Journal of Systems Engineering and Electronics
Vol. 31, No. 4, August 2020, pp.712 – 721

An improved de-interleaving algorithm of radar pulses based on SOFM with self-adaptive network topology

JIANG Wen, FU Xiongjun*, CHANG Jiayun, and QIN Rui

School of Information and Electronics, Beijing Institute of Technology, Beijing 100081, China

Abstract: As a core part of the electronic warfare (EW) system, de-interleaving is used to separate interleaved radar signals. As interleaved radar pulses become more complex and denser, intelligent classification of radar signals has become very important. The self-organizing feature map (SOFM) is an excellent artificial neural network, which has huge advantages in intelligent classification of complex data. However, the de-interleaving process based on SOFM is faced with the problems that the initialization of the map size relies on prior information and the network topology cannot be adaptively adjusted. In this paper, an SOFM with self-adaptive network topology (SANT-SOFM) algorithm is proposed to solve the above problems. The SANT-SOFM algorithm first proposes an adaptive proliferation algorithm to adjust the map size, so that the initialization of the map size is no longer dependent on prior information but is gradually adjusted with the input data. Then, structural optimization algorithms are proposed to gradually optimize the topology of the SOFM network in the iterative process, constructing an optimal SANT. Finally, the optimized SOFM network is used for de-interleaving radar signals. Simulation results show that SANT-SOFM could get excellent performance in complex EW environments and the probability of getting the optimal map size is over 95% in the absence of priori information.

Keywords: de-interleaving, self-organizing feature map (SOFM), self-adaptive network topology (SANT).

DOI: 10.23919/JSEE.2020.000046

1. Introduction

With the development of electronic warfare (EW), the emission signals intercepted by the EW system have been interleaved into complex pulse trains. To separate the trains effectively, an advanced de-interleaving theory has become very critical.

Literature contains many de-interleaving approaches, which can be divided into two categories: pulse repetition interval (PRI) analysis and feature clustering. Mahmoud [1] introduced the PRI analysis into the de-interleaving process. Mostafa [2] introduced a technique based on cumulative differences (C-DIF) of PRI histograms and Milojevic [3] and Mostafa [4] improved this method by using sequential differences (S-DIF) of histograms which need less calculation. On the other hand, some of the scholars utilized hidden Markov models [5] and pulse correlation [6] for this problem. In the latest research, Torun and Orhan [7] proposed a method to de-interleave the signals with the stagger PRI and dwell-switch PRI types, which has some reference value to the special de-interleaving process.

As the appearance of complex PRI patterns in evolving modern EW environments, it has been increasingly difficult to use PRI analysis de-interleaving complex pulse trains. Thus, clustering algorithms depending on individual parameters of radar pulses have become an indispensable task to assist the de-interleaving process. Liu and Zhang [8] proposed an improved algorithm based on clustering and S-DIF, which could effectively analyze and estimate radar signals' PRI in complex EW environments. Amini and Amineh [9] introduced the multi-density clustering into the de-interleaving process, which reduced the dependence of clustering results on the initial prototype. Notably, the multi-prototype clustering technique proposed by Silva and Wunsch [10] also achieved good results.

Compared to the conventional clustering approaches, the self-organizing feature map (SOFM) [11,12] has been a more effective method for radar pulses clustering. SOFM is a topology preserving mapping network, which is used to simulate the brain's regional functions, self-organization and neuronal excitability. Granger and Savaria [13] introduced SOFM into the de-interleaving process and evaluated the performance of SOFM. Based on [13], Ata'a [14] and Gencol [15] further improved this process, achieving intelligent and automatic de-interleaving of radar signals. Ata'a [14] used three radar features, including angle-of-arrival (AOA), pulse width (PW) and radio frequency

Manuscript received August 27, 2019.

*Corresponding author.

This work was supported by the National Natural Science Foundation of China (61571043) and the 111 Project of China (B14010).

(RF), while Gencol [15] added estimated pulse amplitude (PA) as the fourth information source to assist the clustering process. Notably, Dai and Lei [16] proposed a noise suppression algorithm and realized the SOFM-based de-interleaving in noisy environments.

The studies mentioned above all focus on how to use the SOFM algorithm, ignoring the improvement of the algorithm itself. In SOFM, the network topology implies the impact on clustering performance, making the map size and the network structure important factors affecting SOFM. However, the conventional network has the following problems: the setting of the map size depends on prior information and the network structure cannot be self-adjusted. Therefore, the conventional network topology could not meet the needs of the modern electronic warfare, and a new network with a self-adaptive topology has become the key to solving these problems.

In order to address the above dilemmas, a new network, presented as SOFM with self-adaptive network topology (SANT-SOFM), is proposed to make the network self-adaptive. The new network [17 – 19] possesses the following innovations. Firstly, the topology of the SOFM network is no longer fixed, but dynamically adjusted as the input changes. The network is initialized to a topology with a small map size and gradually optimized by the proposed algorithms. Secondly, an adaptive proliferation algorithm is proposed to adjust the map size, making the map size no longer depend on prior information but gradually adjust itself with the input data. Thirdly, structural optimization algorithms (including neuronal elimination, merging and division) are used to optimize the topology of the SOFM network, constructing an optimal SANT. Simulation results show that the proposed algorithm could not only adapt to the complex and variable EW environments, but also obtain better clustering effects, thus improving the de-interleaving performance effectively.

2. Related work

SOFM has been used to solve various problems, such as vector quantization [20], clustering [21,22], and classification [23,24]. In this section, we briefly introduce the SOFM-based de-interleaving algorithm.

2.1 SOFM network

SOFM consists of an input layer and an output layer, as shown in Fig. 1. The input layer contains N nodes, corresponding to N inputs. The output layer contains $M = m_1 \times m_2$ neurons, which means the map size is $m_1 \times m_2$.

The input vectors are mapped from the input layer to the output layer, and SOFM uses a large amount of inputs to adjust neuronal weights and eventually makes the neurons sensitive only to specific patterns.

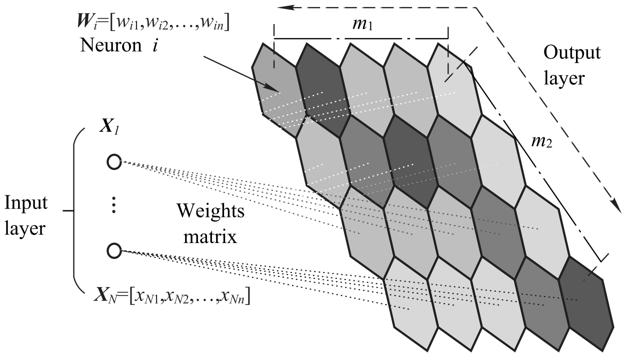

W_i=[w_{i1},w_{i2},…,w_{in}]
Neuron i
m_1
Output layer
m_2
X_1
Input layer
Weights matrix
X_N=[x_{N1},x_{N2},…,x_{Nn}]

Fig. 1 SOFM network

2.2 SOFM-based de-interleaving

The SOFM-based de-interleaving algorithm is shown as follows.

Step 1 Parameter settings: $\boldsymbol{X}_1, \boldsymbol{X}_2, \dots, \boldsymbol{X}_N$ are input radar feature vectors, $\boldsymbol{Y}_1, \boldsymbol{Y}_2, \dots, \boldsymbol{Y}_M$ are the neurons and the weight of $\boldsymbol{Y}_i$ is $\boldsymbol{W}_i$.

Step 2 Initialize the map size $m_1 \times m_2$, weights $\boldsymbol{W}_i$ $(i = 1, 2, \dots, M)$, iteration times T, the learning coefficient $\eta(t)$, and the neighborhood radius $N_g(t)$.

Step 3 Normalize the input and weights.

$$\overline{\boldsymbol{X}}_k = \frac{\boldsymbol{X}_k}{\|\boldsymbol{X}_k\|_2}, \quad k = 1, 2, \dots, N \tag{1}$$

$$\overline{\boldsymbol{W}}_i = \frac{\boldsymbol{W}_i}{\|\boldsymbol{W}_i\|_2}, \quad i = 1, 2, \dots, M \tag{2}$$

Step 4 Input $\boldsymbol{X}_k$.

Step 5 Select a winning neuron g by finding the minimum distance d_g:

$$d_g = \min_i [d_i], \quad i = 1, 2, \dots, M,$$
$$d_i = \|\overline{\boldsymbol{X}}_k - \overline{\boldsymbol{W}}_i\|_2. \tag{3}$$

Step 6 Adjust the neuronal weights in g's neighborhood.

$$\boldsymbol{W}_i(t+1) = \begin{cases} \overline{\boldsymbol{W}}_i(t) + \boldsymbol{D}_i, & i \in N_g(t) \\ \overline{\boldsymbol{W}}_i(t), & \text{other} \end{cases} \tag{4}$$

$$\boldsymbol{D}_i = \eta(t)[\overline{\boldsymbol{X}}_k - \overline{\boldsymbol{W}}_i(t)] \tag{5}$$

Step 7 Normalize the new weights.

$$\overline{\boldsymbol{W}}_i(t+1) = \frac{\boldsymbol{W}_i(t+1)}{\|\boldsymbol{W}_i(t+1)\|_2}, \quad i \in N_g(t) \tag{6}$$

Step 8 If $k = N$, go to Step 9; otherwise, $k = k + 1$ and return to Step 4.

Step 9 Update $\eta(t)$ and $N_g(t)$.

$$\eta(t) = \left(1 - \frac{t}{T}\right)\eta(0) \tag{7}$$

$$N_g(t) = \mathrm{e}^{-\frac{t}{T}} \cdot N_g(0) \tag{8}$$

Step 10 Let $t = t + 1$ and $k = 1$, return to Step 4 until $t = T$.

Step 11 The final weights $\boldsymbol{W}_1, \dots, \boldsymbol{W}_M$ would be the clustering centers.

3. Problem statement

Numerous studies have shown that network topology of SOFM implies the impact on the clustering performance, making the map size and the network structure important factors affecting SOFM. However, the conventional network has the following problems.

(i) Initialization of the map size relies on prior information. In conventional SOFM, the map size can only be preset to a fixed value based on prior information. In modern EW environments, it is difficult for the interceptor to obtain sufficient prior information to initialize the map size, which seriously weakens the role of SOFM as an optimal vector quantizer. In order to initialize the map size without prior information, the network topology should be self-adaptive under un-supervised learning conditions.

(ii) The network structure cannot be self-adaptively adjusted. In modern EW environments, the emission signals are complex and variable, resulting in irregular changes in intercepted pulse trains. The conventional network topology cannot be adaptively adjusted with these changes, which seriously affects the performance of SOFM. Therefore, it has been very critical to construct a new SOFM network with an adaptive topology.

4. SANT-SOFM algorithm

To solve the above problems, the SANT-SOFM algorithm is proposed. The algorithm realizes the dynamic adjustment of the network topology by regulating the neurons. First, the SOFM network is initialized to a topology with a small map size. Second, a proliferation algorithm is proposed to promote the rapid growth of neurons while preserving the data topology, preliminarily determining the map size and the distribution of neurons. Third, the Kohonen algorithm [11] is used to train the network. Finally, structural optimization algorithms are used to optimize the topology of the network.

4.1 Proliferation algorithm

In order to effectively initialize the map size of the SOFM network, the distribution of neurons should be determined. Therefore, this paper designs a proliferation algorithm to initially determine the topology of the SOFM network. The algorithm first initializes the network to a smaller topology, and then gradually expands the network based on the degree of similarity between the input data and the winning neuron. In order to effectively measure the similarity between the input and the winner, this paper designs a similarity threshold that can be adaptively adjusted using fuzzy operations [19]. If the similarity between the input and the winner is higher than the threshold, the input signal will be classified into the winning category, and the winning neuron will also be updated; otherwise, the input signal will be regarded as a new signal category, and the network needs to be expanded. Through the proliferation algorithm, the topology of the SOFM network is gradually adapted to the input data, thereby effectively determining the map size under the condition that the prior information is missing. The proliferation algorithm is divided into the following steps.

Step 1 Initialize the network.

The output layer is initialized to a small network, where the map size is $m_1 \times m_2$, the neurons are $\boldsymbol{Y}_1, \boldsymbol{Y}_2, \dots, \boldsymbol{Y}_M$ and the weight of $\boldsymbol{Y}_i$ is $\boldsymbol{W}_i$.

Step 2 Input data.

$\boldsymbol{X}_k = [x_{k1}, x_{k2}, \dots, x_{kn}]$ $(k = 1, 2, \dots, N)$ is input feature vector used to train the network, N is the vector' number and n is the dimension.

Step 3 Normalize the input and weights.

$$\overline{\boldsymbol{X}}_k = \frac{\boldsymbol{X}_k}{\|\boldsymbol{X}_k\|_2}, \quad k = 1, 2, \dots, N \tag{9}$$

$$\overline{\boldsymbol{W}}_i = \frac{\boldsymbol{W}_i}{\|\boldsymbol{W}_i\|_2}, \quad i = 1, 2, \dots, M \tag{10}$$

Step 4 Select the winner g.

$$d_g = \min_i [d_i], \quad i = 1, 2, \dots, M,$$

$$d_i = \|\overline{\boldsymbol{X}}_k - \overline{\boldsymbol{W}}_i\|_2. \tag{11}$$

Step 5 Calculate the complement codes.

Since complement-coding can effectively suppress the excessive proliferation of neurons [19], this paper complement-codes the input and the winning neurons before calculating the similarity between the two vectors. The complement code of $\overline{\boldsymbol{X}}_k$ is

$$\boldsymbol{I} = (\overline{\boldsymbol{X}}_k, \overline{\boldsymbol{X}}_k^c) =$$

$$(\overline{x}_{k1}, \overline{x}_{k2}, \dots, \overline{x}_{kn}, \overline{x}_{k1}^c, \overline{x}_{k2}^c, \dots, \overline{x}_{kn}^c) \tag{12}$$

where $\overline{x}_{ki}^c = 1 - \overline{x}_{ki}$ $(i = 1, 2, \dots, n)$. Similarly, the complement code of $\overline{\boldsymbol{W}}_g$ is

$$\boldsymbol{T} = (\overline{\boldsymbol{W}}_g, \overline{\boldsymbol{W}}_g^c). \tag{13}$$

Step 6 Calculate the distance $\boldsymbol{d} = (d_1, \dots, d_{2n})$.

$$d_i = |T_i - I_i|, \quad i = 1, 2, \dots, 2n \tag{14}$$

Step 7 Calculate the similarity threshold ρ of the winning neuron g.

$$\rho=\frac{1}{2n}\sum_{i=1}^{2n}\left(1-\mathrm{e}^{-md_i}\right) \tag{15}$$

The threshold ρ is adaptively adjusted as the distance d_i changes. The closer the input and the winning neurons are, the lower the threshold will be, and vice versa.

Step 8 Judge the similarity according to the following similarity criterion:

$$\frac{\|\boldsymbol{I}\wedge\boldsymbol{T}\|_1}{\|\boldsymbol{I}\|_1}\geqslant\rho. \tag{16}$$

"$\wedge$" is the fuzzy operator AND, which is an effective method for measuring the similarity between two vectors [19]. The operation rules are shown as follows:

$$\boldsymbol{p}\wedge\boldsymbol{q}=(\min(p_1,q_1),\min(p_2,q_2),\dots,\min(p_n,q_n)). \tag{17}$$

Step 9 If the similarity criterion is satisfied, the weight of the winning neuron should be updated to

$$\overline{\boldsymbol{W}}_g^{\text{new}}=\beta\boldsymbol{\varDelta}_g+(1-\beta)\overline{\boldsymbol{W}}_g^{\text{old}},$$

$$\boldsymbol{\varDelta}_g=(\overline{\boldsymbol{X}}_k\wedge\overline{\boldsymbol{W}}_g^{old}) \tag{18}$$

where $\overline{\boldsymbol{W}}_g^{\text{old}}$ and $\overline{\boldsymbol{W}}_g^{\text{new}}$ are the normalized weights of the winner g before and after the update, respectively; β is the initial learning rate of the SOFM network, which is expressed as

$$\beta=\eta(0). \tag{19}$$

Step 10 If the similarity criterion fails, a set of neurons would be generated. The rules of the generation are shown as follows.

Assume that the neighboring neurons of the winner g are $\{\boldsymbol{y}_i, i=1,2,\dots,6\}$, and the normalized weight of $\boldsymbol{y}_i$ is $\overline{\boldsymbol{w}}_i$. The similar neighboring neuron l can be obtained by

$$d_l=\min_i[d_i],\quad i=1,2,\dots,6,$$

$$d_i=\|\overline{\boldsymbol{W}}_g-\overline{\boldsymbol{w}}_i\|_2. \tag{20}$$

As shown in Fig. 2, a new neuron is generated between neurons g and l, and the weight of the new neuron is initialized to

$$\overline{\boldsymbol{W}}_{M+1}=\overline{\boldsymbol{X}}_k. \tag{21}$$

The voids created by the generation are filled by neurons, and the weight of the filled neuron i is initialized to

$$\overline{\boldsymbol{W}}_i=(1,\dots,1)_{1\times n}. \tag{22}$$

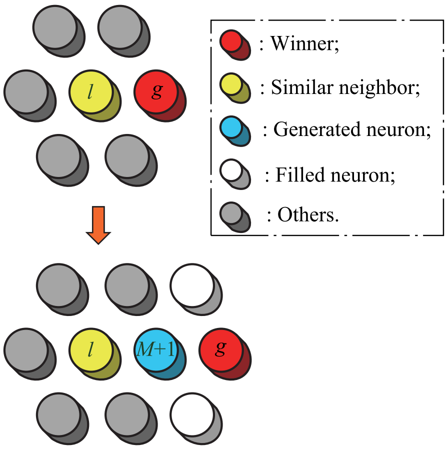


Fig. 2 Proliferation algorithm

4.2 Adaptive adjustment of network topology

The map size of the expanded SOFM network has been adapted to the input data, but the network topology is not optimal. For example, there are a large number of redundant neurons that are not activated. Therefore, the topology of the network needs to be further optimized. This paper first uses the traditional Kohonen algorithm [11] to train the network to map the input in an orderly manner. When the training is completed, the algorithms in this section will be used to gradually optimize the network topology.

4.2.1 Parameter settings

The parameters used to optimize the network topology are shown as follows.

(i) M is the neuronal number in the current network. The value of M changes as the network changes.

(ii) D_{Th} is the distance threshold of neighboring neurons, which is used to limit the similarity of neighboring neurons.

Due to the normalization of input vectors, the similarity criterion ignores the contribution of the vector length to the vectors' differences. In this section, the distance threshold D_{Th} is introduced as a criterion for network optimization to further distinguish the differences in vectors:

$$D_{\mathrm{Th}}=\frac{\sum\limits_{k=1}^{N}\sum\limits_{i=1}^{N}d_{ki}}{n^{\mathrm{af}}},$$

$$d_{ki}=\|\boldsymbol{X}_k-\boldsymbol{X}_i\|_2. \tag{23}$$

af is the adjustment coefficient [18], which is used to adjust the distance threshold. A large number of experiments show that when A is set to 0.3, the proposed algorithm could obtain good sorting effect. n is the number of

Euclidean distances between different vectors:

$$n = (N-1)N. \tag{24}$$

(iii) $C^{(1)}(i)$ $(i = 1, 2, \dots, M)$ is used to record the winning number of the neuron i.

(iv) $C^{(1)}_{\text{Th}}$ is the neuronal extinction threshold. If the winning number of the neuron i is less than the extinction threshold, the signal class represented by the neuron i is considered to be disappeared, and i needs to be eliminated.

(v) $C^{(2)}(i)$ $(i = 1, 2, \dots, M)$ is used to record the number of inputs that are classified to the neuron i, but differ from i (the distance is greater than D_{Th}).

(vi) $C^{(2)}_{\text{Th}}$ is the neuronal splitting threshold. If $C^{(2)}(i) > C^{(2)}_{\text{Th}}$, the neuron i corresponds to multiple signal classes. Therefore, the neuron i should be divided.

4.2.2 Neuronal elimination

If the winning number of the neuron i is less than $C^{(1)}_{\text{Th}}$, i.e., it satisfies

$$C^{(1)}(i) < C^{(1)}_{\text{Th}}, \tag{25}$$

it means that the neuron i cannot represent any of the current signal classes. In other words, the signal class represented by the neuron i has disappeared. Therefore, the neuron i should be eliminated to achieve the forgetting function, as shown in Fig. 3.

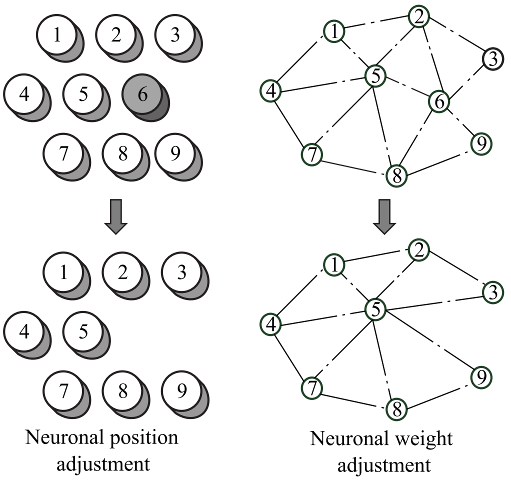


Fig. 3 Neuronal elimination

4.2.3 Neuronal merging

If the distance between neighboring neurons i and k is less than the distance threshold, i.e., it satisfies

$$D_{ik} = \|\boldsymbol{W}_i - \boldsymbol{W}_k\|_2 \leqslant D_{\text{Th}}, \tag{26}$$

it means that the two neurons represent the same signal class, and neurons i and k should be merged. The rules of merging is to eliminate neurons i and k, and generate a new neuron at the neuron i's position, as shown in Fig. 4. The weight of new neuron is

$$\boldsymbol{W}_i = \frac{(\boldsymbol{W}_i + \boldsymbol{W}_k)}{2}. \tag{27}$$

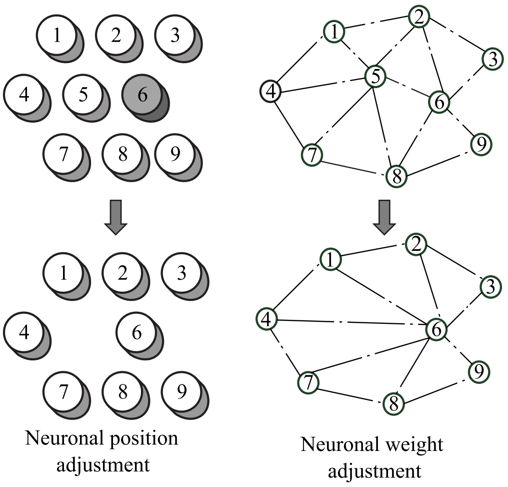


Fig. 4 Neuronal merging

4.2.4 Neuronal division

If the data in the class i, represented by the neuron i, differ a lot, i.e., they satisfy

$$C^{(2)}(i) > C^{(2)}_{\text{Th}}, \tag{28}$$

it means that the class i contains multiple signal classes due to the excessive merging. Therefore, a new neuron should be split from the neuron i, to represent the new signal class.

The division rule is to find the neuron i's similar neighboring neuron l (the method is the same as in (20)), and to grow a new neuron $M+1$ between i and l, as shown in Fig. 5. The weight of the new neuron is initialized to

$$\boldsymbol{W}_{M+1} = \frac{\boldsymbol{W}_i + \boldsymbol{W}_l}{2}. \tag{29}$$

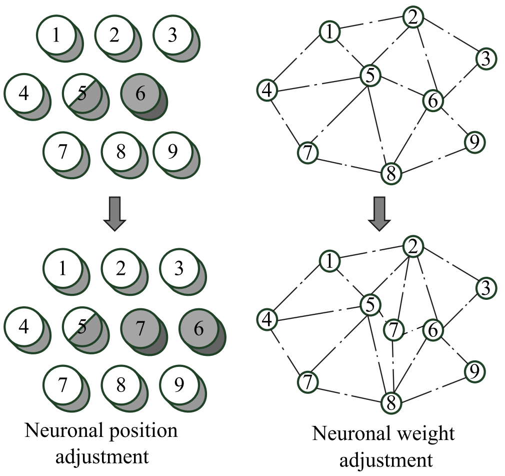


Fig. 5 Neuronal division

5. Simulation and discussion

In the following section, a comparative study of SOFM [15] and SANT-SOFM is accomplished to evaluate the radar pulse de-interleaving performance of the two algorithms. The performance evaluation involves the network topology, the number of sample hits and the clustering result. For each experiment in the comparative study, it has been repeated 100 times under the condition of random input orders. The result of each experiment is the mean of the 20 repetitions.

In these experiments, the learning coefficient is initialized to 0.6 and the iteration times is set to 100. For other parameters, grid searches are used for parameter tuning. The interval [0,3] with a step size of 1 is used for searching optimal initial neighborhood radius. The 2-dimensional grid search [15] based on the principal component analysis (PCA) is used to initialize the SOFM network.

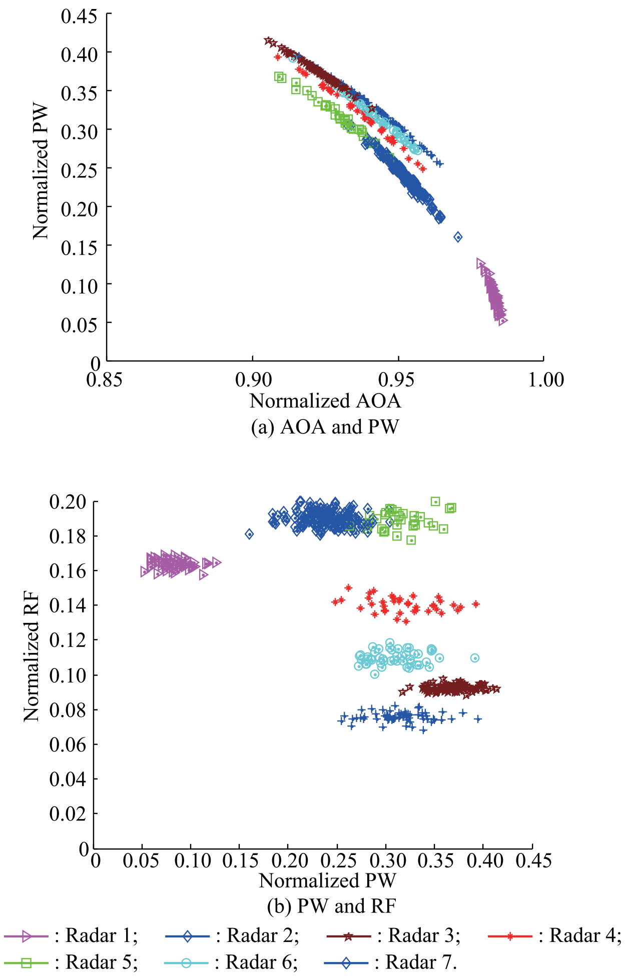
For SANT-SOFM, the initial map size is 2×2. $C_{\mathrm{Th}}^{(1)}$ and $C_{\mathrm{Th}}^{(2)}$ are set to 5 and 50, respectively.

5.1 Experimental data

The parameters of experimental data and the distribution of normalized radar features are shown in Table 1 and Fig. 6, respectively.

Table 1 Signal parameters

Mode	Radar	AOA/(°)	PW/μs	RF/MHz	Pulse loss/%	PRI
Multi-mode	1	29.5 – 31.5	9.5 – 10.5	2.28 – 2.52	10	Constant: 400 μs
	2	29.5 – 31.5	9.5 – 10.5	3.32 – 3.68	10	Constant: 400 μs
	3	29.5 – 31.5	9.5 – 10.5	4.28 – 4.73	10	Jitter: 10%; Center: 400 μs
	4	29.5 – 31.5	9.5 – 10.5	5.7 – 6.3	10	Jitter: 30%; Center: 400 μs
Single-mode	5	38 – 42	9.5 – 10.5	7.6 – 8.4	5	Jitter: 20%; Center: 100 μs
	6	47.5 – 52.5	19 – 21	4.75 – 5.25	5	Hopping: 10%; Center: 190 μs
	7	57 – 63	4.75 – 5.25	9.5 – 10.5	5	Grops: 300/330/370/420 μs



(a) AOA and PW

(b) PW and RF

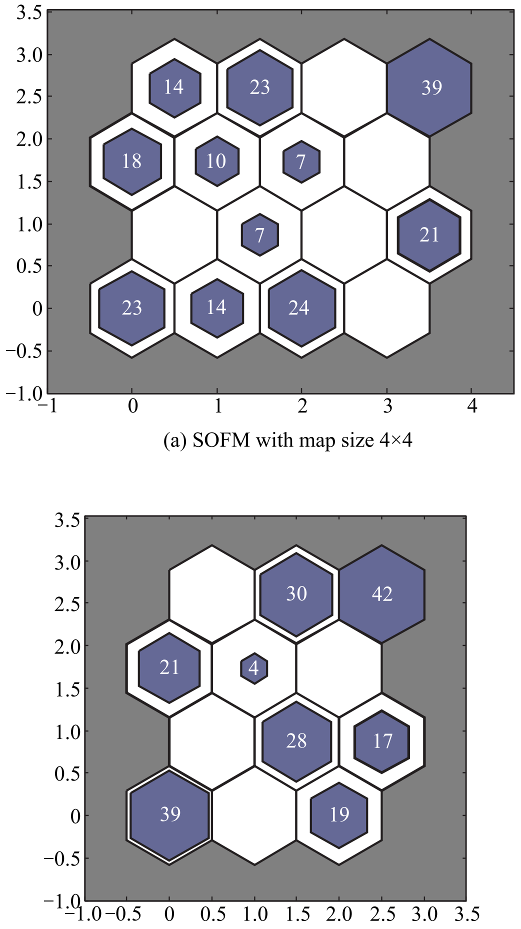
Fig. 6 Distribution of radar features

In order to simulate the dynamic and variable emission signals in the complex EW environments, the parameters of radar models constructed in this paper are all dynamically changed, as shown in Table 1. Signals 1 – 4 are used to simulate the four sub-signals of multi-mode and multi-parameter radars, and signals 5 – 7 are used to simulate single-mode radars. It is worth pointing out that the parameters of these radar models have certain similarities. The similarity of the parameters has great advantages in accurately evaluating clustering performance of the two algorithms.

In order to investigate the performance of SANT-SOFM algorithms, the simulations in this paper are implemented in noisy and noise-free environments. In Section 5.2, simulations are performed in noise-free environments; 15% of the Gaussian noise signals are added to the experimental data.

5.2 Result and discussion

Simulation results of the comparative study are shown in Fig. 7 to Fig. 10. Fig. 7(a) and Fig. 7(b) show the topological map of SOFM with the map sizes 4×4 and 4×3, respectively. The activated neurons and the hit number of each neuron have been marked in the topology. Fig. 8(a) shows the topological map that is expanded from a 2×2 initial network using the proliferation algorithm. The topology in Fig. 8(a) is optimized using the optimization algorithms proposed in this paper, and the result of the optimization is shown in Fig. 8(b). The comparative study of the two algorithms is shown in Fig. 9 to compare clustering effects, and the clustering results of SANT-SOFM in noise environments are shown in Fig. 10.



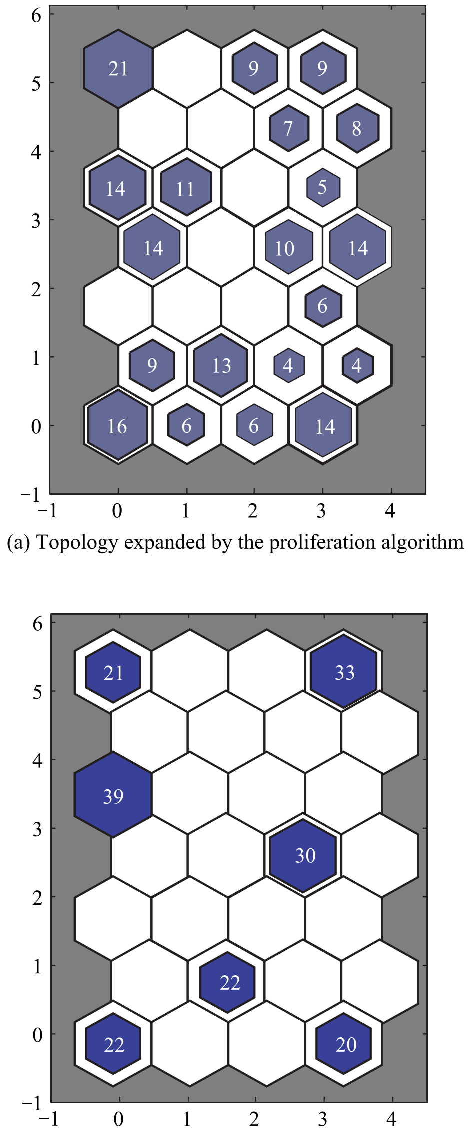
(a) SOFM with map size 4×4

(b) SOFM with map size 4×3

Fig. 7 Topology of SOFM and its hits' number

Fig. 7 shows that the network topology of SOFM implies the impact on mapping results. When the map size is set to 4×4, as shown in Fig. 7(a), 11 neurons in the network are activated while only seven signal sources exist; when the map size is 4×3, as shown in Fig. 7(b), the number of activated neurons is reduced to eight. The results illustrate that the network topology is an important factor and different map sizes could bring different clustering effects.

To find the best topology, the conventional SOFM relies on prior information to initialize the map size of the network. However, in most cases, it is very difficult to get sufficient prior information for choosing an optimal map size. Therefore, this paper designs a network with an adaptive topology, and achieves the automatic acquisition of the optimal map size.



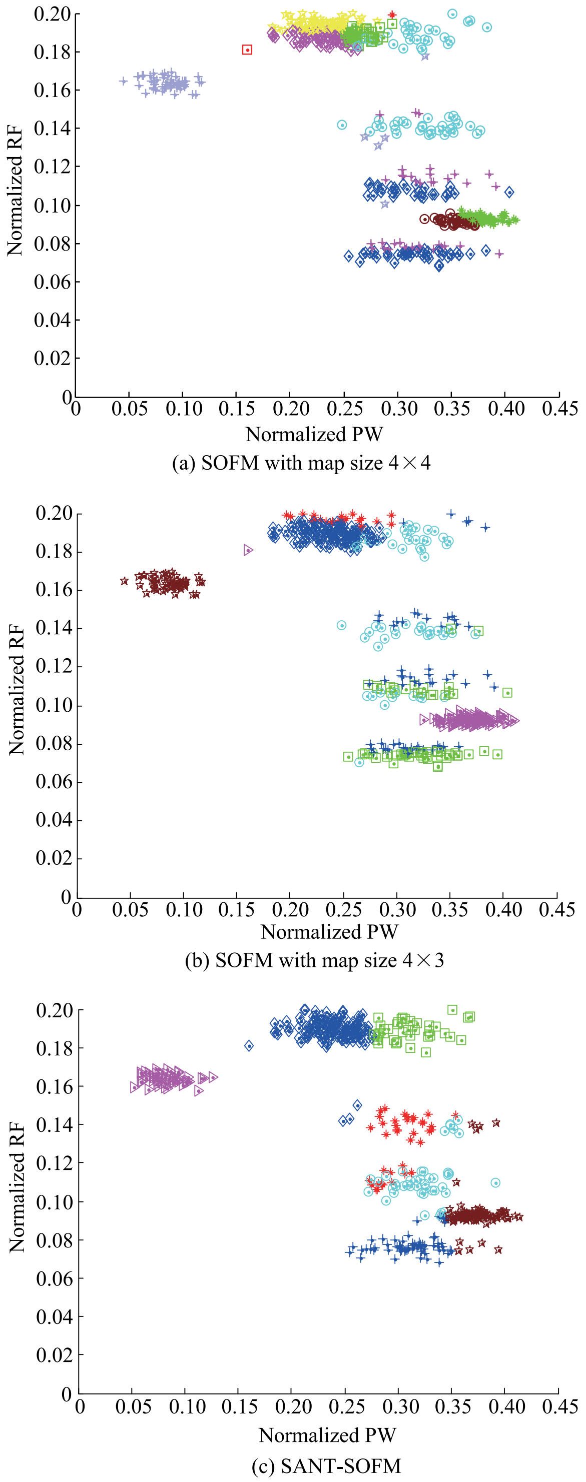
(a) Topology expanded by the proliferation algorithm

(b) Topology optimized by the optimization algorithm

Fig. 8 Topology of SANT-SOFM and its hits' number

The new topology is firstly initialized to a small network and then expanded by the proposed proliferation algorithm. After the proliferation, as shown in Fig. 8(a), the map size is gradually adapted to the input data, and the activated neurons are distributed along the outer edge of the topological map. The more times a neuron is hit by the samples, the more active it would be.

Based on the neuronal activity, the neuronal optimization algorithm is introduced to further adjust the network topology, and the result of the optimization is shown in Fig. 8(b). As shown in Fig. 8(b), the number of the activated neurons is stabilized at seven, which is equal to the quantity of signal sources, thus obtaining the dynamically optimal topology.



(a) SOFM with map size 4×4

(b) SOFM with map size 4×3

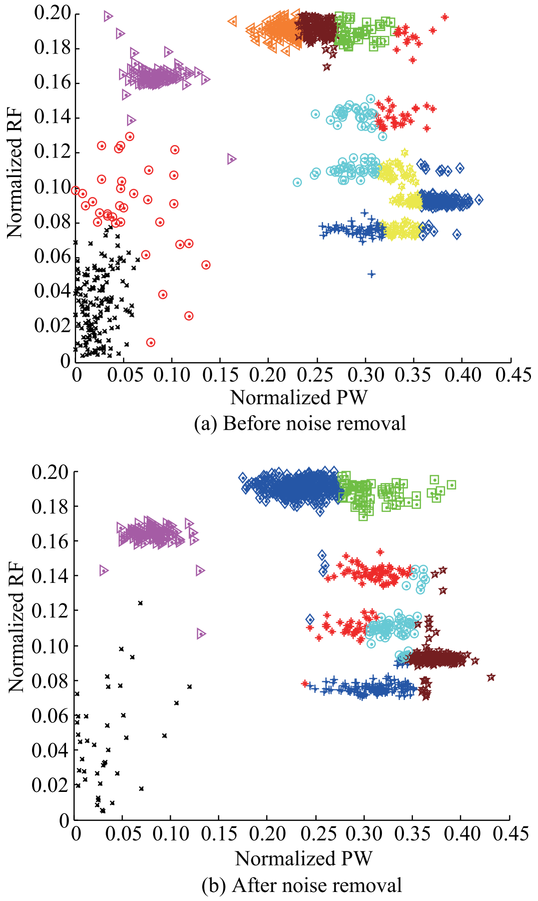
(c) SANT-SOFM

Fig. 9 Clustering results

The comparative study of SOFM and SANT-SOFM is shown in Fig. 9 to compare the clustering effects of different algorithms. Due to the high similarity and the complex variability of the signal sources used in this article, the clustering performance of the two algorithms is significantly disturbed, as shown in Fig. 9.

For SOFM, it is seriously affected by the disturbance. When the map size is 4×4, as shown in Fig. 9(a), the seven signal sources are classified into 11 categories, and the classified categories appear to overlap significantly. When the map size is adjusted to 4×3, as shown in Fig. 9(b), the number of clusters drops to eight, but the category overlapping problem is even more serious. The results illustrate that the traditional SOFM has a significant drop in the clustering performance when sorting complex and variable signal sources. Since the network topology of SOFM is fixed, these signals are easily misclassified regardless of the adjustment of the initial map size, causing the classified categories to overlap each other.

As for SANT-SOFM, its topology could be self-adaptively adjusted with the structure of the input data, making input data gradually approach clustering centers during the mapping process. Therefore, SANT-SOFM effectively improves the overlapping problem, as shown in Fig. 9(c), resulting in better clustering performance.



(a) Before noise removal

(b) After noise removal

Fig. 10 Performance in noisy environments

In order to evaluate the performance of the proposed algorithm in different environments, simulations in Fig. 10 are performed in Gaussian noisy environments. As shown in Fig. 10(a), the clustering quality of SANT-SOFM is significantly reduced in noise environments. Since the noise signals have changed the structure of the input data, the topology of the network could not be optimized to the optimal state, resulting in signal misclassification. Therefore, it is very important to use de-noising methods to remove the noise before the clustering process. In this paper, the algorithm in [16] is used to remove noise signals, and de-noised clustering results are shown in Fig. 10(b). As shown in Fig. 10(b), most of the noise has been removed and the remaining noise is classified as a new category. The clustering performance of the proposed algorithm is significantly improved due to the noise removal. Simulation results show that the SANT-SOFM algorithm has excellent performance in both noisy and noise-free environments.

De-interleaving performance is analyzed above, and then we will further analyze the computational complexity. The complexity (the symbols in parentheses represent the calculation amount) is derived from network initialization C_I, neuronal proliferation C_P, network optimization C_O and Kohonen training $N \times C_K$. Therefore, the computational complexity of the two algorithms is shown in Table 2.

Table 2 Computational complexity

Algorithm	Computational complexity
SOFM	$N_1 \times C_K + C_I$
SANT-SOFM	$N_2 \times C_K + C_I + C_P + C_O$

In these four parts, Kohonen training is the most important part affecting the complexity. The computation amount in Kohonen training is related to the number N of the activated neurons. The larger the value of N is, the higher the calculation amount will be.

The number of the activated neurons in the SOFM N_1 is greater than the number of signal sources, while in SANT-SOFM, the number N_2 is approximately equal to that of signal sources. Therefore, in Kohonen training, the computational complexity of SOFM is higher than that of SANT-SOFM. However, adaptive adjustment of the topology has a certain cost, and both proliferation and optimization consume a certain amount of computing resources. Therefore, compared to SOFM, the computational complexity of SANT-SOFM has not changed significantly.

In view of the above advantages, SANT-SOFM has great advantages in clustering and vector quantization, which makes it suitable for de-interleaving similar signals. With the development of modern EW, emission signals have become complex and variable, making the signals intercepted by EW systems interleave into complex pulse trains. To separate the pulse trains effectively, it needs an advanced de-interleaving algorithm, and SANT-SOFM has a huge advantage in this respect. Therefore, the research on SANT-SOFM is very promising.

6. Conclusions

This work presents the idea of using the self-adaptive network to replace the traditional output layer, showcased in SANT-SOFM, which can solve the initialization dilemma of map size and structural adjustment problems. It is completed by using neuronal proliferation and adjustment algorithms to optimize the network topology. Simulation results show that the proposed algorithm could improve the de-interleaving performance and address the dilemmas in SOFM-based de-interleaving systems.

References

[1] KESHAVARZI M, AMIRI D, PEZESHK A M, et al. A novel method of de-interleaving pulse repetition interval modulated sparse sequences in noisy environment. IEICE Trans. on Fundamentals of Electronics Communications and Computer Sciences, 2014, 97(5): 1136 – 1139.

[2] BAGHERI M, SEDAAGHI M H. A new approach to pulse de-interleaving based on adaptive thresholding. Turkish Journal of Electrical Engineering and Computer, 2017, 25(5): 3827 – 3838.

[3] MILOJEVIC D J, POPOVIC B M. Improved algorithm for the de-interleaving of radar pulse. IEEE Proceedings of Radar and Signal Processing, 1992, 139(1): 98 – 104.

[4] BAGHERI M, SEDAAGHI M H. A new method for detecting jittered PRI in histogram-based methods. Turkish Journal of Electrical Engineering and Computer, 2018, 26(3): 1214 – 1224.

[5] LOGOTHETIS A, KRISHNAMURTHY V. An interval-amplitude algorithm for de-interleaving stochastic pulse train sources. IEEE Trans. on Signal Processing, 1998, 46(5): 1344 – 1350.

[6] GE Z P, SUN X, REN W J, et al. Improved algorithm of radar pulse repetition interval de-interleaving based on pulse correlation. IEEE Access, 2019, 7: 30126 – 30134.

[7] TORUN O, KOCAMIS M B, ABACI H, et al. De-interleaving of radar signals with stagger PRI and dwell-switch PRI types. Proc. of the 25th Signal Processing and Communications Applications Conference, 2017: 89 – 97.

[8] LIU Y C, ZHANG Q Y. Improved method for de-interleaving radar signals and estimating PRI values. IET Radar, Sonar and Navigation, 2018, 12 (5): 506 – 514.

[9] AMINI A, SABOOHI H, HERAWAN T, et al. MuDi-Stream: a multi density clustering algorithm for evolving data stream. Journal of Network and Computer Applications, 2016, 59: 370 – 385.

[10] SILVA L E B, WUNSCH D C. Multi-prototype local density-based hierarchical clustering. Proc. of the International Joint Conference on Neural Networks, 2015: 1 – 9.

[11] KOHONEN T. Self-organizing maps. Neural Networks, 2006, 19: 723 – 733.

[12] ZHENG Z Y, CHEN Y Y. An improved pre-processing algorithm of radar signal sorting based on SOFM clustering. Aerospace Electronic Warfare, 2013, 29(3): 42 – 45.

[13] GRANGER E, SAVARIA Y, LAVOIE P, et al. A comparison of self-organizing neural networks for fast clustering of radar

pulses. Signal Processing, 1998, 64(3): 249 – 269.

[14] ATA'A A W, ABDULLAH S N. De-interleaving of radar signals and PRF identification algorithms. IET Radar, Sonar and Navigation, 2007, 1(5): 340 – 347.

[15] GENCOL K, KARA A. Improvements on de-interleaving of radar pulses in dynamically varying signal environments. Digital Signal Processing, 2017, 69: 86 – 93.

[16] DAI S B, LEI W H, CHENG Y Z, et al. Clustering of DOA data in radar pulse based on SOFM and CDbw. Journal of Electronics, 2014, 31(2): 107 – 114. (in Chinese)

[17] GORZAŁCZANY M B, RUDZIŃSKI F. Generalized self-organizing maps for automatic determination of the number of clusters and their multi-prototypes in cluster analysis. IEEE Trans. on Neural Networks and Learning Systems, 2018, 29(7): 2833 – 2845.

[18] MAJEED S, GUPTA A, RAJ D, et al. Uncertain fuzzy self-organization based clustering: interval type-2 fuzzy approach to adaptive resonance theory. Information Sciences, 2018, 424: 69 – 90.

[19] AHMED A, RUI Z, MEHDI N, et al. An effective density-based clustering and dynamic maintenance framework for evolving medical datastreams. International Journal of Medical Informatics, 2019, 126: 176 – 186.

[20] ADAM G. Preprocessing and analysis of ECG signals: a self-organizing maps approach. Expert Systems with Applications, 2011, 38(7): 9008 – 9013.

[21] DINO I, KALLIMANI V P, LAM H L. Using the self organizing map for clustering of text documents. Expert Systems with Applications, 2009, 36(5): 9584 – 9591.

[22] EVERTON B L, CARLOS A B. Segmentation of connected handwritten digits using self-organizing maps. Expert Systems with Applications, 2013, 40(15): 5867 – 5877.

[23] UNGLERT K, RADIĆ K, JELLINEK A M, et al. Principal component analysis vs. self-organizing maps combined with hierarchical clustering for pattern recognition in volcano seismic spectra. Journal of Volcanology and Geothermal Research, 2016, 320(15): 58 – 74.

[24] ANIMA M, LAXMIDHAR B, VENKATESH K. Emotion recognition from geometric facial features using self-organizing map. Pattern Recognition, 2014, 47(3): 1282 – 1293.

Biographies

JIANG Wen was born in 1991. He received his master's degree from Zhengzhou University, China, in 2016. Currently, He is pursuing his Ph.D. degree in the School of Information and Electronics, Beijing Institute of Technology (BIT). His research interests include radar signal processing and radar pulses deinterleaving.
E-mail: jwen912@126.com

FU Xiongjun was born in 1978. He received his B.E. and Ph.D. degrees from Beijing Institute of Technology (BIT), China, in 2000 and 2005 respectively. He is currently the vice dean of the School of Information and Electronics, BIT, and an associate professor and Ph.D. supervisor with BIT. His current research interests include radar system, radar signal processing, waveform design and automatic target recognition.
E-mail: fuxiongjun@bit.edu.cn

CHANG Jiayun was born in 1989. She received her master's degree from Beijing Institute of Technology (BIT), China, in 2016. Currently, she is pursuing her Ph.D. degree in the School of Information and Electronics, BIT. Her research interests include automatic target recognition and radar signal processing.
E-mail: 824400828@qq.com

QIN Rui was born in 1990. He received his master's degree from Wuhan University of Technology, China, in 2015. Currently, he is pursuing his Ph.D. degree in the School of Information and Electronics, Beijing Institute of Technology (BIT). His research interests include radar signal processing, SAR image processing and pattern recognition.
E-mail: qinrui90@163.com