Front. Comput. Sci., 2016, 10(4): 673–688
DOI 10.1007/s11704-015-4457-7

RESEARCH ARTICLE

A three-way incremental-learning algorithm for radar emitter identification

Xin XU (✉)[1], Wei WANG[2], Jianhong WANG[1]

1 Science and Technology on Information System Engineering Laboratory, Nanjing Research Institute of Electronic Engineering (NRIEE), Nanjing 210007, China

2 State Key Laboratory for Novel Software and Technology, Nanjing University, Nanjing 210093, China

Abstract Radar emitter identification has been recognized as an indispensable task for electronic intelligence system. With the increasingly accumulated radar emitter intelligence and information, one key issue is to rebuild the radar emitter classifier efficiently with the newly-arrived information. Although existing incremental learning algorithms are superior in saving significant computational cost by incremental learning on continuously increasing training samples, they are not adaptable enough yet when emitter types, features and samples are increasing dramatically. For instance, the intra-pulse characters of emitter signals could be further extracted and thus expand the feature dimension. The same goes for the radar emitter type dimension when samples from new radar emitter types are gathered. In addition, existing incremental classifiers are still problematic in terms of computational cost, sensitivity to data input order, and difficulty in multi-emitter type identification. To address the above problems, we bring forward a three-way incremental learning algorithm (TILA) for radar emitter identification which is adaptable for the increase in emitter features, types and samples.

Keywords radar emitter identification, incremental learning, classification, data mining

1 Introduction

Radar emitter identification has been recognized as an indispensable task for electronic intelligence system. By identifying radar emitter types, the potential platforms and associate threat levels of the radar emitter could be inferred. With the increasingly accumulated radar emitter intelligence and information, one key issue is to rebuild the radar emitter classifier efficiently using the newly-arrived information. The batch learning classifiers need to re-process the whole data whenever new training samples are available and thus unable to finish the task in a reasonable amount of time and memory space.

Incremental learning approaches are recognized as superior in saving significant computational cost by incremental learning on continuously increasing training samples [1–3]. However, existing incremental data stream methods are still problematic to be applied for three-way incremental emitter identification.

Firstly, as for increment in the sample dimension, quite a large number of benchmark incremental learning algorithms [4–8] are mostly based on binary partitions and are not adaptable enough yet for radar emitter multi-class classification. Besides, some incremental learning algorithms, such as incremental decision tree [9] and neural network [10–12], are based on a heuristic scheme due to complex model architectures and very sensitive to the training sample input order. Secondly, as for increment in the feature dimension, most data stream methods [13–16] are restricted to discrete features only, such as text streams where new words or phrases are evolving. Finally, as for increment in the class dimension, some data stream class evolution algorithms [17] are unsupervised and unable to apply for supervised multi-class

Received October 16, 2014; accepted August 18, 2015

E-mail: flora.xin.xu@gmail.com

classification. In addition, existing data stream class evolution methods [3,15,16,18] have not paid sufficient attention to the minority class. In such a case, the information of the minority class may be lost when the data chunk gets obsolete and the cohesion of test samples in the minority class is hard to be found.

In the practical application of incremental radar emitter identification, emitter types and features as well as the samples are increasing. Majority of the emitter features are continuous. For example, for the same radar emitter sample, intra-pulse characters of emitter signals, such as the wavelet transformation characters, could be further extracted and thus its continuous feature number increases. And so does the number of radar emitter types when samples from new radar emitter types are detected. Unfortunately, classic emitter identification methods [19,20] have paid little attention to the above problems yet. Even though a small number of incremental radar emitter identification methods [21–24] have been proposed, they are generally restrictive for sample increment only.

Therefore, we bring forward a three-way incremental learning algorithm (TILA) for radar emitter identification which is adaptable for the increase in emitter features, types and samples. Experimental results showed that TILA is robust to sample input order, superior in computational efficiency while yielding competitive identification accuracy when compared with the peers. We make the following contributions accordingly:

- TILA is able to adapt to the simultaneous increase in radar emitter features, both discrete and continuous, and types as well as the increase in radar emitter samples.
- TILA is much more robust to the training sample input orders when compared with the incremental learning heuristics.
- The time complexity of TILA is low, approximately linear w.r.t. the number of new emitter samples and quadratic w.r.t. the number of new features.

The rest of paper is organized as follows. We review related work in Section 2. Our three-way incremental learning algorithm (TILA) for radar emitter identification is formally presented in Section 3. In Section 4, we present our experimental results. Finally, We conclude in Section 5.

2 Related work

We partition the related work into three categories for increment in the sample, feature and class dimension respectively.

• Related work for increment in the sample dimension

Existing incremental learning algorithms are usually heuristics and are very sensitive to the training data input order.

Representatives of heuristic incremental learning algorithms include incremental neural network [10–12,25,26] and incremental decision tree [1,9,27]. Both algorithms suffer from high complexity in architecture and uncertainty in output during incremental learning. For instance, incremental neural network typically leads to different adaptive weights and different predictions for different orderings of the same training data.

Binary incremental learning heuristics include online perceptron [5], SVM [4,6,7], and maximum-margin [8] methods. The corresponding representatives are margin-perceptron [5], SGD-SVM [6], pegasos SVM [7] and ROMMA [8]. The perceptron algorithm trains a linear classifier to separate the data into two groups using a decision hyperplane. SGD SVM applies a stochastic gradient descent scheme [28,29] on SVM and obtains numerical rate of convergence. Pegasos SVM, on the other hand, alternates between stochastic gradient descent and projection steps when optimizing SVM functions. ROMMA is referred as the relaxed online maximum margin algorithm. ROMMA works by maintaining a relatively simple relaxation of a number of linear constraints, and could be viewed as an approximation to the online classifiers with maximum margin. To optimize both preference ranking and regression performance, efficient and effective combined regression and ranking strategies, such as [30], would be applied simultaneously with the above online classification algorithms.

As can be seen, the heuristic nature, constraint of binary partition scheme, and sensitivity to data input order of the above incremental learning algorithms make them unable to meet the demands of accuracy and efficiency in practical emitter identification.

• Related work for increment in the feature dimension

Most data stream feature evolution algorithms [13–16] are restricted to discrete features. These methods generally cope with text streams, in which new features of words or phrases appear as the text stream progresses. For instance, when a new document arrives, it would be checked whether there is any new word in the document. The new word will be added to a vocabulary, and the word statistics will be updated. However, in the application of incremental emitter identification, the majority of emitter features are continuous. In Refs. [3,18], different feature spaces would be constructed for dif-

ferent models in the ensemble. However, in that case, a global feature ranking required in the practical application of emitter identification would be difficult to obtain. In Ref. [31], a novel online feature selection framework for streaming features is constructed, whereas the number of training examples has to remain fixed.

• Related work for increment in the class dimension

In order to detect the novel class, a "one class" data stream method is proposed in Ref. [17] by building a normal model of the data using clustering. However, the method is unsupervised and unable to apply for multi-class emitter identification. Alternatively, to deal with the novel class discovery for multi-class classification, "MineClass and XMiner" [15] and DXMiner [16] build an ensemble of models to classify the unlabeled data and conduct outlier detection. Novel classes could be inferred when strong cohesion is discovered among outliers. Variants of [15,16] have been further brought forward in Refs. [3,18] where a flexible decision boundary for outlier detection is utilized to reduce the false alarm rate and increase the detection rate. However, these algorithms are not suitable for incremental emitter identification for minority emitter type identification. The information of minority emitter types in the obsolete data chunks may be lost. Also, due to the small sample size in test data, strong cohesion of outliers would be hard to be found. An incremental learning algorithm for new concept class discovery from unbalanced datasets, Learn++.UDNC, has been studied in Ref. [32], but it has not considered the unbalanced class distribution either.

In the field of incremental emitter identification, none of the existing approaches [21–24] is able to update the classification models incrementally to make the best use of the new features yet. For instance, besides the traditional radar emitter signal parameters of radio frequency (RF), pulse repetition interval (PRI) and pulse width (PW), suppose some new intra-pulse fine features have been extracted by the autocorrelation and inverse spectrum techniques in the time and frequency fields respectively, all existing incremental emitter identification approaches need to rebuild the classifier on all the available features just like the batch ones.

As discussed above, existing incremental learning approaches in both areas of machine learning and emitter identification are problematic when applied for three-way incremental learning in emitter identification.

3 Method

In this section, we formally present our incremental three-way incremental learning algorithm for radar emitter type identification (TILA). TILA deals with three basic cases of incremental learning: 1) *sample increment* where the new training samples all belong to the original radar emitter types, 2) *type increment* during which apart from the original radar emitter types, some new training samples belong to some new types, and 3) *feature increment* that new features for the original training samples become available, such as additional intra-pulse fine features. TILA would incrementally update the data description variables, upon which, discriminating features would be selected and identification would be made. Please note that the increment of radar emitter intelligence could proceed in any combinations of the above three basic ways.

3.1 Data description

Suppose the original radar emitter training data D is composed of m samples from k different types, denoted as Ω ($|\Omega| = m$), each radar emitter sample with n features ($n = nc + nd$, nc continuous and nd discrete). Each radar emitter sample is denoted as $x_s = \{x_c(s,1), x_c(s,2), ..., x_c(s,nc), x_d(s,1), x_d(s,2), ..., x_d(s,nd), cls_s\}$, where $1 \leqslant s \leqslant m$, x_c denotes the continuous features, x_d the discrete features, and cls_s indicates the type of the sth radar emitter sample. Each radar emitter sample belongs to one of the k different types $\{c_1, c_2, ..., c_k\}$. Each type j, where $1 \leqslant j \leqslant k$, has m_j ($\sum_{j=1}^{k} m_j = m$) different samples, denoted as Ω_j where $|\Omega_j| = m_j$ and $\cup_{j=1}^{k} \Omega_j = \Omega$. The incrementally accumulated radar emitter samples are independent from each other and may taken from different radar emitters.

We construct four data description matrices for both continuous and discrete features. In this way, the discriminating power of each feature could be evaluated, the best features could be selected, and an optimal classifier could be built. Below is the data description of continuous and discrete radar emitter features respectively:

3.1.1 Description of continuous feature

For continuous radar emitter features such as the pulse description word of radio frequency, pulse width, pulse amplitude, direction of arrival and time of arrival, we define the feature-type sum matrix, feature-type square sum matrix and feature-type pair matrix.

Definition 1 (feature-type sum matrix $\Sigma(i,j)$) We define each matrix element $\Sigma(i,j)$ as the sum of values of continuous feature i on all the radar emitter training samples in type

j, where $1 \leqslant i \leqslant nc$ and $1 \leqslant j \leqslant k$. Mathematically speaking,

$$\Sigma(i, j) = \Sigma_{s\in\Omega_j} x_c(s, i). \tag{1}$$

Definition 2 (feature-type square matrix $\Sigma^2(i, j)$) We define each matrix element $\Sigma^2(i, j)$ as the sum of square values of continuous feature i on all the radar emitter training samples in type j, where $1 \leqslant i \leqslant nc$ and $1 \leqslant j \leqslant k$. Mathematically speaking,

$$\Sigma^2(i, j) = \Sigma_{s\in\Omega_j} x_c(s, i)^2. \tag{2}$$

Definition 3 (feature-type pair matrix $\Sigma^{pair}(p, q, j)$) We define each matrix element $\Sigma^{pair}(p, q, j)$ as the sum of product of values of continuous feature p and feature q on all the radar emitter training samples in type j, where $1 \leqslant p, q \leqslant nc$ and $1 \leqslant j \leqslant k$.

$$\Sigma^{pair}(p, q, j) = \Sigma_{s\in\Omega_j} x_c(s, p)x_c(s, q). \tag{3}$$

We calculate the $\Sigma(i, j)$ and $\Sigma^2(i, j)$ for all the $nc\times k$ feature-type combinations, thus forming a feature-type sum matrix $[\Sigma(i, j)]_{nc\times k}$ and a feature-type square matrix $[\Sigma^2(i, j)]_{nc\times k}$. Likewise, we calculate $\Sigma^{pair}(p, q, j)$ for all the $C_n^2 k$ feature pair-type combinations and form a feature-type pair matrix $[\Sigma^{pair}(p, q, j)]_{nc\times nc\times k}$.

The mean value and sample standard deviation of each continuous feature i on each radar emitter type j could be finely calculated based on the above three matrices. The mean value of continuous feature i on type j is equal to the ratio of $\Sigma(i, j)$ to the number of radar emitter samples in type j, m_j, as indicated in Eq. (4):

$$\overline{X_{ji}} = E(x_c(s, i))_{s\in\Omega_j} = \frac{\Sigma(i, j)}{m_j}. \tag{4}$$

The radar emitter sample standard deviation of each continuous feature i on each type j could also be inferred upon the feature-type sum matrix and the feature-type square matrix, as shown in Eq. (5).

$$\begin{aligned} &stdev(X_{ji}) \\ &= stdev(x_c(s, i))|_{s\in\Omega_j} \\ &= \sqrt{\frac{\sum_{s\in\Omega_j}(x_c(s, i) - E(x_c(s, i)))^2}{m_j - 1}} \\ &= \sqrt{\frac{\sum_{s\in\Omega_j} x_c(s, i)^2}{m_j - 1} - \frac{(\sum_{s\in\Omega_j} x_c(s, i))^2}{m_j(m_j - 1)}} \\ &= \sqrt{\frac{\Sigma^2(i, j)}{m_j - 1} - \frac{(\Sigma(i, j))^2}{m_j(m_j - 1)}}. \end{aligned} \tag{5}$$

The estimated radar emitter sample covariance coefficient between feature p and q for type j is calculated by Eq. (6):

$$\begin{aligned} &cov(p, q, j) \\ &= \frac{\sum_{s\in\Omega_j}(x_c(s, p) - E(x_c(s, p)))(x_c(s, q) - E(x_c(s, q)))}{n_j - 1} \\ &= \frac{\Sigma^{pair}(p, q, j) - \frac{\Sigma(p, j)\Sigma(q, j)}{m_j}}{m_j - 1}. \end{aligned} \tag{6}$$

As can be seen clearly, the radar emitter sample covariance coefficient between any two continuous features p and q, as well as the mean value and the standard deviation of each continuous feature i, are determined by the three data description matrices.

3.1.2 Description of discrete feature

Some radar emitter features, such as radio frequency type and intra-pulse modulation type, are discrete or categorical. To better accommodate the discrete radar emitter features for radar emitter identification, we further define the feature-type frequency matrix below:

Definition 4 (feature-type frequency matrix $freq(i, j, v)$) We define each matrix element $freq(i, j, v)$ as the occurrence frequency of each value v of discrete feature i on all the training samples from type j, where $1 \leqslant i \leqslant nd$ and $1 \leqslant j \leqslant k$.

Mathematically speaking,

$$freq(i, j, v) = SUM_{s\in\Omega_j}(x_d(s, i) == v). \tag{7}$$

Given the discrete value v of discrete feature i from a radar emitter test sample, the membership probability of the discrete value v to each type j could be inferred according to the feature-type frequency matrix $freq(i, j, v)$. The membership probability of discrete value v of feature i to type j is computed as the ratio of the occurrence frequency of each discrete value v of discrete feature i on training samples in type j to the occurrence frequency of each discrete value v of discrete feature i on training samples in all the types, as denoted in Eq. (8).

$$member(i, j, v) = \frac{freq(i, j, v)}{\Sigma_u freq(i, u, v)}. \tag{8}$$

3.2 Feature evaluation and selection

Upon the four data description matrices for both continuous and discrete features, we are able to evaluate and select the top covering discriminating features.

3.2.1 Feature evaluation

The discriminating power of continuous features for each type pair $c_g - c_h$ is evaluated by the welch t-test. In the pairwise welch t-test, a t-statistics t_{ghi} is computed for each type pair $c_g - c_h$ on feature i, where $1 \leqslant g, h \leqslant k$ and $1 \leqslant i \leqslant nc$, $X_{gi} = \{x_c(s,i)\}_{s\in\Omega_g}$ denotes feature i in type g, $X_{hi} = \{x_c(s,i)\}_{s\in\Omega_h}$ denotes feature i in type h, m_g is the number of samples in type c_g and m_h is the number of samples in type c_h.

$$t_{ghi} = \frac{\overline{X_{gi}} - \overline{X_{hi}}}{\sqrt{\frac{stdev^2(X_{gi})}{m_g} + \frac{stdev^2(X_{hi})}{m_h}}}. \tag{9}$$

The degree of freedom of t_{ghi} is computed as:

$$df = \frac{(\frac{stdev^2(X_{gi})}{m_g} + \frac{stdev^2(X_{hi})}{m_h})^2}{\frac{stdev^4(X_{gi})}{m_g^2(m_g-1)} + \frac{stdev^4(X_{hi})}{m_h^2(m_h-1)}}. \tag{10}$$

The two-tailed p-values are further adjusted by multiplying the number of type pairs being compared. It can be inferred that the inter-type t-statistics t_{ghi} and its degree of freedom for type pair $c_g - c_h$ on feature i are solely determined by $\Sigma(i,g)$, $\Sigma^2(i,g)$, $\Sigma(i,h)$ and $\Sigma^2(i,h)$. If the associate p-value of the welch t-statistic is below 0.05, the feature is assumed to be discriminating for type pair $c_g - c_h$.

The welch t-test was adopted in our work to judge the discriminating power of continuous features. This is because in the field of emitter parameter analysis, the emitter parameter values are generally assumed to follow normal distributions. In the case of non-parametric analysis, such as wilcoxon signed-ranks test, our incremental learning strategies can be applied effectively, and other feature selection strategies could probably be applied too.

The discriminating power of a discrete radar emitter feature i for each type pair $c_g - c_h$ is evaluated, on the other hand, by the sum of overlapping membership probabilities *overlap*, as calculated in Eq. (11), where v is any possible value of discrete feature i.

$$overlap = \sum_v \min(member(i,g,v), member(i,h,v)). \tag{11}$$

Similar to the continuous radar emitter features, a threshold of 0.05 is applied and only those discrete features whose sum of overlapping membership probabilities *overlap* is below 0.05 would be considered discriminating for type pair $c_g - c_h$.

As we can see, the feature evaluation procedure traces back to the data description matrices as well. The pairwise welch t-test on continuous radar emitter features is solely determined by the feature-type sum matrix and the feature-type square matrix. The membership probability using discrete radar emitter features could be finely calculated from the feature-type frequency matrix.

3.2.2 Feature selection

Our feature selection strategy is specially designed to select the top covering discriminating feature for each emitter type pair. In this way, each type pair could be finely discriminated, and the "siren pitfall" problem could be finely avoided. Our feature selection procedure is composed of the following steps:

1) For each feature i, either continuous or discrete, calculate the set of type pairs (denoted as $CPSet_i$) that it is discriminating for, and the average p-values obtained on these type pairs (denoted as $AveP_i$).
2) Rank all the $n = nc + nd$ features first in the descending order of the size of discriminated type pair set and next in the ascending order of average p-value, denoted as $ord = f_1 \prec f_2 \prec \cdots \prec f_n$.
3) Initialize the top covering discriminating feature of each type pair $c_g - c_h$ as NA, $TopF_{c_g-c_h} = NA$.
4) Start from the first ranked feature a in ord onwards, repeat until no feature a could be located:
   - Find all redundant features given a whose discriminated type pair set is a subset of a, $\forall b \succ a$ and $CPSet_a \supseteq CPSet_b$ and prune them;
   - Assign a as the top covering discriminating feature for the type pair in its discriminated type pair set: $\forall c_g - c_h \in CPSet_a$ we set $TopF_{c_g-c_h} = a$ if $TopF_{c_g-c_h} = NA$;
   - Set a as the next ranked feature in ord in the remaining features.
5) Output the top covering discriminating features in $TopFS = \{TopF_{c_g-c_h}\}_{\forall c_g-c_h}$.

At Step 1, for any feature, continuous or discrete, as long as it passes the type pair evaluation test for type pair $c_g - c_h$, $c_g - c_h$ would be included in its discriminated type pair set, $c_g - c_h \in CPSet_i$. The ord order at Step 2 makes sure that the features are ranked by the discriminating power. We assume that the more type pairs the feature is able to discriminate, the more discriminating it is. When the number of type pairs discriminated by two or more features is the same, we assume the feature with lower average p-value is more discrim-

inating. The *ord* order helps flexibly locate the top covering discriminating feature for each type pair.

We define the *top covering discriminating feature* of each type pair $c_g - c_h$ as the top ranked feature in *ord* which is able to discriminate $c_g - c_h$. At Step 3, the top covering discriminating feature of each type pair $c_g - c_h$ is initialized as *NA*. We define a feature b as redundant given another feature a, if the type pairs feature b discriminates could all be discriminated by feature a and feature a ranks before feature b. At Step 4, the top covering discriminating features are identified and redundant features pruned iteratively.

At Step 5, the identified top covering discriminating features would be output for classifier construction.

3.3 Classifier construction

Upon the identified top covering discriminating features $TopFS$, a fuzzy classifier is constructed in line upon the data description matrices. Suppose among the original *nc* continuous and *nd* discrete radar emitter features, *sc* continuous and *sd* discrete features have been selected into $TopFS$, denoted as Ψ_c ($|\Psi_c| = sc$) and Ψ_d ($|\Psi_d| = sd$) respectively, $TopFS = \Psi_c \cup \Psi_d$.

A prototype classifier would be constructed based on Mahalanobis distance and the geometric mean of the membership probability, because it is generally assumed that the emitter parameter values comply with normal distributions approximately. Another reason to adopt such a prototype classifier is that the simplicity and high flexibility make it adaptable to the simultaneous expansion in all the three dimensions of sample, class and feature. The similarity of a radar emitter test sample to each radar emitter type is estimated by Mahalanobis distance and membership probability.

We denote the *sc* number of selected continuous features of the radar emitter test sample t as $t_c = (t_1, t_2, ..., t_{sc})$ and the mean vector of each existing radar emitter type j on the *sc* number of selected continuous features as

$$mean_j = [\Sigma(i, j)]_{i \in \Psi_c} / m_j. \tag{12}$$

Then, the Mahalanobis distance [33] between test sample t and radar emitter type j is calculated as shown in Eq. (13).

$$MD_{t,j} = ((t_c - mean_j)^T C^{-1} (t_c - mean_j))^{\frac{1}{2}}, \tag{13}$$

where C is the estimated radar emitter sample covariance matrix. For multivariate normally distributed data, the values are approximately chi-square distributed with *sc* degrees of freedom (χ^2_{sc}). Note that the radar emitter sample covariance coefficient between selected continuous feature p and q for type j is calculated according to Eq. (6). We denote the p-value of the Mahalanobis distance $MD_{t,j}$ as $pval_{t,j}$.

Meanwhile, for the *sd* number of selected discrete radar emitter features, we denote the corresponding value of radar emitter test sample t of each selected discrete feature i as $v(t, i)$.We calculate the geometric mean of the membership probability of the test sample t to type j, denoted as $gm_{t,j}$, as illustrated in Eq. (14).

$$gm_{t,j} = (\prod_{i \in \Psi_d} member(i, j, v(t, i)))^{\frac{1}{sd}}. \tag{14}$$

The radar emitter test sample is classified according to Eq. (15) as follows:

$$cls(t) = \begin{cases} \text{Null}, & \forall j\, pval_{t,j} \leqslant 0.05; \\ \text{Null}, & \forall j\, gm_{t,j} \leqslant 0.05; \\ argmax_j \dfrac{pval_{t,j} \cdot sc + gm_{t,j} \cdot sd}{sc + sd}, & \text{otherwise.} \end{cases} \tag{15}$$

When either the continuous features or the discrete features of a radar emitter test sample are significantly different from those of all the emitter classes, the test sample would be classified to none existing emitter types. In that case, a label of "Null" would be output instead.

Again, the classifier construction procedure traces back to the data description matrices. The Mahalanobis distance between the test sample and each radar emitter type is determined by the feature-type sum matrix $[\Sigma(i, j)]$ and the feature-type pair matrix $[\Sigma^{pair}(p, q, j)]$, while the geometric mean of the membership probability of the test sample to each type is calculated from the feature-type frequency matrix $[freq(i, j, v)]$.

3.4 Incremental learning

Suppose the original radar emitter training samples consist of m emitter samples and each sample has n features (*nc* continuous and *nd* discrete) from k types. After incorporating new radar emitter intelligence and information, the new radar emitter training data set is expanded to m' ($m' \geqslant m$) radar emitter training samples, each with n' features, nc' ($nc' \geqslant nc$) continuous and nd' ($nd' \geqslant nd$) discrete, from k' ($k' \geqslant k$) types, as shown in Table 1. The newly-arrived information is highlighted in light gray.

During the incremental update of the data description matrices, we keep a buffer of size L to record up to L/k latest training samples for each of the k existing emitter types instead of keeping the whole available data set. In addition, when an emitter feature is convinced undiscriminating, it would not be stored in the buffer. The set of latest training

samples for each emitter type j is denoted as Ω_j^l, $|\Omega_j^l| \leqslant L/k$. The newly-arrived training samples for type j are denoted as $\Delta\Omega_j$.

Table 1 Updated radar emitter training data set

Sample	f_1	...	f_n	f_{n+1}	...	$f_{n'}$	Type
1	$x(1,1)$	...	$x(1,n)$	$x(1,n+1)$	...	$x(1,n')$	c_1
2	$x(2,1)$	...	$x(2,n)$	$x(2,n+1)$	...	$x(2,n')$	...
...	...	...	...	...	...	...	...
m	$x(m,1)$	...	$x(m,n)$	$x(m,n+1)$	...	$x(m,n')$	c_k
$m+1$	$x(m+1,1)$	...	$x(m+1,n)$	$x(m+1,n+1)$	...	$x(m+1,n')$	...
...	...	...	...	...	...	...	...
m'	$x(m',1)$	...	$x(m',n)$	$x(m',n+1)$	...	$x(m',n')$	$c_{k'}$

3.4.1 Incremental update of data description matrices

According to the study in previous sections, it can be inferred that the selected features and the resulted classification model all derive from the data description matrices. It can be further inferred that as long as the data description matrices are incrementally updated in time, the radar emitter classification model can be updated incrementally in line.

The incremental update of radar emitter data description matrices proceeds in three dimensions iteratively as below:

• Update of feature dimension

In the first step, we check whether some new radar emitter features are available, i.e., $nc' > nc$ or $nd' > nd$. If so, the feature dimension of the feature-type sum matrix $[\Sigma(i,j)]_{nc\times k}$, the feature-type square matrix $[\Sigma^2(i,j)]_{nc\times k}$ and the feature-type pair matrix $[\Sigma^{pair}(p,q,j)]_{nc\times nc\times k}$ for the continuous radar emitter features and the feature-type frequency matrix $[freq(i,j,v)]_{nd\times k\times V}$ for the discrete radar emitter features would be expanded. During the update of feature dimension, the information of new features for the original radar emitter training samples would be extracted and incorporated into the new data description matrices.

For each newly-arrived continuous radar emitter feature p, one at a time, the corresponding new matrix element $\Sigma(p,j)$, $\Sigma^2(p,j)$ and $\Sigma^{pair}(p,q,j)$ are updated iteratively, where $nc < p \leqslant nc'$ and q $(1 \leqslant q < p)$ is one of the existing continuous radar emitter features, as indicated in Eqs. (16)–(18):

$$\Sigma(p,j) = \sum_{s\in\Omega_j^l} x_c(s,p), \tag{16}$$

$$\Sigma^2(p,j) = \sum_{s\in\Omega_j^l} x_c(s,p)^2, \tag{17}$$

$$\Sigma^{pair}(p,q,j) = \sum_{s\in\Omega_j^l} x_c(s,p)x_c(s,q) \tag{18}$$

Eqs. (16) and (17) indicate the expansion of the feature dimension for the feature-type sum matrix and feature-type square matrix. Eq. (18) illustrates that whenever a new continuous radar emitter feature p arrives, the product of the new feature p and one existing feature q in the buffer will be computed and incorporated into the feature-type pair matrix.

For each newly-arrived discrete radar emitter feature i $(nd+1 \leqslant i \leqslant nd')$, the elements of the feature-type frequency matrix $freq(i,j,v)$ for new feature i on each discrete value v is updated as below:

$$freq(i,j,v) = \mathrm{SUM}_{s\in\Omega_j^l}(x_d(s,i) == v), \tag{19}$$

where j is the radar emitter type that existing radar emitter sample s belongs to.

• Update of sample and type dimension

The next update is in sample and type dimension. There are two different cases of update in sample and type dimension. In one case, the new radar emitter samples all belong to the original k types. In the other case, the new radar emitter samples belong to the $k'-k$ new types.

In case one, the elements of feature-type sum matrix $\Sigma(i,j)$, feature-type square matrix $\Sigma^2(i,j)$ and feature-type pair matrix $\Sigma^{pair}(p,q,j)$ for features i, p, q and type j, where $1 \leqslant i,p,q \leqslant nc'$ and $1 \leqslant j \leqslant k$, are updated respectively as shown in Eqs. (20)–(22):

$$\Sigma(i,j) = \Sigma(i,j) + \sum_{s\in\Delta\Omega_j} x_c(s,i), \tag{20}$$

$$\Sigma^2(i,j) = \Sigma^2(i,j) + \sum_{s\in\Delta\Omega_j} x_c(s,i)^2, \tag{21}$$

$$\Sigma^{pair}(p,q,j) = \Sigma^{pair}(p,q,j) + \sum_{s\in\Delta\Omega_j} x_c(s,p)x_c(s,q). \tag{22}$$

Similarly, for each newly-arrived radar emitter sample belonging to each existing type j, the elements of the feature-type frequency matrix $freq(i,j,v)$ for the discrete value v of discrete feature i and associated type j are updated as below:

$$freq(i,j,v) = freq(i,j,v) + \mathrm{SUM}_{s\in\Delta\Omega_j}(x_d(s,i) == v), \tag{23}$$

where j is any radar emitter type that the radar emitter sample s belongs to, and $\Delta\Omega_j$ indicates the new samples of radar emitter type j.

In case two, the elements of feature-type sum matrix $\Sigma(i,j)$, feature-type square matrix $\Sigma^2(i,j)$ and feature-type pair matrix $\Sigma^{pair}(p,q,j)$ for each new type j, where $k+1 \leqslant j \leqslant k+\Delta k$, are updated respectively as illustrated in Eqs. (24)–(26):

$$\Sigma(i,j) = \sum_{s\in\Delta\Omega_j} x_c(s,i), \tag{24}$$

$$\Sigma^2(i,j) = \sum_{s\in\Delta\Omega_j} x_c(s,i)^2, \tag{25}$$

$$\Sigma^{pair}(p,q,j) = \sum_{s\in\Delta\Omega_j} x_c(s,p)x_c(s,q). \tag{26}$$

For the newly-arrived radar emitter samples of new types, the elements of the feature-type frequency matrix $freq(i,j,v)$ on the discrete value v of discrete feature i for new type j are updated as below:

$$freq(i,j,v) = \mathrm{SUM}_{s\in\Delta\Omega_j}(x_d(s,i) == v). \tag{27}$$

Similarly, the data description matrices would be updated when a training sample is obsolete and taken out of the buffer. After the incremental update of data description matrices, the latest information of radar emitter features, samples and types would be extracted efficiently into the data description matrices for the later classification model building.

3.4.2 Incremental update of classifier

As discussed above, the updated data description matrices include the feature-type sum matrix $[\Sigma(i,j)]$, the feature-type square matrix $[\Sigma^2(i,j)]$ and the feature-type pair matrix $[\Sigma^{pair}(p,q,j)]$ for continuous emitter features, and the feature-type frequency matrix $[freq(i,j,v)]$ for discrete emitter features. Based on these updated data description matrices, all available radar emitter features are re-evaluated, the top covering discriminating features are updated and our fuzzy classifier is incrementally updated all along with the updated top covering discriminating features. Specifically, the procedure of incremental update of the fuzzy classifier includes three major steps: description variable update, feature update and classifier update.

- Description variable update
  1) Update the mean value of each continuous feature i on each radar emitter type j, $\overline{X_{ji}}$, with the feature-type sum matrix $[\Sigma(i,j)]$, as indicated in Eq. (4);
  2) Update the radar emitter sample standard deviation $stdev(X_{ji})$ of each continuous feature i on each type j with the feature-type sum matrix $[\Sigma(i,j)]$ and the feature-type square matrix $[\Sigma^2(i,j)]$, as indicated in Eq. (5);
  3) Update the estimated radar emitter sample covariance coefficient $cov(p,q,j)$ between continuous feature p and q for each emitter type j with the feature-type pair matrix $[\Sigma^{pair}(p,q,j)]$ and the feature-type sum matrix $[\Sigma(i,j)]$, as indicated in Eq. (6);
  4) Update the membership probability $member(i,j,v)$ of value v for discrete feature i on emitter type j with the feature-type frequency matrix $[freq(i,j,v)]$, as indicated in Eq. (8).
- Feature update
  1) Update the t-statistics t_{ghi} for each type pair $c_g - c_h$ on feature i based on the updated mean value of each continuous feature i on each radar emitter type j, $\overline{X_{ji}}$, and radar emitter sample standard deviation $stdev(X_{ji})$ of each continuous feature i on each type j, as indicated in Eq. (9);
  2) Update the associated degree of freedom df of statistic t_{ghi} for each type pair $c_g - c_h$ on feature i with the updated radar emitter sample standard deviation $stdev(X_{ji})$, as indicated in Eq. (10);
  3) Update the resulted two-tailed p-values of the welch t-test and re-judge whether continuous feature i is discriminating for type-pair $c_g - c_h$;
  4) Update the sum of overlapping membership probabilities, $overlap$, upon the updated membership probability $member(i,j,v)$ of value v for discrete feature i on emitter type j, and re-judge whether discrete feature i is discriminating for type-pair $c_g - c_h$, as indicated in Eq. (11);
  5) Update the top covering discriminating features $TopFS$, either continuous or discrete, as illustrated in Section 2;
- Classifier update
  1) Update the structure and parameter values of mean vector $mean_j$ of each existing radar emitter type j in Mahalanobis distance calculation (Eq. (13)) upon the selected continuous features in the updated top covering discriminating features $TopFS$ with the updated $\overline{X_{ji}}$;
  2) Update the structure and parameter values of the estimated radar emitter sample covariance matrix C in Mahalanobis distance calculation (Eq. (13)) upon the selected continuous features in the updated top covering discriminating features $TopFS$ with the updated estimated radar emitter sample covariance coefficient $cov(p,q,j)$;
  3) Update the structure and parameter values of the geometric mean of the membership probability of the test sample t to type j, $gm_{t,j}$, with the updated

membership probability $member(i, j, v)$ of value v for selected discrete feature i to emitter type j, as indicated in Eq. (14).

The above incremental update order between the matrices, variables for the fuzzy classifier is illustrated in Fig. 1. The involved incremental update relationships could be summarized with a network of arrows, pointing from the variables or matrices first updated to the variables or matrices updated afterwards. There are accordingly four layers of variables and matrices corresponding to the incremental update of description matrices, description variables, features and the final fuzzy classifier respectively. The arrows in solid lines indicate incremental update of variables or matrices within the same or between adjacent layers, while the arrows in dash lines indicate incremental update of variables or matrices across unadjacent layers.

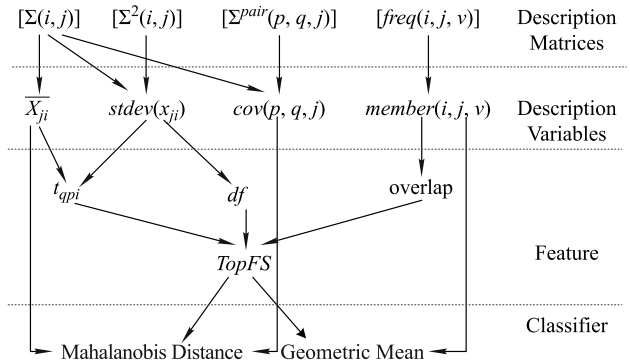

Fig. 1 Incremental update order of TILA

3.4.3 Time complexity analysis

Assume initially, there are m training samples, nc number of continuous features, nd number of discrete features and k different types in the original radar emitter training data, and later Δm number of new samples, Δnc number of new continuous features, Δnd number of new discrete features and Δk number of new types become available. In this way, the radar emitter training data is expanded to $m' = m + \Delta m$ samples, each with $nc' = nc + \Delta nc$ continuous features and $nd' = nd + \Delta nd$ discrete features, from $k' = k + \Delta k$ different types.

To cope with the expanded radar emitter training data set, we need to update the data description matrices in feature, sample and type dimensions at the first step. To update the feature-type sum matrix $\Sigma(i, j)$ and feature-type square matrix $\Sigma^2(i, j)$, it takes a time complexity of $O(m\Delta nc)$ for the update on the Δnc new continuous features of the existing m training samples and a time complexity of $O(\Delta m(nc + \Delta nc))$ for the update on all the $nc + \Delta nc$ continuous features of Δm new training samples. The update of feature-type pair matrix $\Sigma^{pair}(p, q, j)$ on the Δnc new continuous features on the updated $m + \Delta m$ training samples takes a time complexity of $O((nc + \Delta nc)^2(m + \Delta m) - nc^2 m)$. Similarly, the time complexity of incremental update of the feature-type frequency matrix $freq(i, j, v)$ is $O(m\Delta nd + \Delta m(nd + \Delta nd))$. So, the time complexity of data description matrix update is $O(m\Delta nc + \Delta m(nc + \Delta nc) + (nc + \Delta nc)^2(m + \Delta m) - nc^2 m + m\Delta nd + \Delta m(nd + \Delta nd)) = O(nc^2\Delta m + nc\Delta nc(m + \Delta m) + \Delta nc^2(m + \Delta m) + \Delta m(nd + \Delta nd) + m\Delta nd)$.

Upon the updated data description matrices, the discriminating power of $nc + \Delta nc$ continuous features and $nd + \Delta nd$ discrete features would be evaluated for each of the $(k + \Delta k)(k + \Delta k - 1)/2$ type pairs. The time complexity of feature evaluation is $O((nc + \Delta nc + nd + \Delta nd)(k + \Delta k)^2)$. The time complexity of top covering discriminating feature selection is within $O((nc + \Delta nc + nd + \Delta nd)^2)$. Therefore, the time complexity of incremental feature evaluation and selection is $O((nc + \Delta nc + nd + \Delta nd)(k + \Delta k)^2 + (nc + \Delta nc + nd + \Delta nd)^2)$.

The final step is the update of classification model upon the selected sc' continuous features and sd' discrete features. The time complexity of the update of Mahalanobis distance model on the sc' selected continuous features for each type is $O(sc'^2(k + \Delta k))$, and that of the update of the geometric mean model on the sd' selected discrete features for each type is $O(sd'(k + \Delta k))$. As $sc' \leqslant nc'$ and $sd' \leqslant nd'$, the time complexity of classification model update would be within $O(((nc + \Delta nc)^2 + nd + \Delta nd)(k + \Delta k))$.

Please note that when the numbers of continuous features, discrete features and types are fixed, i.e., $\Delta nc = \Delta nd = \Delta k = 0$, and the number of newly-arrived radar emitter sample is positive, $\Delta m > 0$, the total time complexity is $O(\Delta m(nc^2 + nd) + (nc + nd)k^2 + (nc + nd)^2 + nc^2 k)$, which is proportional to the number of new radar emitter samples Δm and irrelevant with the original training data set size m. Likewise, when the numbers of samples, discrete features and types are fixed, i.e., $\Delta m = \Delta nd = \Delta k = 0$, and the number of newly-arrived continuous features is positive, $\Delta nc > 0$, then the total time complexity would be approximately in the second order of the number of new continuous features Δnc.

4 Results

We evaluated TILA on two dynamically accumulated real-life radar emitter data sets, airborne radar emitter data set and ground radar emitter data set. The sizes of the two real-life

data sets double in every six months approximately.

The radar emitter samples of the two real-life data sets were collected from the observatory stations periodically. Some new emitter samples might come from a new emitter type different from existing ones. Meanwhile, all existing emitter samples and newly collected samples were being analyzed in the information centers and some new features may become available from time to time. In this way, there would be a simultaneous increment in all the three dimensions of sample, type and feature.

The airborne radar emitter data set had 13194 training samples, coming from six different radar emitter types, "I", "F", "M", "C", "G1" and "G2". Accordingly, an independent test data set of 2400 samples was provided. The ground radar emitter data set had 40 000 training samples and 8 000 independent test samples from six different emitter types, "U1", "U2", "U3", "FP", "B1" and "B2".

On each real-life radar emitter data set, there were eight continuous features and three discrete features. The eight continuous radar emitter features were the minimum carrier frequency "minCF", the maximum carrier frequency "maxCF", the minimum repetitive frequency "minRF", the maximum repetitive frequency "maxRF", the minimum pulse width "minPW", the maximum pulse width "maxPW", the minimum pulse interval "minPI" and the maximum pulse interval "maxPI". The three discrete radar emitter features were "carrier frequency type", "repetitive frequency type" and "intra-pulse modulation type". These eleven features were contributed to the incremental update of data description matrices during expansions of feature, sample and type dimensions as discussed in Section 3.4.1.

The parameter L for buffer size was set as $20k$. The experiments were conducted on a Dell PC running Microsoft Windows XP with a Pentium dual-core CPU of 2.6GHz and a 4G RAM.

We designed three experimental series on purpose for evaluation of both three-way feature selection and three-way incremental learning ability, as indicated in Table 2. Experimental series 1 was designed for sample and type dimension expansion, during which the number of features was fixed at eleven while the number of training samples and emitter types increased incrementally. During experimental series 2, all the three dimensions of feature, sample and type expanded randomly. Experimental series 3 was specifically designed for feature dimension expansion during which the number of training samples was fixed while the number of features increased by about three at each time.

4.1 Evaluation of three-way incremental feature selection

We evaluated the incremental feature selection ability of TILA by the three experimental series in Table 2. We compared the discovered top covering emitter features against the ground-truth discriminating ones given by the information centers. These globally discriminating features could guide the information centers to extract more discriminating emitter features in the future.

We reported the hamming distance between the set of selected features and the ground-truth ones at each time point for each experimental series, as presented in Fig. 2. For each feature, a boolean value of TRUE would be assigned when the feature was selected and a value of FALSE would be given otherwise. In this way, a vector of length eleven was

Table 2 Three experimental series

		Airborne			Ground		
Series	Time	#Sample	#Feature	Expansion	#Sample	#Feature	Expansion
1	0	500	11	Initial	2 000	11	Initial
	1	4 000	11	Sample & type	10 000	11	Sample & type
	2	7 500	11	Sample & type	20 000	11	Sample & type
	3	11 000	11	Sample	30 000	11	Sample
	4	13 194	11	Sample	40 000	11	Sample
2	0	0	0	Initial	0	0	Initial
	1	3 500	6	Feature & sample & type	10 000	6	Feature & sample & type
	2	7 500	9	Feature & sample & type	20 000	9	Feature & sample & type
	3	11 000	11	Feature & sample	30 000	11	Feature & sample
	4	13 194	11	Sample	40 000	11	Sample
3	0	13 194	1	Initial	40 000	1	Initial
	1	13 194	3	Feature	40 000	3	Feature
	2	13 194	6	Feature	40 000	6	Feature
	3	13 194	9	Feature	40 000	9	Feature
	4	13 194	11	Feature	40 000	11	Feature

generated for each set of selected features and the ground-truth discriminating features. As can be seen, the hamming distances between the top covering features discovered by TILA and the ground-truth discriminating features within the three experimental series all converged to zero. This indicates that the top covering features were consistent with the ground-truth ones. We shuffled the input data orders, repeated the experiments dozens of times, and the conclusion still holds.

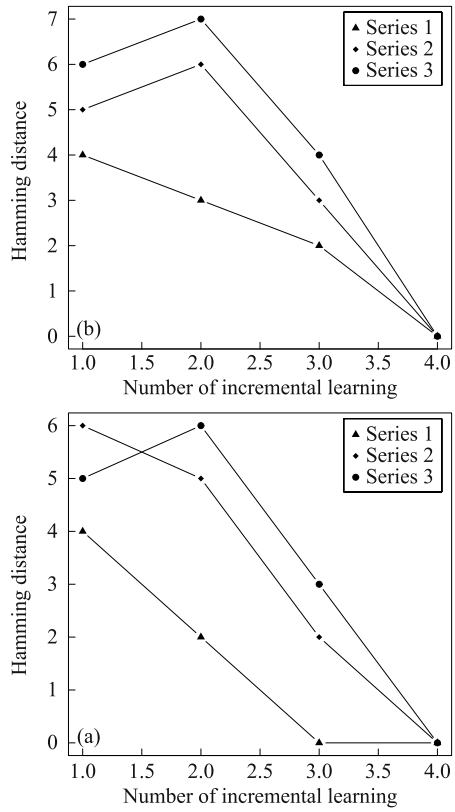

Fig. 2 Evaluation of three-way incremental feature selection. (a) Airborne emitter data set; (b) ground emitter data set

Most benchmark incremental learning algorithms [1,4,6–8] deal with a static feature space. Though some data stream feature evolution algorithms [13,14,16] are able to conduct feature selection on a dynamic feature space, their feature selection strategies are only applicable for discrete text features and thus are inappropriate for our continuous emitter features. Other benchmark data stream feature evolution algorithms [3,18] build an ensemble of models on different feature space but are incapable of evaluating the global discriminating power of emitter features. Therefore, none of the existing incremental learning algorithms is able to discover the ground-truth discriminating emitter features.

4.2 Evaluation of three-way incremental learning

We evaluated the incremental learning ability of TILA on the three experimental series as well. TILA was first trained incrementally with the newly arriving training samples and then evaluated on the independent test samples.

It is well-known that "Siren pitfall" is a common problem in multi-class classification [23], i.e., the majority classes are predicted well while the minority classes are predicted poorly. To better address the above "Siren pitfall" problem, average true positive rate (*AveTPR*), instead of classification accuracy, was chosen for evaluation of the incremental learning ability. As another benefit, *AveTPR* is adaptable for evaluation of both multi-class classification and binary classification problems. Comparatively, other evaluation metrics such as ROC and F-measure are restricted to evaluation of binary classification only. The calculation of *AveTPR* is given below:

$$AveTPR = \text{Average}\left(\frac{\text{SUM}_{s\in\Omega_c}(predict(s)=c)}{|\Omega_c|}\right)_{\forall c}. \quad (28)$$

When the predicted class of a test sample was Null and the true class label of the test sample was c, the true positive rate for class c would decrease while the true positive rates of other classes would not be affected. Therefore, the *AveTPR* would decrease.

4.2.1 Comparison against benchmark batch algorithms

We compared the *AveTPR* of TILA against those of five traditional batch machine learning algorithms, gray relational analysis algorithm (GRA), neural network (NN), support vector machine with polynomial kernel (SVM-poly), support vector machine with Gaussian kernel (SVM-Gaussian) and Random Forest on the two real-life radar emitter data sets during sample dimension expansion. We varied the number of training samples as indicated in experimental series 1 in Table 2. During each expansion, 3 500 and 10 000 new training samples were obtained respectively for the airborne and ground emitter data.

We only reported the *AveTPR*s when the corresponding algorithm was able to finish training within ten minutes and did not run out of memory. We did not report the *AveTPR*s of GRA, NN, SVM-poly and SVM-Gaussian for all the training sizes due to the training time constraint, and also did not report that of Random Forest due to the memory problem. As can be seen from Fig. 3, TILA has outperformed these batch algorithms in term of either classification true positive rate or computational cost.

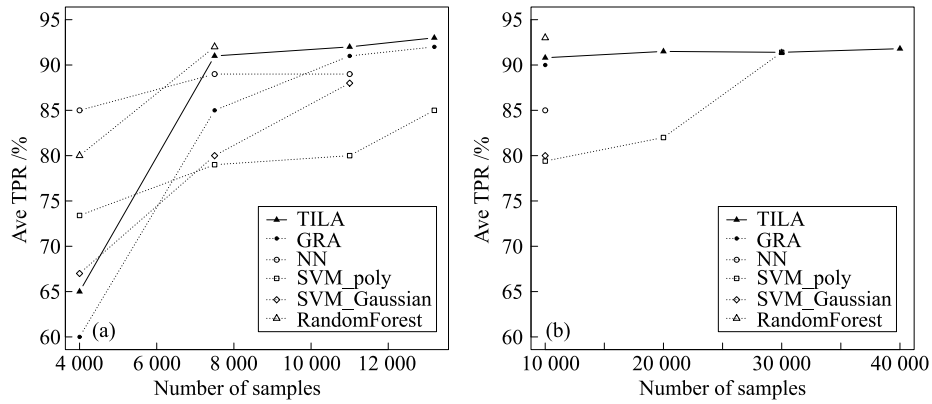

Fig. 3 Comparison of TILA against benchmark batch algorithms. (a) Airborne emitter data; (b) ground emitter data

4.2.2 Comparison against benchmark binary incremental learning algorithms

In addition, we compared the incremental learning ability of TILA against four benchmark binary incremental learning algorithms implemented in R package "RSofia"[1)], including margin-perceptron [5], SGD-SVM [6], pegasos SVM [7] and ROMMA [8] on the two real-life data sets. The four benchmark binary incremental learning algorithms were evaluated under the default parameter settings: the numeric scalar lambda was set as 0.1, the number of iterations was set as 100 000, the type of sampling loop to use for training was set as "stochastic", the probability to take a rank step (as opposed to a standard stochastic gradient step) in a combined ranking or combined ROC loop was set as 0.5 and the size of buffer to use in reading/writing to files was set as 40MB. We specified the type of sampling loop for training as "combined-ranking" (combined regression and ranking) [30].

Please note that these four incremental learning algorithms are all restricted to binary class partition as discussed in Section 2, and unable to cope with a dynamic feature space. As a result, they could not apply for multi-type radar emitter identification directly. Furthermore, the heuristic nature of the four incremental learning algorithms may take a risk by sacrificing the prediction accuracy due to the bias to the ground-truth distribution. For example, the SGD-SVM algorithm directly minimizes the empirical risk with the randomly sampled examples in each iteration, rather than with the expected risk using the whole set of examples.

For fair comparison, we reported the *AveTPR*s of the four incremental learning algorithms in binary partitions of one versus other types while reporting the *AveTPR* of TILA on the partition on all the six emitter types, as indicated in Figs. 4 and 5 respectively. We shuffled the data input orders ten times for each partition and calculated the *AveTPR* of each algorithm two times for each data input order. The *AveTPR* of a binary partition was computed as the average proportion of emitter samples correctly ascribed to the associate partition in the two partitions, while that of "All Six" was calculated as the average proportion of emitter samples correctly ascribed to the associate partition in all the six partitions.

As can be seen from Figs. 4 and 5, it turned out that the output of TILA was rather stable. On the contrary, the performance of margin-perceptron, SGD-SVM, pegasos SVM and ROMMA was significantly unstable and the *AveTPR*s varied abruptly even given the same input data set. Specifically, the *AveTPR*s of TILA were fixed around 92.4% and 91.8% respectively on the airborne and ground emitter data while those of margin-perceptron, SGD-SVM, pegasos SVM and ROMMA varied around a median value of only 50% – 60%. The whiskers in boxplots of the four binary algorithms were long, while no whiskers were found in boxplots of TILA. Obviously, TILA outperformed margin-perceptron, SGD-SVM, pegasos SVM and ROMMA in terms of both classification accuracy and insensitivity to data input orders.

4.2.3 Comparison against benchmark data stream algorithms

We compared the *AveTPR*s of TILA against those of the benchmark data stream algorithms during the three experimental series. The data stream methods in Refs. [15,16] were inappropriate for our emitter data sets, as their feature selection strategy was designed for discrete features only. The "one class" data stream method [17] was inappropriate either, as it is unable to deal with multi-class classification. As a result, we only compared TILA against the latest data stream approach introduced in Refs. [3,18], denoted as *DStream*.

[1)] http://code.google.com/p/sofia-ml/

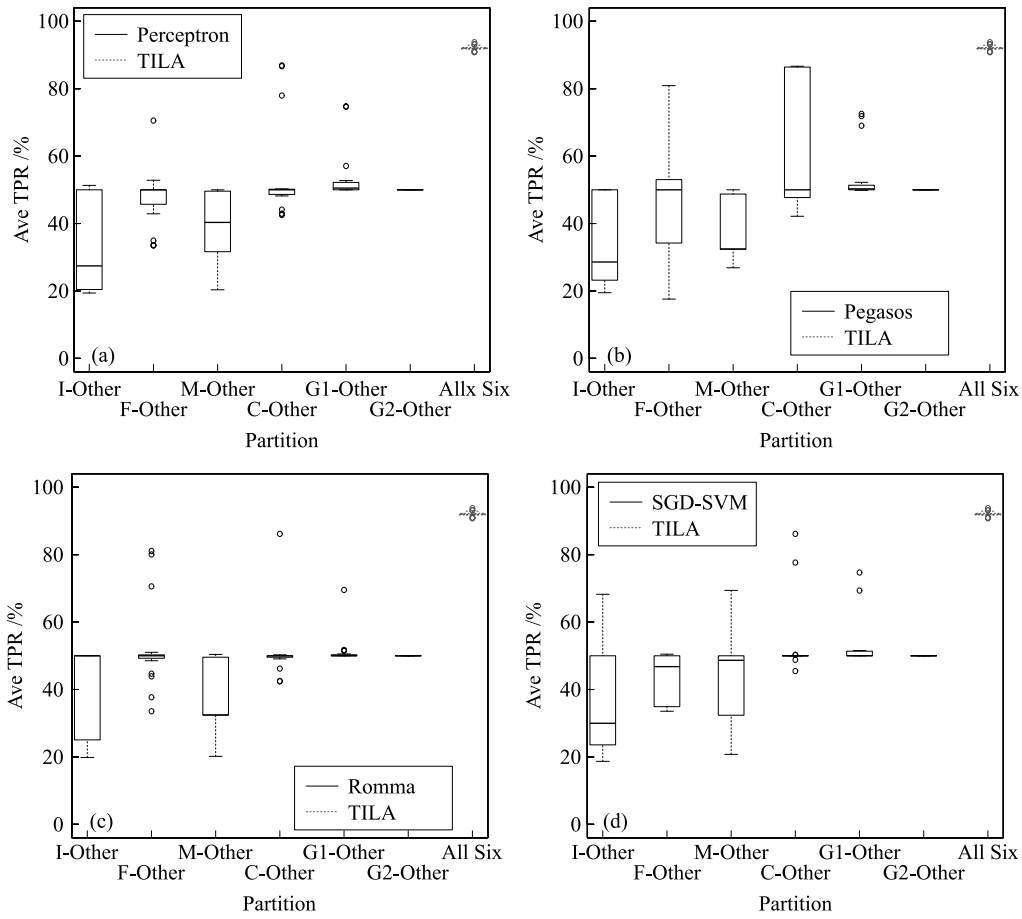

Fig. 4 Comparison of TILA against benchmark binary incremental algorithms on airborne emitter data. (a) Margin-perceptron; (b) Pegasos; (c) ROMMA; (d) SGD-SVM

In *DStream*, the latest L data samples in the buffer were divided into four equal-sized data chunks. With each data chunk, a model would be built using a semi-supervised K-means clustering. In this way, there were exactly four models in the ensemble at any given point of time.

Figure 6 indicates the *AveTPR* averaged on all the radar emitter types at each point of the three experimental series. Although the expansion paths of the three series were quite different, the *AveTPR*s of the three series all converged around 92.4% and 91.8% respectively on the airborne and ground emitter data. Comparatively, the *AveTPR*s of *DStream* on the ground emitter data were getting lower. This is because the training samples from minority emitter types were discarded with the update of data chunks, and the cohesion among the test samples in minority classes was also hard to be found. As a result, *DStream* obtained a relatively low true positive rate for the minority emitter types.

As can be observed, TILA has adapted well to the expansion of either samples, types or features.

4.3 Evaluation of computational efficiency

We compared the runtime of TILA against that of the five benchmark batch algorithms (GRA, NN, SVM-poly, SVM-Gaussian and Random Forest), the benchmark binary incremental learning algorithms and the data stream algorithm *DStream* on the two real-life radar emitter data sets during sample dimension expansion. As the runtimes of the four benchmark binary incremental learning algorithms were quite similar to each other, we only reported the training time of margin-perceptron for space limitation.

During experiments, we incrementally varied the number of training samples from 4 000 to 13 000 for the airborne emitter data and from 10 000 to 40 000 for the ground emitter data. During each sample increment, approximately 3 500 and 10 000 new emitter training samples were obtained respectively. We reported the corresponding training time if the algorithm was able to finish within ten minutes and had no memory problems.

As can be seen from Fig. 7, the incremental learning

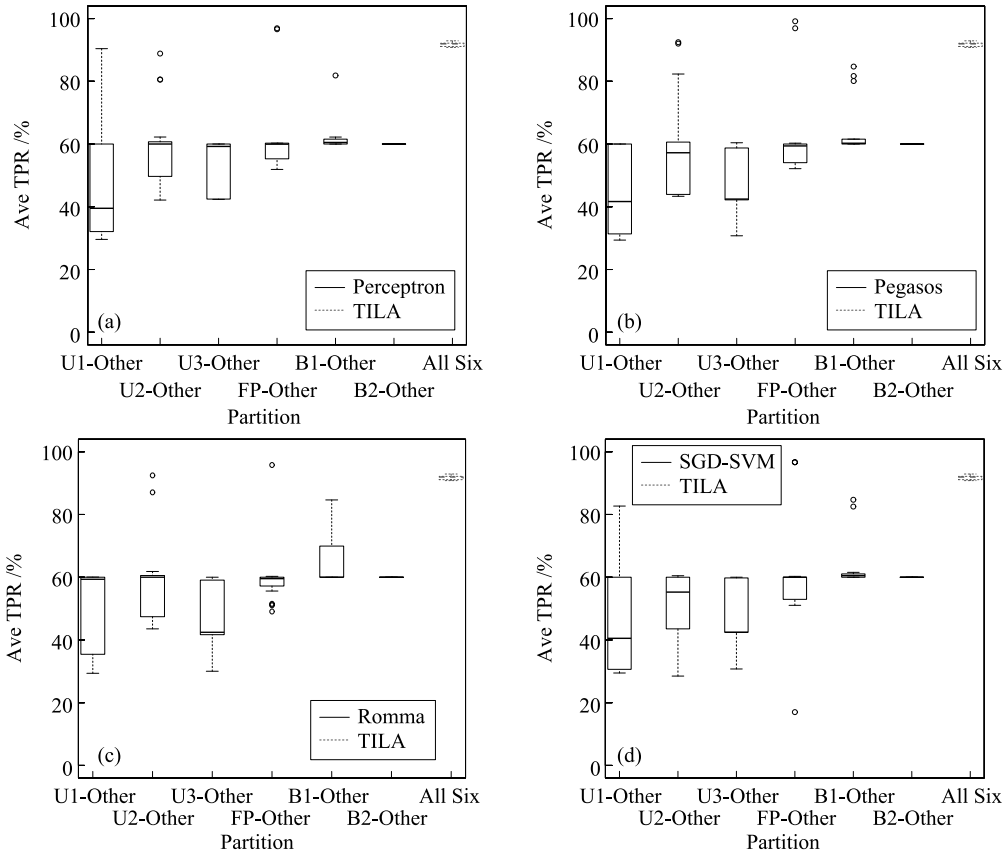

Fig. 5 Comparison of TILA against benchmark binary incremental algorithms on ground emitter data. (a) Margin-perceptron; (b) Pegasos; (c) ROMMA; (d) SGD-SVM

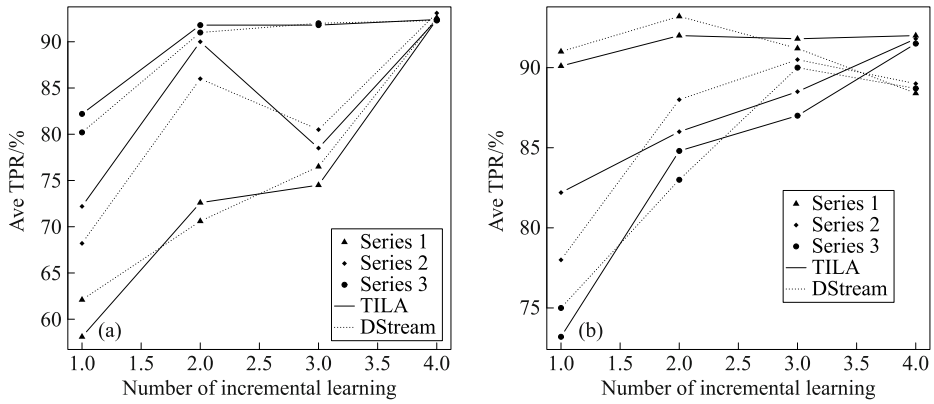

Fig. 6 Comparison of TILA against benchmark data stream algorithms. (a) Airborne emitter data; (b) ground emitter data

time of TILA, margin-perceptron and *DStream* was approximately fixed during each sample increment. TILA has outperformed *DStream* and was also competitive with the benchmark binary incremental learning peers. This is because when no new features are available ($\Delta nc = \Delta nd = 0$), the time complexity of TILA was proportional to the number of new training samples Δm, as discussed in Section 3. On the contrary, the runtime of the batch GRA, NN, SVM-poly, SVM-Gaussian and Random Forest increased substantially with the increase of training samples.

5 Conclusion

In this paper, we have proposed a three-way incremental learning algorithm for radar emitter type identification (TILA). Rather than restricted to the expansion in sample and type dimensions as most incremental learning algorithms do, TILA is able to adapt to all expansions in features, samples and types simultaneously. Unlike previous heuristic classifiers such as neural network and support vector machine,

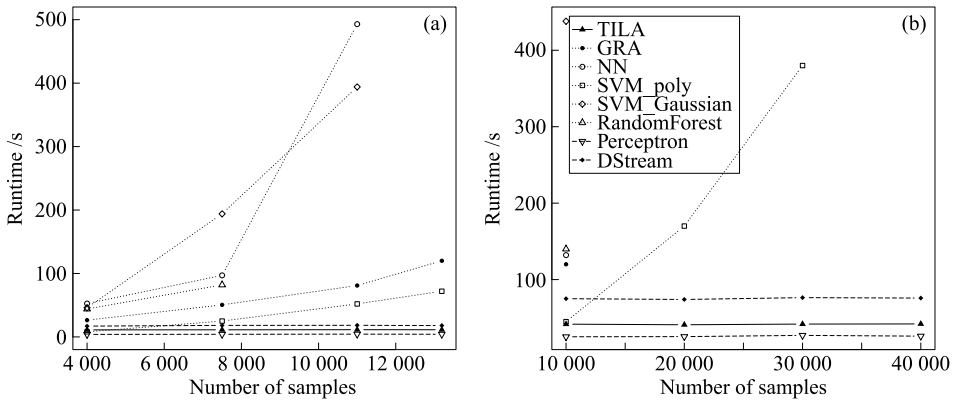

Fig. 7 Comparison of runtime with varying number of samples. (a) Airborne emitter data; (b) ground emitter data

TILA is insensitive to the data input order. Compared with the batch peers, TILA is superior in terms of computational efficiency, whose training time is approximately linear w.r.t. the number of newly-arrived training samples. Also, TILA outperforms the data stream peers in discovering a set of globally discriminating features and coping with unbalanced class distribution.

Acknowledgements This work was supported by the National Natural Science Foundation of China (Grant Nos. 61402426, 61373129) and the Collaborative Innovation Center of Novel Software Technology and Industrialization.

References

1. Zhou Z H, Chen Z Q. Hybrid decision tree. Knowledge-Based Systems, 2002, 15(8): 515–528
2. Domingos P, Hulten G. A general framework for mining massive data streams. Journal of Computational and Graphical Statistics, 2003, 12(14): 945–949
3. Masud M M, Chen Q, Khan L, Aggarwal C C, Gao J, Han J, Srivastava A, Oza N C. Classification and adaptive novel class detection of feature-evolving data streams. IEEE Transaction on Knowledge and Data Engineering, 2013, 25(7): 1484–1497
4. Bordes A, Bottou L. The huller: a simple and efficient online SVM. Lecture Notes in Computer Science, 2005, 3720: 505–512
5. Barak O, Rigotti M. A simple derivation of a bound on the rerceptron margin using singular value decomposition. Neural Computation, 2011, 23(8): 1935–1943
6. Zhang T. Solving large scale linear prediction problems using stochastic gradient descent algorithms. In: Proceedings of the 21st International Conferenc on Machine Learning. 2004, 919–926
7. Shalev-Shwartz S, Singer Y, Srebro N, Cotter A. Pegasos: primal estimated sub-gradient solver for SVM. Mathematical Programming, 2011, 127(1): 3–30
8. Li Y, Long P M. The relaxed online maximum margin algorithm. Machine Learning, 2002, 46: 1–3
9. Hulten G, Spencer L, Domingos P. Mining time-changing data streams. In: Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2001, 97–106
10. Carpenter G A, Grossberg S, Markuzon N, Reynolds J H, Rosen D B. Fuzzy ARTMAP: a neural network architecture for incremental supervised learning of analog multidimensional maps. IEEE Transactions on Neural Networks, 1991, 3(5): 698–713
11. Polikar R, Udpa L, Udpa S S, Honavar V. Learn++: an incremental learning algorithm for supervised neural networks. IEEE Transactions on Systems, Man, and Cybernetics, 2001, 31(4): 497–508
12. Sheng W, Banta L E. Parameter incremental learning algorithm for neural networks. IEEE Transactions on Neural Networks, 2006, 17(6): 1424–1438
13. Katakis I, Tsoumakas G, Vlahavas I. Dynamic feature space and incremental feature selection for the classification of textual data streams. Knowledge Discovery from Data Streams, 2006: 107–116
14. Wenerstrom B, Giraud-Carrier C. Temporal data mining in dynamic feature spaces. In: Proceedings of the 13th IEEE International Conference on Data Mining. 2006, 1141–1145
15. Masud M M, Gao J, Khan L, Han J, Thuraisingham B. Integrating novel class detection with classification for concept-drifting data streams. Lecture Notes in Computer Science, 2009, 5782: 79–94
16. Masud M M, Chen Q, Jing G, Latifur K, Jiawei H, Bhavani T. Classification and novel class detection of data streams in a dynamic feature space. Machine Learning and Knowledge Discovery in Databases, 2010, 6322: 337–352
17. Spinosa E J, de Leon F, de Carvalho A P, Gama J. Cluster-based novel concept detection in data streams applied to intrusion detection in computer networks. In: Proceedings of the 2008 ACM Symposium on Applied Computing. 2008: 976–980
18. Sminu N R, Jemimah S. Feature based data stream classification (FBDC) and novel class detection. International Journal of Engineering Research and Applications (IJERA), 2014, 25(7): 28–32
19. Zhu B, Jin W D. Radar emitter signal recognition based on EMD and neural network. Journal of Computers, 2012, 7(6): 1413–1420
20. Yang Z, Wu Z, Yin Z, Quan T, Sun H. Hybrid radar emitter recognition based on rough k-means classifier and relevance vector machine. Sensors, 2013, 13(1): 848–864
21. Liu H J, Liu Z, Jiang W L, Zhou Y Y. Incremental learning approach based on vector neural network for emitter identification. IET Signal Processing, 2010, 4(1): 45–54
22. Kauppia J P, Martikainenb K, Ruotsalainena U. Hierarchical classifica-

tion of dynamically varying radar pulse repetition interval modulation patterns. Neural Networks, 2010, 23(10): 1226–1237

23. Xu X, Wang W. An incremental gray relational analysis algorithm for multiclass classification and outlier detection. International Journal of Pattern Recognition and Artificial Intelligence, 2012, 26(6): 1250011
24. Montazer G A, Khoshniat H, Fathi V. Improvement of RBF neural networks using Fuzzy-OSD algorithm in an online radar pulse classification system. Applied Soft Computing, 2013, 13(9): 3831–3838
25. Tang K, Lin M, Minku F L, Yao X. Selective negative correlation learning approach to incremental learning. Neurocomputing, 2009, 72(13–15): 2796–2805
26. Sudo A, Sato A, Hasegawa O. Associative memory for online learning in noisy environments using self-organizing incremental neural network. IEEE Transactions on Neural Networks, 2009, 20(6): 964–972
27. Chao S, Wong F. An incremental decision tree learning methodology regarding attributes in medical data mining. In: Proceedings of IEEE International Conference on Machine Learning and Cybernetics. 2009, 3: 1694–1699
28. Bottou L. Large-scale machine learning with stochastic gradient descent. In: Procceedings of OMPSTAT. 2010, 177–186
29. Bottou L, Bousquet O. The tradeoffs of large scale learning. Advances in Neural Information Processing Systems, 2008, 20: 161–168
30. Sculley D. Combined regression and ranking. In: Proceedings of the 16th ACM SIGKDD Conference. 2010, 979–988
31. Wu X, Yu K, Ding W, Wang H, Zhu X. Online feature selection with streaming features. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 2013, 35(5): 1178–1192
32. Ditzler G, Rosen G, Polikar R. Incremental learning of new classes from unbalanced data. In: Proceedings of International Joint Conference on Neural Networks (IJCNN). 2013, 33–42
33. Filzmoser P. A multivariate outlier detection method. In: Proceedings of the 7th International Conference on Computer Data Analysis and Modeling. 2004, 18–22

Xin Xu received her PhD in School of Computing from National University of Singapore, Singapore in 2006. She is currently a Senior Research Engineer in Science and Technology on Information System Engineering Laboratory, China. Her research interests are in the area of data mining and data fusion.

Wei Wang received his PhD in electrical and computer engineering from National University of Singapore, Singapore in 2008. He is currently an associated professor in Department of Computer Science and Technology, Nanjing University, China. His research interests are in the area of wireless sensor networks and data fusion.

Jianhong Wang received his PhD in College of Automation Engineer from Nanjing University of Aeronautics and Astronautics, China. He is currently an associate professor in Jingdezhen Ceramic Institute, China. His research interests include real-time distributed control, optimization and system identification.