IOP Publishing Measurement Science and Technology
Meas. Sci. Technol. **31** (2020) 065017 (14pp) https://doi.org/10.1088/1361-6501/ab6a46

Source number estimation based on a novel multi-view meta-hierarchical classification framework

Wu Yun[1,2,3,4], Li Xiukun[1,3,4] and Cao Zhimin[2]

[1] Acoustic Science and Technology Laboratory, Harbin Engineering University, Harbin, Heilongjiang 150001, People's Republic of China
[2] School of Electronic Science, Northeast Petroleum University, Daqing, Heilongjiang, People's Republic of China
[3] Key Laboratory of Marine Information Acquisition and Security, Ministry of Industry and Information Technology, Harbin Engineering University, Harbin 150001, People's Republic of China
[4] College of Underwater Acoustic Engineering, Harbin Engineering University, Harbin 150001, People's Republic of China

E-mail: lixiukun@hrbeu.edu.cn

Received 30 August 2019, revised 25 November 2019
Accepted for publication 10 January 2020
Published 9 April 2020

Abstract

Accurately measuring the direction of arrival (DOA) is one of the most important issues in multiple sensor/antenna array monitoring scenarios. However, as a necessary parameter of almost all state-of-the-art DOA estimation methods, the source number is normally hard to determine using the traditional Akaike information criterion or minimum description length methods, especially in low or very low signal-to-noise ratio (SNR) conditions. In this paper, we propose to estimate the source number in a data-driven manner by employing a novel multi-view meta-hierarchical classification framework. Specifically, there are two collaborative views and two hierarchical classification layers employed for generating meta-features. Then, the obtained meta-features are re-learned by the final meta-classification layer to estimate the final prediction of the source number. Experimental results illustrated that the proposed method can estimate the source number accurately and reliably even in low SNR conditions.

Keywords: DOA, source number, data-driven, hierarchical classification, meta classification

(Some figures may appear in colour only in the online journal.)

1. Introduction

Direction of arrival (DOA) estimation in array signal processing has been one of the most important issues in various fields of location measurement applications, such as airspace radar array [1–3], underwater sonar array [4], speech recognition [5], wireless sensor networks [6], etc.

In order to achieve the objective of accurate DOA estimation, many well-known high-resolution DOA estimation methods have been proposed in the literature. Methods such as singular value decomposition (SVD) to the covariance matrix of the array outputs, estimation of signal parameters via rotational invariance techniques [7], multiple signal classification (MUSIC) [8] and their derived subspace-based algorithms can really work well with known source number [9–13]. Similarly, the performances of the advanced sparse representation or sparse Bayesian learning-based methods [14–19] are also heavily dependent on the correct estimation of the source number. Besides these mainstream techniques, it has been demonstrated that promising results can be achieved by employing a hypothesis testing technique [20, 21]. In general, all of these methods achieve success by analyzing the physical array model in different views. Therefore, the general drawbacks of these methods are that their performance is sensitive to unavoidable array errors and influences from system and random noise.

For the source number estimation problem, Akaike information criterion (AIC) [22] and minimum description

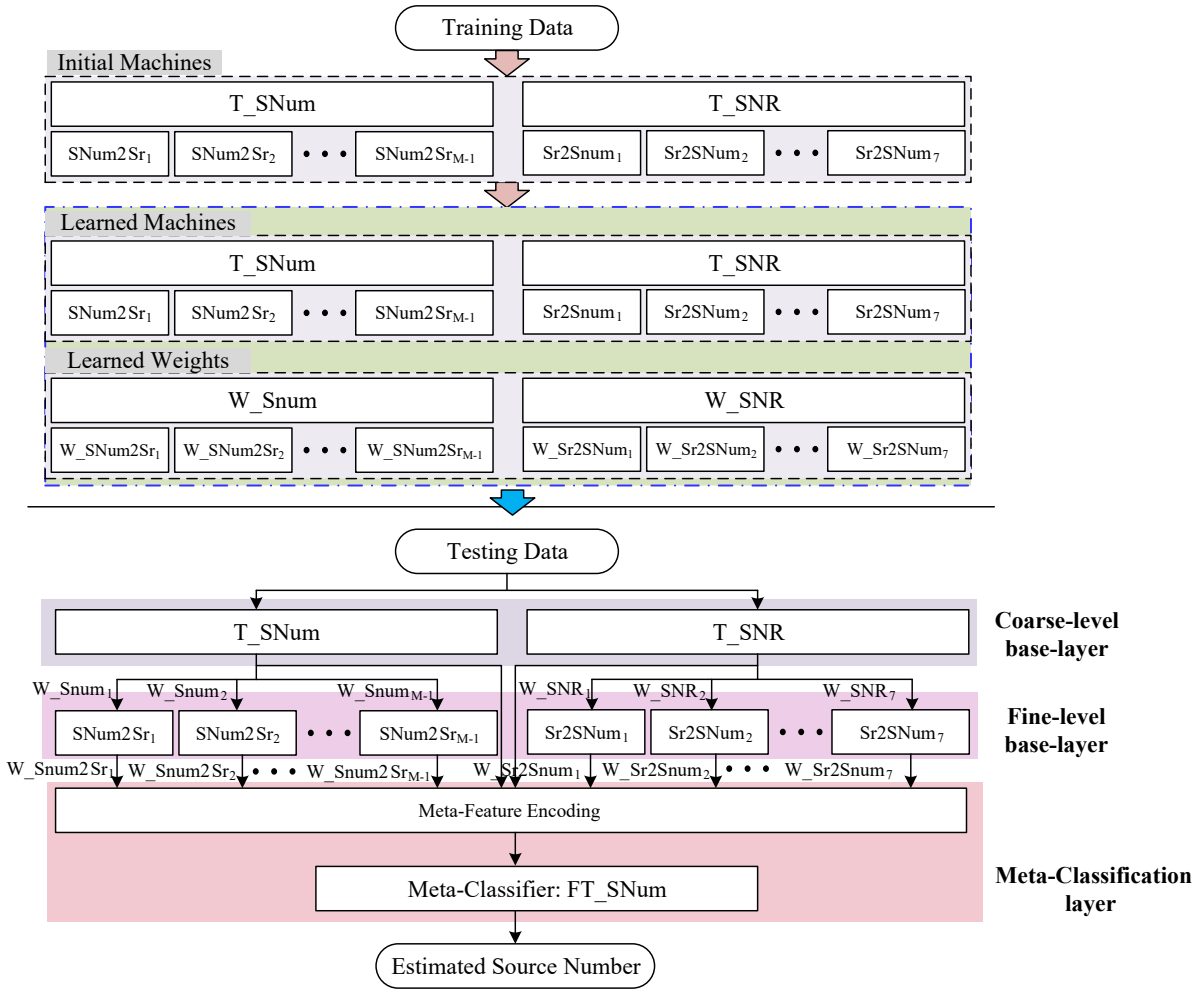

Figure 1. Block diagram of the proposed source number estimation method (the upper part is for offline training, and the lower part is the source number estimation part).

length (MDL) [23] are two well-known information-based source number estimation techniques that are commonly used in practice. However, the corresponding performance of these two information-based methods will drop dramatically with small sample size and low signal-to-noise ratio (SNR). In order to tackle this problem, there are mainly two kinds of methods that exist in the literature. The first method is designed from some new perspectives to avoid the source number estimation problem. For example, [24] proposed a novel MUSIC-like DOA estimation method, which is not based on subspace decomposition, and thus can avoid estimating the source number during the DOA estimation procedure. The secondmethod is to adopt novel signal models for source number estimation by introducing new technologies that are not commonly used for traditional source number estimation methods. For example, by employing a hypothesis testing technique, [21] proposed a method to jointly estimate the source number and target DOAs by solving a sequential generalized likelihood ratio test. However, the complexity and performance of these methods are far from expected without directly employing knowledge about the source number, and their performance is not always satisfactory in the low SNR cases.

Recently, in order to address this problem, there have been studies in the literature that employ data-driven technologies, such as deep learning for DOA estimation [25–27] or hierarchical artificial intelligent units for source number estimation [28]. However, the performance in cases of low or very low SNR (<−10 dB) have not been verified enough in the current deep learning-based methods. Our previous work [28] hierarchically employed different artificial intelligence (AI) units to estimate the SNR and source number simultaneously. Promising results can be achieved compared to the well-known AIC or MDL methods, especially at low SNR and from more than four sources. However, because the hierarchical AI units were performed iteratively in an interactive verification manner, the corresponding network structure is relatively complex. Furthermore, the information obtained by each sub-machine employed was integrated in a simple empirical hypothesis-test manner. Therefore, the corresponding source number estimation accuracy is also relatively low in the case of low SNR, especially for cases with more than three sources. Therefore, in order to improve the performance of the state-of-the-art methods, a novel data-driven-based source number estimation method is proposed. Specifically, in order to achieve an optimal feature representation for revealing the

relationship between array outputs and the corresponding source number, the source number estimation problem is tackled with a novel two-view meta-hierarchical classification/pattern recognition framework in this paper, which is illustrated in figure 1.

The rest of this paper is organized as follows. In section 2, the problem formulation section of the task in this paper is given. The proposed source number estimation method is presented in detail in section 3. Simulation experiments were conducted, and the corresponding experimental results are discussed in section 4. Finally, some remarkable conclusions from this paper are drawn in section 5.

2. Classical physical model

Without loss of generality, assume that K narrowband far-field equal-powered signals from distinct directions $\boldsymbol{\theta} = \{\theta_1, \theta_2, \cdots, \theta_K\}$ impinge onto an array with M $(M > K)$ omnidirectional sensors. The output of this array with N_s snapshots, which is corrupted by additive circular complex Gaussian white noise, can be modeled as

$$\mathbf{y}(t) = \sum_{i=1}^{K} \mathbf{a}(\theta_i)\,\mathbf{s}_k(t) + \mathbf{n}(t),\ t = 1, 2, \cdots, N_s, \tag{1}$$

where $\mathbf{y}(t) = [y_1(t), y_2(t), \cdots, y_M(t)]^T$ is the array output, $\mathbf{s}(t) = [\mathbf{s}_1(t), \mathbf{s}_2(t), \cdots, \mathbf{s}_K(t)]^T$ is the impinging signals, $\mathbf{n}(t)$ is the corresponding independent unit Gaussian noise with zero mean and variance matrix $\sigma^2\mathbf{I}$, where σ^2 can be used to define the SNR value with the normalized source power (e.g. 1) as below:

$$SNR = 10 \cdot \log_{10}\left(\frac{1}{\sigma^2}\right). \tag{2}$$

$\mathbf{a}(\theta_i)$ is the steering vector for the ith DOA and the given array geometry, which can be formulated as

$$\mathbf{a}(\theta_i) = \left[e^{j\omega_c\tau_{i,1}}, e^{j\omega_c\tau_{i,2}}, \cdots, e^{j\omega_c\tau_{i,M}}\right]^T, \tag{3}$$

where ω_0 is the center angular frequency of the impinging signals. $\tau_{i,j}$ $(j = 1, 2, \ldots, M)$ is the time delay associated with the jth senor and the ith impinging signal.

With the above definitions, let

$$\mathbf{A}(\boldsymbol{\theta}) = [\mathbf{a}(\theta_1), \mathbf{a}(\theta_2), \cdots, \mathbf{a}(\theta_K)]. \tag{4}$$

The signal model (1) can therefore be formulated in the following matrix form:

$$\mathbf{Y} = \mathbf{A}(\boldsymbol{\theta})\,\mathbf{S} + \mathbf{N}, \tag{5}$$

where $\mathbf{Y} = [\mathbf{y}(t_1), \cdots, \mathbf{y}(t_N)]^T$, $\mathbf{S} = [\mathbf{s}(t_1), \cdots, \mathbf{s}(t_N)]^T$, and $\mathbf{N} = [\mathbf{n}(t_1), \cdots, \mathbf{n}(t_N)]^T$. Then, the array covariance matrix is given by

$$\mathbf{R} = E\{\mathbf{y}(t)\,\mathbf{y}^H(t)\} = \mathbf{A}(\boldsymbol{\theta})\,\mathbf{P}\mathbf{A}^H(\boldsymbol{\theta}) + \mathbf{Q}, \tag{6}$$

where $E\{\cdot\}$ is the statistical expectation operator. $\{\cdot\}^H$ denotes the Hermitian transpose operator. $K \times K$ matrix $\mathbf{P} = diag\{\lambda_1^2, \lambda_2^2, \cdots, \lambda_K^2\}$ is the diagonal covariance matrix of the impinge signals. $M \times M$ matrix $\mathbf{Q} = \sigma^2\mathbf{I}$ is the covariance matrix of Gaussian white noises for the given array.

By performing SVD to the array covariance matrix $\mathbf{R}$, the corresponding eigenvalues will be $\{\lambda_1^2 + \sigma^2, \cdots, \lambda_K^2 + \sigma^2, \underbrace{\sigma^2, \cdots, \sigma^2}_{M-K}\}$. Therefore, subspace for the first K largest eigenvalues can be seen as signal subspace. And subspace for the rest of the small eigenvalues can be seen as the noise one. Obviously, the sorted eigenvalues have important mathematical and physical implications for providing information about the source number as well as the SNR conditions.

In practice, the array covariance matrix can be estimated as below:

$$\hat{\mathbf{R}} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{y}(i)\,\mathbf{y}^H(i). \tag{7}$$

Obviously, if the SNR value for a given array output is not very low, then the difference between signal and noise subspaces can be found relatively easily. However, in contrast, if the SNR is very low, it will be difficult to distinguish the signal subspace from the noise subspace by analyzing the sorted eigenvalues. In other words, it is difficult to determine the source number in low SNR cases. More seriously, the SNR value and corresponding source number are two inter-acting factors. Therefore, estimating one of these factors separately without considering the other one will become very difficult and implausible.

Theoretically, with the obtained sorted eigenvalues $\mathbf{x}$ as input, and the objective source number K as output, the relationship between them can be formulated as below:

$$K = \begin{cases} f_f(\mathbf{x}, SNR), & Known\,SNR \\ f_c(\mathbf{x}), & UnKnown\,SNR \end{cases}, \tag{8}$$

where $f_f(\mathbf{x}, SNR)$ and $f_c(\mathbf{x})$ are fine-scale (with known SNR) and coarse-scale (with unknown SNR) mapping functions from the sorted eigenvalues to the source number K.

Similarly, from another view, the relationship between the sorted eigenvalues $\mathbf{x}$ and SNR value will be

$$SNR = \begin{cases} g_f(\mathbf{x}, K), & Known\,K \\ g_c(\mathbf{x}), & UnKnown\,K \end{cases}, \tag{9}$$

where $g_f(\mathbf{x}, SNR)$ and $g_c(\mathbf{x})$ are fine-scale (with known K) and coarse-scale (with unknown K) mapping functions from the sorted eigenvalues to the SNR value.

3. Source number estimation method

3.1. Multi-view meta-hierarchical classification framework

From equations (8) and (9), it can be seen that the source number K and SNR value can be seen as two different views related to the given sorted eigenvalues $\mathbf{x}$, and these two views are

affected by each other. Obviously, if we can accurately reveal the veils of $f_c(\mathbf{x})$ and $g_c(\mathbf{x})$ only with the sorted eigenvalues $\mathbf{x}$, then K and SNR will be perfectly solved simultaneously. However, in practice, the dynamic ranges of K and SNR are both very large, therefore, the corresponding fluctuations of $\mathbf{x}$ with different K and SNR are very serious. Obviously, faced with this problem, it is impossible to derive the exactly explicit physical model for $f_c(\mathbf{x})$ and $g_c(\mathbf{x})$. And this is why so many AI-based methods have been proposed to solve this problem with promising performance exhibited. In other words, in order to derive the non-linear mappings $f_c(\mathbf{x})$ and $g_c(\mathbf{x})$, it can be seen as a pattern recognition problem, which can be addressed in a data-driven manner by employing the power of machine learning techniques. Specifically, the patterns for the source number estimation problem are of course all possible solutions of K, for example, seven patterns for eight elements ULA. Therefore, it is a multi-class classification problem.

For the multi-class classification problem, traditional ensemble methods of one-vs-rest or one-vs-one using thresholding decision strategies are computationally complex, and the corresponding manner in which the information/knowledge is represented is too simple to match the human decision-making process [29]. To address these problems, hierarchical classification methods or meta-learning methods have been successfully employed in many applications [30–35]. Specifically, the principle of hierarchical classification is to divide a multi-class classification problem into many sub-fine-scale classification problems organized in a top-down or coarse-to-fine manner. In this way, a complex problem with a large dynamic range can be reduced to a much simpler sub-problem with stable dynamic range, and this strategy matches the multi-scale nature of the human perception process. From another aspect, meta-learning is viewed as an understanding and adaptation of learning itself on a higher level than merely acquiring subject knowledge. In order to follow the human decision-making process, the principle of this kind of ensemble learning method is to learn from learned knowledge (achieved by base classifiers).

Obviously, from equations (8) and (9) it can be seen that both these interacted views can be addressed by employing a hierarchical classification technique. This means that coarse-level base knowledge about $f_c(\mathbf{x})$ and $g_c(\mathbf{x})$ can be learned at first. Then, the corresponding fine-level base knowledge about $f_f(\mathbf{x}, SNR)$ and $g_f(\mathbf{x}, K)$ can be learned. Obviously, with all of this base knowledge, the meta-classification technique, which can be used to accurately make a decision for the target pattern from the learned knowledge, is an ideal tool here. Therefore, the representation of high-level meta-samples can be constructed by encoding the obtained base knowledge. The final prediction of the source number can be achieved by using a meta-classifier.

In general, in this paper, as illustrated in figure 1, the proposed source number estimation task is addressed by a multi-view meta-hierarchical (MVMH) classification method. Next, in the MVMH method, there are two basic problems that should be well tackled: (1) selecting machine learning methods for both the base classifiers and meta-classifier, and (2) how to generate the representation of meta-samples.

3.2. Machine learning methods

From figure 1, it can be seen that there are many different classifiers employed in the proposed framework. (1) *T_SNum* and *T_SNR* are two coarse-level base classifiers for estimating the initial values (i.e. coarse-level base knowledge) of the source number and the SNR. (2) $SNum2Sr_i, (i = 1, 2, \cdots, M-1)$ and $Sr2SNum_j, (j = 1, 2, \cdots, N-1)$ are fine-scale base classifiers for estimating the SNR and source number (i.e. fine-scale base knowledge) with the initial estimated results from the coarse layer. (3) *FT_SNum* is the final meta-classifier for determining the final estimated value of the source number. In order to simplify the final structure of the proposed framework with reasonable performance, well-known shallow machines: artificial neural network (ANN) and support vector machine (SVM), simple ensemble machines random forest (RF), and fully connected deep neural network (DNN) are considered for realizing one or more of the classifiers mentioned above.

3.2.1. Artificial neural network and deep neural network. ANN [36–39] is an information processing paradigm that is inspired by biological nervous systems. It is composed of a large number of highly interconnected processing elements called neurons. Normally, for the classical ANN architecture illustrated in figure 2(a), there are always three layers composed of a different number of neurons: input layer, hidden layer and output layer. Each neuron in ANN receives a number of inputs, and an activation function is applied to these inputs, which results in the activation level of the neuron (output value of the neuron). The mathematical model of the neuron is shown in figure 2(b). Obviously, the classical ANN is a shallow machine for a given task (classification or regression). The optimal configuration of parameters in the ANN can be obtained by performing the back-propagation optimization technique, for example, back-propagation neural network. By adding more than one additional hidden layer, the deep version of ANN and DNN [40–42] can be constructed, as shown in figure 2(c). In DNN, all layers are also composed of cell neurons. Specifically, the jth neuron at the lth layer is a linear combination of all the neurons of the former layer:

$$\nu_{lj} = \sum_{i=1}^{n_{l-1}} \omega_{lji} x_{l-1,i} + b_{lj}, \tag{10}$$

where ω_{lji} is the weight value for $x_{l-1,i}$ to ν_{lj} and b_{lj} is the corresponding bias value for ν_{lj}. For the entire DNN networks, let $\mathbf{W} = \bigcup_{l=1}^{L-1} \bigcup_{j=1}^{n_l} \bigcup_{i=1}^{n_{l-1}} \omega_{lji}$ be the collection of weight values and $\mathbf{B} = \bigcup_{l=1}^{L-1} \bigcup_{j=1}^{n_l} b_{lj}$ be the collection of biases. Therefore, the output of the DNN can be fully determined by these two collections. Then, with a given training data set $\mathbf{d}_t$, the objective of DNN training is to minimize some loss function $L(\mathbf{d}_t | \mathbf{W}, \mathbf{B})$.

3.2.2. Support vector machine. The SVM is one of the most well-known non-linear mapping machines in the machine learning society [43–46]. The structure of the SVM is illustrated in figure 3. Normally, the SVM is used as a powerful

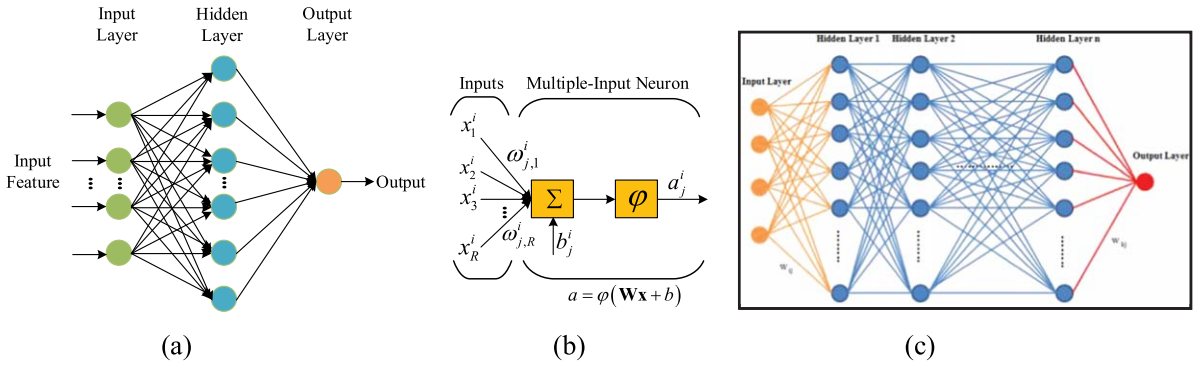

Figure 2. Illustration of the architecture of ANN and DNN: (a) ANN architecture, (b) neuron model, (c) DNN architecture.

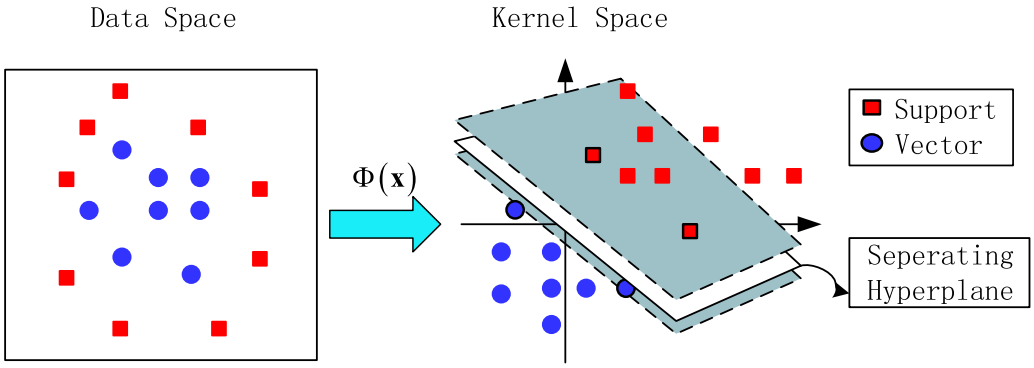

Figure 3. Illustration of the mechanism of the SVM.

two-class classifier. Specifically, for a given labelled training data set: $\{(\mathbf{x}_1,y_1),(\mathbf{x}_2,y_2),\cdots,(\mathbf{x}_n,y_n)\}$ with $y_i\in\{-1,1\}$, it is used to find a hyperplane that provides the smallest classification risk for novel unlabeled data. In order to achieve this objective, based on the principle of structured risk minimization, input (feature) data is always mapped into an unknown high-dimensional feature space. Then, the optimal classification hyperplane in high-dimensional space can be found by solving the following quadratic programming problem:

$$\begin{aligned}&\max\ W(\alpha)=-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j K(\mathbf{x}_i,\mathbf{x}_j)+\sum_{i=1}^{n}\alpha_i\\ &s.t.\quad\begin{cases}\sum_{i=1}^{n}\alpha_i y_i=0,\ i=1,2,\cdots,n\\ 0\leqslant\alpha_i\leqslant C\end{cases},\end{aligned}\tag{11}$$

where $K(\mathrm{gg})$ is an inner product kernel function satisfying Mercer's theorem; constant parameter $C>0$ is used to determine the trade-off between the training risk and model robustness. Target parameters $\alpha_i, i=1,2,\cdots,n$ are used to construct the optimal hyperplane in the mapped high-dimensional space. By solving equation (11) using optimization techniques, the final decision function with obtained support vectors **SV** will be

$$f(x)=\mathrm{sgn}\{\boldsymbol{\omega}\cdot\mathbf{x}+b\}=\mathrm{sgn}\left\{\sum_{x_i\in\mathbf{SV}}\alpha_i y_i K(\mathbf{x},x_i)+b\right\}.\tag{12}$$

3.2.3. Random forest. RF is an ensemble machine learning technique that integrates multiple decision trees, which are constructed following the classification and regression trees model [47–52]. The main structure of the RF algorithm is shown in figure 4. Specifically, for a given labelled training data set: $\mathbf{D}=\{(\mathbf{x}_1,y_1),(\mathbf{x}_2,y_2),\cdots,(\mathbf{x}_n,y_n)\}$ with $y_i\in\{1,2,,R\}$, where R is the number of class. *ntree* data sets with the same number of the original data set are randomly generated from **D** using the bootstrap resampling technique. Then, with these bootstrap data sets, *ntree* regression decision trees are learned. For each bootstrap data set, approximately one-third of the samples, which are called out-of-bag (OOB) data, are not selected for the corresponding decision tree training. Actually, the OOB data is used to evaluate the performance of the decision tree. Obviously, this built-in cross-validation process ensures that the RAF can achieve an unbiased estimation of the generalization error without using external data. Therefore, for a certain decision tree, the corresponding prediction error can be obtained using the OOB data:

$$E=\frac{\sum_{i=1}^{N_{\mathrm{OOB}}}(\widehat{y}_i-y_i)}{N_{\mathrm{OOB}}},\tag{13}$$

where N_{OOB} is the number of samples of the OOB data. Finally, with the OOB error equation (13) as the optimization target, the RAF model can be learned by gradually increasing the depth of the RAF. In general, (1) the random bootstrap

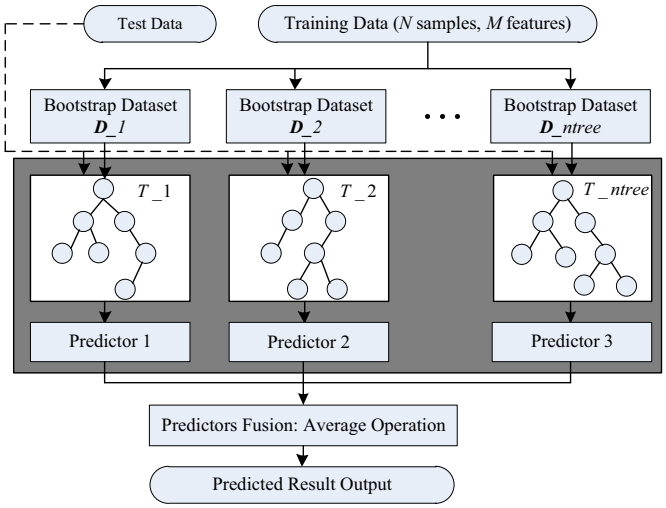

Figure 4. Structural diagram of RFss.

sampling strategy reduces the similarity of trees grown from different bootstrap samples, and therefore the RAF cannot easily be affected by noise and outliers; (2) with OOB samples, the final ensemble result by bagging of the ones from each tree is practical for mitigating both underfitting and overfitting [51, 52].

3.3. Detailed configurations for the employed machines

As mentioned above, there are two-channel and three hierarchical layers in the proposed MVMH-based source number estimation method. The first layer (coarse-level base layer) is used to roughly estimate the source number and SNR. And by dividing both the source number and SNR into sub-regions, there are two parallel channels in the second layer (fine-level base layer) constructed to accurately estimate the source number and SNR, respectively. Finally, by encoding the estimated results from both the first layer and second layer, meta-samples can be generated, and then the last layer (meta-layer) is employed to perform the final pattern decision task.

3.3.1. Coarse-level base layer. In this layer, there is no prior information about the source number or SNR that can be used. Therefore, the corresponding AI machines (*T_SNum* and *T_SNR*) employed here should be learned with large-scale training data covering all the possible values/patterns. Specifically, the inputs of these machines are the sorted eigenvalues $\mathbf{x} \in \mathbf{R}^{1\times M}$, and the outputs of these two machines are the corresponding sub-regions of the source number or SNR. For *T_SNum*, the outputs are all possible source numbers, e.g. $Y_{T\_SNum} = k \in [1, 2, \cdots, M-1]$. For *T_SNR*, the outputs are all possible SNR sub-regions defined below:

$$Y_{T\_SNR} = \begin{cases} 1,\ SNR \geq 15dB \\ 2,\ 10dB \leq SNR < 15dB \\ 3,\ 5dB \leq SNR < 10dB \\ 4,\ 0dB \leq SNR < 5dB \\ 5,\ -5dB \leq SNR < 0dB \\ 6,\ -10dB \leq SNR < -5dB \\ 7,\ SNR < -10dB \end{cases}. \quad (14)$$

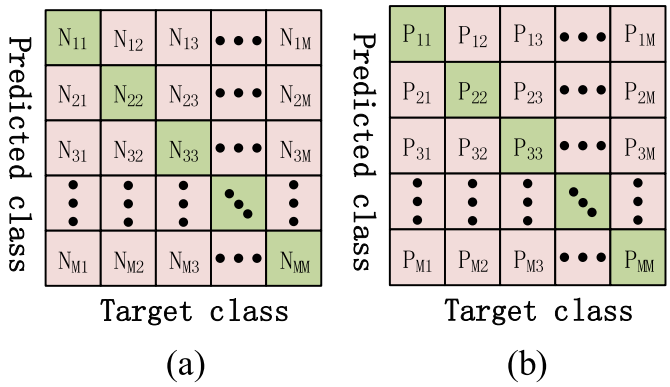

Figure 5. Illustration of the structure of classification of the confusion matrix.

Then, by training these two machines, the corresponding confusion matrixes $\mathbf{C}_{T\_SNum}$ and $\mathbf{C}_{T\_SNR}$ can be acquired. The structure of the confusion matrix is illustrated in figure 5(a). In figure 5(a), N_{ij} indicates the number of samples predicted as the *i*th class, which is actually the *j*th target class. From the obtained confusion matrix, the weighting matrix shown in figure 5(b) of the predicted results of the corresponding machine can be calculated as below:

$$P_{ij} = \frac{N_{ij}}{\sum_k N_{ik}}. \quad (15)$$

3.3.2. Fine-level base layer. With the outputs of the first layer as prior information, the machines in the second layer can be used to predict the fine-scale source number and SNR in parallel. Specifically, the machines $\{\mathrm{SNum2Sr}_i\}_{i=1,2,\cdots,M-1}$ are used to accurately estimate the SNR sub-region with a known source number. That is, these machines are learned with training data having the same number of sources and labeled by different SNR sub-regions. On the other hand, the machines $\{\mathrm{Sr2SNum}_i\}_{i=1,2,\cdots,7}$ are used to estimate the source number with a known SNR sub-region. Namely, these machines are learned with training data having the same SNR sub-region and labeled by different source number.

Obviously, by employing more information in this layer, the predicted results are more reliable and more accurate than the ones from the first coarse layer. However, because the outputs of the first layer are not accurate enough, especially in the low SNR case, it is always difficult to determine the real source number by the outputs of the second layer directly. Nevertheless, they are also clues or evidence for finding out the answer of the real source number. Therefore, an additional layer, for example, the last meta-classification layer, is added to perform the final pattern recognition task.

3.3.3. Meta-classification layer. In order to make better use of the information obtained by the first two layers, a specified representation of the meta-sample is designed, as illustrated in figure 6.

From figure 6, it can be seen that there are three sections in the proposed representation of the meta-sample. The first

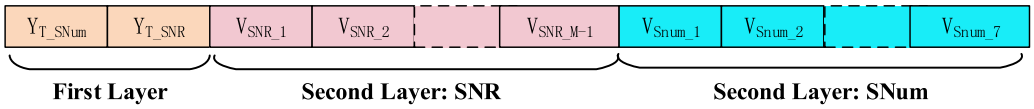

Figure 6. Structure of the proposed representation of the meta-sample.

section has two values: predicted results of the coarse source number Y_{T_SNum} and the coarse SNR sub-region Y_{T_SNR}. The second section contains $M-1$ components to record the voting results of the predicted fine-scale source number. The last section contains components to record the voting results of the predicted fine-scale SNR sub-regions. Specifically, the recorded voting results for each component of the last two sections can be acquired by performing the following procedure:

Meta-sample Extraction

Inputs: Testing data $\mathbf{x}$, first layer machines: *T_SNum* and *T_SNR*, first layer weight matrixes: $\mathbf{W}_{T\_SNum}$ and $\mathbf{W}_{T\_SNR}$, second layer machines: $\{\text{SNum2Sr}_i\}_{i=1,2,\cdots,M-1}$ and $\{\text{Sr2SNum}_i\}_{i=1,2,\cdots,7}$, second layer weight matrixes: $\{\mathbf{W}_{\text{SNum2Sr}_i}\}_{i=1,2,\cdots,M-1}$ and $\{\mathbf{W}_{\text{Sr2SNum}_i}\}_{i=1,2,\cdots,7}$
Step 1: Initializing the meta-sample as $\mathbf{HF}=0^{(M+8)\times 1}$;
Step 2: Obtaining the two components of the first section of the meta-sample:
1: $Y_{T\_SNum}=predict\,(T\_SNum,\mathbf{x})$;
2: $Y_{T\_SNR}=predict\,(T\_SNR,\mathbf{x})$.
3: Updating the high-order feature: $\mathbf{HF}\,(1)=Y_{T\_SNum}$, $\mathbf{HF}\,(2)=Y_{T\_SNR}$
Step 3: Obtaining the weight vectors from the first layer:
1: $\mathbf{tW}_{T\_SNum}=\mathbf{W}_{T\_SNum}(Y_{T\_SNum},:)$;
2: $\mathbf{a}(\theta_i)=[e^{j\omega_c\tau_{i,1}},e^{j\omega_c\tau_{i,2}},\cdots,e^{j\omega_c\tau_{i,M}}]^T$,.
Step 4: Obtaining the second section of the meta-sample:
For $k=1$ to $M-1$
1: Letting the current machine is SNum2Sr_k;
2: Getting the outputs of the current machine: $Y_{SNum2Sr\_k}=predict\,(\text{SNum2Sr}_k,\mathbf{x})$;
3: Getting the current weighting vector corresponding to $Y_{SNum2Sr\_k}$: $\mathbf{tW}_{SNum2Sr\_k}=\mathbf{W}_{SNum2Sr_k}\,(Y_{SNum2Sr\_k},:)$;
4: Updating the second section of the high-order feature:

$$\mathbf{HF}\,(3:M+1)=\mathbf{HF}\,(3:M+1)+\mathbf{tW}_{SNum2Sr\_k}\cdot\mathbf{tW}_{T\_SNum}(k) \tag{16}$$

End
Step 5: Obtaining the last section of the meta-sample:
For$k=1$ to 7
1: Letting the current machine is Sr2SNum_k;
2: Getting the outputs of the current machine: $Y_{Sr2SNum\_k}=predict\,(\text{Sr2SNum}_k,\mathbf{x})$;
3: Getting the current weighting vector corresponding to $Y_{Sr2SNum\_k}$: $\mathbf{tW}_{Sr2SNum\_k}=\mathbf{W}_{Sr2SNum_k}\,(Y_{Sr2SNum\_k},:)$;
4: Updating the second section of the meta-sample:

$$\mathbf{HF}(M+2:end)=\mathbf{HF}(M+2:end)+\mathbf{tW}_{Sr2SNum\_k}\cdot\mathbf{tW}_{T\_SNR}(k) \tag{17}$$

End
Step 6: Outputting the meta-sample **HF**.

Note: $predict\,(machine,\mathbf{x})$ referred to predict the output pattern of the input $\mathbf{x}$ using the given machine.

Finally, with the obtained meta-sample as input, the final meta-classification machine is constructed to give the final output of the predicted source number.

4. Experimental results

In order to evaluate the performance of the proposed MVMH method for source number estimation, several experiments were conducted with two simulated data sets generated by the well-known DOA tools downloaded from https://github.com/morriswmz/doa-tools [53] provided by Dr Wang Mianzhi, Washington University, and by the phased array system toolbox of MATLAB (e.g. phased.ULA + sensorsig functions). In the following sections, these two data sets are called DOA-tools data set and MATLAB data set, respectively.

Without loss of generality, all the simulated data were generated for 8-element ULA with element spacing 0.5λ, and all the impinging signals are equal-powered narrow-band ones with Gaussian white noise. Then, for each case of the given source number or SNR sub-range (i.e. Snum $=1$, SNR ≥ 15 dB), 10 000 simulated data samples were randomly generated. As a result, 490 000 simulated data samples in each of the two data sets were employed for conducting the following experiments. For all of the experiments, the key

Table 1. Key parameters of the used machines.

Machines	Parameters
ANN	Hidden layer: 48, 64, 96 neurons; cost function: cross-entropy
SVM	Kernel: RBF (Gaussian radial basis); gamma: 0.05; epsilon: 1×10^{-3}; tenfold cross-validation; C = 1, 10, 50
RF	Trees: 32, 48, 64; MinLeafSize: 1
DNN	Hidden layer: three hidden layers with 32, 48, 64 neurons per hidden layer

Table 2. Comparison of the classification performance of different classifiers: MATLAB data set (%).

	ANN			SVM			RF			DNN		
	ANN_1	ANN_2	ANN_3	SVM_1	SVM_2	SVM_3	RF_1	RF_2	RF_3	DNN_1	DNN_2	DNN_3
T_SNR	91.2	91.6	92.7	92.4	92.2	91.7	95.3	96.2	**97.4**	93.1	93.1	93.5
T_SNUM	72.3	73.7	73.9	–	–	–	75.8	76.1	**76.4**	73.5	73.2	73.6
$Sr2SNum_1$	84.1	84.3	84.4	84.4	84.6	83.9	88.4	88.8	**90.3**	71.9	75.6	77.1
$Sr2SNum_2$	85.1	85.2	85.2	85.3	85.5	85.6	89.1	89.4	**89.7**	71.2	74.7	75.2
$Sr2SNum_3$	84.6	84.8	84.9	85.0	85.3	85.2	87.4	88.0	**88.6**	70.1	71.3	71.9
$Sr2SNum_4$	82.4	82.7	82.6	–	–	–	85.6	85.9	**86.3**	69.6	69.7	69.6
$Sr2SNum_5$	80.3	80.8	80.9	–	–	–	81.8	82.0	**82.6**	67.1	68.2	68.7
$Sr2SNum_6$	71.8	72.2	72.5	–	–	–	71.6	72.5	**72.9**	63.5	63.8	63.9
$Sr2SNum_7$	41.2	42.5	**42.5**	–	–	–	38.8	40.3	41.0	36.4	37.5	37.8
$SNum2Sr_1$	94.3	94.5	94.3	95.1	95.0	95.4	100	100	**100**	95.6	95.7	95.9
$SNum2Sr_2$	94.0	94.1	94.1	94.6	94.7	94.5	100	100	**100**	95.4	95.6	95.5
$SNum2Sr_3$	96.5	96.2	96.8	95.9	96.3	96.6	99.2	99.5	**99.5**	95.7	96.2	96.2
$SNum2Sr_4$	95.1	95.7	95.7	96.8	96.7	96.9	98.9	99.1	**99.3**	96.8	97.0	97.1
$SNum2Sr_5$	92.4	92.5	92.7	93.2	93.7	93.6	98.2	98.6	**98.8**	94.7	94.5	94.8
$SNum2Sr_6$	93.9	93.8	94.0	94.1	94.1	94.0	97.3	97.2	**97.5**	94.1	94.3	94.5
$SNum2Sr_7$	96.5	96.5	96.5	96.9	97.1	97.0	98.6	98.6	**98.9**	97.2	97.2	97.5

Note: the symbol '–' denotes that the machine was trained without convergence.

Table 3. Comparison of the classification performance of different classifiers: DOA-tools data set (%).

	ANN			SVM			RF			DNN		
	ANN_1	ANN_2	ANN_3	SVM_1	SVM_2	SVM_3	RF_1	RF_2	RF_3	DNN_1	DNN_2	DNN_3
T_SNR	92.3	92.6	93.1	92.8	93.4	93.7	95.7	96.5	**97.6**	95.1	95.4	95.6
T_SNUM	78.5	78.9	79.0	–	–	–	79.7	80.1	**80.4**	78.3	78.6	78.7
$Sr2SNum_1$	96.8	97.1	97.2	97.6	97.6	97.9	98.5	98.7	**99.3**	98.8	98.8	98.7
$Sr2SNum_2$	97.1	97.2	97.2	97.6	97.5	97.7	98.6	98.8	**99.2**	98.7	98.8	98.8
$Sr2SNum_3$	97.0	97.1	97.1	97.6	97.7	97.7	98.5	99.0	**99.2**	98.6	98.8	98.8
$Sr2SNum_4$	96.3	96.3	96.5	97.2	97.0	97.0	98.4	98.5	**98.7**	98.0	98.1	98.3
$Sr2SNum_5$	80.3	80.8	80.9	96.3	96.2	96.2	98.1	98.5	**98.6**	97.1	97.2	97.7
$Sr2SNum_6$	92.6	92.7	93.1	92.4	92.1	92.0	92.9	93.2	**93.3**	92.0	92.4	92.6
$Sr2SNum_7$	68.1	68.4	69.2	–	–	–	85.7	86.1	**86.5**	58.2	58.3	58.4
$SNum2Sr_1$	94.3	94.0	94.2	95.1	95.0	95.4	100	100	**100**	95.1	95.2	95.5
$SNum2Sr_2$	94.0	93.8	94.0	94.6	94.7	94.5	100	100	**100**	95.3	95.4	95.7
$SNum2Sr_3$	96.5	96.6	96.8	95.9	96.3	96.6	100	100	**100**	96.4	96.5	96.8
$SNum2Sr_4$	96.9	96.9	97.4	96.8	96.7	96.9	99.2	99.5	**99.6**	97.0	97.1	97.3
$SNum2Sr_5$	97.0	97.2	97.8	97.2	97.7	97.6	98.2	98.4	**98.8**	96.6	96.7	96.9
$SNum2Sr_6$	92.5	92.8	93.1	94.1	94.1	94.0	96.8	97.1	**97.2**	94.2	94.3	94.7
$SNum2Sr_7$	96.1	96.1	96.4	96.9	97.1	97.0	98.7	98.7	**98.9**	97.3	97.5	97.6

parameters of the compared machines are given in table 1 below.

4.1. Experiment 1—machine selection for the first two layers

In this experiment, ANN, SVM, RF and DNN with different configurations were used to act as machines in the first two layers for comparison. Specifically, ANN_1 to ANN_3, SVM_1 to SVM_3, RF_1 to RF_3 and DNN_1 to DNN_3 refer to machines with different configurations corresponding to the settings shown in table 1. In order to validate the efficiency of the methods using these machines, classification accuracy was applied as the quantitative criterion. Detailed experimental results are given in tables 2 and 3 for the MATLAB data set and

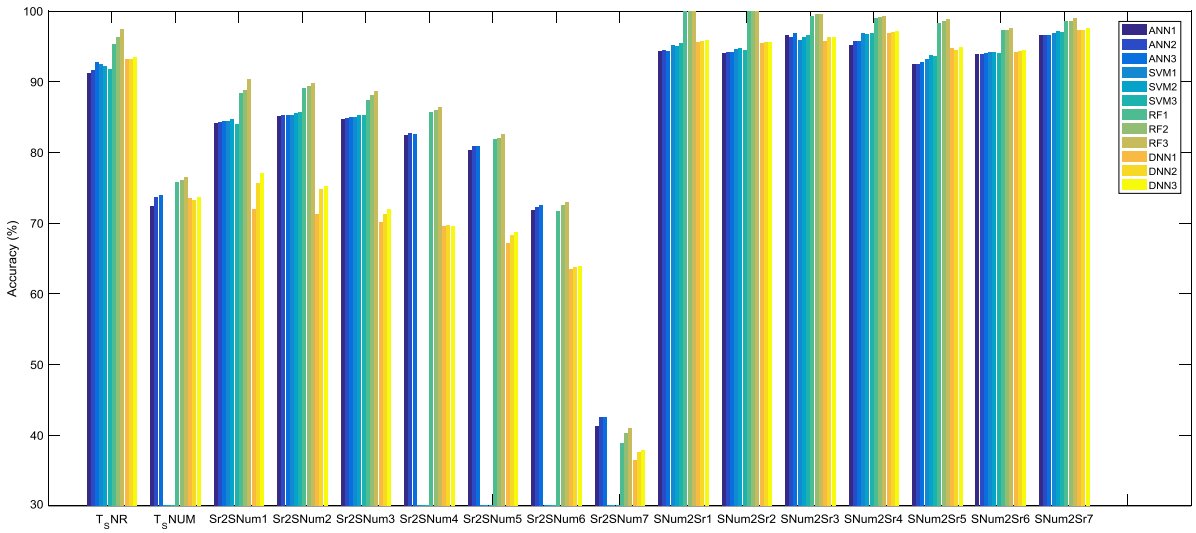

Figure 7. Bar-plot of the comparative results of classification accuracy achieved by different machines: MATLAB data set.

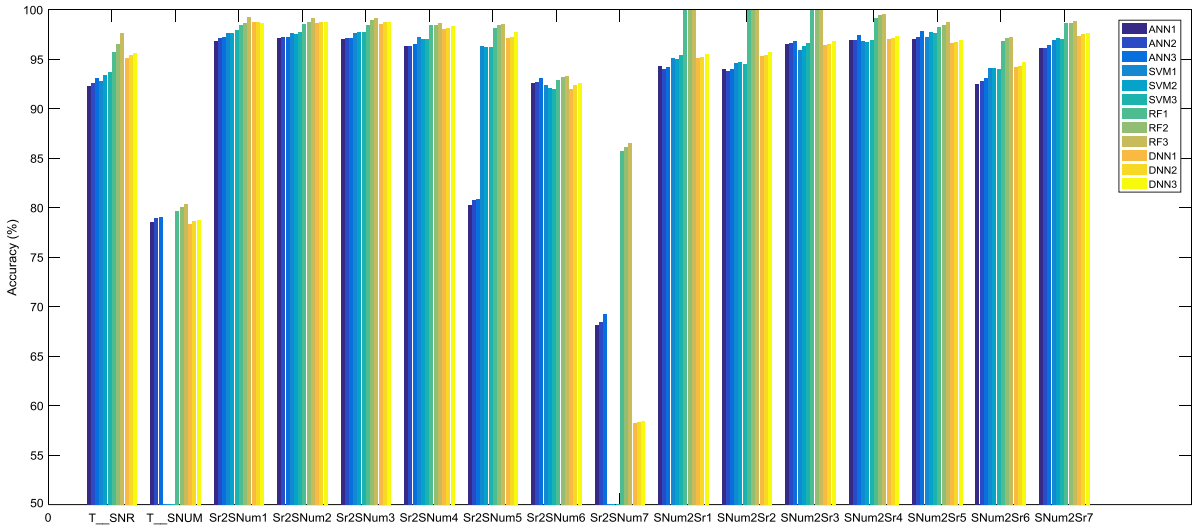

Figure 8. Bar-plot of the comparative results of classification accuracy achieved by different machines: DOA-tools data set.

DOA-tools data set, respectively. The corresponding bar-plots are illustrated in figures 7 and 8, respectively.

4.2. Experiment 2—machine selection for the meta-classification layer

In this experiment, ANN_1 to ANN_3, SVM_1 to SVM_3, RF_1 to RF_3 and DNN_1 to DNN_3 were also employed for comparison to validate the final pattern recognition performance. At first, RF with 64 trees was used as all of the machines in the first two layers to generate the representation of the meta-sample used for training and testing the final meta-classification machine. 400 000 generated meta-samples were used to train the corresponding machines, and the other 90 000 meta-samples were used to test the performance of these machines. Experimental results for the final source number estimation performance are given in tables 4 and 5 for the two data sets used.

The corresponding confusion matrixes of ANN_3, RF_3 and DNN_3 are shown in figures 9 and 10.

4.3. Experiment 3—performance comparison with other source number estimation methods

In this experiment, the performance of source number estimation achieved by the proposed MVMH method (with RF as all of the machines in the proposed methods) was compared with AIC, MDL and our previous source number estimation method HAUSSEM_V (HAU for short). Detailed experimental results are given in tables 6 and 7 for the two data sets used, respectively. In order to obviously illustrate the performance difference among these methods, the corresponding estimation accuracy curves with different SNR and different source numbers are shown in figures 11 and 12.

4.4. Experiment 4—performance evaluation of the effect with preprocessing technique employed

In this section, the performance of the proposed MVMH method with an additional denoising technique introduced

Table 4. Comparative results of source number estimation with different machines: MATLAB data set (%).

	ANN			SVM			RF			DNN	
ANN_1	ANN_2	ANN_3	SVM_1	SVM_2	SVM_3	RF_1	RF_2	RF_3	DNN_1	DNN_2	DNN_3
94.4	94.6	94.7	–	–	–	95.4	95.7	**96.0**	95.4	95.4	95.5

Table 5. Comparative results of source number estimation with different machines: DOA-tools data set (%).

	ANN			SVM			RF			DNN	
ANN_1	ANN_2	ANN_3	SVM_1	SVM_2	SVM_3	RF_1	RF_2	RF_3	DNN_1	DNN_2	DNN_3
95.1	95.1	95.3	–	–	–	96.2	96.7	**97.1**	95.2	95.5	95.6

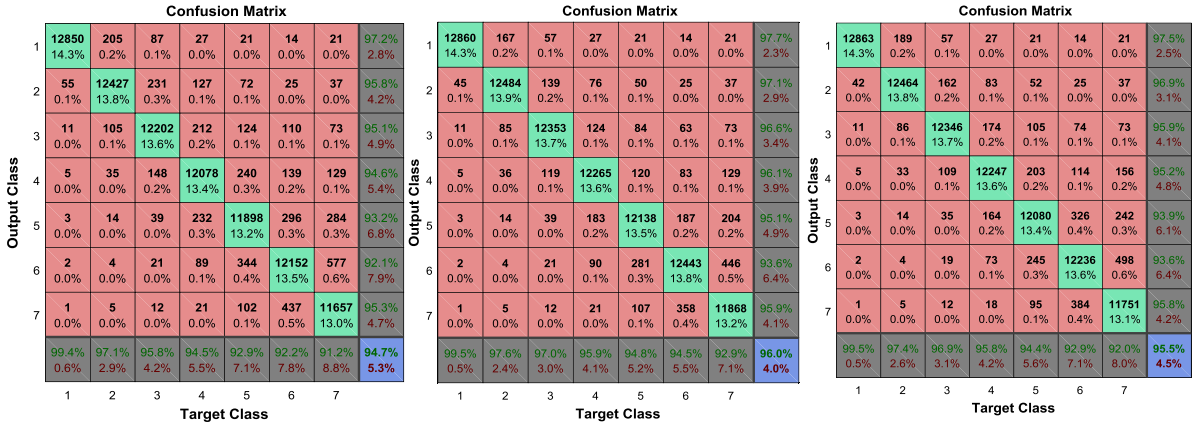

Figure 9. Illustration of the confusion matrixes achieved by different machines for final source number estimation of the MATLAB data set using the high-order feature: (a) ANN_3, (b) RF_3, (c) DNN_3.

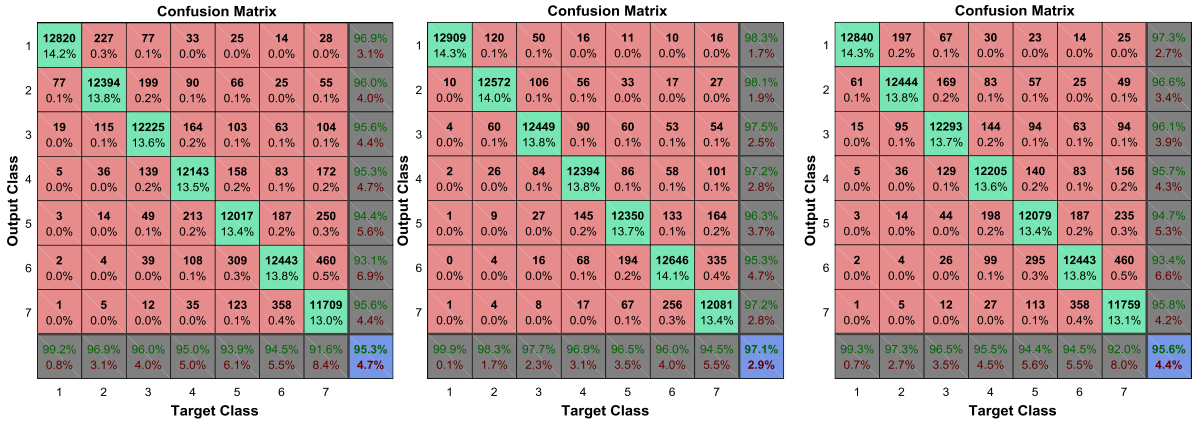

Figure 10. Illustration of the confusion matrixes achieved by different machines for source number estimation of the DOA-tools data set using the high-order feature: (a) ANN_3, (b) RF_3, (c) DNN_3.

was evaluated. Without loss of generality, since the wavelet-based denoising method [54–58] is usually used for array signal denoising, the denoising method proposed in [55] was employed here. Specifically, by performing a denoising operation to the array signals before extracting their sorted eigenvector, new MATLAB-derived data sets were generated, including data sets with and without denoising preprocessing. And then, using the same training and testing data sets, source number estimation performance achieved by different methods was evaluated. The detailed experimental results are given in table 8. In table 8, -DN denotes the corresponding results derived from the denoised data set.

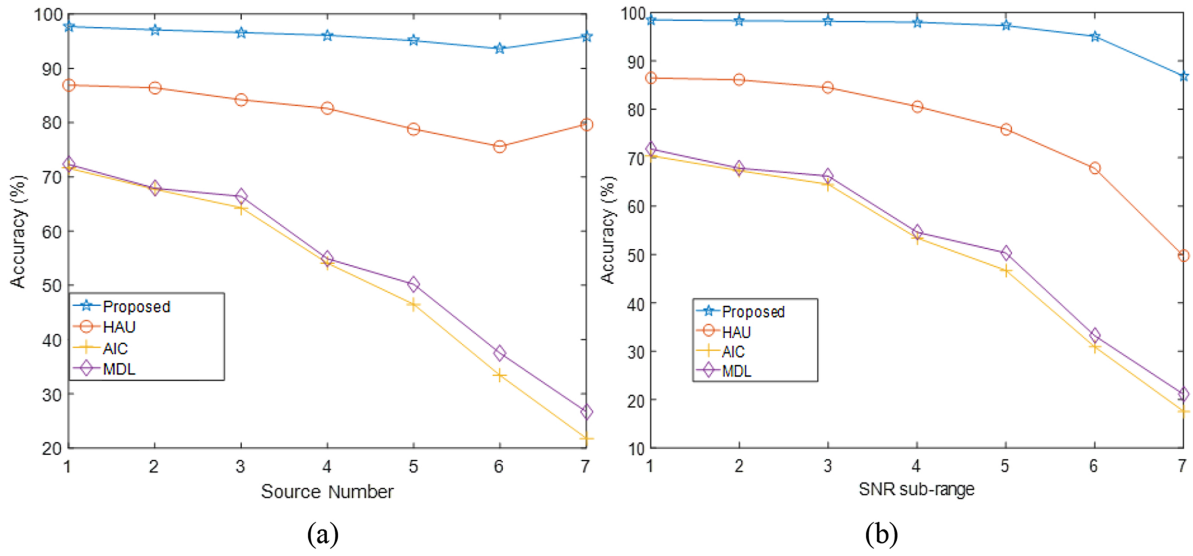

Figure 11. Illustration of the final source number estimation accuracy of the MATLAB data set: (a) curves for accuracy with different source number, (b) curves for accuracy with SNR sub-range.

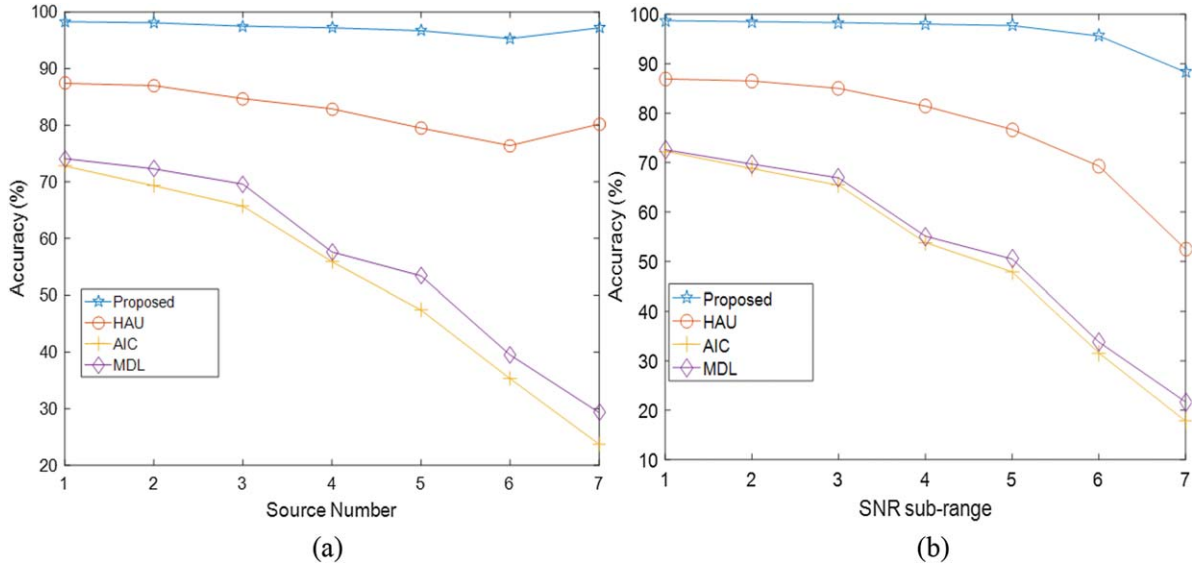

Figure 12. Illustration of the final source number estimation accuracy of the DOA-tools data set: (a) curves for accuracy with different source number; (b) curves for accuracy with SNR sub-range.

Table 6. Source number estimation results from the comparison of different methods for the MATLAB data set (%).

AIC	MDL	HAU	Proposed-MVMH
47.6	49.6	75.9	**96.0**

Table 7. Source number estimation results from the comparison of different methods for the DOA-tools data set (%).

AIC	MDL	HAU	Proposed-MVMH
50.6	52.3	78.7	**97.1**

4.5. Discussions

In the above sections, several experiments were conducted to validate the performance of the proposed source number estimation method. In the first experiment, ANN, SVM, RF and DNN were employed to implement the AI machines for the first two hierarchical layers of the proposed MVMH method, and the corresponding experimental results were used to select the optimal one with the highest accuracy. Specifically, better results can be achieved by using RF and ANN

Table 8. Source number estimation results from comparison of un-denoised and denoised data sets (%).

	T_SNUM	T_SNR	SSr1	SSr3	SSr5	SSr7	SSn1	SSn3	SSn5	SSn7	FSnum
MVMH-DN	65.3	92.8	97.9	98.6	97.4	91.5	79.1	79.5	77.3	71.8	87.3
MVMH	75.0	96.6	99.3	99.7	99.7	97.6	89.6	88.8	85.0	80.7	**93.8**
HAU-DN	–	–	–	–	–	–	–	–	–	–	70.1
HAU	–	–	–	–	–	–	–	–	–	–	73.4
MDL-DN	–	–	–	–	–	–	–	–	–	–	43.7
MDL	–	–	–	–	–	–	–	–	–	–	47.4
AIC-DN	–	–	–	–	–	–	–	–	–	–	42.9
AIC	–	–	–	–	–	–	–	–	–	–	45.8

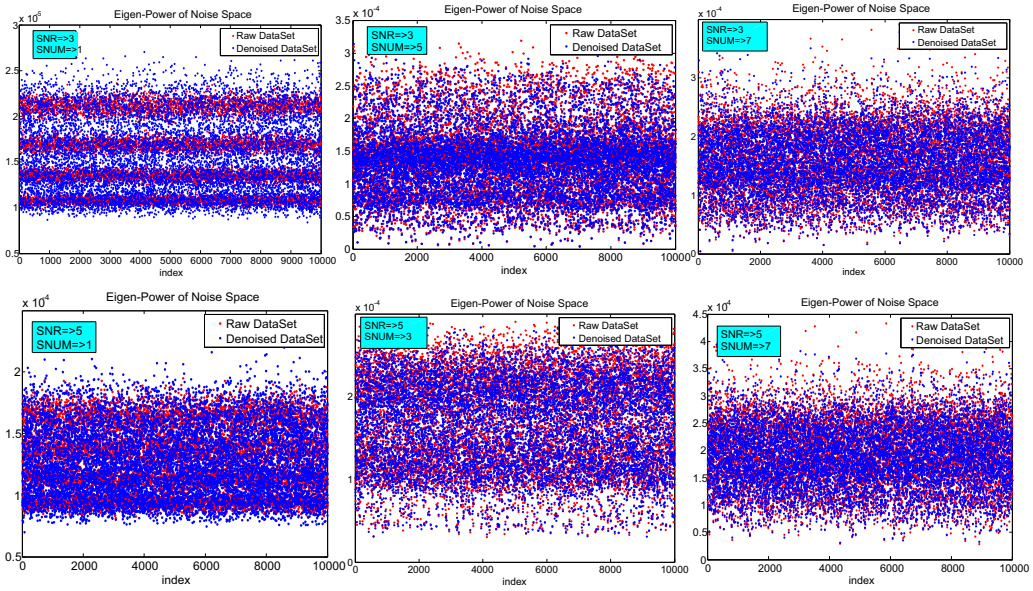

Figure 13. Illustration of some comparisons of the distributions of eigenpower of noise space.

compared to SVM and DNN. In addition, the corresponding time consumption for these machines varies greatly: both the training and testing time of SVM or DNN are ten times or even more than those of ANN and RF (this is the reason time consumption data was not illustrated in the specified experiments.). Between ANN and RF, RF with 64 trees is the best choice with the highest accuracy and relatively less time consumption. Obviously, because the intrinsic value of the input feature (e.g. the sorted eigenvalues of array outputs) used to distinguish the signal space and noise space is the magnitude relationship among components of the input feature, this is more consistent with the classification nature of RF composed of several decision trees. In the second experiments, RF with 64 trees is also the final winner for the last meta-classification layer of the proposed method. In the last experiment, source number estimation performances among the proposed method and some other well-known methods were given. From the experimental results, it can be seen that the proposed MVMH method can dramatically improve the source number estimation performance of any cases with different SNR or source numbers. In particular, the advantages of the proposed method are more obvious in cases with more than three sources or SNR under 0 dB. In the last experiment, the effects of adding the denoising technique were evaluated. From the experimental results, it can be seen that the performance of the proposed method will be weakened by the introduction of an additional denoising procedure. By directly checking the feature space, as illustrated in figure 13, it was found that the denoising operation made the feature space too diverse to reflect the intrinsic information about the source number and SNR. Obviously, in order to further improve the performance of the proposed method by using some data or feature enhancement strategy, it is must be specifically designed by taking the architecture of the proposed method into account. In general, promising results were illustrated for the proposed method.

5. Conclusion

In this paper, a novel multi-view meta-hierarchical classification-based source number estimation method was proposed. From the experimental results given in this paper, there are at least two conclusions that can be made. At first,

by employing multi-view hierarchical classification and meta-learning techniques, more reasonable performance enhancement for source number estimation can be achieved than traditional source number estimation methods. Second, for the two simulated data sets used with 490 000 samples in this paper, hierarchically integrating shallow machines are enough to tackle the task of source number estimation even in low SNR or more source number cases. Obviously, information hidden in the eigenvalues of the covariance matrix associatedg with the array outputs can readily be used for revealing the actual input scenario. However, the desired effects are not achieved by employing DNNs in this paper. There are two possible reasons: (1) the number of samples, e.g. 490 000, is smaller than the requirement of deep learning techniques, and (2) there is much information loss for the deep learning method by only employing the sorted eigenvectors as the input feature. Therefore, in future, more powerful feature representations will be exploited to reveal more accurately the truths between array outputs and the corresponding source number or SNR value. In addition, features suitable for deep learning techniques will be studied. In general, source number or even the detailed DOAs estimated in a data-driven manner is very promising.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant Nos. 11774073, 51279033 and 51574087.

ORCID iD

Cao Zhimin https://orcid.org/0000-0002-3679-6288

References

[1] Bekkerman I and Tabrikian J 2006 Target detection and localization using MIMO radars and sonars *IEEE Trans. Signal Process.* **54** 3873–83
[2] Shi J, YangD S *et al* 2017 Spatial time-frequency DOA estimation based on joint diagonalization using Jacobi rotation *Appl. Acoust.* **116** 24–32
[3] Shi S-G, Li Y *et al* 2019 Real-valued robust DOA estimation method for uniform circular acoustic vector sensor arrays based on worst-case performance optimization *Appl. Acoust.* **148** 495–502
[4] Chen H W and Zhao J W 2005 Coherent signal-subspace processing of acoustic vector sensor array for DOA estimation of wideband sources *Signal Process.* **85** 837–47
[5] Kikuchi T, Yamaoka T and Hamada N. 1999. Microphone array system with DOA estimation by using harmonic structure of speech signals *IEICE Technical Report* DSP98-164
[6] Wang S Q and Yang J 2007 Decentralized acoustic source localization with unknown source energy in a wireless sensor network *Meas. Sci. Technol.* **18** 3768
[7] Roy R and Kailath T 1989 Esprit-estimation of signal parameters via rotational invariance technique *IEEE Trans. Acoust. Speech Signal Process.* **37** 984–95
[8] Schmidt R 1986 Multiple emitter location and signal parameter estimation *IEEE Trans. Antennas Propag.* **34** 276–80
[9] Fridlander B 1993 The root-music algorithm for direction finding with interpolated arrays *Signal Process.* **30** 15–29
[10] Zoltowski M D, Harrdt M and Mathews C P 1996 Closed-form 2D angle estimation with rectangular arrays in element space or beamspace via unitary esprit *IEEE Trans. Signal Process.* **44** 316–28
[11] Abeida H and Delmas J P 2007 Efficiency of subspace based DOA estimators *Signal Process.* **87** 2075–84
[12] Liu A F and Liao G S 2018 An eigenvector based method for estimating DOA and sensor gain-phase errors *Digit. Signal Process.* **79** 116–24
[13] Liu A F, Yang D S *et al* 2018 Augmented subspace MUSIC method for DOA estimation using acoustic vector sensor array *IET Radar Sonar Navig.* **13** 969–75
[14] Malioutov D, Cetin M and Willsky A S 2005 A sparse signal reconstruction perspective for source localization with sensor arrays *IEEE Trans. Signal Process.* **53** 3010–22
[15] Huang Q H, Xiang L F and Liu K 2017 Off-grid DOA estimation in real spherical harmonics domain using sparse Bayesian inference *Signal Process.* **137** 124–34
[16] Yang Z, Xie L and Zhang C 2013 Off-grid direction of arrival estimation using sparse Bayesian inference *IEEE Trans. Signal Process.* **61** 38–43
[17] Wu X H, Zhu W P, Yan J and Zhang Z Y 2018 Two sparse-based methods for off-grid direction-of-arrival estimation *Signal Process.* **142** 87–95
[18] Liu Q, So H C and Gu Y T 2017 Off-grid DOA estimation with nonconvex regularization via joint sparse representation *Signal Process.* **140** 171–6
[19] Dai J S and So H C 2018 Sparse Bayesian learning approach for outlier-resistant direction-of-arrival estimation *IEEE Trans. Signal Process.* **66** 744–56
[20] Xiao M L, Wei P and Tai H M 2012 Estimation of the number of sources based on hypothesis testing *J. Comm. Netw.* **14** 481–6
[21] Izedi M K and Derakhtian M 2017 Joint DOA and source number detection for arrays with arbitrary geometry *Signal Process.* **140** 149–60
[22] Wax M and Kailath T 1985 Detection of signals by information theoretic criteria *IEEE Trans. Acoust. Speech Signal Process.* **33** 387–92
[23] Wax M and Ziskind I 1989 Detection of the number of coherent signals by the MDL principle *IEEE Trans. Acoustics Speech Signal Process.* **37** 8–15
[24] Zhang Y and Ng B P 2010 MUSIC-like DOA estimation without estimating the number of sources *IEEE Trans. Signal Process.* **58** 1668–76
[25] Wang D S, Zou Y X and Wang W W 2018 Learning soft mask with DNN and DNN-SVM for multi-speaker DOA estimation using an acoustic vector sensor *J. Franklin Inst.* **355** 1692–709
[26] Huang H J, Yang J *et al* 2018 Deep learning for super-resolution channel estimation and DOA estimation based massive MIMO system *IEEE Trans. Veh. Technol.* **67** 8549–60
[27] Liu Z M, Zhang C W and Yu P S 2018 Direction-of-arrival estimation based on deep neural networks with robustness to array imperfections *IEEE Trans. Antennas Propag.* **66** 7315–27
[28] Wu Y, Li X K *et al* 2018 Joint signal-to-noise ratio and source number estimation based on hierarchical artificial intelligence units *Meas. Sci. Technol.* **29** 095104(11pp)
[29] Wang X L, Zhao H and Lu B L 2014 A meta-top-down method for large-scale hierarchical classification *IEEE Trans. Knowl. Data Eng.* **26** 500–13
[30] Lemke C, Budka M and Gabrys B 2015 Metalearning: a survey of trends and technologies *Artif. Intell. Rev.* **44** 117–30

[31] Hirt R, Kühl N and Satzger G 2019 Cognitive computing for customer profiling: meta classification for gender prediction *Electron. Mark.* **2019** 1–14

[32] Cesa-Bianchi N, Itgentile D U *et al* 2006 Incremental algorithms for hierarchical classification *J. Mach. Learn. Res.* **7** 31–54

[33] Alshamaa D, Chehade F M and Honeine P 2018 A hierarchical classification method using belief functions *Signal Process.* **148** 68–77

[34] Fu R, Li B, Gao Y *et al* 2018 CNN with coarse-to-fine layer for hierarchical classification *IET Comput. Vis.* **12** 892–9

[35] Deepika S S and Geetha T V 2018 A meta-learning framework using representation learning to predict drug-drug interaction *J. Biomed. Inform.* **84** 136–47

[36] Barnard E 1992 Optimization for training neural nets *IEEE Trans. Neural Netw.* **3** 232–40

[37] Chen C H and Honavar V 1999 A neural network architecture for syntax analysis *IEEE Trans. Neural Netw.* **10** 94–114

[38] Ham F M and Kostanic I 2001 *Principles of Neuro-computing for Science and Engineering* (New York: McGraw-Hill)

[39] Hinton G E, Osindero S and Teh Y W 2006 Reducing the dimensionality of data with neural networks *Science* **313** 1527–54

[40] Haykin S 2018 *Neural Networks and Learning Machines* (Zurich: Pearson Schweiz AG)

[41] Hinton G E, Deng L *et al* 2012 Deep neural networks for acoustic modeling in speech recognition *IEEE Signal Process. Mag.* **29** 82–97

[42] Sze V, Chen Y H *et al* 2017 Efficient processing of deep neural networks: a tutorial and survey *Proc. IEEE* **105** 2295–329

[43] Cortes C and Vapnik V 1995 Support vector networks *Mach. Learn.* **20** 273–97

[44] Vapnik V, Golowich S and Smola A 1997 Support vector method for function approximation, regression estimation, and signal processing *Advances in Neural Information Processing Systems* vol 9 (Cambridge, MA: MIT Press) pp 281–7

[45] Hearst M A 1998 Support vector machines *IEEE Intell. Syst.* **8** 18–20

[46] Artemio S O, Marco A A F *et al* 2013 Forecast urban air pollution in Mexico City by using support vector machines: a kernel performance approach *Int. J. Intell. Sci.* **3** 126–35

[47] Breiman L 2001 Random forests *Mach. Learn.* **45** 5–32

[48] Jiang X, Abdel-Aty M *et al* 2016 Investigating macro-level hotzone identification and variable importance using big data: a random forest models approach *Neurocomputing* **181** 53–63

[49] Ghasemian N and Akhoondzadeh M 2018 Introducting two random forest based methods for cloud detection in remote sensing images *Adv. Space Res.* **62** 288–303

[50] Sukhija S and Narayanan C K 2019 Supervised heterogeneous feature transfer via random forests *Artif. Intell.* **268** 30–53

[51] Altman N M and Martin K 2017 Ensemble methods: bagging and random forests *Nat. Methods* **14** 993–4

[52] Leo B 2001 Random forests *Mach. Learn.* **45** 5–32

[53] Wang M, Zhang Z and Nehorai A 2018 Performance analysis of coarray-based MUSIC in the presence of sensor location errors *IEEE Trans. Signal Process.* **66** 3074–85

[54] Xue Y, Wang J and Liu Z 2005 Application of wavelet array denoising to ESPRIT algorithm *TENCON 2005 2005 IEEE Region 10 IEEE* **1** 1–4

[55] Anil M R and Douglas L J 2000 A denoising approach to multisensor signal estimation *IEEE Trans. Singal Process.* **5** 1225–34

[56] Shu X L and Han S P 2009 Improvement of DOA estimation using wavelet denoising. *1st Int. Conf. on Information Science and Engineering (ICISE2009)* **1** 587–90

[57] Sathish R and Anand G V 2003 Wavelet denoising for plane wave DOA estimation by MUSIC *TENCON 2003. Conf. on Convergent Technologies for Asia-Pacific Region. IEEE* **1** 104–8

[58] Paramod N C and Anand G V 2004 Nonlinear wavelet denoising for DOA estimation by music *2004 Int. Conf. on Signal Processing & Communications (SPCOM)* **1** 388–92