Function approximation model ensembles and their application to the simultaneous determination of sample categories and positions

Gao Daqi, Sun Xiaoning

Abstract: *This paper uses multiple approximation model ensembles to solve a multi-input multi-output learning task. An ensemble is on behalf of a specified class, and composed of several multi-input single-output (MISO) approximation models. An MISO model may be either a multivariable cubic polynomial, or a multivariable quartic polynomial, or a single-hidden-layer perceptron. The number of ensembles is equal to that of the existing classes, and all the members in an ensemble are trained only by the samples from the represented category. The ensemble in which all the members have the most identical viewpoint finally determines the label and position of one sample. The "most identical viewpoint" can be scaled by the corrected relative standard deviation. The proposed method is verified to be effective by a synthetic dataset.*

I. Introduction

The task of the classifier component proper of a full system is to use the feature vector provided by the feature extractor to assign the object to a category [1]. Similarly, the role of the function approximation model is to predict the position of an object on the existing curves using the given feature vector [2]. Therefore, different kinds of classifiers and approximation models come into being, including multivariable linear and polynomials [3], neural networks [4], support vector machines (SVMs) [5], and others. Correspondingly, different kinds of learning algorithms, such as least mean squared (LMS), back-propagation (BP), quadratic programming (QP), etc, are presented [3-5].

When classifying, we always choose one and only one from all the output units of the classifier to label an object, and in general, the unit with the largest output [3-4]. When estimating the position of an object, we must consider all the real values of the output units in an approximation model. In a robot control system, for instance, several such output units of an approximation model may express the sizes of the forward, upward, right, and speeded-up actions, respectively [6]. Therefore, an approximation model with multiple output units will give multiple predicted values for a feature vector, and we can not simply select one and abandon the others.

In the real life, however, we often need to simultaneously know which category a sample belongs to and what its "position" is. Although all the multiple outputs given by a multi-input multi-output (MIMO) system for a specified vector have definite physical meanings, we must select one and only one from among them. A typical application is about the simultaneous determination of odor classes and concentrations [7-8]. Suppose that there exist several kinds of odors and several concentrations in each, after a sample is sensed by a gas sensor array and a feature vector is thus got, the MIMO system needs to determine which kind of odor the sample belongs to and what its concentration is. Since it is so, which of the multiple outputs of the MIMO system ought to be selected? It is indeed a dilemma. It is meaningless to consider which of the aforementioned outputs to be the largest or to have the highest approximation accuracy.

The above problem is how to simultaneously determinate the classes and locations of samples if their physical significance is left aside. There are the following three approaches to try.

(A) Treat such a problem as an MIMO classification problem, one point for one class, and employ either an MIMO or multiple multi-input single-output (MISO) classifiers [9] to solve it. Suppose there are n_1 categories and n_2 points in each, $n_1 \times n_2$ output units are needed. When $n_1 \times n_2$ is very large, such classifiers are often complicated in structure. What can we do if a certain sample is just located at the midpoint of two training samples?

(B) Look upon such a problem first as an MIMO classification problem and then as multiple MISO approximation problems [10]. One of main drawbacks of the solution is that the decision boundaries are often too complicated to be realized, and thus the structure of the resulting models is quite complicated. What is more serious is that often the learning process of classifiers is not convergent.

(C) Regard such a problem as an MIMO approximation problem, and use either an MIMO or multiple MISO models to solve it [8]. Does an MISO model trained with the dataset from class ω_j have a predicted value for a certain sample from class ω_k? Yes! And the reverse is true. That is where the trouble lies. Therefore, the two solutions are not yet ideal in effect, because a single approximation model does not know to say "No".

Classifier ensembles came into being at the required moment, and different combination rules thus were presented in the 1990s [1]. Consequently, there is only one more kind of classifier for selection. That is all there is to it. When the simultaneous determination of categories and positions of samples is regarded as an MIMO approximation problem, however, the situation completely changes [8]. At the moment, the function approximation model ensembles become the only selection. Not only that, hardly can any one of the existing classifier combination strategies be directly applied to the function approximation model ensembles, and the "weak" components of combining classifiers will not be suitable to the approximation model ensembles.

This paper considers the simultaneous determination of sample categories and positions as an MIMO approximation problem. An MIMO problem is first decomposed into

This work is supported by the National Science Foundation of China (NSFC) under Grant No. 60275017, 60373073, 60575027, the high-tech development program of china (863) under Grant No. 2006AA10Z315; the Doctor Unit Foundation of the Education Ministry of China under Grant No. 20060251013, Shanghai Science Foundation of China under Grant No. 06ZR14026, and the open project program of the State Key Laboratory of Bioreactor Engineering

Daqi Gao is with the Department of Computer Science, East China University of Science and Technology, Shanghai 200237, China (phone: +86-21-64246970; fax: +86-21-64252830; email: gaodaqi@8163.net.cn).

Sun Xiaoning is with the Department of Computer Science, East China University of Science and Technology, Shanghai 200237, China (phone: +86-21-64253780; Fax: +86-21-64252830; email: sunxn@ecust.edu.cn).

multiple MISO problems, and multiple ensembles are then employed to solve them one by one. An ensemble consists of several MISO models or members, and represents a specified class. All the members in an ensemble are only trained by the samples from the represented class. If one pattern is indeed from a certain represented class, the viewpoints of all the members in the representative ensemble are quite identical; otherwise quite divergent. The ensemble with the most identical viewpoint will finally label the sample, and the average predicted value of all the experts in the ensemble will determine its position.

II. Approximation ensembles and combination strategies

2.1 *Combination strategies*

Let us take Fig. 1 as an example to illustrate how the approximation model ensembles and combination strategies work. There are three ensembles, which are on behalf of 3 curves, marked in red, blue and cyan, respectively. And there are three MISO models or *Experts* signed with '□', 'o', and '◊', respectively, in each. Our purpose is to predict which curve a specified sample $\boldsymbol{x}$ belongs to and what its position is. The prerequisite is that $\boldsymbol{x}$ can be assigned to one and only one curve. Each of the 9 experts will give its own prediction. At the moment we are unable to determine the label and location of $\boldsymbol{x}$ only depending upon a single expert, and it is also meaningless to only consider which of experts to have the highest prediction accuracy. With our visual senses, we can infer that $\boldsymbol{x}$ belongs to *Curve* 2, because it looks as if all the three experts in *Ensemble* 2 have the most identical viewpoint, and $\boldsymbol{x}$ may be equal to their average predicted value.

The above example illustrates the following facts. An ensemble is made up of several experts, and on behalf of a specified class. If there are n classes, there are n ensembles, one for one. In an ensemble, all the members, or the many-to-one models, are used to simultaneously estimate the position of $\boldsymbol{x}$. If $\boldsymbol{x}$ is indeed from a certain class, say ω_j, the outputs of all the members in *Ensemble j* will simultaneously be close in on a predicted point; otherwise quite divergent. The ensemble with the most consistent viewpoint ought to be taken to finally determine the label and place of $\boldsymbol{x}$.

The real outputs of *Ensemble j* (j=1, 2,…, n) for a given sample $\boldsymbol{x}_p \in R^m$ are the similar degree $\zeta_p^{(j)}$ to the represented class ω_j as well as the average prediction $\bar{y}_p^{(j)}$. In the learning stage, all the s experts in *Ensemble j* simultaneously fit the relationship between the training samples $X^{(j)}$ from ω_j and the expected outputs $\boldsymbol{d}_k^{(j)}$, namely $f: X^{(j)} \to \boldsymbol{d}_k^{(j)}$ (k=1, 2, …,s), only the samples from ω_j take part in training *Ensemble j*. In the test stage, the n existing ensembles will give n pairs of similar degrees and average predicted values $\left(\zeta_p^{(j)}, \bar{y}_p^{(j)}\right)$. If $\zeta_p^{(j)}$ is the smallest among all the similar degrees, we can decide that $\boldsymbol{x}_p$ belongs to ω_j and its location is $\bar{y}_p^{(j)}$.

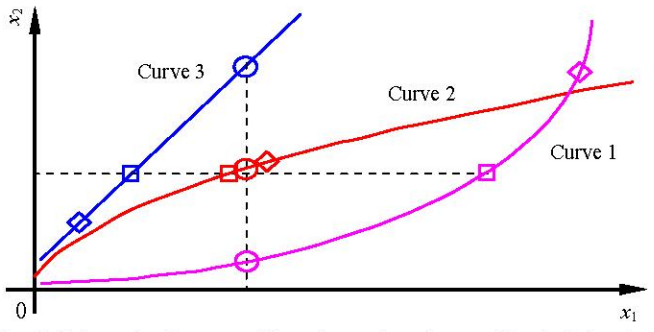


Fig. 1. Schematic diagram of three 3-membered ensembles to determine which curve a sample finally belongs to and what its place is.

The detailed learning process of approximation model ensembles and combination strategies are as follows.

(1) Decompose an MIMO approximation task $f: X \to \boldsymbol{D}$ into multiple MISO tasks $f: X^{(j)} \to \boldsymbol{d}_k^{(j)}$. Here $X \in R^{N\times m}$, $X^{(j)} \in R^{Nj\times m}$, $\boldsymbol{D} \in R^{N\times n}$, $\boldsymbol{d}_k^{(j)} \in R^{Nj\times 1}$, N is the total number of learning samples, and N_j only the number of training samples from ω_j. In fact, N_j is far smaller than N.

(2) Employ s experts, each with the structure of m-to-1, to form *Ensemble j* so as to fit the m-to-1 curve $f: X^{(j)} \to \boldsymbol{d}_k^{(j)}$.

(3) Predetermine the position of $\boldsymbol{x}_p$ in ω_j with *Ensemble j* according to the average combination rule:

$$\bar{y}_p^{(j)} = \frac{1}{s}\sum_{k=1}^{s} y_{pk}^{(j)} \tag{1}$$

Here, $y_{pk}^{(j)}$ is the predicted output of expert k in *Ensemble j*. The similar degree given by *Ensemble j* is

$$\rho_p^{(j)} = \frac{\sigma_p^{(j)}}{\bar{y}_p^{(j)}} = \frac{1}{\bar{y}_p^{(j)}}\sqrt{\frac{1}{s-1}\sum_{k=1}^{s}\left(y_{pk}^{(j)} - \bar{y}_p^{(j)}\right)^2} \times 100\% \tag{2}$$

As a matter of fact, $\rho_p^{(j)}$ is similar to the relative standard derivation (RSD) of the average prediction given by all the experts in *Ensemble j*. In consideration of two such cases that a very small average value, e.g. $\bar{y}_p^{(j)} = 0$, will unduly enlarge the RSD and a negative average value will lead to an incorrect comparison, Eq. (2) is revised as

$$\zeta_p^{(j)} = \frac{\sigma_p^{(j)}}{\left|\bar{y}_p^{(j)}\right| + \sqrt{\sigma_p^{(j)}}} \times 100\% = \frac{1}{\left|\bar{y}_p^{(j)}\right| + \sqrt{\sigma_p^{(j)}}}\sqrt{\frac{1}{s-1}\sum_{k=1}^{s}\left(y_{pk}^{(j)} - \bar{y}_p^{(j)}\right)^2} \times 100\% \tag{3}$$

Where $\sigma_p^{(j)}$ is the standard derivation of the predicted values given by all the members in *Ensemble j*. When $\bar{y}_p^{(j)} = 0$ and $\sigma_p^{(j)}$ =0.01, $\zeta_p^{(j)} = 10\%$. Such a prediction is relatively close to reality. If $\sigma_p^{(j)}$ is far smaller than $\bar{y}_p^{(j)}$, $\zeta_p^{(j)} \cong \sigma_p^{(j)} \Big/ \left|\bar{y}_p^{(j)}\right|$, and (3) is roughly equal to (2). Henceforth, $\zeta_p^{(j)}$ is called the corrected RSD (CRSD).

(4) Determine the final label and location of $\boldsymbol{x}_p$ by means of the minimum rule:

$$\boldsymbol{x}_p \in \omega_j \quad and \quad y_p = \bar{y}_p^{(j)} \qquad if \quad \zeta_p^{(j)} = \min_{1\le r\le n}\left(\zeta_p^{(r)}\right) \tag{4}$$

Equation (4) tells us that if $\zeta_p^{(j)}$ is the smallest among all the n CRSDs given by the n ensembles, $\boldsymbol{x}_p$ belongs to the represented category of *Ensemble j*, namely ω_j, and it is located at the average predicted value $\bar{y}_p^{(j)}$.

In form, the above combination rules used in function approximation are analogous to the synthesis of the average and the minimum rules used in classification. In fact, both of them are quite different from each other.

2.2 *Function approximation models and their learning*

Single-hidden-layer perceptrons (called MLPs hereafter) have good function approximation capacity, so they are good candidates. However, linear and quadric polynomial models are not suitable to be selected as candidates, because their approximation performance is not good enough yet. It is well-known that the over-fitting phenomenon [3] will come out if the order of a polynomial is unduly high. Not only that, a high-ordered polynomial in the high-dimensional space will result in too many awaiting parameters. Therefore, this paper proposes such a condition for a model to be chosen that it is able to approximate a single-periodic sine curve good enough. Therefore, we select multivariate cubic and quartic polynomials, called as MVCP and MVQP models later, as other two components. The later experiments will show that MVCP and MVQP models meet the above conditions, but one-variable cubic and quartic polynomials don't. If a many-to-one curve is very meandering, we will have to decompose it into several simpler segments so as to be approximated part by part. In that way, there are three members in each ensemble, namely an MVCP, an MVQP, and an MLP, all with the structure of single output.

Now let us have a simple analysis about the characteristics of the above three approximation models. When the structures and parameters of MLPs, say the weights and biases, are determined, their output values are always in a limited range, no matter how changeable the input variables may be. An MLP with the standard sigmoid activation function (SAF) $f(\varphi)=(1+\exp(-\varphi))^{-1}$, for example, always has its outputs in the range of (0, 1). However, the real outputs of MVCP and MVQP models are infinite in theory. If an MVCP is taken as a "positive" member, an MVQP is a "negative" member. And the reverse is true. This kind of difference is just used to make such a decision that a certain sample doesn't belong to the represented class.

2.2.1 *Multivariate cubic and quartic polynomial models*

MVCP j realizes the many-to-one mapping $f: \boldsymbol{X}^{(j)} \to \boldsymbol{d}_1^{(j)}$ in order to minimize the root-mean-square (RMS) error $\varepsilon_1^{(j)} = \sqrt{\frac{1}{2N_j}}\left\|\boldsymbol{d}_1^{(j)} - \boldsymbol{y}_1^{(j)}\right\|_2$. The real output vector of MVCP j for $\boldsymbol{X}^{(j)}$ is

$$\begin{aligned}\boldsymbol{y}_1^{(j)} &= \boldsymbol{d}_1^{(j)} + \boldsymbol{\varepsilon}_1^{(j)} = \left\{\left(X^{(j)}\right)^3,\ \left(X^{(j)}\right)^2,\ X^{(j)}\right\}\boldsymbol{\alpha}^{(j)} + \alpha_0^{(j)}\boldsymbol{1} + \boldsymbol{\varepsilon}_1^{(j)} \\ &= \widetilde{X}^{(j)}\widetilde{\boldsymbol{\alpha}}^{(j)} + \boldsymbol{\varepsilon}_1^{(j)} \end{aligned} \tag{5}$$

Here, $\boldsymbol{d}_1^{(j)} \in R^{N_j \times 1}$, $\boldsymbol{\alpha}^{(j)} \in R^{3m}$ is a coefficient vector, $\alpha_0^{(j)}$ a constant, $\boldsymbol{1}$ a vector of length N_j consisting solely of 1's, $\boldsymbol{\varepsilon}_1^{(j)}$ a residual error vector, and $\widetilde{X}^{(j)} = \{(X^{(j)})^3, (X^{(j)})^2, X^{(j)}, \boldsymbol{1}\} \in R^{N_j \times (3m+1)}$ the augmented matrix of $\boldsymbol{X}^{(j)}$. In the practical application, $\boldsymbol{X}^{(j)}$ is first normalized.

The $3m$+1-dimensional parameter vector $\widetilde{\boldsymbol{\alpha}}^{(j)} = \{\boldsymbol{\alpha}^{(j)},\ \alpha_0^{(j)}\}$ is determined by the pseudo-inverse method [3]

$$\widetilde{\boldsymbol{\alpha}}^{(j)} = \left(\left(\widetilde{X}^{(j)}\right)^T \widetilde{X}^{(j)}\right)^{-1}\left(\widetilde{X}^{(j)}\right)^T \boldsymbol{d}_1^{(j)} \tag{6}$$

The essential prerequisite for (6) to be used is that $\left(\widetilde{X}^{(j)}\right)^T \widetilde{X}^{(j)} \in R^{(3m+1)\times(3m+1)}$ is nonsingular. In order to surmount the singular difficulty of $\left(\widetilde{X}^{(j)}\right)^T \widetilde{X}^{(j)}$, this paper takes the following steps to determine $\widetilde{\boldsymbol{\alpha}}^{(j)}$.

(A) Normalize the original input matrixes $\boldsymbol{X}^{(j)} \in R^{N_j \times m}$ (j=1,…,n), i.e., the entry $x_{pi}^{(j)}$ in $\boldsymbol{X}^{(j)}$ is transformed into

$$x_{pi}^{(j)} \leftarrow \frac{x_{pi}^{(j)} - min\left(X^{(j)}\right)}{max\left(X^{(j)}\right) - min\left(X^{(j)}\right)} \tag{7}$$

Here, $max\left(X^{(j)}\right)$ and $min\left(X^{(j)}\right)$ are the maximum and minimum entries of $\boldsymbol{X}^{(j)}$, respectively.

(B) Bring about the augmented matrix $\widetilde{X}^{(j)} = \{(X^{(j)})^3, (X^{(j)})^2, X^{(j)}, \boldsymbol{1}\} \in R^{N_j \times (3m+1)}$.

(C) Add a noise matrix $0.0001N(\boldsymbol{\theta}, \boldsymbol{1})$ to $\{(X^{(j)})^3, (X^{(j)})^2, X^{(j)}\} \in R^{N_j \times 3m}$ except the constant column $\boldsymbol{1}$ consisting solely of 1's, if $\left(\widetilde{X}^{(j)}\right)^T \widetilde{X}^{(j)}$ is singular. Here $N(\boldsymbol{\theta}, \boldsymbol{1})$ is an $N_j \times 3m$ matrix with random entries chosen from a normal distribution with mean 0.0 and variance 1.0. $\widetilde{X}^{(j)}$ keeps unchanged if nonsingular.

(D) Calculate $\widetilde{\boldsymbol{\alpha}}^{(j)} \in R^{3m+1}$ according to (6).

The learning process of MVQP j is almost the same as that of MVCP j, except the augmented matrix becomes $\widetilde{X}^{(j)} = \{(X^{(j)})^4, (X^{(j)})^3, (X^{(j)})^2, X^{(j)}, \boldsymbol{1}\} \in R^{N_j \times (4m+1)}$, omitted here.

2.2.2 *Single-hidden-layer perceptrons*

Let the structure of MLP j be m-h-1, which works for the many-to-one mapping $f: \boldsymbol{X}^{(j)} \to \boldsymbol{d}_3^{(j)}$. Here, h is the number of hidden nodes. Every MLP uses the batch-learning BP algorithm to adjust its weights and biases. The learning process of MLP j is to minimize the RMS error $E_3^{(j)}(\tau) = \sqrt{\frac{1}{2N_j}}\left\|\boldsymbol{d}_3^{(j)} - \boldsymbol{y}_3^{(j)}(\tau)\right\|_2$ between $\boldsymbol{d}_3^{(j)}$ and the real output $\boldsymbol{y}_3^{(j)}(\tau)$. The iterative algorithm requires taking a weight vector $\boldsymbol{w}^{(j)}(\tau)$ at iteration τ and updating it as

$$\boldsymbol{w}^{(j)}(\tau+1) = \boldsymbol{w}^{(j)}(\tau) + \eta\Delta\boldsymbol{w}^{(j)}(\tau) \tag{8}$$

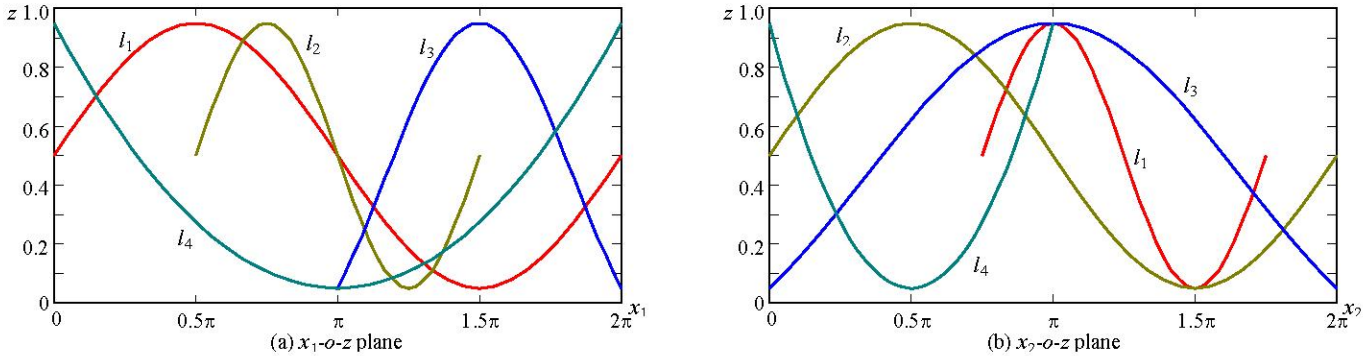
Here, η is a learning rate. It is shown in theory that variable scales and activation functions have a great influence on the convergence and generalization of MLPs [20]. In our experiment, the activation functions are $f(\varphi) = 3(1+\exp(-\varphi/3))^{-1}$ in the hidden and output layers, and all the input variables and the target values scaled in proportion to the ranges of [0.0, 6.0] and [0.05, 2.95], respectively [20].

III. Experimental Results-An Artificial Data

Four 3-dimensional curves are

$$l_1: \begin{cases} z = \frac{9}{20}\sin(x_1) + \frac{1}{2} \\ x_2 = \frac{3}{4}\pi + \frac{x_1}{2} \end{cases} \qquad x_1 \in [0, 2\pi] \tag{9}$$

$$l_2: \begin{cases} z = \frac{9}{20}\sin(x_2) + \frac{1}{2} \\ x_1 = \frac{x_2 + \pi}{2} \end{cases} \qquad x_2 \in [0, 2\pi] \tag{10}$$



(a) x_1-o-z plane (b) x_2-o-z plane

Fig. 2. Projection of four 3-dimensional curves on the x_1-o-z and x_2-o-z planes

$$l_3:\begin{cases} z = 0.9\dfrac{\sin(x_2-\pi)}{x_2-\pi}+0.05 \\ x_1 = \pi+\dfrac{x_2}{2} \end{cases} \qquad x_2 \in [0,2\pi] \qquad (11)$$

$$l_4:\begin{cases} z = 0.9\dfrac{(x_1-\pi)^2}{\pi^2}+0.05 \\ x_2 = \dfrac{x_1}{2} \end{cases} \qquad x_1 \in [0,2\pi] \qquad (12)$$

Let the sampling points be $(i\text{-}1)\pi/18$, i=1,2,…,37 for $x_1(i)$ in (9) and (12) or $x_2(i)$ in (10) and (11) for the training sets, and $(i\text{-}0.5)\pi/18$, i=1,2,…,36 for the test sets. Consequently, there are 4×37=148 samples in the original training set, and 4×36=144 samples in the test set. Figs. 2(a) and 2(b) show their projections on the x_1-o-z and x_2-o-z planes.

We select an MVCP, an MVQP and an MLP as the component units of each ensemble. Therefore, there are 3 members in every ensemble. While training a certain model, it is enough to only use the 37 samples from the represented class, and there is no need to use the original 148 samples.

According to Eqs. (9)-(12), each $\widetilde{X}^{(j)}$ is singular. For this reason, when determining the parameters of MVCR and MVQR models using the pseudo-inverse method, we have to added a noise matrix $0.0001N(\boldsymbol{\theta}, \boldsymbol{1})$ to $\widetilde{X}^{(j)}$, except the constant column $\boldsymbol{1}$ consisting solely of 1's. Table I gives the final parameters $\widetilde{\boldsymbol{\alpha}}^{(j)}$ (j=1,…,4) of the four MVCR and MVQR models for fitting the 4 curves. The range of the final weights is between -76.00 and 99.25, and the case of unduly large and small parameters is successfully avoided.

Let the number of hidden nodes in every MLP be h=6, the learning rate η=0.025, the maximum iteration step τ_{max}= 50000, and the allowable RMS error ε^*=0.025. The real iteration steps of MLPs are between 6910 and 50000, and the duration of time used are between 0.533 and 3.823 *sec* (given by a PC with 2.6G CPU, 256M RAM, the same below).

Figures 3 to 6 give the predicted results of four 3-membered ensembles for the four curves expressed by (9) - (12). For the purpose of clarity, the predicted values in the figures are forced to be 1.0 if over 1.0, and 0.0 if below 0.0. In fact, the largest predicted value even comes up to 32.3494, which is given by the MVCP expert in *Ensemble* 3 for predicting the 36th sample in l_2; and the smaller predicted value is lower to -40.1283, given by the MVQP model in *Ensemble* 2 for predicting the 36th sample in l_4. Relatively speaking, MVCPs are poor in approximation capability among the three models. Figs. 3 to 6 clearly and forcefully support the following viewpoint. When one sample is really from a certain represented class, the predicted outputs of all experts in the representative ensemble will be quite identical; otherwise quite divergent. According to Figs. 4(d) and 5(d), the predicted CRSDs for a small part of the test samples in l_2 and l_3 given by *Ensemble* 4 maybe are close to that given by *Ensembles* 2 and 3, as shown by dashed circles. Table II gives the detailed decision process for 2 probably confusable samples. Take the 6th sample in l_2 as an example. The predicted STD given by *Ensemble* 4 is 0.0105, which is smaller than that given by *Ensembles* 2, namely 0.0170. However, the predicted CRSD given by the latter is 1.6628%, less than half of that given by the former, or 3.8021%. We can thus make such a decision that the sample belongs to l_2 and its predicted value is 0.8942, a little larger than its target value 0.8686. Obviously, the decision is correct. Figs. 3 to 6 still verify that MVCP and MVQP models constitute a good complemented pair each other. According to Figs. 3-4,

TABLE I

FINAL PARAMETERS $\alpha_i^{(j)}$ OF MVCR AND MVQR MODELS FOR THE 4 CURVES

Model j		Curve	$\alpha_8^{(j)}:\left(x_2^{(j)}\right)^4$	$\alpha_7^{(j)}:\left(x_1^{(j)}\right)^4$	$\alpha_6^{(j)}:\left(x_2^{(j)}\right)^3$	$\alpha_5^{(j)}:\left(x_1^{(j)}\right)^3$	$\alpha_4^{(j)}:\left(x_2^{(j)}\right)^2$	$\alpha_3^{(j)}:\left(x_1^{(j)}\right)^2$	$\alpha_2^{(j)}:x_2^{(j)}$	$\alpha_1^{(j)}:x_1^{(j)}$	$\alpha_0^{(j)}$
MVCR	1	l_1	-	-	10.0567	0.4640	-20.4171	20.4594	-11.8978	18.6492	-9.4648
	2	l_2	-	-	44.9373	4.4930	23.1483	-29.3865	-49.047	19.7230	10.5442
	3	l_3	-	-	-76.0044	9.4290	-13.8606	28.3502	67.7030	5.25436	-20.9140
	4	l_4	-	-	0.0020	-0.0115	3.5807	0.0690	-2.5464	-2.1041	0.9498
MVQR	1	l_1	2.7208	-41.0150	15.6637	15.3688	0.9503	-46.8771	8.8977	29.9179	-4.1968
	2	l_2	42.3513	-2.1957	-58.3746	11.2377	-44.0759	3.3340	5.5198	17.5262	2.5484
	3	l_3	99.2464	50.7233	37.5308	-37.5315	-35.5395	31.7711	19.0891	-15.3316	-0.0375
	4	l_4	-0.0265	0.1810	-0.0283	0.4301	3.1875	1.6028	-3.1398	-0.9187	0.9503

TABLE II
PROCESS OF DECISION MAKING USING FOUR 3-MEMBERED ENSEMBLES FOR 2 TEST SAMPLES IN THE 4 CURVES

Original number and class	Practical value	Ensemble						Decision making
		No.	MVCP	MVQP	MLP	Predicted value	STD	RSD (%)	
No. 6 in l_2	0.8686	2	0.9076	0.8999	0.8750	0.8942	0.0170	**1.6628**	To l_2
		4	0.1804	0.1624	0.1810	0.1746	0.0105	3.8021	
No. 12 in l_3	0.7690	3	0.7740	0.7715	0.7561	0.7672	0.0097	**1.1184**	To l_3
		4	0.1634	0.1390	0.1233	0.1419	0.0202	7.1194	

MVCP and MVQP models are quite ideal candidates for approximating a single-periodic sine curve in the 3-dimensional space and above.

Let us pay a serious attention to a smaller average prediction value appeared in Fig. 5(c). For the 1st sample in l_3, the average predicted value and the standard deviation given by *Ensemble* 3 are $\bar{y}_1^{(3)}$=0.0587 and $\sigma_1^{(3)}$=0.0317, respectively, so $\sigma_1^{(3)} / \bar{y}_1^{(3)}$ = 54.01%, $\sigma_1^{(3)} / \left(\bar{y}_1^{(3)} + \sigma_1^{(3)}\right)$ =35.07%, and $\sigma_1^{(3)} / \left(\bar{y}_1^{(3)} + \sqrt{\sigma_1^{(3)}}\right)$ =13.39%. The CRSDs given by the other 3 ensembles for the same sample with (3) are 124.81%, 86.44%, and 31.75% in order. Consequently, the sample is still assigned to l_3, and located at 0.0587, a little smaller than its target value 0.0757. Such a predicted result is relatively close to reality, see Fig. 5 for details.

As a result, the final correct rate given by the proposed approximation model ensembles is 100% for simultaneously determining the labels and position of all the test samples. Such a predicted accuracy is quite satisfactory.

To regard the problem as an MIMO classification problem won't work, because that kind of method can only label but not locate samples. If the problem is wholly taken as an MIMO approximation problem, none of MIMO models is itself able to solve it yet, because we don't know which of the multiple outputs to be selected. For the purpose of comparison, let's try to look upon it first as an MIMO classification problem and then as 4 MISO approximation ones. We employ an MIMO MLP and 4 MISO MLPs to solve the MIMO classification problem respectively, and the follow-up MISO approximation models are the same as that used in the aforementioned ensembles. The structure and learning parameters of MLPs are h=6, η=0.01, τ_{max}=100000, and ε^*=0.10. And furthermore, we use 4 MISO SVMs to solve the classification problem, and then 4 MISO SVMs to solve the approximation problem. In SVMs, the width factor γ=0.25, the insensitivity constant ε=0.005, and the capacity constant C=1000 [5, 21]. The duration of learning time of MLPs is the longest, bur the storage requirement of SVMs is the largest. SVMs have even to store about 675 equivalent samples, which is discouraging. An SVM is originally a good candidate, but finally got rid of for the sake of its undue storage requirement. Obviously, the proposed approximation models and combination strategies have advantages on such aspects as small training subsets, low storage requirement, short learning time, and high predicted accuracy.

IV. Conclusions

This paper regards the simultaneous determination of sample classes and positions as a MIMO approximation problem, and employs multiple approximation model ensembles to solve it. An ensemble consists of several MISO approximation models, and is trained only by the samples from the represented class. The ensemble with the most identical view, namely the minimum CRSD, finally determines the label and position of a sample. Whether or not an approximation model is suitable to be selected as a component part of ensembles depends upon its capability to approximate a single-periodic sine curve. The experimental result for a synthetic dataset shows that the proposed approximation model ensembles and combination strategies are quite effective for simultaneously estimating the classes and positions of samples.

References

[1] J. Kittier, M. Hatef, R.P.W. Duin, et al, On combining classifiers, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1998, 20(3): 226-239.

[2] W. David, D. Lngram, D. Mclean, et al, On global-local artificial neural networks for function approximation, *IEEE Transactions on Neural Networks*, 2006, 17(4): 942-952.

[3] R.O. Duda, P. E. Hart, D. G. Stork, *Pattern classification*, John Wiley & Sons, Inc, New York, 2000.

[4] C.M. Bishop, Neural networks for pattern recognition, *Clarendon Press, Oxford*, 1995.

[5] V.N. Vapnik, The nature of statistical learning theory, *Springer-Verlag, New York*, 2000.

[6] S.S Ge, C. Wang, Adaptive neural control of uncertain MIMO nonlinear systems, *IEEE Transactions on Neural Networks*, 2004, 15: 674-692.

[7] E. Liobet, J. Brezmes, X. Vilanova, et al, Qualitative and quantitative analysis of volatile organic compounds using transient and steady-state responses of a thick-film tin oxide gas sensor array, *Sensors and Actuators B*, 1997, 41(1): 13-21.

[8] G. Daqi, C. Wei, Simultaneous estimation of odor classes and concentrations using an electronic nose with function approximation model ensembles, *Sensors and Actuators B*, 2007, 120(2): 584-594.

[9] B.C. Eduardo, A.D. Nancy, MIMO SVMs for classification and regression using the geometric algebra framework, In: *Proceedings of the 2005 International Joint Conference on Neural Networks* (IJCNN' 05), 895-900, Jul 31-Aug 4, 2005, Montreal, Quebec, Canada.

[10] M.D. Nam, T.C. TRanh, Approximation of function and its derivations using radial basis function networks, Applied Mathematical Modelling, 2003, 27: 197-220.

[11] G. Daqi, Y. Genxing. Influences of variable scales and activation functions on the performances of multilayer feedforward neural networks. *Pattern Recognition*. 2003, 36(4): 869-878.

[12] S. Ferrari, R.F. Slengel, Smooth function approximation using neural networks, *IEEE Transactions on Neural Networks*, 2005, 16(1): 24-38.

[13] T.C. Pearce, S.S. Schiffman, H.T. Nagle, J.W. Gardner (Eds.), Hand of Machine Olfaction, Wiley-VCH Press, 2003.

[14] L.K. Hansen, P. Salamon, Neural network ensembles, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1990, 12(10): 993-1001.

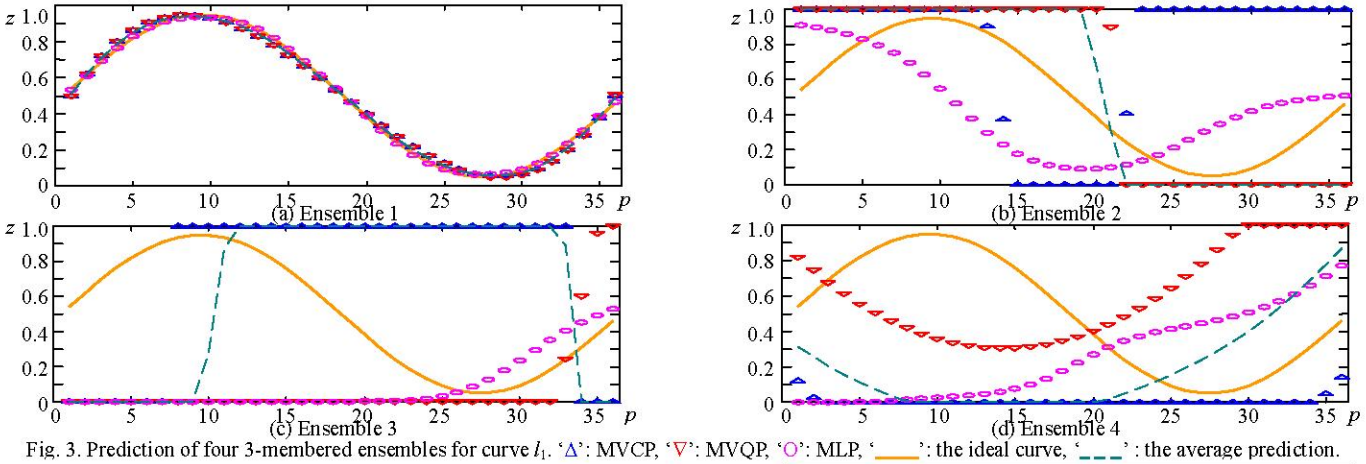


Fig. 3. Prediction of four 3-membered ensembles for curve l_1. 'Δ': MVCP, '∇': MVQP, 'O': MLP, '____': the ideal curve, '- - -': the average prediction.

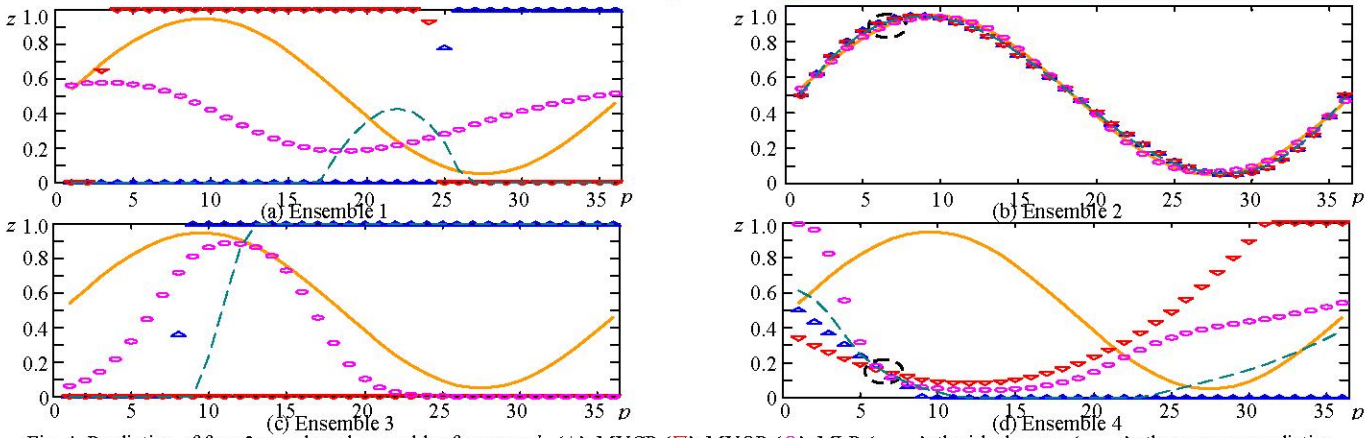


Fig. 4. Prediction of four 3-membered ensembles for curve l_2. 'Δ': MVCP, '∇': MVQP, 'O': MLP, '____': the ideal curve, '- - -': the average prediction.

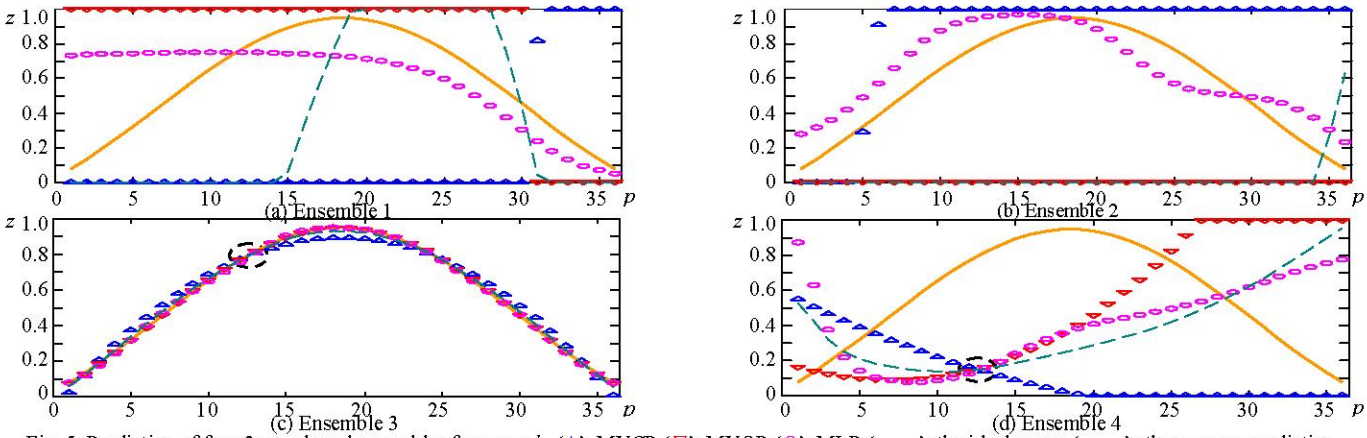


Fig. 5. Prediction of four 3-membered ensembles for curve l_3. 'Δ': MVCP, '∇': MVQP, 'O': MLP, '____': the ideal curve, '- - -': the average prediction.

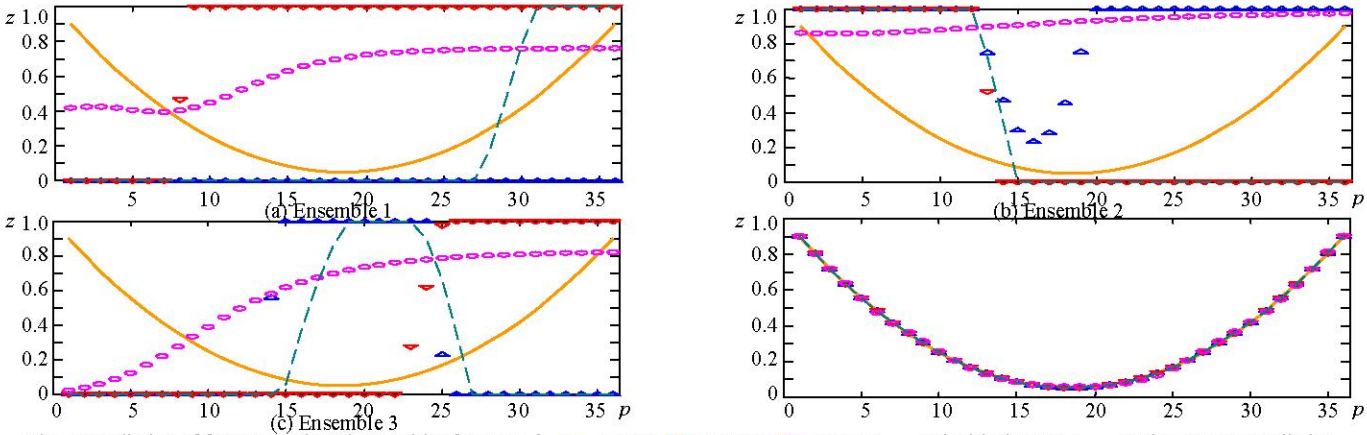


Fig. 6. Prediction of four 3-membered ensembles for curve l_4. 'Δ': MVCP, '∇': MVQP, 'O': MLP, '____': the ideal curve, '- - -': the average prediction.