSO's performance:

1) Regression: The first 2000 data points of the Mackey-Glass data, sampled every 15 samples ($\Delta t = 0.1$), were used as a both the training data set and the test set.

2) Time series prediction: Starting with the first 2000 data samples, sampled every 15 samples ($\Delta t = 0.1$), the CMPSO was trained and the following 1 to 30 sample prediction horizons were estimated and compared to truth data. This process was repeated 50 times, moving the starting sample training sample point by 100. The resulting ensemble of prediction horizon data is then measured for accuracy and consistency.

A sample results is shown in Figure 2 - the blue curve is the output of CMPSO; the red curve is the output of SMO, with an ε set to a tighter tolerance of 0.01 (user selectable). The numeric results are given in Table 1.

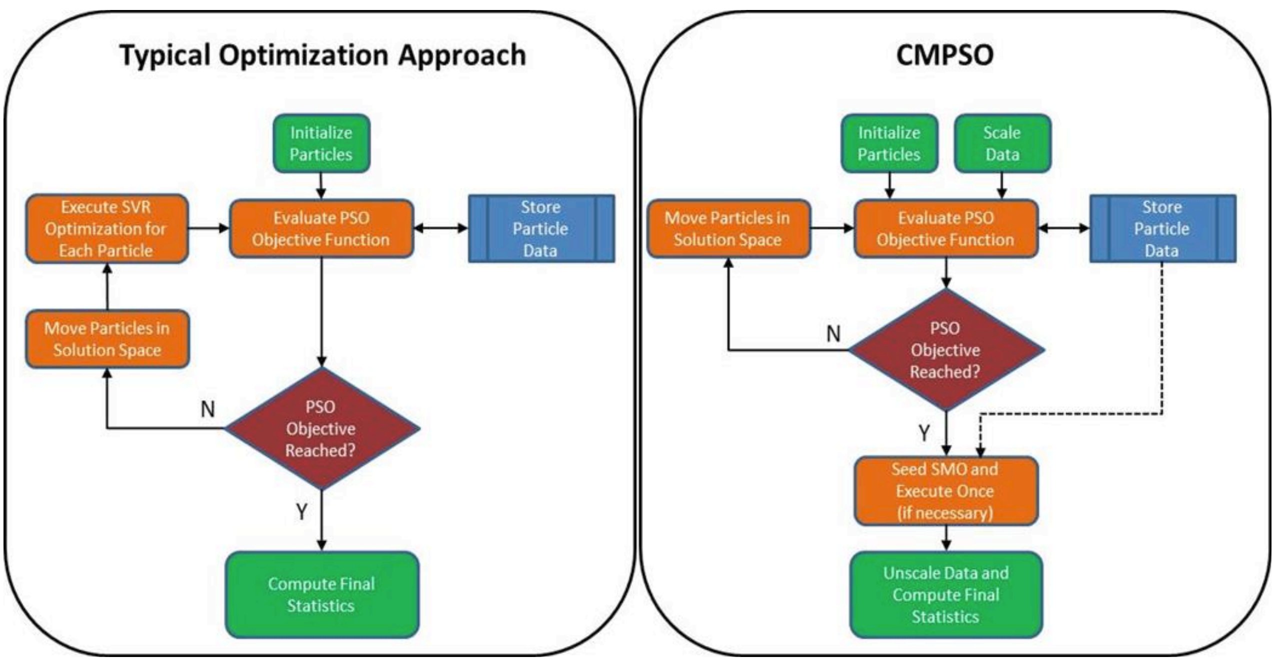

Figure 1: CMPSO vs. Other Optimization Approaches

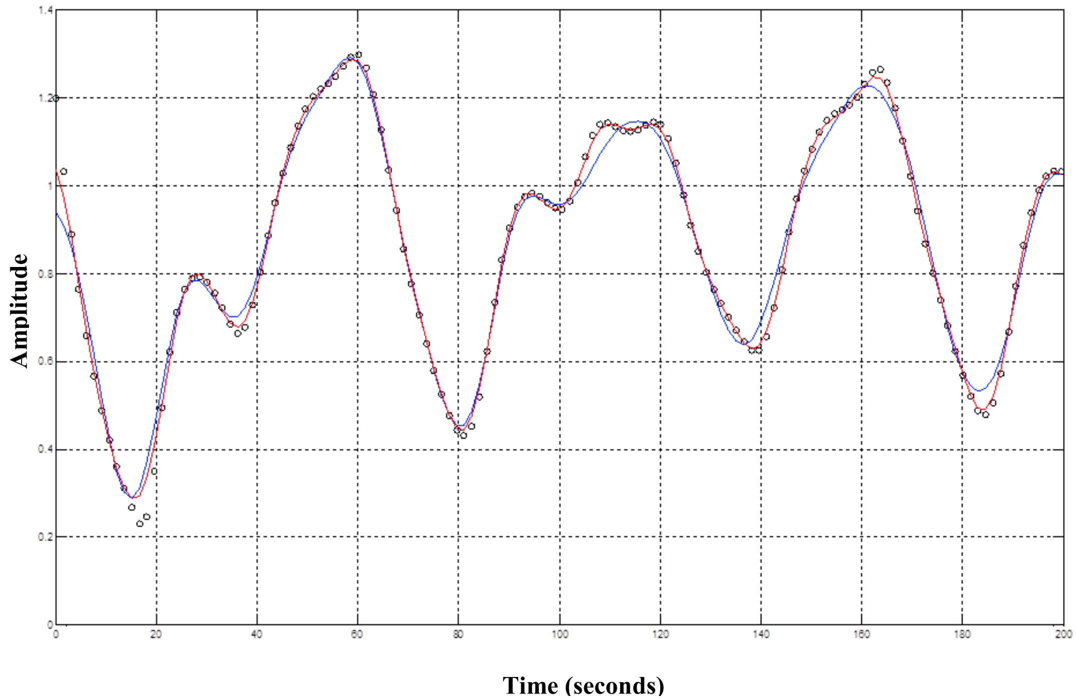

Figure 2: CMPSO Estimation – Mackey-Glass Data
(Blue Curve: pre-SMO Output, Red Curve: post-SMO Output, Black Circles: Training Points)

TABLE 1: CMPSO Estimation Statistics: Mackey-Glass Data

Metric	MSE	NMSE	RMSE
Prediction Horizon			
Regression	0.0006	0.0055	0.0199
+1 Sample	0.0006	0.0120	0.0254
+5 Samples	0.0014	0.0268	0.0379
+10 Samples	0.0032	0.0610	0.0569
+15 Samples	0.0062	0.1176	0.0787
+20 Samples	0.0104	0.1988	0.1018
+30 Samples	0.0214	0.4184	0.1461

Using the Mackey-Glass benchmark, CMPSO is compared to other published findings that also use SVR time series prediction techniques (refer to [13], [14], [15], and [16]) with the results shown in Table 2:

TABLE 2: CMPSO Performance Comparison to Other SVR Techniques Using Mackey-Glass Benchmark Data

Prediction Horizon (Samples)	**Regression**	**+1**	**+5**	**+10**
Technique	**MSE**	**MSE**	**MSE**	**MSE**
CMPSO	0.0006	0.0006	0.0014	0.0032
ROLS [13]	-	0.1064	0.1696	0.3248
SVR [14]	0.0004	-	-	-
SVM+RT [15] (NMSE of test set in Table 1 of ref [15])	-	0.5158	-	-
KDM [16]	-	0.0066	-	-

The results shown in Table 1 illustrate CMPSO performance as both a regression technique as well as a prediction technique using non-linear benchmark data. Also, Table 2 shows CMPSO performance (using MSE as a metric) as compared to other published SVR related time series prediction techniques. CMPSO is comparable and in most cases superior to the other techniques using the same type of Mackey-Glass non-linear time series benchmark data.

V. CONCLUSION

Non-linear time series regression and prediction is a very challenging field of research. There are many non-linear time series prediction applications and methodologies designed for these types of applications, including SVR. CMPSO takes advantage of both the time series regression and prediction capabilities of SVR and the parameter optimization capabilities of PSO by fusing the two processes into one combined approach. Initial results shown in this paper illustrate the effectiveness of this technique for both regression and prediction applications on Mackey-Glass non-linear time series benchmark data.

The CMPSO approach has several distinct advantages relative to other discussed techniques:

1) Guaranteed SVR feasible solution for every particle in the solution space for every movement update.

2) QP solver only executed once - if deemed necessary - and is seeded with partial solution.

3) The CMPSO motion "constraints" improve computational efficiency by not only guaranteeing solution feasibility, but forcing the motion of the particles towards ever better solutions.

4) General CMPSO framework highly adaptable to a wide range of real-world applications as demonstrated by the stated results.

Future research should include the optimization of the PSO free parameters as well as CMPSO performance against other real world applications such as financial market forecasting and electric utility power load prediction.

REFERENCES

[1] N. Sapankevych and R. Sankar, "Time Series Prediction Using Support Vector Machines: A Survey", IEEE Computational Intelligence, vol. 4, num. 2, May 2009, pg 24-38

[2] V. N. Vapnik, " The Nature of Statistical Learning Theory", Springer, 1995

[3] V. N. Vapnik, "Statistical Learning Theory", John Wiley and Sons, 1998.

[4] A. J. Smola and B. Schölkopf, "A Tutorial on Support Vector Regression", NeuroCOLT Technical Report, Royal Holloway College, London, UK, 1998

[5] N. Cristianini and J. Shawe-Taylor, "An Introduction to Support Vector Machines and Other Kernel-based Learning Methods", Cambridge University Press, 2000

[6] John C. Platt, "Sequential Minimization Optimization: A Fast Algorithm for Training Support Vector Machines", Technical Report MSR-TR-98-14, Microsoft Research, 1998

[7] J. Kennedy and R. Eberhart, "Particle Swarm Optimization", Proceedings of the IEEE International Conference on Neural Networks, vol. 4, 1995, pg 1942–1948

[8] R. Poli, J. Kennedy, and T. Blackwell, "Particle Swarm Optimization: An Overview", Swarm Intell, Vol. 1, pg. 33-57, 2007

[9] J. Robinson and Y. Rahmat-Samii, "Particle Swarm Optimization in Electromagnetics", IEEE Transactions on Antennas and Propagation, Vol. 52, No. 2, February 2004

[10] R. Poli, "An Analysis of Publications on Particle Swarm Optimization Applications", University of Essex, UK, Department of Computer Science, Technical Report CSM-469, May 2007, Revised November 2007

[11] U. Paquest and A. P. Engelbrecht, "Training Support Vector Machines with Particle Swarms", Proceedings International Joint Conference on Neural Networks, pg. 1593-1598, 2003

[12] F. E. H. Tay and L. J. Cao, "Application of Support Vector Machines in Financial Time Series Forecasting", Omega, Vol. 29, pages 309-317, 2001

[13] K. L. Lee and S. A. Billings, "Time Series Prediction Using Support Vector Machines, the Orthogonal and Regularized Orthogonal Least-Squares Algorithms", International Journal of Systems Science, Volume 33, Number 10, pages 811-821, 2002

[14] S. Mukherjee, E. Osuna, and F. Girosi, "Nonlinear Prediction of Chaotic Time Series Using Support Vector Machines", Proceedings of the 1997 IEEE Workshop – Neural Networks for Signal Processing VII, pages 511-520, 24-26 September 1997

[15] J. M. Górriz, C. G. Puntonet, M. Salmerón, R. Martin-Clemente, and S. Hornillo-Mellado, "Using Confidence Interval of a Regularization Network", Proceedings of the 12th IEEE Mediterranean Electrotechnical Conference (MELECON 2004), pages 343-346, May 12-15, 2004

[16] L. Ralaivola and F. d'Alche-Buc, "Dynamical Modeling with Kernels for Nonlinear Time Series Prediction", Neural Information Processing Systems (NIPS 2003), 8-13 December 2003I. S. Jacobs and C. P. Bean, "Fine particles, thin films and exchange anisotropy," in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.