ARTICLE IN PRESS

Support vector machines experts for time series forecasting

Lijuan Cao*

Institute of High Performance Computing, 89C Science Park Drive #02-11/12 118261 Singapore

Abstract

This paper proposes using the support vector machines (SVMs) experts for time series fore-casting. The generalized SVMs experts have a two-stage neural network architecture. In the first stage, self-organizing feature map (SOM) is used as a clustering algorithm to partition the whole input space into several disjointed regions. A tree-structured architecture is adopted in the partition to avoid the problem of predetermining the number of partitioned regions. Then, in the second stage, multiple SVMs, also called SVM experts, that best fit partitioned regions are constructed by finding the most appropriate kernel function and the optimal free parameters of SVMs. The sunspot data, Santa Fe data sets A, C and D, and the two building data sets are evaluated in the experiment. The simulation shows that the SVMs experts achieve significant improvement in the generalization performance in comparison with the single SVMs models. In addition, the SVMs experts also converge faster and use fewer support vectors. © 2002 Published by Elsevier Science B.V.

Keywords: Non-stationarity; Support vector machines; Self-organizing feature map; Mixture of experts

1. Introduction

Recently, support vector machines (SVMs) have been proposed as a novel technique in time series forecasting [14–16]. SVMs are a very specific type of learning algorithms characterized by the capacity control of the decision function, the use of the kernel functions and the sparsity of the solution [6,28,29]. Established on the unique theory of the structural risk minimization principle to estimate a function by minimizing an upper bound of the generalization error, SVMs are shown to be very resistant to the

*Institute of High Performance Computing, 1 Science Park Road 01-01 the Capricorn, Science Park II 117528 Singapore.

E-mail address: caolj@ihpc.nus.edu.sg (L. Cao).

1   over-fitting problem, eventually achieving high generalization performance in solving various time series forecasting problems [2,23–26]. Another key property of SVMs is
3   that training SVMs is equivalent to solving a linearly constrained quadratic programming problem so that the solution of SVMs is always unique and globally optimal,
5   unlike other networks' training which requires non-linear optimization with the danger of getting stuck into local minima.

7   In the modeling of time series, two of the key problems are noise and non-stationarity. The noisy characteristic refers to the unavailability of complete information from the
9   past behaviour of the time series to fully capture the dependency between the future and the past. The information that is not included in the model is considered as noise.
11 The noise in the data could lead to the over-fitting or under-fitting problem. The obtained model will have a poor level of performance when applied to new data patterns.
13 The non-stationarity implies that the time series switch their dynamics between different regions. This will lead to gradual changes in the dependency between the input and
15 output variables. In general, it is hard for a single model including SVMs to capture such a dynamic input–output relationship inherent in the data. Furthermore, using a
17 single model to learn the data is somewhat mismatch as there are different noise levels in different input regions—before the single model starts to extract features in some
19 region (local under-fitting), it potentially could have extracted in another region (local over-fitting).

21   A potential solution to the above problems is to use a mixture of experts (ME) architecture [9,10,33,34]. Inspired by the so-called "divide-and-conquer" principle that
23 is often used to attack a complex problem by dividing it into simpler problems whose solutions are combined to yield a solution to the complex problem, the well-known
25 ME consists of a set of expert networks and a gating network that cooperate with each other to solve a complex problem (Fig. 1). Specifically, the expert networks are used
27 to solve different input regions which are softly decomposed from the whole input space by a softmax based gating network. Then the outputs of the expert networks are
29 combined by the softmax based gating network to obtain the solution of the problem. The motivation of the ME is that individual expert networks can focus on specific
31 regions and attack them well.

    Based on the same idea of using different experts for different input regions, Mi-
33 lidiu et al. [13] generalize the ME architecture into a two-stage architecture to handle the non-stationarity in the data. As shown in Fig. 2, in the first stage, the Isodata
35 clustering algorithm is used to partition the whole input space into several disjointed regions. Then, in the second stage, a mixture of experts including partial least squares,
37 K-nearest neighbors and carbon copy are competed to solve partitioned regions. For each particular region, only the expert that best fits it is used for the final prediction. By
39 taking this strategy, the proposed method has an adaptive architecture in the sense any model can be chosen as the expert candidate. Furthermore, by applying the most ade-
41 quate model to each partitioned region, this generalized ME architecture significantly improves prediction performance in comparison with using a single expert model to
43 learn the whole input space.

    This paper generalizes the ME into SVMs for time series forecasting. The idea of
45 generalizing SVMs into the ME architecture has been simply discussed in [12]. Based
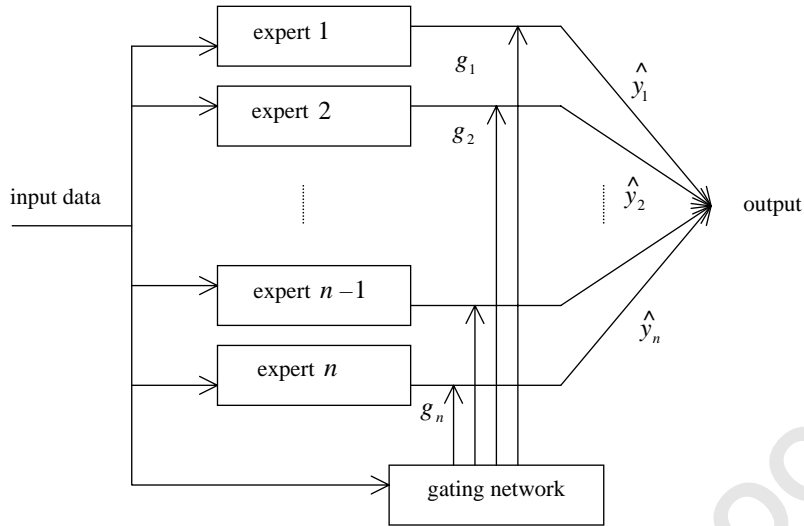
Fig. 1. A typical mixture of experts.

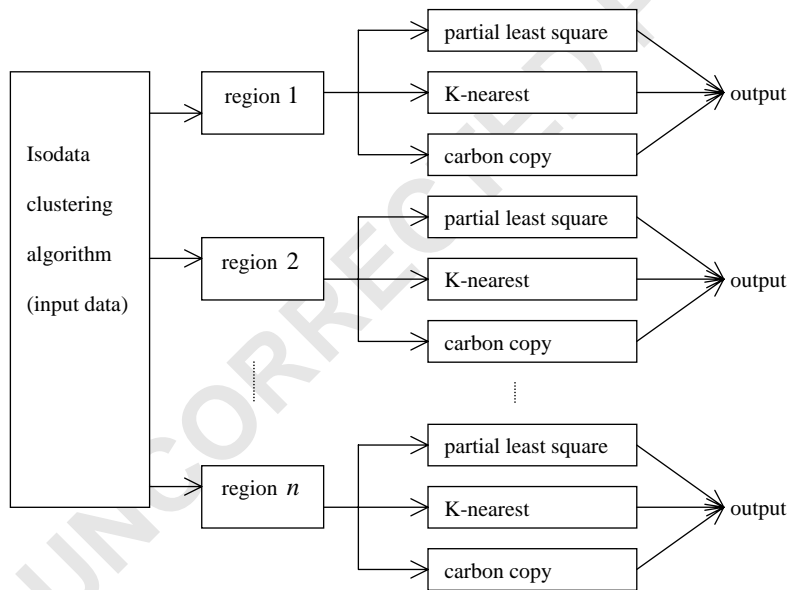Fig. 2. A generalized two-stage mixture of experts.

1   on the original ME architecture (Fig. 1), Kwok directly uses multiple SVMs as expert
    networks, resulting in a weighted quadratic programming (QP) problem. As the weights
3   are functions of input vectors, it is very difficult to solve this complex QP problem.
    Motivated by Milidiu's work (Fig. 2), this paper incorporates the ME architecture into
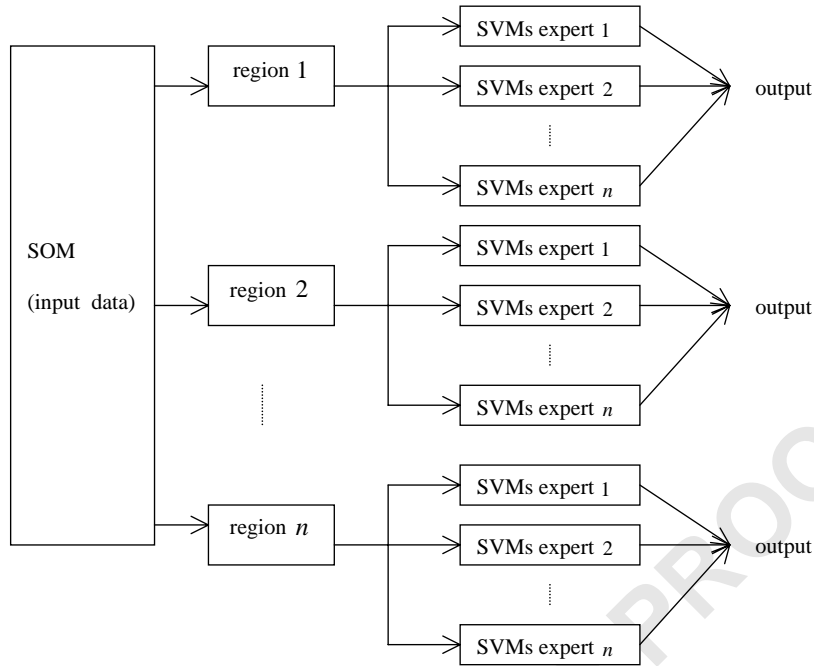
Fig. 3. The generalized SVMs experts.

1   SVMs by using a two-stage neural network architecture. As illustrated in Fig. 3, in
    the first stage, self-organization feature map (SOM) is used to partition the whole
3   input space into several disjointed regions. Then, in the second stage, different SVMs
    experts are competed to tackle partitioned regions. Same as in Milidiu's work, for
5   each particular region only the SVMs expert that is the most adequate one is used for
    the final prediction. There are two rationales in the proposed method. First, as SOM
7   is an unsupervised clustering algorithm based on the competitive learning algorithm
    [11], the training data points which have similar characteristics in the input space
9   will be classified into the same region. As the partitioned regions have more uniform
    distributions than that of the whole input space, it will become easier for a SVMs expert
11 to capture such a more stationary input–output relationship. Second, different choices
    of the kernel function in SVMs will define different types of feature space resulting in
13 different solutions [29]. As different partitioned regions have different characteristics,
    by taking this architecture the SVMs experts that best fit particular regions by choosing
15 the most appropriate kernel function and the optimal learning parameters of SVMs will
    be used for the final prediction. This is very different from a single SVMs model that
17 learns the whole input space globally and thus cannot guarantee that each local input
    region is the best learned. The SVMs experts are illustrated experimentally by using
19 the sunspot data set, Santa Fe competition time series and the building data sets. The
    simulation shows that there is great improvement in prediction performance by using
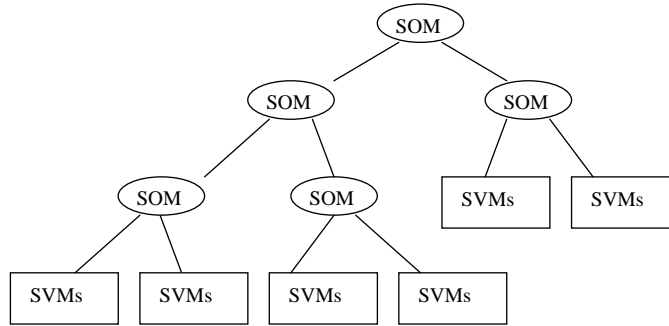
Fig. 4. A typical tree-structured architecture generated by the hybrid system. There are 5 non-terminal nodes located by SOM and 6 leaves located by SVMs experts.

1   the SVMs experts to learn the data. In addition, the SVMs experts also converge faster and use fewer support vectors.

3   This paper is organized as follows: Section 2 describes the detailed architecture of the SVMs experts. A learning algorithm is also developed in the same section. Section

5   3 presents the experimental results. Section 4 discusses the related work, followed by the conclusions drawn from this study in the last section.

7   **2. Architecture and the learning algorithm**

    The basic idea underlying the SVMs experts is to use SOM for partitioning the

9   whole input space into several regions and to use SVMs experts for solving these partitioned regions. As there is no prior knowledge about how many regions could

11   be partitioned from the whole input space, the tree-structured architecture proposed by [3,4] is adopted here for partition, which recursively partitions a large input space

13   into two regions until the partition condition is not satisfied. The main advantage of the tree-structured architecture is that by specifying a partition condition, it could

15   automatically find a suitable network structure and size for partitioning a large problem without predetermining the number of partitioned regions.

17   As illustrated in Fig. 4, each SOM sits at the non-terminal node of the tree and plays a "divide" role to heuristically partition a large input space into two regions, and then

19   each SVMs expert sits at the leave of the tree and plays a "conquer" role to tackle each partitioned region. For a data set ψ, a terminal node is created and located by the

21   data set. A SOM is developed to automatically partition the data set into two regions according to the input space of the data set. If the number of training data points in

23   the partitioned regions is both larger than a predetermined threshold value $N_{\text{threshold}}$ (partition condition), the terminal node for the data set becomes a non-terminal node,

25   and it is replaced with the SOM. Two new terminal nodes are created and located by the two regions. As a result, the data set is partitioned into two non-overlapping

27   regions ψ_1 and ψ_2, where $\psi_1 \cap \psi_2 = \phi$ and $\psi_1 \cup \psi_2 = \psi$ (ϕ denotes the null set). The aforementioned procedures are applied in the partitioned regions until the partition

condition that the number of training data points in the following partitioned regions is both larger than $N_{\text{threshold}}$ is violated in all the regions. Finally, all the terminal nodes of the tree become leaves, and they are located by the SVMs experts which are appropriately constructed to deal with each region.

A learning algorithm for the proposed architecture is outlined as follows.

(1) Create a terminal node and put the training data set at it. Set a minimum number of training data points $N_{\text{threshold}}$.

(2) Let ψ denotes the data set located at the terminal node. Present the input spaces of ψ (the data set ψ without outputs) to a SOM which will automatically partition ψ into two regions ψ_1 and ψ_2 ($\psi_1 \cap \psi_2 = \phi$ and $\psi_1 \cup \psi_2 = \psi$).

(3) Calculate the number of training data points in each region $\{N_i\}_{i=1}^2$. If $\{N_i\}_{i=1}^2$ is both larger than $N_{\text{threshold}}$, change the terminal node as a non-terminal node and locate it by the SOM. Create two new terminal nodes and locate them by the two regions ψ_1 and ψ_2. Otherwise, merge the two regions ψ_1 and ψ_2 and stop.

(4) Repeat from (2)–(3) until the partition cannot be proceeded in all the regions.

(5) Classify the validation set into the partitioned regions by the trained SOM based on the input space.

(6) Train SVMs experts for the partitioned regions. Choose the most adequate SVMs expert that produces the smallest error on the partitioned validation set for each partitioned region. Locate it at the terminal node of the tree.

For an unknown data point in testing, it is first classified into one of the partitioned regions by multiple SOM traversing path downs to leaves of the tree. Then its output is produced by the corresponding SVMs expert.

3. Experimental results

3.1. Sunspot data

The sunspot data set has long served as a benchmark and been well studied in the previous literature [5,7,8,17,18,27,30,32]. To make results comparable, this study uses the same experimental setup as used in [27,32]. The only difference is that in our experiment, the data points from 1921–1955 are used as the validation set to select the optimal parameters of SVMs. The details of the experimental setup are described in Appendix A.

Furthermore, the normalized mean square error (NMSE) is used to measure the performance of SVMs. The NMSE of the test set is calculated as follows:

$$\text{NMSE} = \frac{1}{\delta^2 n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2, \tag{1}$$

$$\delta^2 = \frac{1}{n-1} \sum_{i=1}^{n} (y_i - \bar{y})^2, \tag{2}$$

Table 1
The converged NMSE in sunspot data

Methods	NMSE
Single SVMs_Polynomial	0.1780
Single SVMs_Tangent	0.2065
Single SVMs Gaussian	0.2682
SVMs experts	0.1541
Benchmarks	0.28 [27]
	0.35 [32]

1   where *n* represents the total number of data points in the test set. $\hat{y}$ represents the predicted value. $\bar{y}$ denotes the mean of the actual output values.

3   The SOM software used is directly taken from Matlab5.3.1 neural network toolbox. In each of our used SOM, there are only two output neurons representing two cat-

5   egories. After training by randomly presenting the input spaces of the training data set, SOM automatically classifies the training data set into two regions according to

7   the winner neuron. The value of $N_{threshold}$ is chosen experimentally. The used value of $N_{threshold}$ and the number of partitioned regions are given in Appendix B, together

9   with the results for the other studied data sets. In addition, the number of training data points in each partitioned region and its inter-class distance are also illustrated in the

11 appendix. Obviously, the inter-class distance in each partitioned region is much smaller than that of the whole input space, demonstrating the clustering characteristic of SOM.

13 For training SVMs, the sequential minimal optimization algorithm solving the regression problem [21,22] is implemented in this experiment and the program is developed

15 by using VC^{++} language. The investigated kernel functions are restricted into three categories: the polynomial kernel, the Gaussian kernel and the two-layer tangent ker-

17 nel. Thus, for each partitioned region, three SVMs experts are firstly developed. For each SVMs expert, the optimal values of the kernel parameters, C and ε are chosen

19 based on the smallest error on the validation set. Then the SVMs expert with the kernel function, the kernel parameters, C and ε that produce the smallest error on the valida-

21 tion set is chosen as the final expert. To assure there is the best prediction performance in the single SVMs models, the validation set is also used to select the optimal kernel

23 parameters, C and ε.

The results of the SVMs experts and of the single SVMs models are given in Table 1.

25 The table shows that in the single SVMs models the polynomial kernel achieves better performance than the Gaussian kernel and the two-layer tangent kernel. The converged

27 NMSE by using the polynomial kernel is much smaller than the benchmarks reported in [27,32]. Furthermore, the SVMs experts could achieve a smaller NMSE than the

29 best single SVMs model by using the polynomial kernel.

Fig. 5(a) illustrates the predicted and actual values. The solid line is the actual value.

31 The thick solid line is the predicted value of the SVMs experts, and the dotted line is the predicted value of the best single SVMs model. From the figure, it can be observed

33 that the SVMs experts forecast more closely to the actual values than the best single
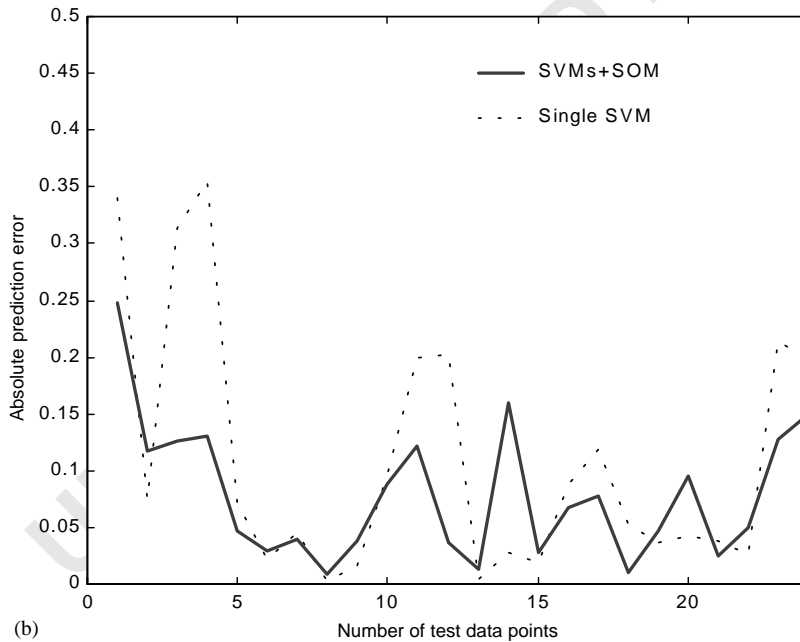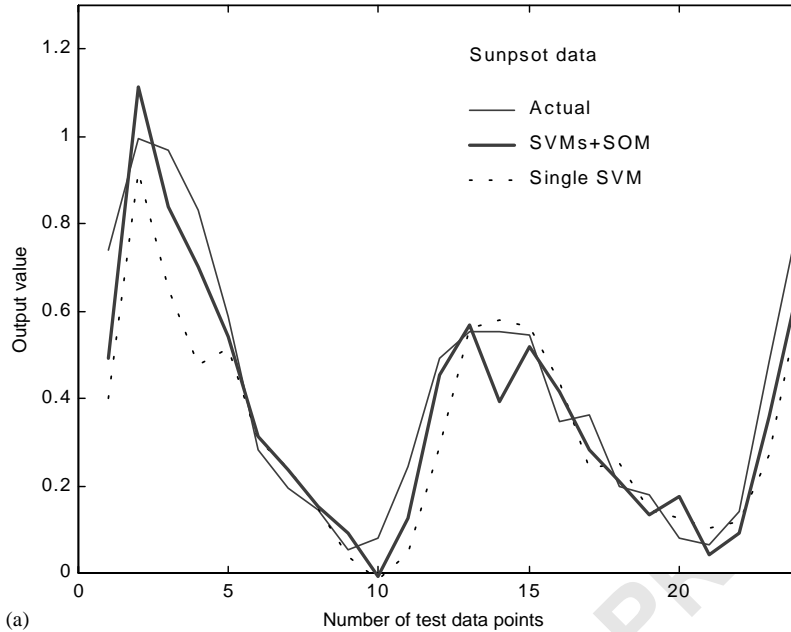
Fig. 5. (a) The predicted and actual values in sunspot data, (b) the absolute prediction errors in the SVMs experts and the best single SVMs model.

Table 2
The results in Santa Fe time series

Methods	Santa Fe-a (NMSE)	Santa Fe-c (NMSE)	Santa Fe-d (RMSE)
Single SVMs_Polynomial	0.6570	0.2249	0.0224
Single SVMs_Tangent	0.4895	0.6974	0.0259
Single SVMs_Gaussian	0.0188	0.2186	0.0212
SVMs experts	0.0061	0.1607	0.0188
Benchmarks	0.0073 [13]	0.2419 [13]	0.0418–0.0559 [15] 0.0596 [19]

SVMs model in most of the testing time period. So there are correspondingly smaller absolute prediction errors in the SVMs experts (the thick solid line) than the best single SVMs model (the dotted line), as illustrated in Fig. 5(b).

3.2. Santa Fe competition time series

The data sets A, C and D in Santa Fe competition which is held during the fall of 1990 under the auspices of the Santa Fe Institute [31] are also examined. In Santa Fe data sets A and C, the experimental setup is used as the same as in [13], which is given in Appendix C. There is no validation set, and the parameters of SVMs that produce the smallest NMSE on the test set are used for SVMs, as the same strategy as in [13]. In Santa Fe data set D, the experimental setup is adopted from [15,19]. For the data set D, the root mean square error (RMSE) is used to evaluate the performance of SVMs as this criterion is used in the previous studies [15,19]. The RMSE is calculated by

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}}, \tag{3}$$

where n and $\hat{y}$ have the same meaning as in (1).

The results of the SVMs experts and of the single SVMs models are given in Table 2. For these data sets, the Gaussian kernel performs best among the single SVMs models. The best single SVMs model by using the Gaussian kernel also has better result than the benchmark reported in [13] for the data set C and in [15,19] for the data set D. In the data set A, the best single SVMs model has slightly worse performance than the benchmark reported in [13]. However, among all the methods, the SVMs experts achieve the smallest test error in all the data sets.

Figs. 6(a)–8(a) illustrate the predicted and actual values in each data set. Obviously, the SVMs experts forecast more closely to the actual values than the best single SVMs model by using the Gaussian kernel in most of the testing time period. And there are correspondingly smaller absolute prediction errors in the SVMs experts than the best single SVMs model, as illustrated in Figs. 6(b)–8(b).

(a)

(b)

Fig. 6. (a) The predicted and actual values in Santa Fe-a, (b) the absolute prediction errors in the SVMs experts and the best single SVMs model.

Fig. 7. (a) The predicted and actual values in Santa Fe-c, (b) the absolute prediction errors in the SVMs experts and the best single SVMs model.

(a)

(b)
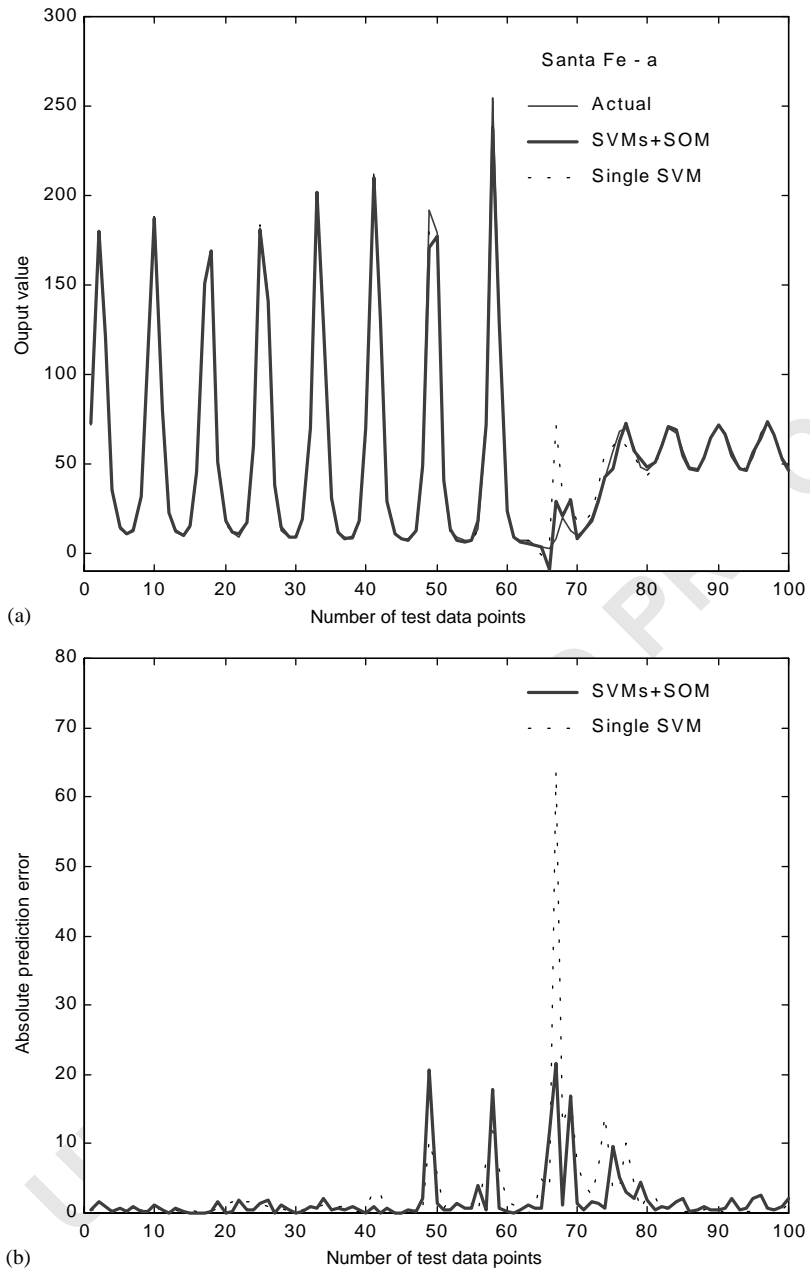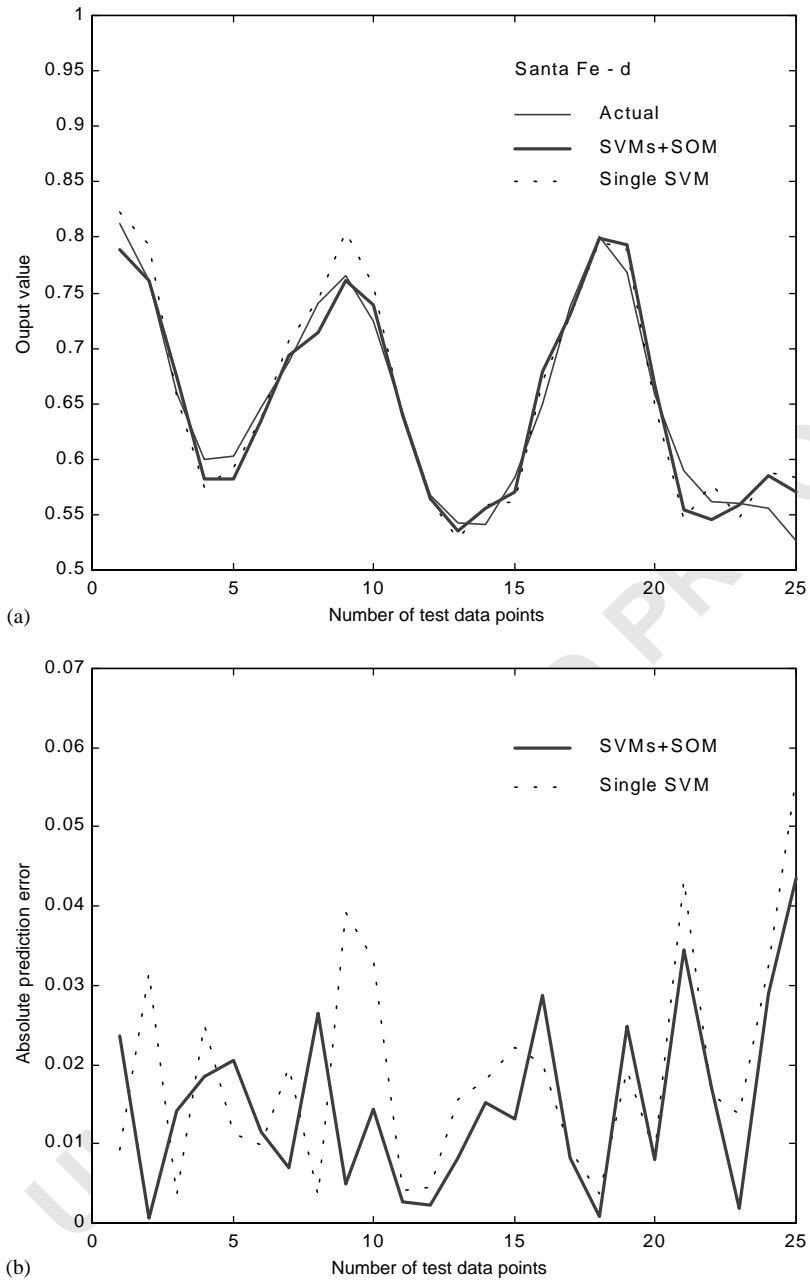
Fig. 8. (a) The predicted and actual values in Santa Fe-d, (b) the absolute prediction errors in the SVMs experts and the best single SVMs model.

Table 3
The converged CV in the building data sets

Methods	Building-1			Building-2
	WBE	CCW	HW	
Single SVMs_Polynomial	19.65	20.54	48.63	13.74
Single SVMs_Tangent	22.10	18.33	52.18	19.48
Single SVMs_Gaussian	18.05	12.94	34.07	6.11
SVMs experts	16.35	12.57	33.25	4.82
Benchmarks	WBE: 10.36–30.91; CCW: 11.65–33.26; HW: 15.24–66.45 ftp.cs.colorado.edu/pub/distribs/energy-shootout			Building 2: 2.75–18.21

WBE: whole building electricity; CCW: chilled cold water; HW: hot water.

3.3. Building data

The two building data sets is taken from the contest of "The Great Energy Predictor Shootout—the First Data Analysis and Prediction" which is organized from December 1, 1992 to April 30, 1993 in Denver, Colorado [20]. For the first building data set, three SVMs experts need to be developed, corresponding to the three dependent variables which are the whole building electricity, the hourly chilled water and the hot water. In both data sets, the last 600 data patterns are used as the validation set to select the optimal parameters of SVMs. Furthermore, the prediction performance of SVMs is evaluated by the coefficient of variation (CV) as this criterion is used in the competition. The criterion of CV is calculated by

$$CV = \frac{\sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}}}{\bar{y}}, \tag{4}$$

where n, $\hat{y}$ and $\bar{y}$ denotes the same meaning as in Eq. (1).

The results of the SVMs experts and of the single SVMs models are given in Table 3. Same as in the Santa Fe data sets, the Gaussian kernel has the best performance among the single SVMs models. The best results in the single SVMs models are among the results reported in the competition. Comparing the results of the SVMs experts with those of the best single SVMs model, it can be observed that the SVMs experts achieve a much smaller CV than the best single SVMs model.

In addition, the used CPU time and the number of converged support vectors in the SVMs experts and the best single SVMs model are also reported in Table 4, which are calculated for all the studied data sets. The table shows that the time spent to find the solution is largely less for in the SVMs experts than the best single SVMs model. Moreover, there are fewer support vectors in the SVMs experts than the best single SVMs model.

Table 4
The used CPU time and the number of support vectors

Data sets	SVMs experts		Best single SVMs model	
	CPU time (s)	# of SV	CPU time (s)	# of SV
Sunspot	53	82	93	98
Santa Fe-a	748	875	34573	899
Santa Fe-c	2	42	51	66
Santa Fe-d	221	1256	66994	1336
Building-1_WBE	344	2049	13688	2117
Building-1_CCW	90	1994	6591	2004
Building-1_HW	52	1977	9228	2046
Building-2	1292	1503	7731	1568

4. Related work

Our proposed method can be considered similar in spirit to the "local learning" algorithm proposed in [1]. In the local learning algorithm, for every test data point, a fixed number of training data points which are closest to it in the input space are found and used to train a neural network. The same amount of neural networks as that of the test data point is established in this method. In our proposed method, the similar test data points in the input space are firstly grouped together, and then the training data points which are closest to them in the input space are used to train a neural network. Thus, in our proposed method, the amount of neural networks that is required to be developed is reduced. The number of training data points for different test data points could vary according to the given training data set. When the number of test data points is reduced to one, our proposed method is equivalent to the local learning algorithm.

5. Conclusions

Based on the principle of "divide-and-conquer", a SVMs experts model is developed by combining SVMs with SOM using a two-stage architecture. In the first stage, multiple SOMs are used to classify a given input into one of the partitioned regions based on a tree-structured architecture. Then, at the second stage, the corresponding SVMs expert is used to produce the output.

There are several advantages in this hybrid system. First, it achieves high prediction performance because different input regions are separately learned by the most appropriate SVMs experts. Second, it allows efficient learning. The time complexity of training SVMs scales approximately between quadratic and cubic in the number of training data points [22]. With the number of training data points getting smaller in each SVMs expert, the convergence speed of SVMs is largely increased. Third, the SVMs experts converges to fewer support vectors. Thus, the solution can be repre-

1 sented more sparsely and simply. The SVMs experts model has been evaluated using an extensive amount of data sets. Its superiority is demonstrated by comparing it with
3 the single SVMs models. All the simulation results show that the SVMs experts model is more effective and efficient in forecasting noisy and non-stationary time series than
5 the single SVM model.

   Although this paper shows the effectiveness of the SVMs experts model, there are
7 more issues need to be investigated. Firstly, due to the "hard" decision used in the current partition, there is deterioration in the performance to some regions, which is also
9 illustrated in Figs. 5–8. The "soft" partition which allows the data to simultaneously lie in multiple regions may be more suitable in these regions. This should be investigated
11 in future work. Secondly, the experiment shows that the performance of the SVMs experts is influenced by the value of $N_{threshold}$. How to determine the optimal value of
13 $N_{threshold}$ is an important issue needs to be studied. Finally, in this study only three kernel functions are investigated. Future work needs to explore more useful kernel
15 functions for further improving the performance of the SVMs experts.

Acknowledgements

Appendix A

21   *Sunspot data*. The data set consists of a total of 280 yearly averaged sunspots recorded from 1700 to 1979. In the experiment, the data points from 1700 to 1920 are
23 used as the training set, and those from 1921 to 1955 are used as the validation set, and the remaining data points from 1956 to 1979 are used as the test set. 12 previous
25 sunspots are used as inputs to predict the current sunspot. So there are a total of 209 data patterns in the training set, 35 data patterns in the validation set and 24 data
27 patterns in the test set.

   Santa Fe data set A. This data set is a laser time series recorded from a Far-Infrared-
29 Laser in a chaotic state, which is approximately described by three coupled non-linear ordinary differential equations. The data set contains 1000 data points, followed by
31 a continued data set containing 9093 data points. The whole data set is used as the training set, and the first 100 data points in the continued data set are used as the test
33 set. 8 lagged data points are used as inputs to predict the current data point. So there are a total of 992 data patterns in the training set and 100 data patterns in the test set.
35   *Santa Fe data set C*. This data set is a tick-wise time record (tick-wise means that samples come at irregular intervals of time) of the high-frequency exchange rates
37 between the Swiss franc and the US dollar from August 7, 1990 to April 18, 1991. The data set contains 10 segments of 3000 data points each. The continued data set is a
39 record of exchange rates at the tick closest to the requested time during the same time

Table 5

Data sets	Sun spot		Santa Fe-a		Santa Fe-c		Santa Fe-d		Building-1		Building-2	
No. of regions	n	d	n	d	n	d	n	d	n	d	n	d
original	209	0.5767	992	122.57	103	1.2381	1880	1.0079	2326	675.19	1744	629.31
1	28	0.2634	36	34.37	21	0.1195	117	0.3992	102	17.52	35	37.53
2	27	0.3734	39	32.86	19	0.1051	106	0.4067	103	17.84	64	44.67
3	30	0.2719	36	21.82	22	0.1006	88	0.5508	85	14.06	69	65.92
4	25	0.3896	27	34.14	14	0.0462	99	0.4982	75	95.56	107	64.61
5	26	0.3620	32	30.36	14	0.1189	54	0.5275	101	78.48	63	128.59
6	23	0.2122	27	18.34	13	0.0423	47	0.5269	103	19.65	37	128.78
7	29	0.2347	33	29.00	—	—	48	0.5471	90	42.31	75	85.97
8	21	0.3241	35	35.36	—	—	41	0.4091	70	68.93	47	114.26
9	—	—	37	32.92	—	—	69	0.4685	87	89.94	52	192.12
10	—	—	38	33.49	—	—	56	0.3406	63	83.64	56	180.17
11	—	—	39	21.90	—	—	73	0.5812	72	102.60	38	192.01
12	—	—	28	44.77	—	—	106	0.3916	66	135.93	43	202.19
13	—	—	27	16.84	—	—	106	0.4744	57	104.76	91	106.60
14	—	—	33	29.99	—	—	53	0.2807	66	100.53	54	126.63
14	—	—	31	32.87	—	—	53	0.2908	73	109.85	49	169.67
16	—	—	33	33.17	—	—	45	0.2901	102	17.01	53	191.10
17	—	—	36	42.85	—	—	59	0.2923	102	17.06	48	169.45
18	—	—	26	28.26	—	—	60	0.3636	86	13.90	57	184.78
19	—	—	30	31.58	—	—	70	0.3582	85	13.91	76	161.58
20	—	—	30	31.16	—	—	100	0.4696	102	16.99	48	165.76
21	—	—	32	37.45	—	—	7	0.4782	101	16.99	69	147.21
22	—	—	35	23.98	—	—	67	0.4211	102	17.13	52	171.38
23	—	—	28	22.12	—	—	59	0.4550	102	17.12	46	194.26
24	—	—	24	53.95	—	—	57	0.4510	98	54.04	32	194.75
25	—	—	32	32.75	—	—	63	0.3883	95	68.09	48	185.14
26	—	—	30	24.59	—	—	65	0.5224	53	75.05	30	193.14
27	—	—	26	27.91	—	—	62	0.4333	85	57.65	58	195.26
28	—	—	38	36.03	—	—	—	—	—	—	31	207.33
29	—	—	31	38.52	—	—	—	—	—	—	47	175.92
30	—	—	29	24.16	—	—	—	—	—	—	58	228.84
31	—	—	34	23.69	—	—	—	—	—	—	48	228.77
32	—	—	—	—	—	—	—	—	—	—	63	210.53
33	—	—	—	—	—	—	—	—	—	—	—	—
$N_{threshold}$	21		24		13		41		53		30	

period, consisting of 60 data points. The daily closing exchange rates are used as the training set, thus reducing the whole time series from 30 000 to 111. 8 lagged exchange rates as well as the week day are used as inputs to predict the present exchange rate. So there are a total of 103 data patterns in the training set and 60 data patterns in the test set.

Santa Fe data set D. This data set is an artificial data generated from a nine-dimensional periodically driven dissipative dynamic system with an asymmetrical four-well

1   potential and a drift on the parameters. The whole data set contains 2 segments of each 5000 data points, followed by a continued data set containing 500 data points. In the
3   last 2000 data points of the data set, the first 1900 data points are used as the training set, and the remaining 100 data points are used as the validation set. In addition, the
5   first 25 data points in the continued data set are used as the test set. 20 lagged data points are used as inputs to predict the current data point. So there are a total of 1880
7   data patterns in the training set, 100 data patterns in the validation set, and 25 data patterns in the test set.

9   *Building data set 1*. This data set is a time record of whole building electricity, hourly chilled water and hot water usage for a four-month period in an institutional
11   building. The hourly usage of whole building electricity, hourly chilled water and hot water for the following two months is to be predicted. There are a total of 2926 data
13   patterns in the training set and 1282 data patterns in the test set. Each data pattern consists of 8 independent variables determined by a time stamp and weather data and
15   the three dependent variables. In this experiment, the training data set is sequentially divided into two parts: the first 2326 data patterns are used for training SVMs, and the
17   remaining 600 data patterns are used as the validation set.

     Building data set 2. This data set is a record of beam radiation during a six-month
19   period. There are a total of 2344 and 900 randomly ordered data patterns respectively in the training set and the test set. Each data pattern consists of 5 independent variables
21   including four solar radiation measurements and one decimal rate. In the experiment, the first 1744 data patterns in the training set are used for training SVMs, and the
23   remaining 600 data patterns are used for validating.

Appendix B

25   The number of partitioned regions, the number of training data points (n) and the inter-class distance (d) in each partitioned region, and the used $N_{\text{threshold}}$ in each data
27   set, as shown in Table 5.

References

29   [1] M.D. Bollivier, W. Eifler, S. Thiria, Sea surface temperature forecasts using on-line local learning algorithm in upwelling regions, Neurocomputing 30 (2000) 59–63.
31   [2] L.J. Cao, F.E.H. Tay, Financial forecasting using support vector machines, Neural Comput. Appl. 10 (2001) 184–192.
33   [3] K.L. Chen, P. Yang, X. Yu, H.S. Chi, A self-generating neural network architecture for supervised learning, Neurocomputing (1997) 33–48.
35   [4] K. Chen, X. Yu, H.S. Chi, Combining linear discriminant functions with neural networks for supervised learning, Neural Comput. Appl. 6 (1997) 19–41.
37   [5] M.B. Cottrell, Y. Girard, M. Mangeas, C. Muller, Neural modeling for time series: a statistical stepwise method for weight elimination, IEEE Trans. Neural Networks 6 (6) (1995) 1355–1364.
39   [6] N. Cristianini, J.S. Taylor, An Introduction to Support Vector Machines: and Other Kernel-Based Learning Methods, Cambridge University Press, New York, 2000.
41   [7] I. Ginzberg, D. Horn, Learning the rule of a time series, Int. J. Neural Systems 3 (2) (1992) 167–177.

[8] C.D. Groot, D. Wurtz, Analysis of univariate time series with connectionist nets: a case study of two classical examples, Neurocomputing 3 (1991) 177–192.

[9] R.A. Jacobs, M.A. Jordan, S.J. Nowlan, G.E. Hinton, Adaptive mixtures of local experts, Neural Comput. 3 (1991) 79–87.

[10] M.I. Jordan, R.A. Jacobs, Hierarchical mixtures of experts and the EM algorithm, Neural Computation 6 (1994) 181–214.

[11] T. Kohonen, Self-organization and Associative Memory, Springer, New York, 1989.

[12] T.J. Kwok, Support vector mixture for classification and regression problems, in: ICPR'98: Proceedings of the 14th International Conference on Pattern Recognition, 1998, pp. 255–258.

[13] R.L. Milidiu, R.J. Machado, R.P. Rentera, Time-series forecasting through wavelets transformation and a mixture of expert models, Neurocomputing 28 (1999) 145–146.

[14] S. Mukherjee, E. Osuna, F. Girosi, Nonlinear prediction of chaotic time series using support vector machines, in: NNSP'97: Neural Networks for Signal Processing VII: Proceedings of the IEEE Signal Processing Society Workshop, 1997, pp. 511–520.

[15] K.R. Muller, A.J. Smola, G. Ratsch, B. Scholkopf, J. Kohlmorgen, Using support vector machines for time series prediction, in: B. Scholkopf, C.J.C. Burges, A.J. Smola (Eds.), Advances in Kernel Methods—Support Vector Learning, 1999, pp. 243–254.

[16] K.R. Muller, J.A. Smola, G. Ratsch, B. Scholkopf, J. Kohlmorgen, V.N. Vapnik, Predicting time series with support vector machines, in: ICANN'97: Proceedings of the sevnth International Conference on Artificial Neural Networks, 1997, pp. 999–1004.

[17] J. Nie, Nonlinear time series forecasting: a fuzzy-neural approach, Neurocomputing 16 (1997) 63–76.

[18] S.J. Nowlan, G.E. Hinton, Simplifying neural networks by soft weight-sharing, Neural Comput. 4 (1992) 473–493.

[19] K. Pawelzik, K.R. Muller, J. Kohlmorgen, Annealed competition of experts for a segmentation and classification of switching dynamics, Neural Comput. 8 (1996) 340–356.

[20] L. Prechelt, PROBEN1—a set of neural network benchmark problems and benchmarking rules, Technical Report 21/94, University of Karlsruhe, Germany, 1994.

[21] A.J. Smola, Learning with kernels, Ph.D. Thesis, GMD, Birlinghoven, Germany, 1998.

[22] A.J. Smola, B. Schölkopf, A tutorial on support vector regression, NeuroCOLT Technical Report NC-TR-98-030, Royal Holloway College, University of London, UK, 1998.

[23] F.E.H. Tay, L.J. Cao, Application of support vector machines in financial time series forecasting, Omega 29 (4) (2001) 309–317.

[24] F.E.H. Tay, L.J. Cao, Modified support vector machines in financial time series forecasting, Neurocomputing, 2001, accepted for publication.

[25] F.E.H. Tay, L.J. Cao, A comparative study of saliency analysis and genetic algorithm for feature selection in support vector machines, Intell. Data Anal. 5 (3) (2001).

[26] F.E.H. Tay, L.J. Cao, Improved financial time series forecasting by combining support vector machines with self-organizing feature map, Intell. Data Anal. 5 (2001) 1–16.

[27] H. Tong, K.S. Lim, Threshold autoregressive, limit cycles and cyclical data, J. Roy. Statist. Soc. B 42 (3) (1980) 245–292.

[28] V.N. Vapnik, An overview of statistical learning theory, IEEE Trans. Neural Networks 10 (5) (1999) 988–999.

[29] V.N. Vapnik, The Nature of Statistical Learning Theory, Springer, New York, 2000.

[30] D.K. Wedding II, K.J. Cios, Time series forecasting by combining RBF networks, certainty factors, and the box-Jenkins model, Neurocomputing 10 (1996) 149–168.

[31] A.S. Weigend, N.A. Gershenfeld, Time Series Prediction: Forecasting the Future and Understanding the Past, Addison-Wesley, Reading, MA, 1992. ftp://ftp.santafe.edu/pub/Time-Series/Competition/

[32] A.S. Weigend, B.A. Huberman, D.E. Rumelhart, Predicting the future: a connectionist approach, Int. J. Neural Systems 1 (1990) 193–209.

[33] A.S. Weigend, M. Manageas, Analysis and prediction of multi-stationary time series, in: Neural Networks in Financial Engineering: Proceedings of the third International Conference on Neural Networks in the Capital Markets, 1995, pp. 597–611.

[34] A.S. Weigend, M. Manageas, A.N. Srivastava, Nonlinear gated experts for time series: discovering regimes and avoiding over-fiting, Int. J. Neural Systems 6 (4) (1995) 373–399.

1

3

5

Cao Lijuan finished her Ph.D. study in National University of Singapore. She is currently working in the Institute of High Performance Computing as a research fellow. Her research area centers on artificial intelligence methods such as neural networks, genetic algorithms, and support vector machines.