Proceedings of the Third International Conference on Machine Learning and Cybernetics, Shanghai, 26-29 August 2004

SUPPORT VECTOR REGRESSION WITH LOCAL ε PARAMETERS WITH THE SUPPORT VECTORS

XUNXIAN WANG, YUNFENG WANG, DAVID BROWN

Intelligent systems and fault diagnostic group, Department of Electronic and Computer engineering, University of Portsmouth, Portsmouth, PO1 3DJ, UK

Abstract:

In support vector machine regression (SVR) a big ε value will give a rough system model with little support vectors and a small ε value will give a accurate system model with many support vectors. The selection of the support vectors will be effected by a small change of the training data. To obtain an accurate model with little support vectors, a method includes two steps is proposed in this paper, in step one, a big ε value is used to select a small number of the support vectors; in step two, by giving these selected support vectors a small value while others a big one, a accurate system model will be obtained. The experimental results demonstrate the efficiency of the proposed method.

Keywords:

Machine learning; regression; support vector machine; Mercer kernel;

1. Introduction

The generalization ability of a learning machine can be measured by its VC confidence [1,2] which tells that in order to obtain a good learning machine by using a given training set, both of the empirical risk and the VC-dimension should be small. Support vector machine classification (SVMC) [1-4] uses linear function as the classification function. By applying the maximum margin principle on the determination of the function parameters, a good classifying machine can be obtained. Expended the idea to regression problem, the error band is used in support vector machine regression (SVMR) [1,2,5,6]. And linear functions are used to do the system regression just as in the SVMC. In the situation where linear regression is improper, a mapping will be seek by which the training data can be mapped into a high dimensional space where the linear regression can be implemented. Due to the difficulties of finding the exactly mapping, Reproducing Kernel Hilbert Space [RKHS] theory [7] is applied and plenty of ideas are used to determine the kernel function, kernel parameter [8-10] as well as to determine the width of the error band denoted by ε [11]. In this paper, we show that even when the proper mapping is known, the obtained system model will not be in the ideal format. In addition, a small change of the training set will result in a different system model. To improve the method, two levels of ε value are used. While the big one is used to determine the position of the support vectors, the small one is used to refine the weight of each support vectors.

2. Basic support vector regression algorithm

Given N pairs of training set $\{(x_k, y_k)\}_{k=1}^{l}$ where x is the m-dimensional input variable, y is the output variable, if the Hyper-plane function is used to approximate the set, the regression function can be stated as

$$y = (w \cdot x) + b \tag{2}$$

With the above linear function as the regression function, the SVMR problem can be represented as the following constraint minimization problem

$$\text{Minimize } \Phi(w, \xi^*, \xi) = \frac{1}{2}(w \cdot w) + C\left(\sum_{i=1}^{l} \xi_i^* + \sum_{i=1}^{l} \xi_i\right)$$

$$\text{Subject to } \begin{array}{l} y_i - (w \cdot x_i) - b \le \varepsilon + \xi_i^* \\ (w \cdot x_i) + b - y_i \le \varepsilon + \xi_i \\ \xi_i^* \ge 0 \\ \xi_i \le 0 \end{array} \quad \text{for } 1 \le i \le l \tag{3}$$

Through its Lagrange format, use KKT condition, its Dual problem can be represented as

Maximize

$$W(\alpha, \alpha^*) = -\frac{1}{2}\sum_{i,j=1}^{l}(\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j)(x_i \cdot x_j)$$

$$-\varepsilon\sum_{i=1}^{l}(\alpha_i^* + \alpha_i) + \sum_{i=1}^{l}(\alpha_i^* - \alpha_i)y_i$$

0-7803-8403-2/04/$20.00 ©2004 IEEE

$$\sum_{i=1}^{l}(\alpha_i^* - \alpha_i) = 0$$

Subject to $0 \le \alpha_i^* \le C, \quad i = 1,\cdots,l$

$$0 \le \alpha_i \le C, \quad i = 1,\cdots,l \qquad (4)$$

Due to the convexity of the problem (3), The dual gap of the above two problems is zero and so the solution of problem (3) can be obtained by solving problem (4) [12].

SVM is advantaged from the dot product of the input value $x_i, x_j, i, j = 1,\cdots,l$ in the above equation. If the hyper-plane is not suitable for the regression problem, a mapping $\varphi(x)$ can be used to map the trained set from its original space to a high dimensional space (sometimes called feature space), where the training data set is expected to be represented by a hyper-plane in the new space. After this mapping $\varphi(x)$, the training set becomes $\{(\varphi(x_k), y_k)\}_{k=1}^{l}$. As a result, the dot product in the above equation becomes $(\varphi(x_i)\cdot\varphi(x_j))$. Not easy to get this map $\varphi(x)$ and sometimes the mapping has too many items to deal with, the Mercer kernel is used to replace the above product by defining

$$K(x_i, x_j) = (\varphi(x_i)\cdot\varphi(x_j))$$

Then the general SVMR problem can be stated as Maximize

$$W(\alpha,\alpha^*) = -\frac{1}{2}\sum_{i,j=1}^{l}(\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j)K(x_i, x_j)$$

$$-\varepsilon\sum_{i=1}^{l}(\alpha_i^* + \alpha_i) + \sum_{i=1}^{l}(\alpha_i^* - \alpha_i)y_i$$

$$\sum_{i=1}^{l}(\alpha_i^* - \alpha_i) = 0$$

Subject to $0 \le \alpha_i^* \le C, \quad i = 1,\cdots,l$

$$0 \le \alpha_i \le C, \quad i = 1,\cdots,l \qquad (5)$$

Solving the above optimisation problem, the resulting learning machine can be represented as

$$f(x) = \sum_{i=1}^{l}(\alpha_i^* - \alpha_i)K(x, x_i) + b \qquad (6)$$

3. Support vector regression analysis by a simple example

The training set is given by using the following cosine function plus noise e as

$$y = \cos(x - x_0) + e \qquad (7)$$

If the mapping $\varphi(x)$ is defined as

$$\varphi(x) = (\cos x, \sin x) = (p, q) \qquad (8)$$

function (7) can be converted from 1-D domain space to a linear function in a 2-D domain space as below

$$\begin{aligned} y &= \cos x\cos x_0 + \sin x \sin x_0 + e \\ &= pp_0 + qq_0 + e \end{aligned} \qquad (9)$$

With mapping φ, the kernel can be written as

$$\begin{aligned} k(x, y) &= <\varphi(x), \varphi(y)> \\ &= <(\cos x, \sin x), (\cos y, \sin y)> \\ &= \cos(x - y) \end{aligned} \qquad (10)$$

As an example, if the training set $\{x_i, y_i\}_{i=1}^{l}$ is given, the regression function will have the following format where $\beta = (\beta_1 \quad \cdots \quad \beta_l)$ will be determined by SVMR algorithm.

$$f(x;\beta) = \sum_{i=1}^{l}\beta_i \cos(x - x_i) \qquad (11)$$

choose the training set as $\{(x_i, y_i) \mid y_i = \cos(x_i), x_i = 0.5*i - 5\}_{i=1}^{20}$. Use C=1, $\varepsilon = 0.1$ in SVMR, a system model with 3 support vectors can be obtained as

$$\begin{aligned} f(x;\beta) &= 0.4520\cos x - 0.2260\cos(x + 3.0) \\ &\quad -0.2260\cos(x - 3.0) \end{aligned} \qquad (12)$$

The figure is shown in Fig.1.a. From the figure, it can be seen that besides the support vector (SV) at $x = 0$, there are other two SV at $x = -3$ and $x = 3$. But a ideal system model should have only one support vector at $x = 0$.

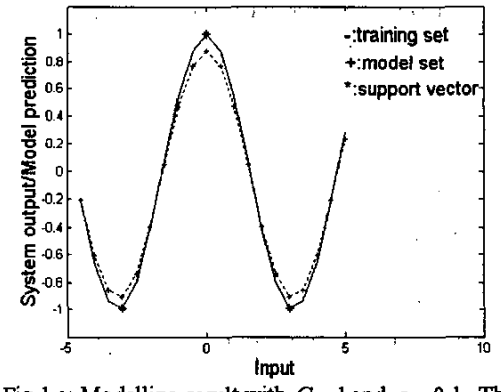

Fig.1.a: Modelling result with $C = 1$ and $\varepsilon = 0.1$. The solid line is the cosine function; dot line with + are the produced system model; * dots are the support vectors (there are total 3 in this situation)

If a single noise is added by set $e(1) = 0.2$. In this situation, the SVR gives a system model with 5 support vectors shown in Fig.1.b. The system model has the following format

$$f(x;\beta) = \cos(x+4.5) - 0.2324\cos(x+4) - 0.0616\cos(x+1) + 0.1708\cos(x+0.5) - \cos(x-2.5) \quad (13)$$

In this situation, the SV in $x = 0$ doesn't exist any more.

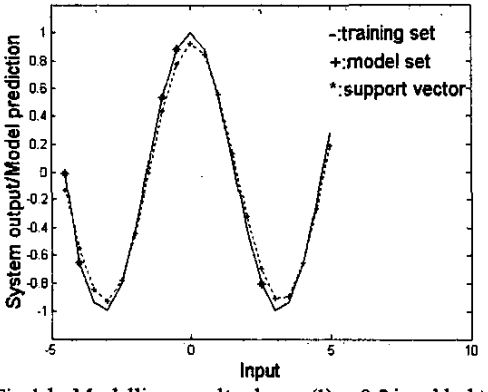

Fig.1.b: Modelling result when $e(1) = 0.2$ is added to the training set (with $C = 1$, $\varepsilon = 0.1$). There are total 5 support vectors in this situation.

The above result shows that a single noise will change the established model a lot.

Keep using the noise model, when use $C = 1$ and $\varepsilon = 0.1, 0.2$ and 0.5 respectively, three models are produced in Fig.2.a. There are 5,3 and 3 support vectors used in the system models.

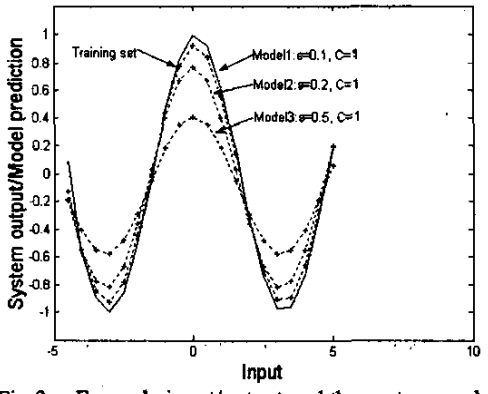

Fig.2.a: Example input/output and the system model produced by SVR with $C = 1$ and when $e(1) = 0.2$ is added to the training set, $\varepsilon = 0.1, 0.2$ and 0.5. There are 5,3 and 3 support vectors used in the system models.

Alternatively, when keep $\varepsilon = 0.1$ and change C=1,0.1 and 0.01 respectively, 5, 12 and 19 support vectors are needed in the system models relatively. The result is shown in Fig2.b.

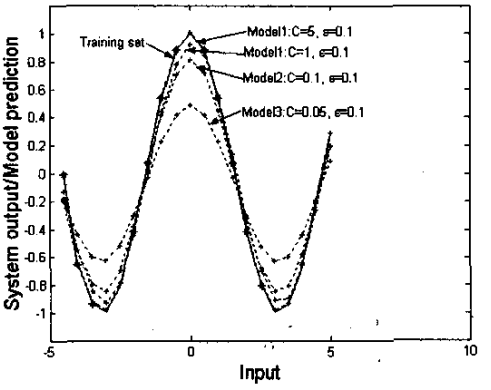

Fig.2.b: Example input/output and the system model produced by SVR with $\varepsilon = 0.1$, $C = 5,1,0.1$ and 0.05 respectively. $e(1) = 0.2$. There are 12,5,12 and 19 support vectors used in the system models.

4 Support vector regression with special ε on the support vectors

Suppressing the influence of the noise as small as possible, keeping the system model as accurate and simple as possible, the following method is proposed. There are two steps: First: determination of the support vectors: be ware that a big ε value can help the system model avoiding the influence of the noise better, a big ε is used to obtain a system model with small number of support vectors; Second, based on the support vectors (SVs) obtained in step 1, change the ε value on these SVs to a small value others unchanged, do the SVMR again, this time, a new system model with much smaller empirical risk will be obtained and the SVs number will not change normally.

The modified the minimization problem is

Minimize

$$\Phi(w,\xi^*,\xi) = \frac{1}{2}(w\cdot w) + C\left(\sum_{i=1}^{l}\xi_i^* + \sum_{i=1}^{l}\xi_i\right)$$

$$\text{Subject to } \begin{cases} y_i - (w \cdot x_i) - b \le \varepsilon_i + \xi_i^* \\ (w \cdot x_i) + b - y_i \le \varepsilon_i + \xi_i \\ \xi_i^* \ge 0 \\ \xi_i \le 0 \end{cases} \text{ for } 1 \le i \le l \tag{3'}$$

Respectively, maximum problem of equation (5) will become

Maximize

$$W(\alpha, \alpha^*) = -\frac{1}{2}\sum_{i,j=1}^{l}(\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j)K(x_i, x_j) - \sum_{i=1}^{l}\varepsilon_i(\alpha_i^* + \alpha_i) + \sum_{i=1}^{l}(\alpha_i^* - \alpha_i)y_i$$

$$\sum_{i=1}^{l}(\alpha_i^* - \alpha_i) = 0$$

$$\text{Subject to } 0 \le \alpha_i^* \le C, \quad i = 1, \cdots, l$$

$$0 \le \alpha_i \le C, \quad i = 1, \cdots, l \tag{5'}$$

The value of ε can be determined as: First, by using equation (5) with a big ε (denoted as $\varepsilon 1$) to obtain a system model. Then change the ε value of these related to the support vectors produced in the former step to a small one, (denoted as $\varepsilon 2$). Now the ε values are

$$\varepsilon = \begin{cases} \varepsilon 1 \text{ for non-support vectors} \\ \varepsilon 2 \text{ for support vectors} \end{cases} \tag{14}$$

To do the SVMR by this new ε, a system model with both small number of support vectors and small empirical risks can be obtained.

As an example, Fig.3 shows the result for the above used training set with single noise e(1) = 0.2. First, $\varepsilon 1 = 0.5$ (C=1) is used and this will produce a system model with 3 support vectors as shown in the figure. Then $\varepsilon 2 = 0.1$ is used to refine the system model. The new model have three support vectors but a much small empirical risk which can be seen easily from the figure. Beside $\varepsilon 2 = 0.1$, $\varepsilon 2 = 0.01$ is tried also to see the effect of the idea. It can be seen that when $\varepsilon 2 = 0.01$ is used, training set and the system model are nearly the same except in the staring period around x=-4.8 where the noise e(1) is available. Actually, The model when $\varepsilon 2 = 0.01$ is

$$f(x, \beta) = 0.4975\cos x - 0.2487(\cos(x+3) + \cos(x-3)) = 0.9942\cos x \tag{15}$$

And the model when $\varepsilon 2 = 0.1$ is

$$f(x, \beta) = 0.4520\cos x - 0.2260(\cos(x+3) + \cos(x-3)) = 0.9034\cos x \tag{16}$$

The one when both $\varepsilon 1 = \varepsilon 2 = 0.5$ is

$$f(x, \beta) = 0.250\cos x - 0.1250(\cos(x+3) + \cos(x-3)) = 0.5\cos x \tag{17}$$

The efficient of the algorithm is obvious.

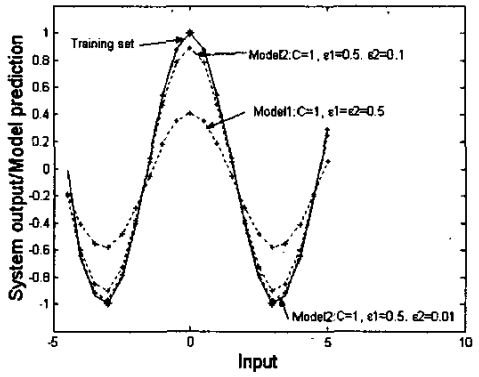

Fig.3: Modelling results with $C = 1, \varepsilon 1 = 0.5$, $\varepsilon 2 = 0.5, 0.1$ and 0.01 respectively.

Next, A more complicated 2-D sinc function added by white zero mean uniform noise is used to show the advantage of the proposed algorithm. The training set is generated by

$$f(x, y) = \frac{5\sin\sqrt{(x-9.99)^2 + (y-9.99)^2}}{\sqrt{(x-9.99)^2 + (y-9.99)^2}} + e \tag{18}$$

where e has uniform (not normal) distribution in the interval [-025,0.25].(Here the Gaussian noise is not used). The input of the training set is given by $\{x_i\}_{i=1}^{10} \times \{y_i\}_{i=1}^{10} = \{2i-10\}_{i=1}^{10} \times \{2i-10\}_{i=1}^{10}$; the test set is given by $\{x_i\}_{i=1}^{50} \times \{y_i\}_{i=1}^{50} = \{0.4i-10\}_{i=1}^{50} \times \{0.4i-10\}_{i=1}^{50}$. (The training set is smaller compared to the test set). Gaussian kernel with the following format is used

$$P(x : x_1, x_2, \sigma) = \exp\left(-\frac{(x-x_1)^2 + (x-x_2)^2}{2\sigma^2}\right) \tag{19}$$

When $\sigma^2 = 2$, C=5, $\varepsilon = 2$ is used with normal SVMR, a system model with 9 support vectors are established shown in Fig.4.a. The used 9 support vectors and the related weight values are listed in Table.1. The pictures are shown in Fig.4.

In Fig.4.a the upper three pictures are training set, test set with uniform noise and the pure test set without noise. The middle three are the system model produced by the SVMR (The middle and the right one are the same). The Lower three pictures are the related errors. From the figure we can see due to the big ε, the errors are big.

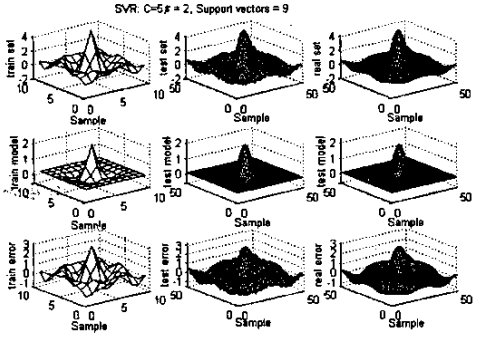

Fig 4.a. The experiment result given by general SVR with C=5, $\varepsilon = 2$. The upper three pictures are the training set, test set with noise and pure set without noise. The middle three are the system model produced by the SVR (The middle and the right one are the same). The Lower three pictures are the related errors

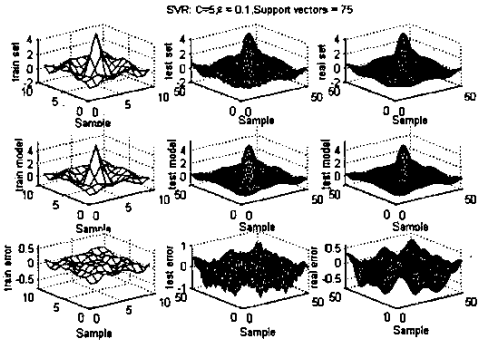

Fig 4.b. The experiment result given by general SVR with C=5, $\varepsilon = 0.1$. The upper three pictures are the training set, test set with noise and pure set without noise. The middle three are the system model produced by the SVR (The middle and the right one are the same). The Lower three pictures are the related errors

When $\sigma^2 = 2$,C=5, $\varepsilon = 0.1$ is used, a system model with 75 support vectors are established shown in Fig.4.b. It is obviously that this model has much small empirical risk than the former one but with much more support vectors (75-9).

When try the 2-steps method proposed in this paper and use $\sigma^2 = 2$,C=5, $\varepsilon 1 = 2$ and $\varepsilon 1 = 0.1$, a system model with 9 support vectors are established shown in Fig. 4.c. With the same support vectors as in Fig4.a, the weight of each support vectors has been changed and result in a system model with much small empirical risk than in Fig.4a.

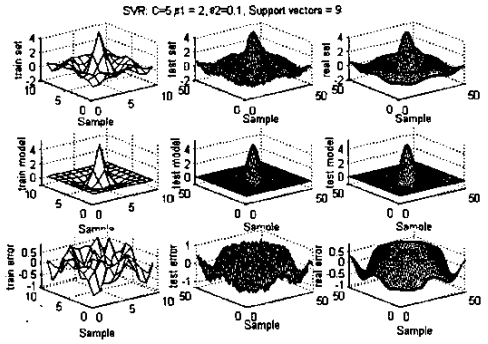

Fig 4.c. The experiment result given by general SVR with C=5, $\varepsilon 1 = 2, \varepsilon 2 = 0.1$. The upper three pictures are the training set, test set with noise and pure set without noise. The middle three are the system model produced by the SVR (The middle and the right one are the same). The Lower three pictures are the related errors

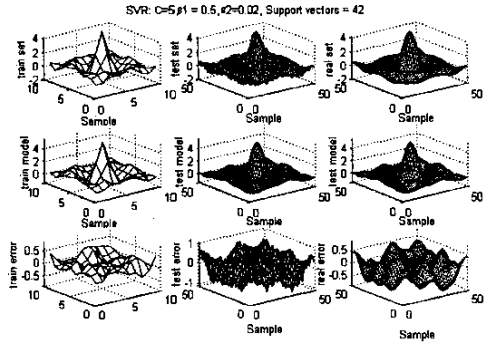

Fig 4.d. The experiment result given by general SVR with C=5, $\varepsilon 1 = 0.5, \varepsilon 2 = 0.02$. The upper three pictures are the training set, test set with noise and pure set without noise. The middle three are the system model produced by the SVR (The middle and the right one are the same).

The Lower three pictures are the related errors. A more result is given to show that when using $\varepsilon 1 = 0.5, \varepsilon 2 = 0.02$, a system model with 42 support vectors can be produced which has similar empirical risk with the model in Fig4.b but the SVs number is much smaller (75-42).

5. Conclusion

The propose method can improve the accuracy of the SVR dramatically as shown in the paper. When same empirical

risk needs to be meet, this method can give a system model with less support vectors and this will result a system model with less regressors.

6. References

[1]. V.N.Vapnik. The nature of statistical learning theory; Springer. 2000

[2]. B. Sch¨olkopf and A. J. Smola. Learning with Kernels. MIT Press, 2002.

[3]. C. J. C. Burges. A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery, 2(2):121–167, 1998

[4]. B.Boser, I. Guyon, and V.N. Vapnik. "A training algorithm for optimal margin classifiers", Fifth Annual Workshop on Computational learning Theory, Pittsburgh ACM, pp 144-152

[5]. V. N.Vapnik, S. Golowich and A. Smola. Support vector method for function approximation, regression estimation and signal processing. In M. C. Mozer, M. I. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems 9*, pages 281-287, Cambridge, MA, 1997. MIT Press

[6]. Smola and B. Sch¨olkopf. A tutorial on support vector regression. Technical Report NC2-TR-1998-030, NeuroCOLT2, 1998.

[7]. N. Aronszajn. Theory of reproducing kernels. Transactions of the American Mathematical Society, 68:337–404, 1950.

[8]. O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee. Choosing multiple parameters for support vector machines. Machine Learning, 46(1):131–159, 2002.

[9]. K. Duan, S.S. Keerthi, and A.N. Poo. Evaluation of simple performance measures for tuning svm hyperparameters. Neurocomputing, 51:41–59, 2003.

[10]. C. S. Ong, A. J. Smola, and R. C. Williamson. Hyperkernels. In Neural Information Processing Systems, volume 15. MIT Press, 2002.

[11]. B. Sch¨olkopf, A. Smola, R. C. Williamson, and P. L. Bartlett. New support vector algorithms. Neural Computation, 12:1207–1245, 2000.

[12]. M.S.Baazraa, H.D.Sherali, C.M.Shetty, "Nonlinear programming: theory and algorithms" second edition, John Wiley & Sons, Inc., 1993

Table.1 The support vectors used in the SVR when $\sigma^2 = 2$,C=5, $\varepsilon = 0.5$

	SV1	SV2	SV3	SV4	SV5	SV6	SV7	SV8	SV9
x_1	-4	-4	-2	-2	0	2	2	4	4
x_2	-2	2	-4	4	0	-4	-4	-2	2
β	-0.1443	-0.0046	0-.1221	-0.3657	1.9950	-0.3901	-0.3187	-0.3021	-0.3474

Table.2 The support vectors used in the SVR when $\sigma^2 = 2$,C=5, $\varepsilon 1 = 0.5, \varepsilon 2 = 0.1$

	SV1	SV2	SV3	SV4	SV5	SV6	SV7	SV8	SV9
x_1	-4	-4	-2	-2	0	2	2	4	4
x_2	-2	2	-4	4	0	-4	-4	-2	2
β	-0.5199	-0.3802	-0.4977	-0.7413	5	-0.7657	-0.6944	-0.6777	-0.7231