

Metamodel selection based on stepwise regression

XiaoJian Zhou¹ · Ting Jiang²

Received: 14 November 2014 / Revised: 2 January 2016 / Accepted: 22 February 2016 / Published online: 25 April 2016
© Springer-Verlag Berlin Heidelberg 2016

Abstract Metamodels are often used to replace expensive simulations of engineering problems. When a training set is given, a series of metamodels can be constructed, and then there are two strategies to deal with these metamodels: (1) picking out the best one with the highest accuracy as an approximation of the computationally intensive simulation; and (2) combining all of them into an ensemble model. However, since the choice of approximate model depends on design of experiments (DOEs), employing of the first strategy thus increases the risk of adopting an inappropriate model. Nevertheless, the second strategy also seems not to be a good choice, since adding redundant metamodels may lead to loss of accuracy. Therefore, it is a necessary step to eliminate the redundant metamodels from the set of the candidates before constructing the final ensemble. Illuminated by the method of variable selection widely used in polynomial regression, a metamodel selection method based on *stepwise regression* is proposed. In our method, just a subset of n ones ($n \leq p$, where p is the number of all of the candidate metamodels) is used. In addition, a new ensemble technique is proposed from the view of polynomial regression in this work. This new ensemble technique, combined with metamodel selection method, has been evaluated using

six benchmark problems. The results show that eliminating the redundant metamodels before constructing the ensemble can provide more ideal prediction accuracy than directly constructing the ensemble by utilizing all of the candidates.

Keywords Metamodel · Surrogate · Ensemble · Stepwise · Design of experiment

1 Introduction

With the continuing updating of CPU and escalation of memory, the computer processing power has been drastically increased, but the computational cost of complex high-fidelity engineering simulations often makes it impractical to exclusively rely on simulation for design optimization (Jin et al. 2001). In order to reduce the computational cost, metamodels are used to replace the expensive simulation models (Queipo et al. 2005; Viana et al. 2010). The most popular metamodels include polynomial model, Kriging (Cressie 1988; Sacks et al. 1989), radial basis functions (RBF) (Fang et al. 2006), support vector regression (SVR), multivariate adaptive regression splines (MARS) (Friedman 1991), least interpolating polynomials (De Boor and Ron 1990), inductive learning (Langley and Simon 1995), and so on.

If only one single predictor is desired, there are two strategies for us to obtain the final prediction metamodel. One is the selection of the most accurate metamodel, which can be done using cross validation (Picard and Cook 1984; Kohavi et al. 1995); The other is the combination of all of the candidate metamodels: Zerpa et al. (2005) proposed a method to determine weight coefficients using prediction variance; Goel et al. (2007) proposed a heuristic method for calculating the weight coefficients, which is known as

✉ Ting Jiang
jiangtinghaha@126.com

XiaoJian Zhou
xjzhou@njupt.edu.cn

¹ School of Management, Nanjing University of Posts and Telecommunications, Nanjing, China

² School of Information Management, Nanjing University, Nanjing, China

PRESS (predicted residual sum of squares) weighted average metamodel; Acar and Rais-Rohani (2009) proposed a combining technique with optimized weight coefficients, which were obtained by solving an optimization problem; Viana et al. (2009) got the solution of the weight via Lagrange multipliers, and replaced the real error covariance matrix C with cross-validation error matrix.

Because the choice of metamodel depends on design of experiments (DOEs), selection strategy thus increases the risk of adopting an inappropriate model. Alternatively, since adding inaccurate metamodellers may lead to loss of accuracy, combination of all the candidate metamodellers also seems not to be a good choice. Therefore, eliminating the redundant metamodellers from the set of the candidates before constructing the ensemble is a necessary step. One of the objectives in this paper is to insert a step, where the redundant metamodellers are removed, before constructing the final ensemble of metamodellers.

Illuminated by the method of variable selection widely used in polynomial regression, a metamodel selection method based on *stepwise regression* is proposed. In this method, just a subset of n ones ($n \leq p$, where p is the number of all of the candidate metamodellers) is used. Although a subset of metamodellers is also DOE-sensitive (the choice of a subset depends on DOE), using of a subset of metamodellers appears to be a more “insure strategy” than selecting one of the “most accurate” metamodel, because there is not less than one individual metamodel used in the construction of the ensemble. Nevertheless, selection of the most accurate metamodel from the candidates only utilizes one metamodel.

The remainder of this paper is organized as follows. In the next section, we present the commonly-used ensemble techniques based on cross validation. In Section 3, the stepwise metamodel selection method is proposed. The arrangement of the experiment is presented in Section 4. The presentation and discussion of results for benchmark functions are displayed in Section 5. And the test for metamodel-based optimization is taken in Section 6. At last, a summary of several important conclusions is discussed in Section 7.

2 Ensemble of metamodellers

2.1 BestPRESS

The traditional method of using an ensemble of metamodellers is to select the best metamodel among all of the considered models. However, once the choice is made, the metamodel is fixed even though the design of experiments is changed. If the choice is refined for each new DOE, we can include it in the strategies for multiple metamodellers, where the model

with least error is assigned a weight of one and all others are assigned zero weight. Just as many works do, we also call this strategy BestPRESS, which corresponds to the selection strategy, i.e. the first strategy, mentioned above.

2.2 Heuristic computation of the weight coefficient

Goel et al. (2007) proposed a heuristic method for calculating the weight coefficients, which is known as PRESS weighted average metamodel, where the weight coefficients are computed as:

$$w_i = w_i^* / \sum_{i=1}^M w_i^*, w_i^* = (E_i + \alpha E_{avg})^\beta, \\ E_{avg} = \frac{1}{n} \sum_{i=1}^n E_i, \beta < 0, \alpha < 1 \quad (1)$$

where E_i is the PRESS error of the i th metamodel, α , β are used to control the importance of averaging and individual PRESS (predicted error sum of squares) respectively. Goel et al. (2007) suggested $\alpha = 0.05$, $\beta = -1$. PRESS is actually the leave-one-out (LOO) cross validation error. We firstly construct a metamodel by leaving the j th sample out from training, and then predict the response y'_j at the j th sample. Finally, we can get the PRESS, $\sum_{j=1}^N (y_j - y'_j)^2$, where N is the number of samples and y_j is the real response.

2.3 Computation of the weights for minimum cross validation (especially LOO cross validation)

The optimal weighted metamodel is obtained by minimizing the MSE as:

$$\min_{\mathbf{w}} \text{MSE} = \mathbf{w}^T \mathbf{C} \mathbf{w} \quad (2)$$

$$\text{s.t. } \mathbf{1}^T \mathbf{w} = 1$$

Using Lagrange multipliers, the solution is obtained as:

$$\mathbf{w} = \frac{\mathbf{C}^{-1} \mathbf{1}}{\mathbf{1}^T \mathbf{C}^{-1} \mathbf{1}} \quad (3)$$

where \mathbf{C} is approximated by the vectors of cross validation errors, e , especially LOO, which are obtained by using $n - 1$ samples to construct a metamodel and to predict the rest one sample. The deviation between true value and the predicted value is LOO cross validation error.

$$c_{ij} \simeq \frac{1}{N} e_i^T e_j \quad (4)$$

and N is the number of data points and the i and j indicate different metamodellers.

Table 1 Parameters used in Hartman function with six variables

i	a_{ij}					c_i	p_{ij}						
1	10.0	3.0	17.0	3.5	1.7	8.0	1.0	0.1312	0.1696	0.5569	0.0124	0.8283	0.5886
2	0.05	10.0	17.0	0.1	8.0	14.0	1.2	0.2329	0.4135	0.8307	0.3736	0.1004	0.9991
3	3.0	3.5	1.7	10.0	17.0	8.0	3.0	0.2348	0.1451	0.3522	0.2883	0.3047	0.6650
4	17.0	8.0	0.05	10.0	0.1	14.0	3.2	0.4047	0.8828	0.8732	0.5743	0.1091	0.0381

Furthermore, Viana et al. (2009) enforced the weight positive by solving (3) using only the diagonal elements of \mathbf{C} . The approach is named OWS_{diag} .

According to the classification in Zhou et al. (2011), the ensemble of metamodels can be classified into two groups: some are based on cross-validation; and the others are based on prediction mean square error. All of the above-mentioned ensemble techniques belong to the first group. The second group is not considered in this paper, because the second group of ensemble techniques requires additional test samples. When the number of samples is not large enough, extracting a part of samples for test from the sample set means that the number of samples used to construct the ensemble of metamodels decrease, and thereby the second group of methods greatly impair the prediction capabilities of ensemble of metamodels.

3 Stepwise metamodel selection

Just as polynomial regression model, we can propose an ensemble of model like this:

$$y = \beta_0 + \beta_1 f_1(\mathbf{x}) + \dots + \beta_p f_p(\mathbf{x}) + \varepsilon \tag{5}$$

Furthermore, if we replace $f_1(\mathbf{x})$ with $M_1(\mathbf{x})$, ..., $f_p(\mathbf{x})$ with $M_p(\mathbf{x})$ respectively, then (5) becomes

$$y = \beta_0 + \beta_1 M_1(\mathbf{x}) + \dots + \beta_p M_p(\mathbf{x}) + \varepsilon \tag{6}$$

where $M_1(\mathbf{x})$, ..., $M_p(\mathbf{x})$ denote individual metamodels. Therefore, (6) can be regarded as a kind of ensemble of metamodels.

Table 2 Summary of training and test data used in each benchmark problem

Benchmark problem	Design variables	Training sets	Design point	Test point
Camelback	2	500	25	441
Hartman	6	500	50	441
ER-9	9	500	55	441
ER-12	12	500	60	441
ER-15	15	500	70	441
DP-15	15	500	70	441

Given a dataset, $S\{X, Y\}$, consisting of N input-output data pairs $\{(\mathbf{x}^{(1)}, y^{(1)}), \dots, (\mathbf{x}^{(N)}, y^{(N)})\}$, where y is the metamodel response at the design sample point, \mathbf{x} , and N is the total number of metamodel runs, the linear regression model becomes a $N \times (p + 1)$ system of equations:

$$\begin{pmatrix} y^{(1)} \\ \vdots \\ y^{(N)} \end{pmatrix} = \begin{pmatrix} 1 & M_1(\mathbf{x}^{(1)}) & \dots & M_p(\mathbf{x}^{(1)}) \\ \vdots & \vdots & \ddots & \vdots \\ 1 & M_1(\mathbf{x}^{(N)}) & \dots & M_p(\mathbf{x}^{(N)}) \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{pmatrix} + \begin{pmatrix} \varepsilon^{(1)} \\ \vdots \\ \varepsilon^{(N)} \end{pmatrix} \tag{7}$$

where $M_i(\mathbf{x}^{(j)})(i = 1, \dots, p; j = 1, \dots, N)$ is the prediction value of the i th metamodel at the j th design sample point. In the process of construction of the i th metamodel, the j th design sample point is left out from training, and the established metamodel is used to predict the value at the omitted j th design sample point. That is, the samples

$$\{(\mathbf{x}^{(1)}, y^{(1)}), \dots, (\mathbf{x}^{(j-1)}, y^{(j-1)}), (\mathbf{x}^{(j+1)}, y^{(j+1)}), \dots, (\mathbf{x}^{(N)}, y^{(N)})\}$$

can be used to construct the metamodels $M_1(\cdot), \dots, M_p(\cdot)$, and then these metamodels are used respectively to predict the values at $\mathbf{x}^{(j)}$, finally the predictions $M_1(\mathbf{x}^{(j)}), \dots, M_p(\mathbf{x}^{(j)})$ are obtained. We do this process on all of the samples, then we get the whole matrix of (7). This process is something like LOO cross validation. Finally, the parameter $\beta = (\beta_0, \beta_1 \dots \beta_p)^T$ can be calculated using *least squares* method.

There are two purposes for using (6) in this paper: (1) We use it to eliminate the redundant metamodels; and (2) We use it as an ensemble. Nevertheless, if we directly use the ensemble (6) to predict the unknown design point, the redundant metamodels would impair the performance of the ensemble.

Stepwise regression is a systematic method for adding and removing terms from a multilinear model based on their statistical significance in a regression. In this paper, we treat metamodels as terms (or variables) and use the *stepwise regression* method to reject the redundant individual metamodels.

Before presenting how *stepwise regression* works to eliminate the redundant metamodells, we detail the process of how to treat these individual metamodells as factors:

Firstly, we use the candidate metamodells (Kriging (denoted by KRG), SVR, RBF and polynomial model (denoted by PRS (polynomial regression surrogate))) to calculate the LOO cross validation predictions, that is, $N - 1$ samples are employed to predict the omitted one. Then the LOO predictions

$$\left\{ \begin{matrix} y_{RBF}^{(1)} \\ \vdots \\ y_{RBF}^{(N)} \end{matrix} \right\}, \left\{ \begin{matrix} y_{SVR}^{(1)} \\ \vdots \\ y_{SVR}^{(N)} \end{matrix} \right\}, \left\{ \begin{matrix} y_{KRG}^{(1)} \\ \vdots \\ y_{KRG}^{(N)} \end{matrix} \right\}, \left\{ \begin{matrix} y_{PRS}^{(1)} \\ \vdots \\ y_{PRS}^{(N)} \end{matrix} \right\}$$

for each of the single metamodells at all of the samples are obtained. Combing with the real responses, we can obtain the following new samples

$$(y_{KRG}^{(1)}, y_{SVR}^{(1)}, y_{RBF}^{(1)}, y_{PRS}^{(1)}; y^{(1)}) \cdots (y_{KRG}^{(N)}, y_{SVR}^{(N)}, y_{RBF}^{(N)}, y_{PRS}^{(N)}; y^{(N)}),$$

where $(y_{KRG}^{(i)}, y_{SVR}^{(i)}, y_{RBF}^{(i)}, y_{PRS}^{(i)})$ for $i = 1, \dots, N$ are regarded as new factors, and $y^{(i)}$ for $i = 1, \dots, N$ are still regarded as responses.

Furthermore, according to these new samples, we use *Stepwise regression* method to eliminate the redundant candidate metamodells.

Finally, the remaining metamodells can be used to construct the ensembles according to the techniques such as BP, EG, OWS, and the regression method proposed in formula (6).

In this process, we use cross validation RMSE (especially LOO) to construct ensembles. The LOO cross validation plays a very important role in eliminating of redundant metamodells. And we use prediction RMSE to test the performances of these ensembles, which means that we need extra samples obtained by LHS to test their performances.

Now, let's go back to how to use *Stepwise regression* to eliminate the redundant metamodells. The method begins with an initial ensemble and then compares the explanatory power of incrementally larger and smaller ensembles. At each step, the p-value of an F-statistic is computed to

test ensembles with and without a potential metamodel. If a metamodel is not currently in the ensemble, the null hypothesis is that the metamodel would have a zero coefficient if added to the ensemble. If there is sufficient evidence to reject the null hypothesis, the metamodel is added to the ensemble. Conversely, if a metamodel is currently in the ensemble, the null hypothesis is that the metamodel has a zero coefficient. If there is insufficient evidence to reject the null hypothesis, the metamodel is removed from the ensemble. The method proceeds as follows:

Algorithm 1 Metamodel selection

Step 1. Fit the initial ensemble. The individual metamodel, which has the smallest LOO cross validation (CV) error, is adopted as the initial ensemble. Notice that the initial metamodel may not be included in the final ensemble, because it could be rejected in the following steps.

Step 2. If any metamodells not in the ensemble have p-values less than an entrance tolerance (that is, if it is unlikely that they would have zero coefficients if added to the ensemble), add the one with the smallest p-value and repeat this step; otherwise, go to *step 3*.

Step 3. If any metamodells in the model have p-values greater than an exit tolerance (that is, if it is unlikely that the hypothesis of a zero coefficient can be rejected), remove the one with the largest p-value and go to *step 2*; otherwise, end.

Depending on the metamodells included in the initial ensemble and the order in which metamodells are moved in and out, the method may build different ensembles from the same set of potential metamodells. The method terminates when no single step improves the ensemble. There is no guarantee, however, that a different initial ensemble or a different sequence of steps will not lead to a better fit. In this sense, stepwise ensembles are locally optimal, but may not be globally optimal.

Here, there are two comments about the entrance/exit tolerance and the ordering:

(1) The entrance/exit tolerance is important to determine which metamodells should be included in the resulting

Table 3 Summary of LOO cross-validation results for the parameters in all of the metamodells, $\mathbf{e} = \mathbf{1}_{1 \times n}$, where n is the number of the dimension of \mathbf{x}

Benchmark problem	d in PRS	$(\theta_0, lob, upb, regression\ model)$ in KRG	(C, ε, σ) in SVR	c in RBF
Camelback	3	$\theta_0 = 5\mathbf{e}, lob = 0.1\mathbf{e}, upb = 1\mathbf{e}, regpoly0$	$C = 1e8, \varepsilon = 1e - 5, \sigma = 3$	3
Hartman	4	$\theta_0 = 12\mathbf{e}, lob = 0.1\mathbf{e}, upb = 10\mathbf{e}, regpoly0$	$C = 10, \varepsilon = 0.015, \sigma = 0.8$	1
ER-9	1	$\theta_0 = 12\mathbf{e}, lob = 0.1\mathbf{e}, upb = 10\mathbf{e}, regpoly1$	$C = 1e8, \varepsilon = 0.01, \sigma = 4$	1
ER-12	1	$\theta_0 = 15\mathbf{e}, lob = 0.1\mathbf{e}, upb = 10\mathbf{e}, regpoly1$	$C = 1e8, \varepsilon = 0.01, \sigma = 4$	1
ER-15	1	$\theta_0 = 4\mathbf{e}, lob = 0.1\mathbf{e}, upb = 10\mathbf{e}, regpoly1$	$C = 1e8, \varepsilon = 0.01, \sigma = 4$	1
DP-15	1	$\theta_0 = 4\mathbf{e}, lob = 0.1\mathbf{e}, upb = 10\mathbf{e}, regpoly1$	$C = 1e8, \varepsilon = 0.01, \sigma = 4$	2

ensemble. The entrance/exit tolerance can be determined by two ways: one is according to the engineer’s experience, and the other is using cross validation. The second way is employed in this paper. Actually, the entrance/exit tolerance is affected by both candidate metamodels and data set. Therefore, the choice of entrance/exit tolerance depends on the case we confront.

(2) Although the ordering effects which metamodels should be chosen, the ordering strategy adopted in this paper is reasonable. Just as done in step 1 of *Algorithm 1*, the metamodel with the lowest CV error is chosen as the first metamodel of the ensemble, and the following ordering of adding or removing of metamodels is affected by the first metamodel. This first-metamodel-choosing strategy is advisable, because the lowest-CV metamodel may be more suitable than the others.

At last, we generalize the selection strategy to all of the ensemble techniques, and summarize the procedure of constructing of ensemble as follows:

Algorithm 2 Construction of ensemble

Step 1. The above-mentioned metamodel selection algorithm (i.e., *Algorithm 1*), can be used to reject the redundant individual metamodels, then a subset of n metamodels ($n \leq p$) is obtained.

Step 2. The subset of metamodels can be used to construct an ensemble. We can employ regression method presented in (6) or other methodologies, such as presented in (1) or (3), to construct ensembles. End.

Two contributions in this paper are:

- (1) Adopting the *stepwise regression* as a metamodel selection method before constructing the final ensemble; and
- (2) Proposing (6) as a new method of construction of ensemble.

4 Experiments

4.1 Benchmark problems

In order to test the proposed techniques in this paper, we choose the following analytic functions that are commonly used as benchmark problems in literature (Goel et al. 2007; Acar and Rais-Rohani 2009; Viana et al. 2009).

1. *CamelBack*:

$$y(x_1, x_2) = (4 - 2.1x_1^2 + \frac{x_1^4}{3})x_1^2 + x_1x_2 + (-4 + 4x_2^2)x_2^2 \tag{8}$$

where $x_1 \in [-3, 3], x_2 \in [-2, 2]$.

2. *Hartman*:

$$y(\mathbf{x}) = - \sum_{i=1}^m c_i \exp[- \sum_{j=1}^n a_{ij}(x_j - p_{ij})^2] \tag{9}$$

where $x_i \in [0, 1]$.

The six-variables ($m = 6$) model of this function is considered. The values of function parameters c_i, p_{ij}, a_{ij} for Hartman model are given in Table 1.

3. *Extended Rosenbrock*(denoted by “ER”):

$$y(\mathbf{x}) = \sum_{i=1}^{m-1} [(1 - x_i)^2 + 100(x_{i+1} - x_i^2)^2] \tag{10}$$

where $-5 \leq x_i \leq 10$.

The nine-variables ($m = 9$), twelve-variables ($m = 12$) and fifteen-variables ($m = 15$) models of this function are considered.

4. *Dixon-Price*(denoted by “DP”):

$$y(\mathbf{x}) = (x_1 - 1)^2 + \sum_{i=2}^m i(2x_i^2 - x_{i-1})^2 \tag{11}$$

where $-10 \leq x_i \leq 10$.

Fifteen-variables ($m = 15$) models of this function is considered in this paper.

4.2 Design and analysis of computer experiments

As stated before, the quality of fit depends on the training data (DOE). The performance measures may vary from DOE to DOE. Thus, for all test problems, a set of 500 different DOEs was used as a way of averaging out the DOE dependence of the results. They were created by the MATLAB Latin hypercube function *lhsdesign*, set with the maxmin option with 5 iterations. Table 2 shows details about the data set generated for each test function. Naturally, the number of points used to fit metamodels increases with dimensionality.

The accuracies of each individual and ensemble model for the benchmark problems are measured using root mean square error (RMSE), average absolute error (AAE), max absolute error (MAE) and correlation coefficient (denoted by R). Their definitions are expressed as:

Root mean square error:

$$RMSE = \sqrt{\sum_{i=1}^{n_{error}} (y_i - \hat{y})^2 / n_{error}}$$

Average absolute error:

$$AAE = \sum_{i=1}^{n_{error}} |y_i - \hat{y}| / n_{error}$$

Max absolute error:

$$MAE = \max |y_i - \hat{y}|, i = 1, \dots, n_{error}$$

Table 4 Setting of the parameters *penter* and *remove* used in function *stepwisefit* for benchmark problems

Benchmark problem	<i>penter</i>	<i>remove</i>
Camelback	<i>penter</i> = 0.0001	<i>remove</i> = 0.0002
Hartman	<i>penter</i> = 0.01	<i>remove</i> = 0.02
ER-9	<i>penter</i> = 0.01	<i>remove</i> = 0.02
ER-12	<i>penter</i> = 0.01	<i>remove</i> = 0.02
ER-15	<i>penter</i> = 0.01	<i>remove</i> = 0.02
DP-15	<i>penter</i> = 0.05	<i>remove</i> = 0.1

Correlation coefficient:

$$R(y, \hat{y}) = \frac{\frac{1}{V} \int_V (y - \bar{y})(\hat{y} - \bar{\hat{y}}) dv}{\delta(y)\delta(\hat{y})}$$

$$\frac{1}{V} \int_V y \hat{y} dv = \sum_{i=1}^{n_{error}} y_i \hat{y}_i / n_{error},$$

$$\bar{y} = \sum_{i=1}^{n_{error}} y_i / n_{error},$$

$$\delta(y) = \sqrt{\sum_{i=1}^{n_{error}} (y_i - \bar{y})^2 / n_{error}}$$

In these four definitions above, *n_{error}* is the number of samples in the test set, *y_i* is the actual response, \bar{y} is average value of actual response, \hat{y} is the metamodel response, $\bar{\hat{y}}$ is the average value of metamodel response.

Because the experiments are repeated 500 times, the mean and the coefficient of variation (CV) of R, RMAE, AAE, and MAE are used to evaluate the prediction accuracies of each individual metamodel and ensemble model.

Table 5 Mean and CV (in parenthesis) of RMSE (based on 500 DOEs) for different metamodels, the least value in each category is shown in bold for ease of comparison

	Camelback	Hartman	ER-9	ER-12	ER-15	DP-15
RBF	8.976 (0.378)	0.295 (0.108)	8.06E+03 (1.556)	1.26E+04 (1.209)	1.60E+04 (1.025)	5.71E+04 (0.231)
SVR	8.140 (0.599)	0.365 (0.364)	5.87E+03 (4.214)	8.73E+03 (3.615)	1.20E+04 (2.843)	3.38E+04 (0.718)
KRG	15.805 (0.314)	0.388 (0.268)	7.08E+03 (2.863)	1.15E+04 (2.583)	1.43E+04 (1.476)	2.86E+04 (0.758)
PRS	15.032 (0.306)	0.219 (0.224)	1.03E+04 (1.965)	1.40E+04 (2.119)	1.73E+04 (1.602)	3.25E+04 (0.679)
–	–	–	–	–	–	–
BP	8.971 (0.506)	0.235 (0.238)	5.24E+03 (2.441)	8.01E+03 (1.956)	1.09E+04 (1.555)	2.83E+04 (0.580)
EG	9.704 (0.355)	0.246 (0.129)	5.16E+03 (3.267)	7.84E+03 (2.823)	1.02E+04 (2.036)	2.21E+04 (0.811)
EG*	8.915 (0.499)	0.214 (0.174)	5.02E+03 (3.166)	7.42E+03 (2.761)	9.46E+03 (2.252)	2.06E+04 (0.900)
OWS	10.151 (0.549)	0.214 (0.179)	2.42E+09 (22.192)	2.05E+12 (16.947)	1.04E+05 (11.847)	2.64E+05 (9.092)
OWS*	9.036 (0.517)	0.213 (0.182)	4.75E+03 (2.537)	7.07E+03 (2.083)	8.82E+03 (1.825)	1.91E+04 (0.937)
Od	9.345 (0.387)	0.230 (0.145)	4.97E+03 (3.183)	7.77E+03 (2.736)	1.02E+04 (1.907)	2.35E+04 (0.721)
Od*	8.872 (0.500)	0.212 (0.178)	4.91E+03 (2.942)	7.38E+03 (2.627)	9.46E+03 (2.060)	2.15E+04 (0.846)
Reg	10.572 (0.463)	0.298 (0.114)	2.83E+16 (22.290)	1.00E+15 (11.166)	2.48E+15 (12.364)	3.84E+15 (20.826)
Reg*	10.080 (0.482)	0.213 (0.179)	4.60e+03 (3.031)	6.82e+03 (4.126)	8.04e+03 (4.202)	1.79e+04 (0.987)

The least values in ensemble models are shown in bold

The definition of CV is expressed as:

$$CV = \delta / \mu$$

where δ is the standard variance of samples, and μ is the mean of samples.

4.3 Individual/ensemble techniques

Four individual metamodels (PRS, KRG, SVR, and RBF), described in “Appendix”, are considered in this paper. These metamodels are used as candidate metamodels.

All the values of the parameters are set using cross validation (LOO is adopted in this paper) such that they minimize the MSE. There are several parameters whose values should be determined in the metamodels mentioned above: the highest order (denoted by *d*) in PRS, the parameter (*c*) in multiquadrics of RBF, (*C*, ϵ , σ) in SVR, and (θ , *lob*, *upb* and *regression model*) in Kriging, where Gauss function is adopted as correlation model.

As to the implementation of Kriging, we use the toolbox proposed by Lophaven et al. (2002), where *lob* and *upb* are defined respectively as lower and upper bounds on θ , which are clearly described in Lophaven et al. (2002).

Just as said in Lophaven et al. (2002), θ_0 is just an initial guess on underlying θ , and it doesn't necessarily be required between *lob* and *upb*. If it is larger than *upb* or smaller than *lob*, there is a function to reset it into [*lob upb*]. And in the train function *dacefit* of Lophaven et al. (2002), the optimal θ -value can be gotten by subfunction *boxmin*.

The LOO cross-validation results are presented in Table 3.

In this paper, we focus on the ensemble techniques based on cross validation: BestPRESS (described in Section 2.1,

heuristic method (described in Section 2.2), optimal weighted metamodel (OWS) (described in Section 2.3), and diagonal optimal weighted metamodel (OWS_{diag}) (described in Section 2.3). Just as the denotation in Zhou et al. (2011): BestPRESS is denoted by BP; The heuristic method of Goel et al. (2007) is labeled as EG; OWS_{diag} in Viana et al. (2009) is denoted by Od; OWS remains unchanged.

Although BestPRESS is classified into ensemble technique, it also can be regarded as a kind of stand-alone metamodel, because the best metamodel varies with DOE, and actually there is only one metamodel used in the construction of ensemble. Except for BestPRESS, the other ensemble techniques, such as heuristic method (Goel et al. 2007), OWS and OWS_{diag} (Viana et al. 2009), can be constructed after eliminating the redundant metamodels using

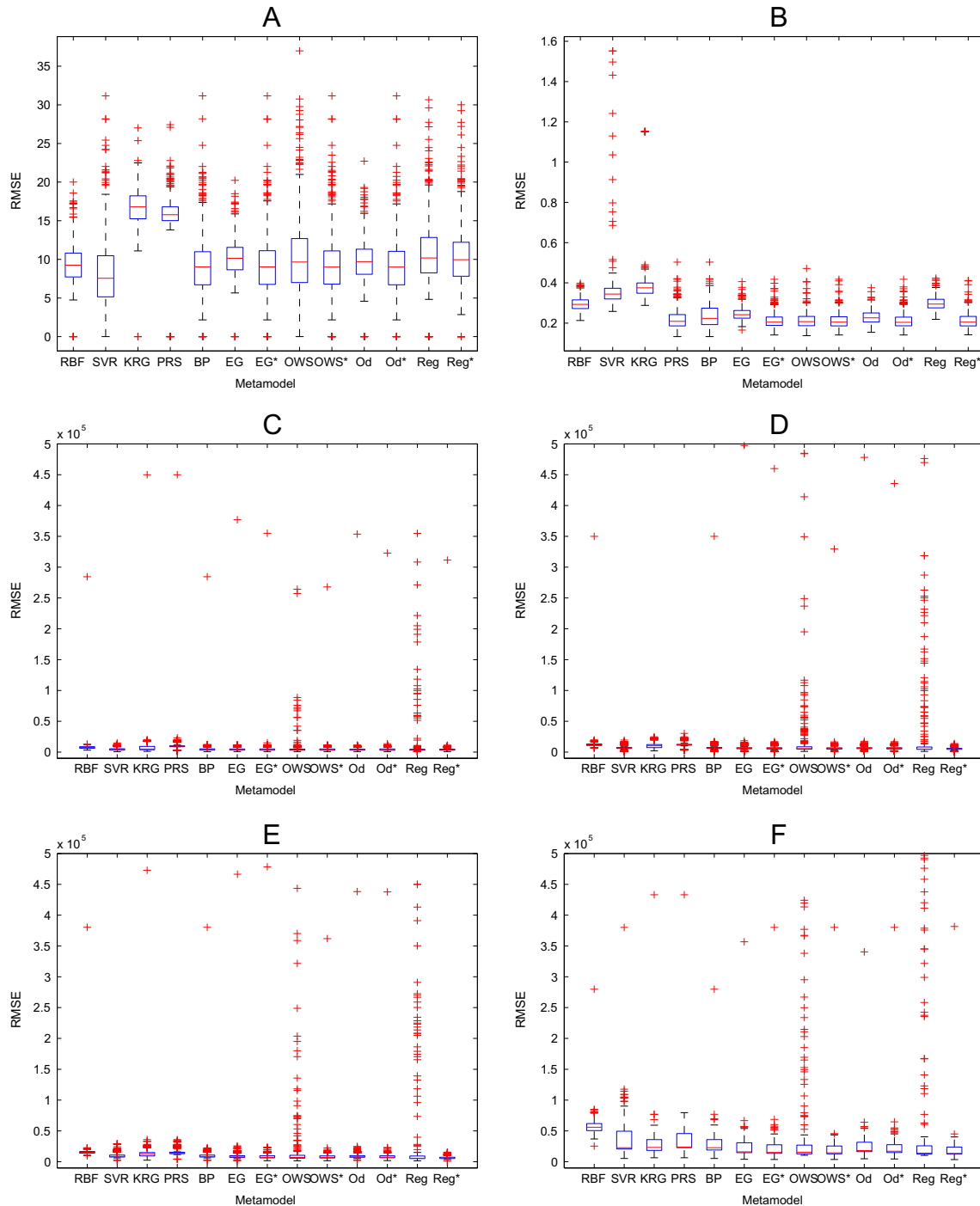


Fig. 1 RMSE for different metamodels. (a: Camelback, b: Hartman, c: ER-9, d: ER-12, e: ER-15, f: DP-15)

the method proposed in this paper. The eliminated ensembles of EG, Od and OWS are denoted by EG*, Od* and OWS* respectively.

In the process of rejecting of the redundant individual metamodels, the function *stepwisefit* in MATLAB is used. The parameters *penter* and *remove*, which are employed in the step 2 and step 3 of *Algorithm 1* in Section 3, are presented in Table 4. They are used to determine which metamodels should be added and which ones should be removed.

5 Results and analysis of experiments

In this paper, we focus on the comparison among the ensembles (including BestPRESS), without comparing these ensembles with the individual metamodels (PRS, KRG, SVR, and RBF). Actually, we don't know which individual metamodel can accurately approximate the underlying relationship in practice, it is thus inappropriate to use a same individual metamodel in all of the DOEs, unless we know the underlying relationship. If we have to use an individual metamodel to approximate the underlying relationship, we often adopt BestPRESS to get the best individual metamodel according to a specific DOE, and BestPRESS varies from DOE to DOE. Therefore, we only need to compare ensembles with BestPRESS rather than with the individual metamodels. Although we don't think that it is necessary to compare the ensembles with the individual metamodels, we still want to present the fact that some of the ensembles even outperform the individual metamodels, which compose the ensembles, in many cases. Therefore, we still present the prediction errors of ensembles and individual metamodels

in a same table/figure, but we only compare the prediction errors among ensembles. In order to distinguish ensembles from individual metamodels, we separate ensembles from individual metamodels with the symbol “-” in these tables.

The results of different benchmark problems are shown with the help of boxplots (the description of boxplot is provided in the “Appendix”), and the means and CVs of the error metrics are presented with several tables.

5.1 RMSE

The RMSEs for all of the benchmark functions using individual metamodels and ensembles are presented in Table 5, where we just compare the performance among the ensembles, and the same is done in the following tables in this paper. From this table we can see: 1) the means of RMSEs for EG*, OWS*, Od* and Reg* are smaller than those for EG, OWS, Od and Reg respectively in all of the benchmark problems (Camelback, Hartman, ER-9, ER-12, ER-15 and DP-15), which indicates that it is necessary to get rid of the redundant individual metamodels from the set of candidate ones before constructing the ensemble of metamodels. 2) Od* has better performance in low dimension benchmark problems: Camelback, Hartman; nevertheless, Reg* has the best performance among all of the metamodels (including ensembles and even individual metamodels) in high dimension benchmark problems: ER-9, ER-12, ER-15 and DP-15. 3) In high dimension benchmark problems (ER-9, ER-12, ER-15 and DP-15), OWS and Reg have apparently larger RMSEs than the other ensembles and individual metamodels; OWS* and Reg*, however, sharply reduce the errors after rejecting the redundant individual metamodels.

Table 6 Mean and CV of AAE for different metamodels

	Camelback	Hartman	ER-9	ER-12	ER-15	DP-15
RBF	4.979 (0.362)	0.228 (0.105)	5.98E+03 (1.606)	9.61E+03 (1.256)	1.24E+04 (1.056)	4.41E+04 (0.242)
SVR	3.929 (0.590)	0.283 (0.386)	4.51E+03 (4.359)	6.64E+03 (3.808)	9.08E+03 (3.022)	2.53E+04 (0.731)
KRG	9.535 (0.316)	0.297 (0.311)	5.53E+03 (2.906)	9.15E+03 (2.626)	1.14E+04 (1.453)	2.28E+04 (0.757)
PRS	11.127 (0.306)	0.169 (0.218)	8.25E+03 (1.945)	1.12E+04 (2.134)	1.39E+04 (1.587)	2.60E+04 (0.676)
-	-	-	-	-	-	-
BP	4.695 (0.508)	0.182 (0.234)	3.99E+03 (2.450)	6.08E+03 (2.032)	8.31E+03 (1.620)	2.23E+04 (0.595)
EG	5.619 (0.346)	0.189 (0.130)	3.97E+03 (3.345)	6.07E+03 (2.948)	7.90E+03 (2.073)	1.73E+04 (0.824)
EG*	4.643 (0.492)	0.164 (0.175)	3.83E+03 (3.228)	5.72E+03 (2.890)	7.34E+03 (2.294)	1.62E+04 (0.900)
OWS	5.645 (0.507)	0.164 (0.180)	1.29E+08 (22.142)	2.03E+12 (16.948)	2.29E+04 (4.664)	2.19E+05 (10.406)
OWS*	4.674 (0.501)	0.164 (0.181)	3.61E+03 (2.553)	5.44E+03 (2.154)	6.91E+03 (1.887)	1.52E+04 (0.932)
Od	5.252 (0.376)	0.176 (0.144)	3.80E+03 (3.258)	5.99E+03 (2.860)	7.88E+03 (1.948)	1.85E+04 (0.735)
Od*	4.611 (0.491)	0.163 (0.178)	3.74E+03 (2.987)	5.69E+03 (2.732)	7.34E+03 (2.097)	1.70E+04 (0.843)
Reg	6.560 (0.410)	0.230 (0.111)	2.79E+16 (22.291)	9.85E+14 (11.145)	2.45E+15 (12.374)	3.82E+15 (20.823)
Reg*	5.931 (0.440)	0.165 (0.179)	3.60e+03 (2.969)	5.15e+03 (3.301)	6.35e+03 (4.086)	1.43e+04 (0.979)

The least values in ensemble models are shown in bold

The intuitional comparison results are presented in Fig. 1. As there are very large outliers in the boxplots of *OWS* and *Reg* within subfigures of C, D, E and F (C: ER-9, D: ER-12, E: ER-15, F: DP-15), we constrain these subfigures into $[0 \ 5 \times 10^5]$ (such treatment is also adopted in AAE and MAE) for the sake of ease of observing, so we can't observe the outliers larger than 5×10^5 . From Fig. 1 we can observe that *OWS* and *Reg* have two very long tails in C, D, E and

F, which indicates that *OWS* and *Reg* have large variations in high dimension problems. Nevertheless, *OWS** and *Reg** greatly shorten their tails.

5.2 AAE

Table 6 shows the AAEs using ensembles and individual metamodels on all of the benchmark problems. From this

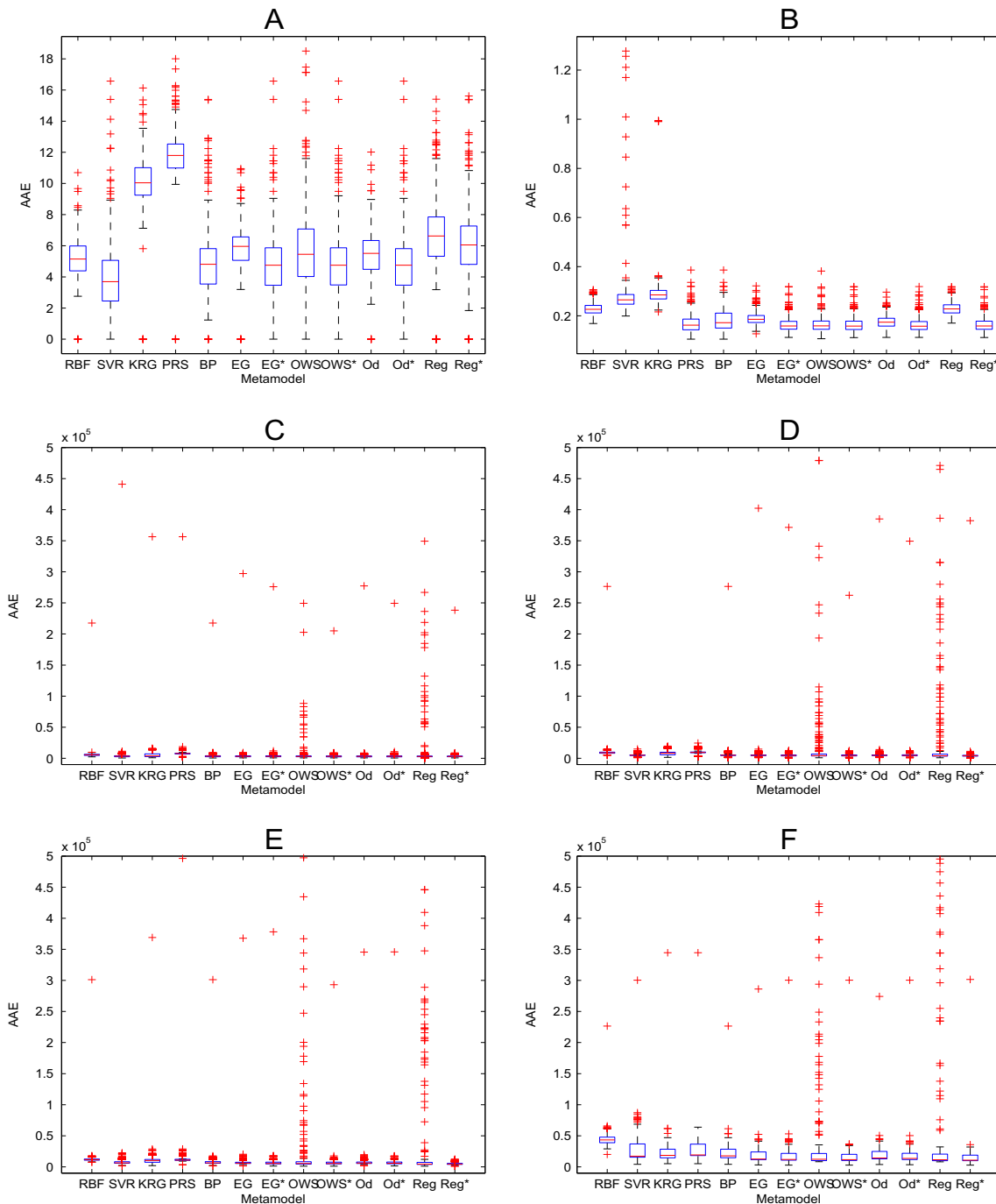


Fig. 2 AAE for different metamodels. (a: Camelback, b: Hartman, c: ER-9, d: ER-12, e: ER-15, f: DP-15)

table we find that EG^* , OWS^* , Od^* and Reg^* have better performance than EG , OWS , Od and Reg respectively in all of the benchmark problems. And we also can observe that Reg^* has the best performance among all of the ensembles, especially in high dimension problems (refer to ER-9, ER-12, ER-15 and DP-15 in this paper), while Od^* is the best ensemble in low dimension problems (refer to Camelback and Hartman in this paper). The corresponding comparison results are shown in Fig. 2. These results are similar to those presented in RMSE.

5.3 MAE

Table 7 and Fig. 3 show the MAEs of different metamodels for different test functions. From this table and figure we find that except for Camelback, EG^* , OWS^* , Od^* and Reg^* still have better performance than EG , OWS , Od and Reg respectively. And Reg^* is still the best ensemble in high dimension problems. Furthermore, this table and figure illustrate that rejecting the redundant individual metamodels is still necessary for getting smaller MAEs.

5.4 Correlation coefficient

The correlation coefficients for different benchmark functions are shown in Table 8 and Fig. 4, from which we can see: 1) Although OWS^* has similar correlation coefficient to Reg^* in ER-9, it has the best performance in high dimension problems. 2) Except for Camelback, Od^* still has better performance than the other ensembles in low dimension

problems. These results demonstrate that we should reject the redundant individual metamodels before constructing the final ensemble.

5.5 Discussions

As for RMSE, AAE, MAE and correlation coefficient, we find that ensemble Reg and OWS have unsatisfying results in ER-9, ER-12, ER-15 and DP-15. Why those happen?

On the one hand, in high dimension benchmark problems, some of individual metamodels may not be suitable for these problems. Therefore, these unsuitable individual metamodels may impair the performances of the ensembles, which can explain to a certain extent why Reg and OWS perform badly in these problems. After eliminating those improper metamodels, Reg^* and OWS^* enhance their performances.

On the other hand, as for Reg , since it doesn't have a constrain,

$$\sum_{i=1}^p \beta_i = 1$$

if these coefficients are not proper, the prediction values obtained with the regression model

$$y = \beta_0 + \beta_1 M_1(\mathbf{x}) + \dots + \beta_p M_p(\mathbf{x})$$

maybe largely deviate from the true values, especially in high-dimensional problems, which perhaps is a drawback of the regression model. But after eliminating the redundant metamodels with *stepwise* method, the regression model

Table 7 Mean and CV of MAE for different metamodels

	Camelback	Hartman	ER-9	ER-12	ER-15	DP-15
RBF	51.708 (0.403)	1.030 (0.166)	3.57E+04 (1.435)	5.13E+04 (0.880)	6.48E+04 (1.040)	2.28E+05 (0.256)
SVR	56.552 (0.656)	1.304 (0.326)	2.44E+04 (3.599)	3.90E+04 (3.111)	5.29E+04 (2.113)	1.52E+05 (0.650)
KRG	83.466 (0.339)	1.417 (0.174)	2.61E+04 (2.473)	3.98E+04 (2.101)	5.07E+04 (1.829)	9.67E+04 (0.804)
PRS	65.176 (0.348)	0.829 (0.280)	3.41E+04 (1.896)	4.57E+04 (1.827)	5.78E+04 (1.849)	1.06E+05 (0.745)
–	–	–	–	–	–	–
BP	56.545 (0.574)	0.873 (0.272)	2.21E+04 (2.374)	3.47E+04 (1.375)	4.75E+04 (1.461)	1.03E+05 (0.545)
EG	56.162 (0.397)	0.895 (0.173)	2.19E+04 (2.555)	3.25E+04 (1.994)	4.27E+04 (2.041)	8.60E+04 (0.753)
EG*	56.724 (0.578)	0.786 (0.230)	2.16E+04 (2.967)	3.08E+04 (2.077)	3.90E+04 (2.226)	7.50E+04 (0.856)
OWS	63.244 (0.668)	0.794 (0.228)	5.04E+10 (22.206)	2.85E+12 (16.825)	1.84E+06 (14.035)	1.29E+06 (12.164)
OWS*	58.080 (0.614)	0.789 (0.242)	2.03E+04 (2.153)	2.98E+04 (1.579)	3.46E+04 (1.580)	6.45E+04 (0.902)
Od	55.609 (0.438)	0.838 (0.188)	2.14E+04 (2.544)	3.27E+04 (1.896)	4.34E+04 (1.911)	9.03E+04 (0.670)
Od*	56.520 (0.581)	0.780 (0.235)	2.13E+04 (2.789)	3.08E+04 (1.902)	3.91E+04 (2.042)	7.78E+04 (0.803)
Reg	60.389 (0.579)	1.046 (0.180)	3.89E+16 (22.277)	1.48E+15 (11.292)	3.38E+15 (12.067)	5.00E+15 (20.872)
Reg*	60.969 (0.597)	0.786 (0.239)	1.80e+04 (3.301)	2.91e+04 (6.585)	2.95e+04 (5.421)	5.83e+04 (0.969)

The least values in ensemble models are shown in bold

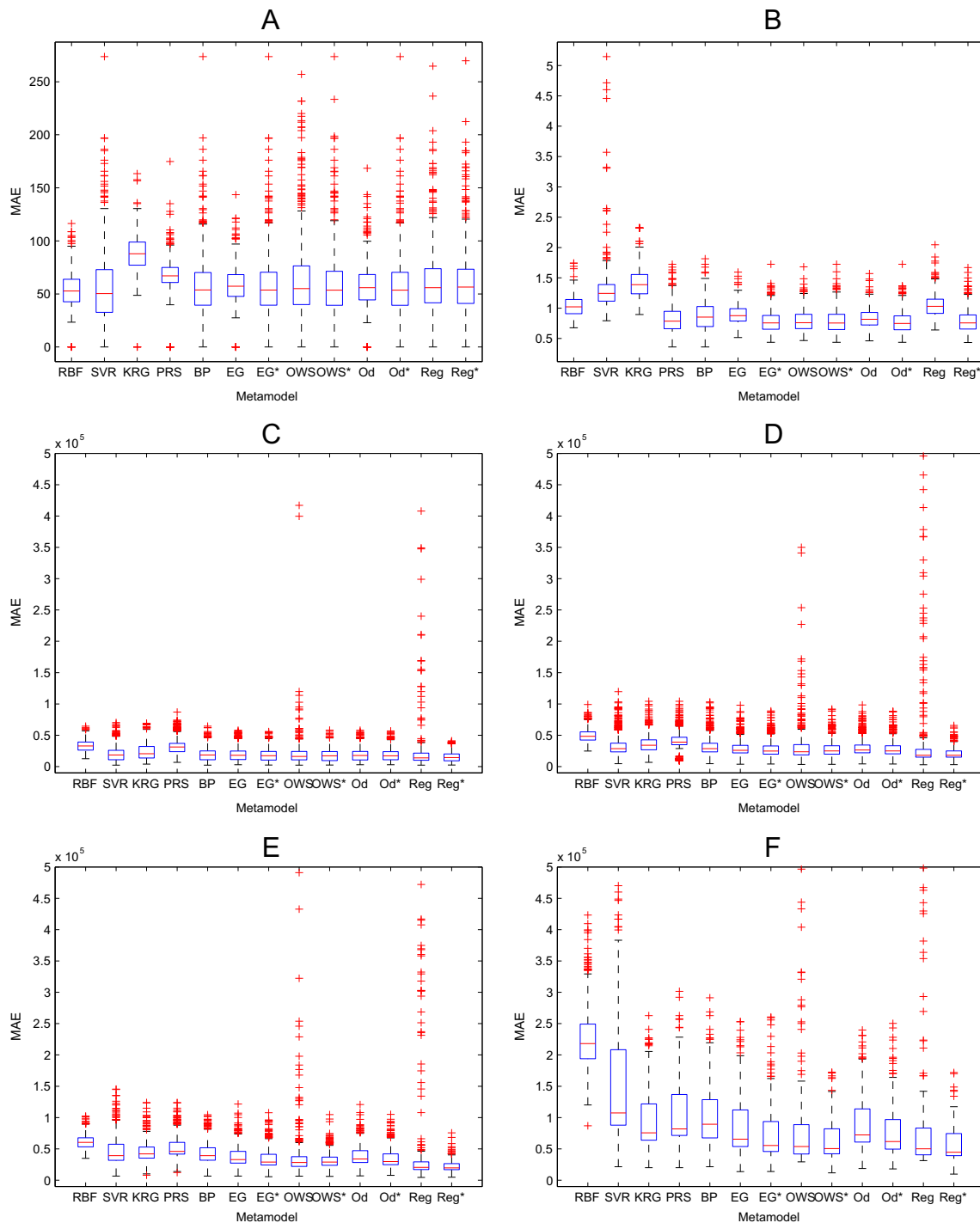


Fig. 3 MAE for different metamodels. (a: Camelback, b: Hartman, c: ER-9, d: ER-12, e: ER-15, f: DP-15)

can significantly improve its performance. Therefore, using *stepwise* method to get rid of the redundant metamodels is a necessary and efficient step not only in construction of regression model but also in any other ensemble construction techniques.

Next, let's analyze the coefficients of variation for these ensembles. After carefully examining the results in Tables 5–8, we find that *Reg** has larger coefficients of variation than the other ensembles (it's still better than *Reg*), especially in high dimension problems, which means

Table 8 Mean and CV of correlation coefficient between actual and predicted response for different metamodells

	Camelback	Hartman	ER-9	ER-12	ER-15	DP-15
RBF	0.8618 (0.2895)	0.9669 (0.0077)	0.9926 (0.0052)	0.9880 (0.0066)	0.9864 (0.0065)	0.9806 (0.0141)
SVR	0.8650 (0.2981)	0.9482 (0.0555)	0.9955 (0.0231)	0.9937 (0.0292)	0.9914 (0.0322)	0.9927 (0.0434)
KRG	0.7160 (0.2974)	0.9335 (0.1146)	0.9925 (0.0178)	0.9881 (0.0197)	0.9856 (0.0136)	0.9931 (0.0469)
PRS	0.7372 (0.2916)	0.9820 (0.0087)	0.9853 (0.0186)	0.9828 (0.0199)	0.9796 (0.0200)	0.9918 (0.0470)
BP	0.8556 (0.2967)	0.9788 (0.0107)	0.9965 (0.0057)	0.9949 (0.0062)	0.9935 (0.0065)	0.9948 (0.0137)
—	—	—	—	—	—	—
EG	0.8512 (0.2901)	0.9776 (0.0063)	0.9967 (0.0096)	0.9952 (0.0126)	0.9940 (0.0092)	0.9963 (0.0293)
EG*	0.8568 (0.2965)	0.9828 (0.0066)	0.9971 (0.0058)	0.9962 (0.0103)	0.9957 (0.0085)	0.9961 (0.0431)
OWS	0.8484 (0.3015)	0.9828 (0.0066)	0.9873 (0.0988)	0.9731 (0.1858)	0.9889 (0.0801)	0.9875 (0.1099)
OWS*	0.8567 (0.2970)	0.9829 (0.0068)	0.9972 (0.0054)	0.9965 (0.0053)	0.9962 (0.0056)	0.9963 (0.0431)
Od	0.8554 (0.2910)	0.9802 (0.0061)	0.9970 (0.0082)	0.9953 (0.0114)	0.9941 (0.0080)	0.9960 (0.0243)
Od*	0.8574 (0.2964)	0.9830 (0.0067)	0.9971 (0.0056)	0.9961 (0.0092)	0.9955 (0.0072)	0.9957 (0.0431)
Reg	0.8599 (0.2919)	0.9666 (0.0081)	0.9931 (0.0885)	0.9748 (0.2029)	0.9669 (0.2435)	0.9830 (0.1647)
Reg*	0.8565 (0.2971)	0.9828 (0.0069)	0.9972 (0.0071)	0.9960 (0.0198)	0.9953 (0.0339)	0.9961 (0.0465)

The largest values in ensemble models are shown in bold

Fig. 4 Correlations between actual and predicted response for different metamodells. (a: Camelback, b: Hartman, c: ER-9, d: ER-12, e: ER-15, f: DP-15)

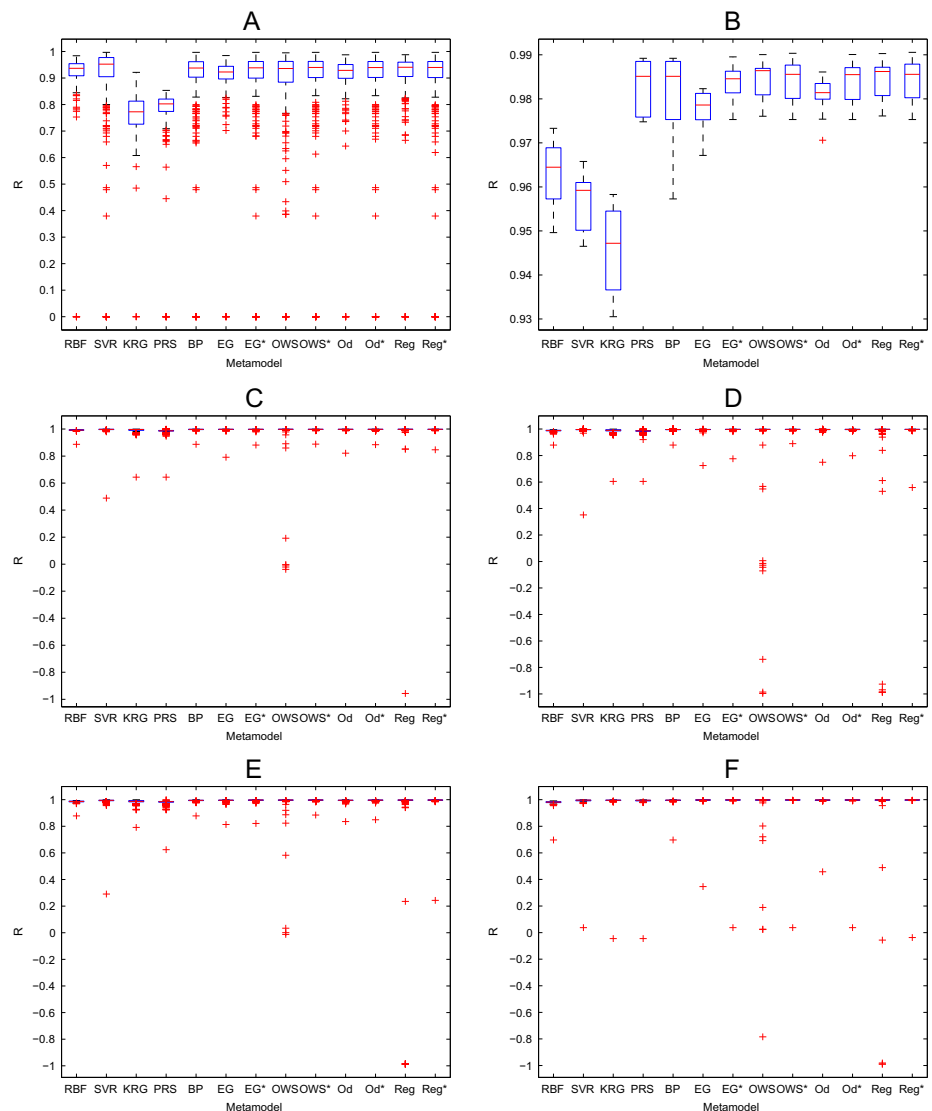


Table 9 The printing process data

i	x_1	x_1	x_1	y_{i1}	y_{i2}	y_{i3}	\bar{y}_i	s_i
1	-1	-1	-1	34	10	28	24.0	12.5
2	0	-1	-1	115	116	130	120.3	8.4
3	1	-1	-1	192	186	263	213.7	42.8
4	-1	0	-1	82	88	88	86.0	3.7
5	0	0	-1	44	178	188	136.7	80.4
6	1	0	-1	322	350	350	340.7	16.2
7	-1	1	-1	141	110	86	112.3	27.6
8	0	1	-1	259	251	259	256.3	4.6
9	1	1	-1	290	280	245	271.7	23.6
10	-1	-1	0	81	81	81	81.0	0.0
11	0	-1	0	90	122	93	101.7	17.7
12	1	-1	0	319	376	376	357.0	32.9
13	-1	0	0	180	180	154	171.3	15.0
14	0	0	0	372	372	372	372.0	0.0
15	1	0	0	541	568	396	501.7	95.5
16	-1	1	0	288	192	312	264.0	63.5
17	0	1	0	432	336	513	427.0	88.6
18	1	1	0	713	725	754	730.7	21.1
19	-1	-1	1	364	99	199	220.7	133.8
20	0	-1	1	232	221	266	239.7	23.5
21	1	-1	1	408	415	443	422.0	18.5
22	-1	0	1	182	233	182	199.0	29.4
23	0	0	1	507	515	434	485.3	44.6
24	1	0	1	846	535	640	673.7	158.2
25	-1	1	1	236	126	168	176.7	55.5
26	0	1	1	660	440	403	501.0	138.9
27	1	1	1	878	991	1161	1010.0	142.5

*Reg** is not very “robust” in high dimension problems compared with the other ensembles, but its average performance is still good and its coefficients of variation are still nice in low dimension problems. In addition, the coefficients of variation of *EG**, *OWS** and *Od** are similar to those of *EG*, *OWS* and *Od* respectively, and even smaller in some cases. Therefore, the values of these coefficients of variation for ensembles with *stepwise*, on the whole, are acceptable.

Table 10 Setting of the parameters *penter* and *premove* used in function *stepwise*fit for Mean and Standard deviation data

Data	<i>penter</i>	<i>premove</i>
Mean data	<i>penter</i> = 0.1	<i>premove</i> = 0.2
Std. Dev. data	<i>penter</i> = 0.2	<i>premove</i> = 0.3

6 Metamodel-based optimization

After presenting the efficiency of these ensemble models using several benchmark problems, we will take a real-life example, printing ink, to furthermore illustrate their performance in metamodel-based optimization problem.

6.1 The printing ink example

The print ink example is widely used in the works of Box and Draper (1987), Vining and Myers (1990), Del Castillo and Montgomery (1993), Lin and Tu (1995), Copeland and Nelson (1996), Vining and Bohn (1998), Kksoy and Yalcinoz (2005), Shaibu and Cho (2009), and Zhou et al. (2012). The purpose of the experiment was to study the effect of speed, x_1 , pressure, x_2 , and distance, x_3 , upon a printing machine’s ability to apply coloring inks upon package labels. The experiment is $3^3(n = 27)$ complete factorial with three runs ($m = 3$) at each design point for a total of 81 runs. These papers all have considered the situation where we wish to minimize the process variance around a target value of 500 for the response. Table 9 summarizes the data. Following the traditional practice in literature, this table reports the specific settings for the factors in terms of the standard -1 to 1 metric.

In this problem, the optimization model is

$$\begin{aligned} \min_x MSE(y) &= E[(y - \zeta)^2] = Dev^2(y) + (E(y) - \zeta)^2. \\ s.t. & 1 \leq \mathbf{x} \leq 1 \end{aligned} \tag{12}$$

where, $E(y)$ denotes the mean response, and $Dev(y)$ denotes the standard deviation response, ζ represents a target value for the response.

After approximating $E(y)$ and $Dev(y)$ using the ensemble of metamodels, we can get the metamodel-based optimization problem:

$$\begin{aligned} \min_x \hat{MSE}(y) &= \hat{Dev}^2(y) + (\hat{E}(y) - \zeta)^2. \\ s.t. & 1 \leq \mathbf{x} \leq 1 \end{aligned} \tag{13}$$

In this experiment, the p-values in stepwise method are set by cross validation and presented in Table 10.

6.2 Optimization results

In this part, the standard optimization function *fmincon* in MATLAB is employed. Table 11 summarizes the estimated MSEs and the recommended settings for all of the ensemble models mentioned above. From this table, we can see that all of the ensemble models with stepwise method have the better performance than those without it in terms of MSE and deviation. In this experiment, we hope the objective func-

Table 11 Comparing solution to the MSE optimization (19)

Metamodels	Optimal setting x^*	MSE	Mean $\hat{\mu}(x^*)$	Std. Dev. $\hat{\sigma}(x^*)$
EG	(1.0000, 1.0000, -0.5018)	69.1690	500.3182	8.3107
EG*	(1.0000, 1.0000, -0.4552)	66.2104	500.2104	8.1408
OWS (Zhou et al. 2012)	(1.0000, 1.0000, -0.4793)	81.4121	500.3119	9.0175
OWS*	(0.9393, 1.0000, -0.4975)	80.8226	500.2356	8.9870
Od	(0.9959, 1.0000, -0.5075)	79.8354	500.3678	8.9275
Od*	(0.9965, 1.0000, -0.4584)	67.1614	500.2144	8.1924
Reg	(1.0000, 1.0000, -0.4455)	1.205e3	501.4542	34.6877
Reg*	(0.9694, 1.0000, -0.4549)	1.036e3	500.3230	32.1883

tion (MSE) and the deviation are as small as possible. We also can find that regression model is not very well, but Reg^* is still better than Reg , which also indicates that using stepwise to reduce the redundant metamodels is a reasonable strategy.

7 Conclusion

Illuminated by variable selection method widely used in polynomial regression, we treat metamodels as a variables in polynomial regression, and then propose a methodology of constructing of ensemble from the view of polynomial regression. In addition, considering that including redundant metamodels could impair the prediction accuracy of ensemble, we adopt the *stepwise regression* method to reject the redundant individual metamodels before constructing the final ensemble of metamodels.

After examining the experiment results, we have the following findings:

1. *Stepwise regression* method is an useful automatic strategy to reject the inappropriate individual metamodels. This strategy can automatically identify how many individual metamodels should be used to construct the final ensemble of metamodels according to a specify DOE. In the existing methods of constructing of ensembles, the number of individual metamodels is determined before-hand; Nevertheless, the method proposed in this paper doesn't need to do that. In fact, we have no idea about how many metamodels are fit for the design problems. In other words, it is unknown that how many metamodels should be added to the ensemble. The experiment results show that getting rid of the redundant individual metamodels before constructing the ensemble can greatly enhance the performance of ensemble.
2. After rejecting the redundant metamodels, using the construction method presented in formula (6) from the view of polynomial regression is an optional strategy, especially in high dimension problems, but it is still

necessary to be careful of the “robustness” (or stability) in spite of its good average performance.

Acknowledgments The funding provided for this study by National Science Foundation for Young Scientists of China under Grant NO.71401080 and NO.201508320059, the University Social Science Foundation of Jiangsu under Grant NO.2013SJB6300072 & NO.TJS211021, the University Science Foundation of Jiangsu under Grant NO.12KJB630002, and the Talent Introduction Foundation of Nanjing University of Posts and Telecommunications under Grant NO.NYS212008 & NO.D/2013/01/104 are gratefully acknowledged.

Appendix A: Several metamodeling techniques

Here, there are four metamodeling techniques (PRS, RBF, Kriging, SVR) are considered.

A.1 PRS

For PRS, the highest order is allowed to be 4 in this paper, but the used order in a specific problem is determined by the selected sample set. When the highest order of a polynomial model is 4, it can be expressed as:

$$\tilde{F}(\mathbf{x}) = a_0 + \sum_{i=1}^N b_i x_i + \sum_{i,j(i \leq j)} c_i x_i x_j + \sum_{i=1}^N d_i x_i^3 + \sum_{i=1}^N e_i x_i^4 \quad (14)$$

where \tilde{F} is the response surface approximation of the actual response function, N is the number of variables in the input vector \mathbf{x} , and a, b, c, d, e are the unknown coefficients to be determined by the least squares technique.

Notice that 3rd and 4th order models in polynomial model do not have any mixed polynomial terms (interactions) of order 3 and 4. Only pure cubic and quadratic terms are included to reduce the amount of data required for model construction. A lower order model (Linear and Quadratic) includes only lower order polynomial terms (only linear and quadratic terms correspondingly).

A.2 RBF

The general form of the RBF approximation can be expressed as:

$$f(x) = \sum_{i=1}^m \beta_i \varphi(\|x - x_i\|) \tag{15}$$

Powell (1987) consider several forms for the basis function $\varphi(\cdot)$:

1. $\varphi(r) = e^{(-r^2/c^2)}$ Gaussian
2. $\varphi(r) = (r^2 + c^2)^{\frac{1}{2}}$ Multiquadrics
3. $\varphi(r) = (r^2 + c^2)^{-\frac{1}{2}}$ Reciprocal Multi-quadrics
4. $\varphi(r) = (r/c^2) \log(r/c)$ Thin-Plate Spline
5. $\varphi(r) = \frac{1}{1+e^{r/c}}$ Logistic

where $c \geq 0$. Particularly, the multi-quadratic RBF form has been applied by Meckesheimer et al. (2002, 2001) to construct an approximation based on the Euclidean distance of the form:

$$\varphi(x) = \beta_0 + \sum_{i=1}^n \beta_i \|x - x_i\| \tag{16}$$

where $\|\cdot\|$ represents the Euclidean norm. Replacing $\varphi(x)$ with the vector of response observations, \mathbf{y} yields a linear system of n equations and n variables, which is used to solve β . As described above, this technique can be viewed as an interpolating process. RBF metamodels have produced good fits to arbitrary contours of both deterministic and stochastic responses (Powell 1987). Different RBF forms were compared by McDonald et al. (2000) on a hydro code simulation, and the author found that the Gaussian and the multi-quadratic RBF forms performed best generally.

A.3 Kriging

For computer experiments, Kriging is viewed from a Bayesian perspective where the response is regarded as a realization of a stationary random process. The general form of this model is expressed as:

$$Y(\mathbf{x}) = \sum_{i=1}^k \beta_i f_i(\mathbf{x}) + Z(\mathbf{x}) \tag{17}$$

which is comprised of the linear model component of k specified function $f_i(\mathbf{x})$ (i.e., the expression of the function is given, which is defined below) with unknown coefficients β_i ($i = 1, \dots, k$), and $Z(\cdot)$ is a stochastic process, commonly assumed to be Gaussian, with mean zero and covariance

$$\begin{aligned} Cov(Z(\mathbf{w}), Z(\mathbf{u})) &= \sigma^2 R(\mathbf{w}, \mathbf{u}) \\ &= \sigma^2 \exp \left\{ -\theta \sum_{i=1}^d (w_i - u_i)^2 \right\}, \end{aligned} \tag{18}$$

where σ^2 is the process variance; parameter θ , which is somewhat like the parameter c in Gaussian basis function of RBF, is estimated using maximum likelihood.

For the set $\mathbf{S} = \{\mathbf{s}_1, \dots, \mathbf{s}_n\}$, we have the corresponding outputs $\mathbf{y}_s = \{y(\mathbf{s}_1), \dots, y(\mathbf{s}_n)\}^T$. Considering the linear predictor

$$\widehat{y}(\mathbf{x}) = \mathbf{c}^T \mathbf{y}_s \tag{19}$$

with $\mathbf{c} = \mathbf{c}(\mathbf{x}) \in \mathbf{R}^n$. Note that the members of the weight vector \mathbf{c} are not constants (whereas β in formula (17) are) but decrease with the distance between the input \mathbf{x} to be predicted and the sampled points $\mathbf{S} = \{\mathbf{s}_1, \dots, \mathbf{s}_n\}$; this $\mathbf{S} = \{\mathbf{s}_1, \dots, \mathbf{s}_n\}$ determines the simulation output vector $\mathbf{y}_s = \{y(\mathbf{s}_1), \dots, y(\mathbf{s}_n)\}^T$. Here, we replace \mathbf{y}_s with the random vector $\mathbf{Y}_s = \{Y(\mathbf{s}_1), \dots, Y(\mathbf{s}_n)\}^T$. In order to keep the predictor unbiased, we demand

$$E(\mathbf{c}^T \mathbf{Y}_s) = E(\mathbf{Y}(\mathbf{x})), \tag{20}$$

and under this condition minimize

$$MSE[\widehat{y}(\mathbf{x})] = E[(\mathbf{c}^T \mathbf{Y}_s - Y(\mathbf{x}))^2]. \tag{21}$$

And then, we have

$$\begin{aligned} E[(\mathbf{c}^T \mathbf{Y}_s - Y(\mathbf{x}))^2] &= (\mathbf{c}^T \mathbf{F} \beta - \mathbf{f}_x^T \beta)^2 \\ &+ [\mathbf{c}^T - 1] \begin{bmatrix} \mathbf{V} & \mathbf{v}_x \\ \mathbf{v}_x^T & \sigma^2 \end{bmatrix} \begin{pmatrix} \mathbf{c} \\ -1 \end{pmatrix}, \end{aligned} \tag{22}$$

where

$$\mathbf{f}_x = (f_1(\mathbf{x}), \dots, f_k(\mathbf{x}))^T, \tag{23}$$

$$\mathbf{V} = (Cov(Y(\mathbf{s}_i), Y(\mathbf{s}_j)))_{n \times n}, \tag{24}$$

$$\mathbf{v}_x = (v(\mathbf{s}_1, \mathbf{x}), \dots, v(\mathbf{s}_n, \mathbf{x}))^T, \tag{25}$$

$$\mathbf{F} = (f_j(\mathbf{s}_i))_{n \times k}. \tag{26}$$

Under the constraint (4), we get

$$\mathbf{c} = \mathbf{A}^{21} \mathbf{f}_x + \mathbf{A}^{22} \mathbf{v}_x, \tag{27}$$

where

$$\mathbf{A}^{21} = \mathbf{V}^{-1} \mathbf{F} (\mathbf{F}^T \mathbf{V}^{-1} \mathbf{F})^{-1}$$

and

$$\mathbf{A}^{22} = \mathbf{V}^{-1} - \mathbf{V}^{-1} \mathbf{F} (\mathbf{F}^T \mathbf{V}^{-1} \mathbf{F})^{-1} \mathbf{F}^T \mathbf{V}^{-1}.$$

Then the MSE-optimal predictor (i.e., the best linear unbiased predictor (BLUP)) is

$$\widehat{y}(\mathbf{x}) = \mathbf{f}_x^T \widehat{\beta} + \mathbf{v}_x^T \widehat{\gamma}, \tag{28}$$

where $\widehat{\beta} = (\mathbf{F}^T \mathbf{V}^{-1} \mathbf{F})^{-1} \mathbf{F}^T \mathbf{V}^{-1} \mathbf{y}_s$ and $\widehat{\gamma} = \mathbf{v}_x^T \mathbf{V}^{-1} (\mathbf{y}_s - \mathbf{F} \widehat{\beta})$. Function $f_i(\mathbf{x})$ in (1) is usually defined with polynomials of orders 0, 1, and 2. More specific, with x_j denoting the j th component of \mathbf{x} ,

Constant, $p = 1$:

$$f_1(\mathbf{x}) = 1 \tag{29}$$

Linear, $p = n + 1$:

$$f_1(\mathbf{x}) = 1, f_2(\mathbf{x}) = x_1, \dots, f_{n+1}(\mathbf{x}) = x_n, \quad (30)$$

Quadratic, $p = \frac{1}{2}(n+1)(n+2)$:

$$\begin{aligned} f_1(\mathbf{x}) &= 1 \\ f_2(\mathbf{x}) &= x_1, \dots, f_{n+1}(\mathbf{x}) = x_n \\ f_{n+2}(\mathbf{x}) &= x_1^2, \dots, f_{2n+1}(\mathbf{x}) = x_1 x_n \\ f_{2n+2}(\mathbf{x}) &= x_2^2, \dots, f_{3n}(\mathbf{x}) = x_2 x_n \\ &\dots \dots f_p(\mathbf{x}) = x_n^2 \end{aligned} \quad (31)$$

The *regpoly0* and *regpoly1* corresponds to (29) and (30) respectively in MATLAB toolbox developed by Lophaven et al. (2002).

A.4 ε -SVR

Given the data set $\{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_l, y_l)\}$ (where l denotes the number of samples) and the kernel matrix $K_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$, and if the loss function in SVR is ε -insensitive loss function

$$L_\varepsilon(f(\mathbf{x}) - y) = \begin{cases} 0, & |f(\mathbf{x}) - y| < \varepsilon \\ |f(\mathbf{x}) - y| - \varepsilon, & \text{other} \end{cases}, \quad (32)$$

then the ε -SVR is written as:

$$\min \Phi(\mathbf{w}, \xi) = \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{i=1}^l (\xi_i^- + \xi_i^+) \quad (33)$$

$$s.t. \begin{cases} f(\mathbf{x}_i) - y_i \leq \varepsilon + \xi_i^+ \\ y_i - f(\mathbf{x}_i) \leq \varepsilon + \xi_i^- \\ \xi_i^-, \xi_i^+ \geq 0, \end{cases} \quad i = 1, \dots, l.$$

The Lagrange dual model of the above model is expressed as:

$$\begin{aligned} \min_{\alpha^{(*)}} & \frac{1}{2} \sum_{i,j=1}^l (\alpha_i - \alpha_i^*) (\alpha_j - \alpha_j^*) K(\mathbf{x}_i, \mathbf{x}_j) \\ & - \sum_{i=1}^l (\alpha_i - \alpha_i^*) y_i + \varepsilon \sum_{i=1}^l (\alpha_i + \alpha_i^*) \end{aligned} \quad (34)$$

$$s.t. \begin{cases} 0 \leq \alpha_i, \alpha_i^* \leq C, i = 1, \dots, l, \\ \sum_{i=1}^l (\alpha_i - \alpha_i^*) = 0. \end{cases}$$

where $K(\cdot, \cdot)$ is kernel function. After being worked out the parameter $\alpha^{(*)}$, the regression function $f(\mathbf{x})$ can be gotten.

The kernel function should be a Mercer kernel which has to be continuous, symmetric, and positive definite. Commonly adopted choices for $K(\cdot, \cdot)$ (Smola and Schlkopf 2004) are

1. $k(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j)$ (linear)
2. $k(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j)^m$ (m degree homogeneous polynomial)

3. $k(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j + c)^m$ (m degree inhomogeneous polynomial)
4. $k(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2})$ (Gaussian)
5. $k(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\sum_{k=1}^l \theta \|\mathbf{x}_i^k - \mathbf{x}_j^k\|^{p_k})$ (Kriging)

No matter which form of $k(\cdot, \cdot)$ is chosen, the technique of finding the support vector remains the same. In all of the kernel functions mentioned above, Gaussian kernel is the most popular kernel function, which is used in this paper.

Appendix B: Box plots

In a box plot, the box is composed of lower quartile (25 %), median (50 %), and upper quartile (75 %) values. Besides the box, there are two lines extended from each end of the box, whose upper limit and lower limit are defined as follows:

$$\text{low_limit} = \max\{Q_1 - 1.5IQR, X_{\text{minimum}}\} \quad (35)$$

$$\text{up_limit} = \min\{Q_3 + 1.5IQR, X_{\text{maximum}}\} \quad (36)$$

where Q_1 is the value of the line at lower quartile, Q_3 is the value of the line at upper quartile, $IQR = Q_3 - Q_1$, X_{minimum} and X_{maximum} are the minimum and maximum value of the data. Outliers are data with values beyond the ends of the lines by placing a “+” sign for each point.

References

- Acar E, Rais-Rohani M (2009) Ensemble of metamodels with optimized weight factors. *Struct Multidiscip Optim* 37:279–294
- Box GEP, Draper NR (1987) *Empirical model building and response surface*. Wiley, New York
- Copeland KAF, Nelson PR (1996) Dual response optimization via direct function minimization. *J Qual Technol* 28:331–336
- Cressie N (1988) Spatial prediction and ordinary kriging. *Math Geol* 20(4):405–421
- De Boor C, Ron A (1990) On multivariate polynomial interpolation. *Constr Approx* 6:287–302
- Del Castillo E, Montgomery DC (1993) A nonlinear programming solution to the dual response problem. *J Qual Technol* 25:199–204
- Fang KT, Li R, Sudjianto A (2006) *Design and modeling for computer experiments*. CRC Press, New York
- Friedman JH (1991) Multivariate adaptive regressive splines. *Ann Stat* 19(1):1–67
- Goel T, Haftka RT, Shyy W, Queipo NV (2007) Ensemble of surrogates. *Struct Multidiscip Optim* 33(3):199–216
- Jin R, Chen W, Simpson TW (2001) Comparative studies of meta-modeling techniques under multiple modeling criteria. *Struct Multidiscip Optim* 23(1):1–13
- Kksoy O, Yalcinoz T (2005) A hopfield neural network approach to the dual response problem. *Qual Reliab Eng Int* 21:595–603
- Kohavi R et al. (1995) A study of cross-validation and bootstrap for accuracy estimation and model selection. In: *Ijcai*, vol 14, pp 1137–1145

- Langley P, Simon HA (1995) Applications of machine learning and rule induction. *Commun ACM* 38(11):55–64
- Lin DKJ, Tu W (1995) Dual response surface optimization. *J Qual Technol* 27:34–39
- Lophaven SN, Nielsen HB, Sndergaard J (2002) Dace-a matlab kriging toolbox. Technical report IMM-TR-2002-12, Technical University of Denmark
- McDonald DB, Grantham WJ, Tabor WL, Murphy M (2000) Response surface model development for global/local optimization using radial basis functions. In: Proceedings of the eighth AIAA/USAF/NASA/ISSMO symposium on multidisciplinary analysis and optimization, Long Beach, CA
- Meckesheimer M, Barton RR, Simpson TW, Limayem F, Yannou B (2001) Metamodeling of combined discrete/continuous responses. *AIAA J* 39(10):1950–1959
- Meckesheimer M, Barton RR, Simpson TW, Booker AJ (2002) Computationally inexpensive metamodel assessment strategies. *AIAA J* 40(10):2053–2060
- Picard RR, Cook RD (1984) Cross-validation of regression models. *J Am Stat Assoc* 79(387):575–583
- Powell MJ (1987) Radial basis functions for multivariable interpolation: a review. In: Algorithms for approximation. Clarendon Press, pp 143–167
- Queipo NV, Haftka RT, Shyy W, Goel T, Vaidyanathan R, Tucker PK (2005) Surrogate-based analysis and optimization. *Prog Aerosp Sci* 41:1–28
- Sacks J, Welch WJ, Mitchell TJ, Wynn HP (1989) Design and analysis of computer experiments. *Stat Sci* 4(4):409–423
- Shaibu AB, Cho BR (2009) Another view of dual response surface modeling and optimization in robust parameter design. *Int J Adv Manuf Technol* 41:631–641
- Smola AJ, Schlkopf B (2004) A tutorial on support vector regression. *Stat Comput* 14(3):199–222
- Viana FA, Gogu C, Haftka RT (2010) Making the most out of surrogate models: tricks of the trade. In: ASME 2010 international design engineering technical conferences and computers and information in engineering conference, American Society of Mechanical Engineers, pp 587–598
- Viana FAC, Haftka RT, Steffen V (2009) Multiple surrogate: how cross-validation errors can help us to obtain the best predictor. *Struct Multidiscip Optim* 39:439–457
- Vining GG, Bohn L (1998) Response surfaces for the mean and variance using a nonparametric approach. *J Qual Technol* 30:282–291
- Vining GG, Myers RH (1990) Combining taguchi and response surface philosophies: a dual response approach. *J Qual Technol* 22:38–45
- Zerpa L, Queipo NV, Pintos S, Salager J (2005) An optimization methodology of alkaline-surfactant-polymer flooding processes using field scale numerical simulation and multiple surrogates. *J Pet Sci Eng* 47:197–208
- Zhou XJ, Ma YZ, Li XF (2011) Ensemble of surrogates with recursive arithmetic average. *Struct Multidiscip Optim* 44(5):651–671
- Zhou XJ, Ma YZ, Tu YL, Feng Y (2012) Ensemble of surrogates for dual response surface modeling in robust parameter design. *Qual Reliab Eng Int* 29(2):173–197