ICIC Express Letters
Volume **11**, Number **11**, November **2017**
 ISSN 1881-803X
pp. **1661–1668**

CONSTRUCTING MULTI-OUTPUT REGRESSION MODEL WITH TUNABLE KERNEL USING GROUP SEARCH OPTIMIZER

Xingyi Chen[1], Lihua Fu[2,*] and Zhihui Liu[2]

[1]Faculty of Information Engineering
[2]School of Mathematics and Physics
China University of Geosciences
No. 388, Lumo Road, Wuhan 430074, P. R. China
chenxyi@sohu.com; zhhliu@cug.edu.cn; *Corresponding author: lihuafu@cug.edu.cn

Received May 2017; accepted July 2017

Abstract. *Most of the multi-output regression models with Gaussian kernel adopt a fixed and predefined scale parameter for all regressors. These models become problematic when fitting noisy samples of a function with time-varying dynamics. This paper proposes to tune the center vector and scale parameter of Gaussian kernels term by term. Specially, we employ a greedy scheme to construct the kernel model incrementally by minimizing norm of the residual matrix using group search optimizer (GSO). Compared with the previous multi-output kernel models, the new algorithm can generate very sparse model with better generality.*
Keywords: Orthogonal forward selection, Multiple output regression, Tunable kernel, Kernel model

1. **Introduction.** In multiple output regression problems, a set of input and target observations $\{\mathbf{x}_k, \mathbf{y}_k\}_{k=1}^N$ is given with $\mathbf{x}_k \in \mathrm{R}^m$, $\mathbf{y}_k \in \mathrm{R}^n$. Unlike the ordinary single output case, the target values $\mathbf{y}_k$'s are vectors rather than scalar quantities, which are considered as the noisy output of a function f: $\mathrm{R}^m \to \mathrm{R}^n$ as

$$\mathbf{y}_k = f(\mathbf{x}_k) + \boldsymbol{\varepsilon}_k, \tag{1}$$

where $\boldsymbol{\varepsilon}_k \in \mathrm{R}^n$ is a measurement noise vector. The multiple regression framework aims to learn a mapping $f(\mathbf{x}_k)$ from the input space to an n-dimensional output. And it has been found in many applications, such as pose estimation [1,2] and viewpoint estimation [3] in computer vision, time series prediction [4], robot control [5,6] and biological data processing [7]. Moreover, some applications, e.g., camera relocalization and cardiac volume estimation can be effectively solved by transferring the original problem into a multi-output regression task [8-10].

Recent developments in kernel methods for vector outputs show great promise to analyze the relationship between input data and vector-valued output [11-15]. They typically assume a linearly weighted model for $f(\mathbf{x}_k)$ as

$$f(\mathbf{x}) = \mathbf{w}_0 + \sum_{k=1}^{N} \mathbf{w}_k K(\mathbf{x}, \mathbf{x}_k). \tag{2}$$

Among all the kernels, the Gaussian function is the most popular because of its good locality and generality

$$K(\mathbf{x}, \mathbf{x}_k) = \exp\left\{-\frac{||\mathbf{x} - \mathbf{x}_k||^2}{2\sigma^2}\right\} \text{ for } k = 1, 2, \ldots, N, \tag{3}$$

where σ is the kernel width. Note that the kernel width σ is invariant for all the training samples in the standard kernel machines for vector-valued outputs [11-14].

The training of multiple output regression model with Gaussian kernel function involves the optimization of three kinds of parameters: kernel centres, kernel scales, and the connecting weights between these kernels. In [11], two general dimensional multiple output support vector regressions (MSVRs) named SOCPL1 and SOCPL2 were proposed, which differ at the first-order and second-order loss. In MSVRs, all the training samples are assigned as the kernel centres while all terms in (2) share a same scale parameter which is decided before training by cross validation. The weight and bias parameters are optimized together. In [12], relevance vector machine (RVM) was generalized to multivariate RVM (MRVM), which decides the parameters by alternatively minimizing the regression loss and estimating the corresponding probabilistic matrix. Although they excel the state-of-the-art algorithms in many aspects, both MRVM and MSVRs suffer the limitations that fail to assign the kernel centres flexibly. Actually, both models choose fixed scale parameters for all the terms before training. However, for a dataset that consists of noisy samples of a function with time-varying dynamics (e.g., Doppler signals or speech signals), choosing a large kernel width will result in a predicted response, which is smoothed in the high-frequency subdomains of the dynamic function. On the other hand, selecting a small kernel width will yield a predicted response, which is over-fitted in the low frequency subdomain of the dynamic function.

In this paper, we propose a novel multi-output tunable kernel model (MTKM) for the regression problem. Instead of (2), MTKM is formulated as

$$f(\mathbf{x}) = \mathbf{w}_0 + \sum_{k=1}^{L} \mathbf{w}_k \phi_k(\mathbf{x}) \tag{4}$$

with

$$\phi_k(\mathbf{x}) = K_k(\mathbf{x}, \boldsymbol{\mu}_k) = \exp\left\{-\frac{||\mathbf{x} - \boldsymbol{\mu}_k||^2}{2\sigma_k^2}\right\} \text{ for } k = 1, \ldots, N. \tag{5}$$

Compared with the traditional kernel machines in (2), MTKM tunes kernel scale and center adaptively. Thus, MTKM is much more flexible to tune the kernel parameters than the traditional kernel models.

Specially, MTKM uses orthogonal forward selection [16] (OFS) to construct a regression model incrementally with a greedy scheme. At each regressor stage, MTKM tunes the optimal kernel width and center to minimize the training error with the aid of group search optimizer (GSO) [17]. GSO is a population-based heuristic optimization algorithm, which is simple to implement but extremely efficient to solve the high dimensional optimization problems, such as neural networks constructing. Other optimization algorithms can be used alternatively [18]. The experimental results on both simulated dataset and real seismic records show the new approach is efficient.

This paper is organized as follows. In Section 2, multi-output orthogonal forward selection is introduced. The detailed description of the proposed algorithm is presented in Section 3. The experimental results are simulated and discussed in Section 4. Finally, conclusions are given in Section 5.

2. Multi-Output Orthogonal Forward Selection.

The multi-output regression model can be formulated as [16]

$$\mathbf{Y} = \boldsymbol{\Phi}\mathbf{W} + \mathbf{E}. \tag{6}$$

Here $\mathbf{Y} = [\mathbf{y}_1\ \mathbf{y}_2 \cdots \mathbf{y}_N]^T$ is the output matrix, $(\boldsymbol{\Phi})_{i,j} = \phi_i(\mathbf{x}_j)$ is the kernel matrix, the weight matrix is denoted by $\mathbf{W} = [\mathbf{w}_1\ \mathbf{w}_2 \cdots \mathbf{w}_L]^T$, and the residual matrix is $\mathbf{E} = [\mathbf{e}_1\ \mathbf{e}_2 \cdots \mathbf{e}_N]^T$. Let an orthogonal decomposition of the regression matrix $\boldsymbol{\Phi}$ be

$$\boldsymbol{\Phi} = \mathbf{P}\mathbf{A}, \tag{7}$$

where $\mathbf{A}$ is an upper triangular matrix with the unit diagonal element and $\mathbf{P} = [\mathbf{p}_1\ \mathbf{p}_2 \cdots \mathbf{p}_M]$ with the orthogonal columns that satisfy $\mathbf{p}_i^T \mathbf{p}_j = 0$ if $i \neq j$.

The regression model (6) can alternatively be expressed as

$$\mathbf{Y} = \mathbf{P}\boldsymbol{\Theta} + \mathbf{E} \tag{8}$$

with the new weight matrix $\boldsymbol{\Theta}$ that satisfies the triangular system $\boldsymbol{\Theta} = \mathbf{AW}$.

The $L_{2,1}$ norm of the error matrix is

$$\|\mathbf{E}\|_{2,1} = \|\mathbf{Y} - \mathbf{P}\boldsymbol{\Theta}\|_{2,1}^2 = \sum_{k=1}^{M} \left\|\mathbf{y}^k - \mathbf{P}\theta^k\right\|_2^2, \tag{9}$$

where the vectors $\mathbf{y}^k$ and θ^k are the k-th columns of $\mathbf{Y}$ and $\boldsymbol{\Theta}$ respectively. And

$$\left\|\mathbf{y}^k - \mathbf{P}\theta^k\right\|_2^2 = \left(\mathbf{y}^k\right)^T \mathbf{y}^k - \left(\mathbf{P}\theta^k\right)^T \left(\mathbf{P}\theta^k\right). \tag{10}$$

At the L-th forward stage, the reduction of $\|\mathbf{E}\|_{2,1}$ is

$$\mathrm{Red}_L = \sum_{k=1}^{M} \mathbf{p}_L^T \mathbf{p}_L \theta_L^k \theta_L^k. \tag{11}$$

The OFS applies a greedy scheme to achieving the sparse kernel model. At each step, we minimize the reduction in (11) to obtain a new model term. To be more specially, at the L-th forward stage, we solve the following optimization problem

$$\min_{\mathbf{u}_L} \mathrm{Red}_L \text{ with } \mathbf{u}_L^T = \begin{bmatrix} \boldsymbol{\mu}_L^T & \sigma_L^T \end{bmatrix}. \tag{12}$$

3. Algorithm.

3.1. Group search optimizer.

In this paper, we apply GSO to obtaining the solution of (12) [16]. GSO is a population-based heuristic optimization algorithm, which is suitable to tackle the high dimensional problem. It adopts the producer-scrounger (PS) model metaphorically for designing optimum searching strategies, inspired by animal foraging behavior. In the GSO scheme, a group consists of three types of members: producers, scroungers and rangers. Producers and scroungers' behaviors include scanning and area copying; and rangers perform random walk. In a searching iteration, the member located in the most promising area is chosen as the producer. The producer scans the environment to seek resources (optima). A number of group members are selected as scroungers, who will keep searching for opportunities to join the resources found by the producer. The rest of the group members called rangers will be dispersed from their current positions. Please refer to [17] for more detail for GSO.

3.2. MTKM with GSO.

The algorithm to select the L-th regression term is described as below.

Initialization

Let $k = 0$. Here k denotes the current generation of GSO. Generate randomly the initial positions $\mathbf{u}_p^0$ with $p = 1, \ldots, Ps$. Here Ps means the size of population.

While $k < Gen$. Here Gen is a preset maximum number of generation.

For each element in current population

Generate Producers

According to (5), generate the regression vector $\boldsymbol{\phi}_p^{(k)}$ for each kernel parameter vector $\mathbf{u}_p^{(k)}$ as the candidate of the L-th regressor, and orthogonalize it to the already-selected regression column vectors $\mathbf{p}_1 \ldots \mathbf{p}_{L-1}$,

$$\alpha_{j,k}^{(p)} = \frac{\mathbf{p}_j^T \boldsymbol{\phi}_k^{(p)}}{\mathbf{p}_j^T \mathbf{p}_j}, \quad 1 \le j < L, \tag{13}$$

$$\mathbf{p}_k^{(p)} = \mathbf{g}_k^{(p)} - \sum_{j=1}^{L-1} \alpha_{j,k}^{(p)} \mathbf{p}_j. \tag{14}$$

For $1 \leq p \leq Ps$, calculate the cost function $J_k^{(p)}$ for each $\mathbf{u}_p^k$. The weight in the orthogonal system can be calculated as

$$\theta_k^{(p)} = \frac{\left(\mathbf{p}_k^{(p)}\right)^T \mathbf{Y}}{\left(\mathbf{p}_k^{(p)}\right)^T \mathbf{p}_k^{(p)}}. \tag{15}$$

And the error in (10) is

$$J_k^{(p)} = J_{L-1} - \frac{1}{N} \left(\mathbf{p}_k^{(p)}\right)^T \mathbf{p}_k^{(p)} \left(\theta_k^{(p)}\right)^T \mathbf{1}. \tag{16}$$

The vector $\mathbf{1}$ denotes an all-ones vector with the proper size.

Perform Producing

There are 4 steps to perform producing

(1) Scan at zero degree

$$\mathbf{u}_z = \mathbf{u}_p^k + r_1 l_{\max} D_p^k \left(\boldsymbol{\varphi}^k\right).$$

(2) Scan at left hand side

$$\mathbf{u}_r = \mathbf{u}_p^k + r_1 l_{\max} D_p^k \left(\boldsymbol{\varphi}^k + \mathbf{r}_2 \beta_{\max}/2\right).$$

(3) Scan at right hand side

$$\mathbf{u}_l = \mathbf{u}_p^k + r_1 l_{\max} D_p^k \left(\boldsymbol{\varphi}^k - \mathbf{r}_2 \beta_{\max}/2\right),$$

where $D_p^k(\cdot)$ is the angle function for the search procedure [17], and $r_1 \in R^1$ is normally distributed random number with mean 0 and standard deviation 1. $\mathbf{r}_2$ is a uniformly distributed random sequence in the range of $[0, 1]$, and $l_{\max} = \sqrt{n}$ is the radix of the search area.

(4) Update the position $\mathbf{u}_p^k$ and head angle $\boldsymbol{\varphi}^k$ as

$$\mathbf{u}_p^{k+1} = \underset{\mathbf{u}_z, \mathbf{u}_r, \mathbf{u}_l, \mathbf{u}_p^k}{\arg\min}\ J \text{ and } \boldsymbol{\varphi}^{k+1} = \boldsymbol{\varphi}^k + \mathbf{r}_2 \alpha_{\max}$$

where $\alpha_{\max}$ is the maximum head angle. After a number of iterations, let us say t, if this procedure cannot obtain a better head angle, just set $\boldsymbol{\varphi}^{k+t} = \boldsymbol{\varphi}^k$.

Perform scrounging

Randomly select 80% from the rest members to perform scrounging, that is $\mathbf{u}_i^{k+1} = \mathbf{u}_i^k + \mathbf{r}_3 \circ (\mathbf{u}_p^k - \mathbf{u}_i^k)$. Here $\circ$ is the Hadamard product. $\mathbf{r}_3$ is a uniformly random sequence in range $(0, 1)$.

Perform dispersion

For the rest members, they will be dispersed from their current position to perform ranging angle and position dispersion:

$$\boldsymbol{\varphi}^{k+1} = \boldsymbol{\varphi}^k + \mathbf{r}_2 \alpha_{\max}, \quad \mathbf{u}_i^{k+1} = \mathbf{u}_i^k + l_i \cdot D_i^k \left(\boldsymbol{\varphi}^{k+1}\right)$$

The fitness function is evaluated for all the new group members.

End For

$$k = k + 1.$$

End While.

The procedure above will produce the L-th kernel parameters $\mathbf{u}_L$, the orthogonalized regression column vector $\mathbf{p}_L$ and column of regression matrix.

4. **Experiments.** In this section, we will compare the proposed MTKM with SOCPL1, SOCPL2 [11] and multivariate relevant vector machines (MRVM) [12] in terms of training accuracy, model sparsity and generality. The parameters in GSO are empirically assigned as below. The population size Ps is 48, and the maximum iteration number $t = \text{round}\left(\sqrt{n+1}\right)$, here round$(\cdot)$ is the nearest integer function and n is the search space dimension. In the step of *Perform Producing*, the initial directional angle $\boldsymbol{\varphi}^0 = (\pi/4, \ldots, \pi/4)$, and maximum head angle $\alpha_{\max} = \beta_{\max} = \pi/t^2$. Please refer to [14] for the more detail of GSO parameter assignment.

4.1. **Complex function regression.** The first target system is a complex function as

$$\hat{f}(x) = \sin(12x)/(12x) + i \cdot \sin 2\pi x, \tag{17}$$

where $i = \sqrt{-1}$. Totally 200 samples were generated by (17) where the input data x_i's are uniformly sampled over the interval $[-1, 1]$. We split randomly the total dataset into two parts with the same size. Half of the samples are used as training dataset and the remaining are used for testing. The Gaussian white noise with zero mean and deviation 0.3 is added to the target value $\hat{f}(x)$. All algorithms are used to produce the models with Gaussian kernel to approximate the real and imaginary parts of the target values. It is a one-input and two-output problem. For SOCPL1, SOCPL2 and MRVM, the parameters are selected by grid search and the results are as below.

Table 1. The parameters of three algorithms in complex function approximation

algorithm	Insensitive $\boldsymbol{\varepsilon}$ bound	Tradeoff parameter	Kernel width
SOCPL1	0.2	100	0.4
SOCPL2	0.3	100	0.3
MRVM	–	–	0.2

Table 2. The averaged results of different methods on complex function approximation

	Training Error	Test Error	Model Size	Time Consuming(s)
SCOPL1	0.093	0.099	191.7	12.4
SCOPL2	0.085	0.101	210.3	11.2
MRVM	0.096	0.098	10.7	4.2
MTKM	0.089	0.091	5.3	0.4

Figure 1 shows the performance of MTKM, SOCPL1, SOCPL2 and MRVM. The circles in the figures show the support vectors (SVs), relevant vectors (RVs) and kernel centers of the regression models. The number of circles in each figure indicates the size of the regression models, which measures the sparseness. This experiment is repeated for 30 times and the average results are listed in Table 2. It indicates that the speed of the proposed MTKM is the greatest in all the algorithms, partly because of the efficient greedy algorithm applied in MTKM. Besides, MTKM produces the sparsest model with good generality, due to the flexible kernel width turning scheme rather than the fix parameter for all the regression terms in the other algorithms.

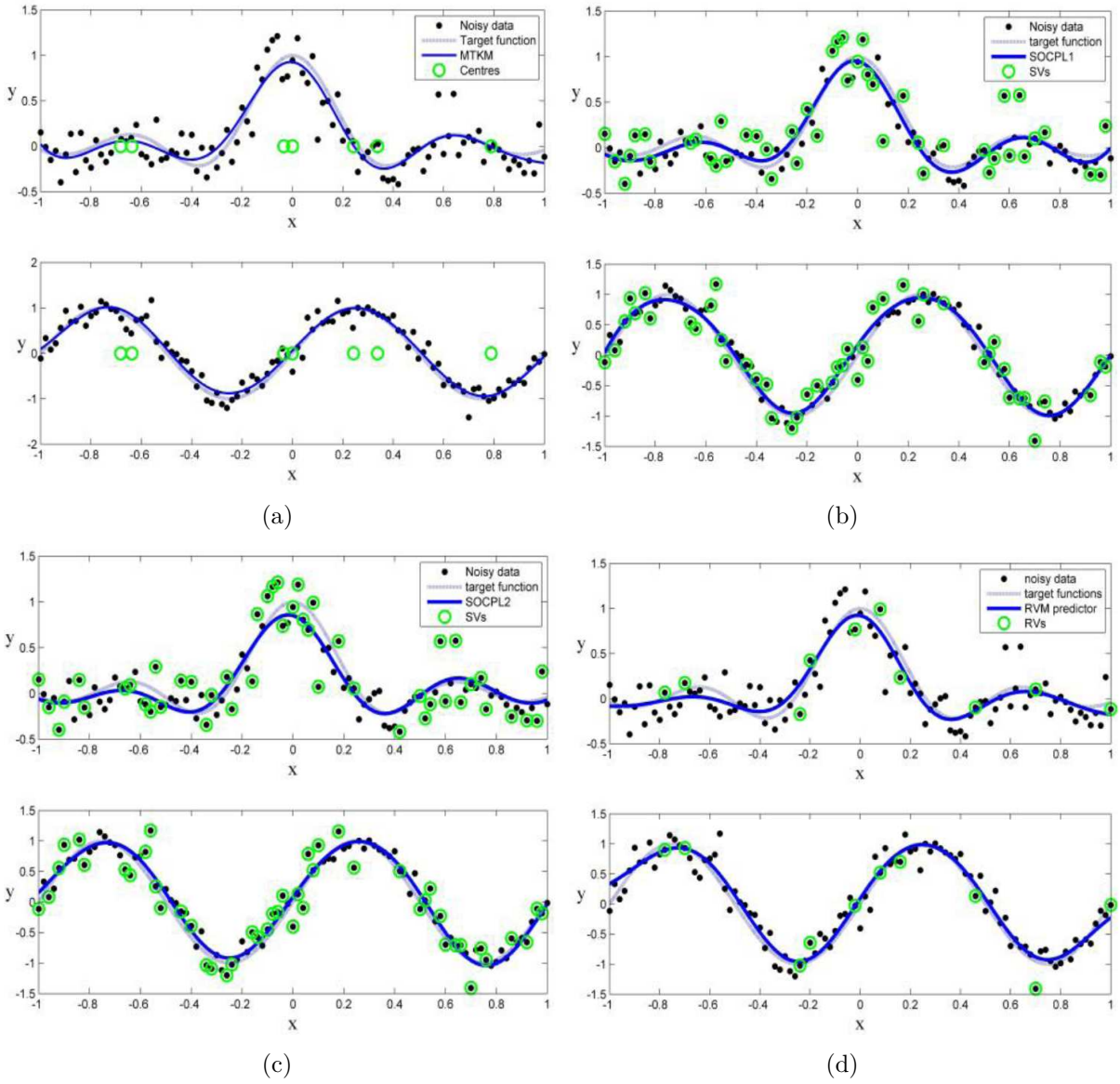

FIGURE 1. The performances of all algorithms in complex function approximation: (a) the performance of MTKM in Experiment 4.1, (b) the performance of SOCPL1 in Experiment 4.1, (c) the performance of SOCPL2 in Experiment 4.1, (d) the performance of MRVM in Experiment 4.1

4.2. **Lorenz data set [19].** The second target system is the Lorenz attractor, which is described as

$$\begin{aligned}\frac{dp_x}{dt} &= \sigma(p_y - p_x),\\ \frac{dp_y}{dt} &= p_x(r - p_z) - p_y,\\ \frac{dp_z}{dt} &= p_x p_y - bp_z.\end{aligned}$$

Here the state of the system is initialized as $\mathbf{x} = [p_x, p_y, p_z]^T = [1, 2, 3]^T$. The system factors are selected as $\sigma = 3$, $r = 26.5$, $b = 1$, $\tau = 0.02s$. This study generates a total of 1600 data samples. And the 600th-999th data points are used as a training set, and the 1000th to the 1166 data samples are used for testing. The first 599 samples are discarded because they are in the transient state and their behaviors are completely different from

the remaining samples. We use the current and previous state $\mathbf{x}_t$ and the previous state $\mathbf{x}_{t-\tau}$ to predict the eight-step-ahead state $\mathbf{x}_{t+8\tau}$. Thus, the target system is a three-output system. We normalize all data point to mean of zero and unit standard deviation. Besides, Gaussian white noise with $\sigma = 0.5$ is added to the samples. As in Experiment 4.1, we repeat this experiment for 30 times, and the averaged results are listed in Table 3.

TABLE 3. The averaged results of different methods on Lorenz system modeling

	Training Error	Test Error	Model Size	Time Consuming (s)
SCOPL1	9.765	10.832	202.7	242.1
SCOPL2	8.061	11.475	361.3	291.2
MRVM	12.261	11.238	95.7	264.4
MTKM	9.716	9.914	17.3	24.5

4.3. **Seismic record data set.** The east Texas seismic record data set [20] is used to evaluate the proposed MTKM algorithm. Seismic records are the reflected signals when the artificial explosion wave propagates through earth layers. The strength of the reflected signal depends on the impedance contrast between adjacent layers. This data set is known to contain high noisy amplitude traces. This is due to the use of bad geophones. Seismic records usually contain the information of the complex earth layers structure, which makes them time-varying dynamics signals. The left panel in Figure 2 shows the seismic section. The horizontal axis represents the offset of each seismic receiver (recorder) from the source where each records a trace with respect to the two-way travel time (vertical axis). There are totally 33 traces in this data set, which result in a 33-output regression problem. The middle and right panels in Figure 2 show the regression model and the corresponding residual. It indicates that MTKM can obtain a good approximation for seismic records.

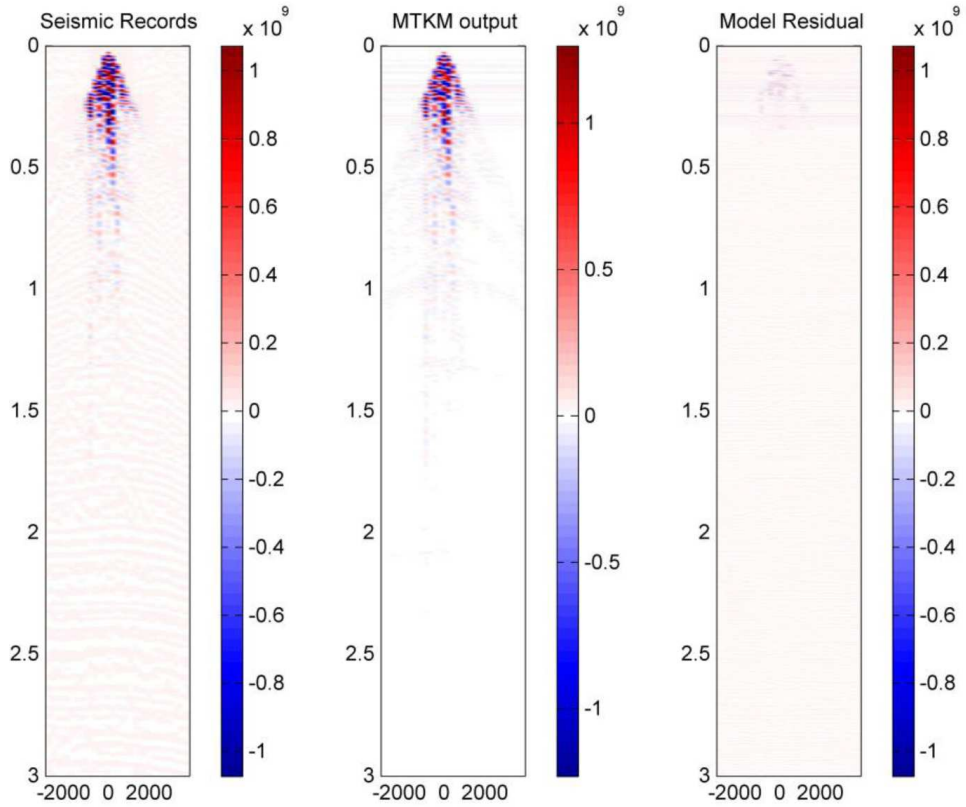

FIGURE 2. Performance of MTKM on seismic records approximation

5. **Conclusions.** This paper introduces a novel multi-output regression model with tunable kernel, MTKM. This new algorithm is capable of approximating the multiple traces of signals simultaneously. Compared with the state-of-the-art methods, the new scheme can result in very sparse kernel model with good generality. As a continuous effort, our future work will focus on the criteria to build the multi-output kernel regression machines, such as L1 and Lp criteria.

Acknowledgment. This work is supported by Open Research Project of the Hubei Key Laboratory of Intelligent Geo-Information Processing (KLIGIP2016A01, KLIGIP2016A-02) and the specific funding for education science research by self-determined research funds of CCNU from the colleges' basic research and operation of MOE with grants 230-20205160288 and CCNU15A05022.

REFERENCES

[1] A. Toshev and C. Szegedy, Deeppose: Human pose estimation via deep neural networks, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.1653-1660, 2014.
[2] B. Hara and R. Chellappa, Growing regression forests by classification: Applications to object pose estimation, *European Conference on Computer Vision (ECCV)*, pp.552-567, 2014.
[3] M. Torki and A. Elgammal, Regression from local features for viewpoint and pose estimation, *International Conference on Computer Vision (ICCV)*, pp.2603-2610, 2011.
[4] C. Li, T. Nguyen, M. Yang, S. Yang and S. Zeng, Multi-population methods in unconstrained continuous dynamic environments: The challenges, *Information Sciences*, vol.296, pp.95-118, 2015.
[5] D. Tuong and J. Peters, Model learning for robot control: A survey, *Cognitive Processing*, vol.12, no.4, pp.319-340, 2011.
[6] Y. Zhou, Z. Fei, S. Yang, J. Kuang, S. Chen and L. Hanzo, Joint angle estimation and signal reconstruction for coherently distributed sources in massive MIMO systems based on 2D unitary ESPRIT, *IEEE Access*, vol.5, pp.9632-9646, 2017.
[7] M. Gonen and S. Kaski, Kernelized Bayesian matrix factorization, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol.36, no.10, pp.2047-2060, 2014.
[8] X. Zhen, Z. Wang, M. Yu and S. Li, Supervised descriptor learning for multi-output regression, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.1211-1218, 2015.
[9] S. Gong, C. Xing, S. Chen and Z. Fei, Secure communications for dual-polarized MIMO systems, *IEEE Trans. Signal Processing*, vol.65, no.16, pp.4177-4192, 2017.
[10] Y. Zhang, J. Wu, Z. Cai, P. Zhang and L. Chen, Memetic extreme learning machine, *Pattern Recognition*, vol.58, pp.135-148, 2016.
[11] W. Chang, J. Kim, H. Lee and E. Kim, General dimensional multiple-output support vector regressions and their multiple kernel learning, *IEEE Trans. Cybernetics*, vol.45, no.11, pp.2572-2584, 2015.
[12] A. Thayananthan, R. Navaratnam, B. Stenger et al., Multivariate relevance vector machines for tracking, *European Conference on Computer Vision*, vol.3953, pp.124-138, 2006.
[13] S. Xu, X. An and X. Qiao, Multi-output least-squares support vector regression machines, *Pattern Recognition Letters*, vol.34, no.9, pp.1078-1084, 2013.
[14] L. Fu, H. Li and M. Zhang, Improved RBF networks with multi-kernel, *ICIC Express Letters*, vol.4, no.4, pp.1331-1336, 2010.
[15] M. Zhang, S. Yu and L. Fu, Generalized Gaussian support vector regression, *ICIC Express Letters*, vol.5, no.12, pp.4525-4534, 2011.
[16] S. Chen, X. Hong and C. Harris, Sparse multi-output radial basis function network construction using combined locally regularised orthogonal least square and D-optimality experimental design, *IEE Proc. of Control Theory and Applications*, vol.150, no.2, pp.139-146, 2003.
[17] S. He, Q. Wu and J. Saunders, Group search optimizer: An optimization algorithm inspired by animal searching behavior, *IEEE Trans. Evolutionary Computation*, vol.13, no.5, pp.973-990, 2009.
[18] W. Gong, Z. Cai and D. Liang, Adaptive ranking mutation operator based differential evolution for constrained optimization, *IEEE Trans. Cybernetics*, vol.45, pp.716-727, 2015.
[19] E. Lorenz, Deterministic nonperiodic flow, *Journal of the Atmospheric Sciences*, vol.20, pp.130-141, 1963.
[20] W. Mousa and A. Al-shuhail, *Proc. of Seismic Reflection Data Using MATLAB*, Morgan & Claypool Publishers, San Rafael, CA, 2012.