Handling Label Noise in Air Traffic Complexity Evaluation Based on Confident Learning and XGBoost

ZHANG Minghua[1], *XIE Hua*[1*], *ZHANG Dongfang*[2], *GE Jiaming*[2], *CHEN Haiyan*[2]

1. College of Civil Aviation, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, P.R. China;
2. College of Computer Science and Technology/College of Artificial Intelligence, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, P.R. China

(Received 8 June 2020; revised 1 July 2020; accepted 22 July 2020)

Abstract: Air traffic complexity is a critical indicator for air traffic operation, and plays an important role in air traffic management (ATM), such as airspace reconfiguration, air traffic flow management and allocation of air traffic controllers (ATCos). Recently, many machine learning techniques have been used to evaluate air traffic complexity by constructing a mapping from complexity related factors to air traffic complexity labels. However, the low quality of complexity labels, which is named as label noise, has often been neglected and caused unsatisfactory performance in air traffic complexity evaluation. This paper aims at label noise in air traffic complexity samples, and proposes a confident learning and XGBoost-based approach to evaluate air traffic complexity under label noise. The confident learning process is applied to filter out noisy samples with various label probability distributions, and XGBoost is used to train a robust and high-performance air traffic complexity evaluation model on the different label noise filtered ratio datasets. Experiments are carried out on a real dataset from the Guangzhou airspace sector in China, and the results prove that the appropriate label noise removal strategy and XGBoost algorithm can effectively mitigate the label noise problem and achieve better performance in air traffic complexity evaluation.

Key words: air traffic complexity evaluation; label noise; confident learning; XGBoost

CLC number: V355 **Document code:** A **Article ID:** 1005-1120(2020)06-0936-11

0 Introduction

With the air transport industry developing rapidly, the surging flight volume and limited airspace impose new challenges on the current air traffic management system and air traffic controllers (ATCos). Many potential safety problems have been raised, such as airspace congestion, flight conflict, and high workload of ATCos. In order to safely regulate air traffic, airspace is divided into several smaller sectors which are in charge of ATCos. However, the ATCos resource is limited, so we need to allocate ATCos resources over different sectors reasonably through advanced techniques, such as re-sectorization or dynamic airspace configuration. The key to these techniques is to accurately evaluate air traffic complexity.

Air traffic complexity is a quantitative indicator to reflect the complexity of air traffic system operation pattern, the relationship between aircraft and uncertainty of evolutionary trend[1-3]. Evaluation of air traffic complexity is not easy because of the numerous complexity related factors and non-linear correlation contained in the formation of air traffic complexity[4].

There are two main methods in the research of air traffic complexity evaluation[5]. The first one focuses on constructing a model or the most relevant indicator, such as conflict probability[6], conflict resolution difficulty[7], Lyapunov Exponent[8], etc. However, as air traffic complexity contains large

*Corresponding author, E-mail address: xiehua@nuaa.edu.cn.

How to cite this article: ZHANG Minghua, XIE Hua, ZHANG Dongfang, et al. Handling label noise in air traffic complexity evaluation based on confident learning and XGBoost[J]. Transactions of Nanjing University of Aeronautics and Astronautics, 2020, 37(6): 936-946.

http://dx.doi.org/10.16356/j.1005-1120.2020.06.011

amounts of information and is embedded with sophisticated relationships, it is unrealistic to perfectly evaluate air traffic complexity by a single indicator or model. The principle of the other method is to consider as more complexity factors as possible to make a comprehensive description for air traffic complexity. The most famous one is the dynamic density method, which calculates complexity as the sum of various complexity factors with different weight[9]. Whereas, due to the inability in depicting non-linear relationship, the dynamic density method tends to get imprecise results in practice. Other non-linear better methods were then put into use. In 2006, Gianazza et al.[10] introduced the idea that the air traffic complexity problem could be considered as a complexity level classification task. They used the backpropagation neural network (BPNN) to capture the non-linear relationship[10]. Later on, more and more advanced methods, such as adaptive boosting learning algorithm[11] and transfer-learning[12], had been employed and acquired fruitful achievements in air traffic complexity evaluation.

All these existing machine learning-based complexity evaluation methods have one same premise assumption that the complexity labels evaluated by air traffic management (ATM) experts are definitely correct. But, in fact, some samples used by machine learning algorithms may have inaccurate labels, especially when the labels are provided by human[13-14]. Air traffic complexity labels marked by ATM experts also have some inaccurate labels. In 2019, Andrasi et al.[15] carried out a comparative experiment on air traffic complexity evaluation between neural network and linear model. Theoretically, neural network may get a better result for its more excellent ability in depicting non-linear relationship. However, the results showed that they only had small difference. The author of Ref.[15] attributed the remaining error as intra-rater or inter-rater unreliability in human experts during labeling, which illustrated that the premise assumption of definitely correct labels may not be appropriate. Hence, we should pay more attention to the incorrect labels, named as label noise, and its impact on air traffic complexity evaluation.

In this paper, we propose a confident learning and XGBoost-based method to evaluate air traffic complexity under label noise, which has not been dealt with in the past. In our method, every sample is calculated in a cross-validation way to get the probability distributions of several classes through different classification algorithms. Under different label probability distributions, label noise detection and cleansing steps based on confident learning are carried out separately to produce several suspected label noise sets. Then these sets are integrated into one total set. Based on the total label noise set, we selectively remove different ratios of label noise samples from the original dataset to get datasets with different cleanliness. Finally, by comparing the performance of XGBoost and other classification algorithms on these cleansed datasets, the optimal label noise removal ratio and the corresponding classification algorithm can be obtained for final air traffic complexity evaluation.

1 Problem Description

This section gives a description of evaluating air traffic complexity by machine learning methods and the problem of label noise we encountered.

Our objective is to evaluate air traffic complexity from a variety of complexity related factors. More specifically, we have its real traffic operational data for every air traffic scenario, such as aircraft speed, heading, longitude, latitude, altitude and so on. According to previous research studies, these data are transformed into complexity related factors to describe air traffic complexity. Air traffic complexity level provided by ATM experts is collected when the real traffic operational data are generated. In the machine learning field, these complexity related factors and air traffic complexity level are known as features and labels, respectively. Based on complexity related features and label information, many scholars have carried out plenty of air traffic complexity researches under machine learning framework[3, 5, 11, 12, 15-17]. The main idea is to construct a

mapping model between these features and the complexity labels. Then, we can use the model to intelligently predict air traffic complexity without ATM experts, when new air traffic data are coming. The whole process is displayed in Fig.1.

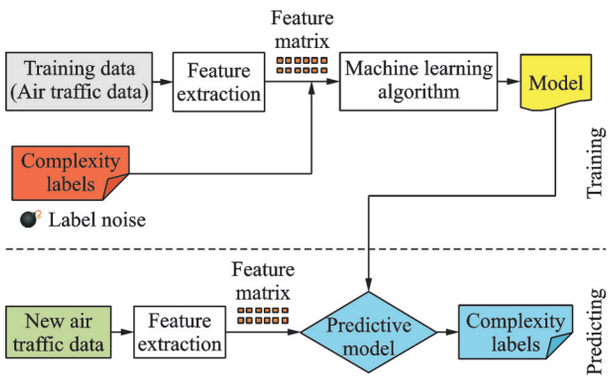

Fig.1 Machine learning framework of air traffic complexity

In the area of supervised machine learning, many benchmark datasets have label noise, which also happens on air traffic complexity datasets. The reason is summarized as human expert inconsistency, which consists of low intra-rater and inter-rater reliability, by Andrasi in 2019[15]. The raters are referred to ATM experts who mark the complexity in the given traffic situations. Intra-rater reliability is the degree of agreement among multiple ratings by a single rater, while intra-rater reliability is the degree of consistency between multiple raters. For instance, even the same traffic situations may be rated with different complexity labels in different circumstances, which induces the problem of label noise.

Label noise may obscure the relationship between the features of a sample and its labels, so as to impact the classification performance of classifiers. Some researchers were aware of the problem, so they managed to get the high quality complexity labels by integrating the ideas of different experts on the same air traffic scenario or conducted more completed and strict process management to alleviate the problem[17]. However, these solutions cannot completely solve the problem of label noise and may even waste limited labeling resources. Therefore, this paper puts forward a label noise sample detection and removal strategy to handle the label noise problem in the evaluation of air traffic complexity.

2 Methodology

2.1 Air traffic complexity representation

Various factors can influence the level of air traffic complexity and have drawn much attention in air traffic complexity research. Kopardekar et al.[9,18-19] has identified nearly 40 air traffic complexity factors since 1963. Delahaye et al.[20] described the intrinsic attributes of air traffic by relative position and relative speed of aircraft pairs, and then constructed a traffic disorder model to analyze the complexity. Lee et al.[21] emphasized heading change of aircraft in response to intrusive aircraft within a sector to calculate air traffic complexity. The probabilistic factor was put forward to measure the midterm traffic complexity by Prandini et al.[6]. In this paper, we choose 24 complexity factors that have been consistently found to be relevant to air traffic complexity. All of the factors are the features we use in the later machine learning process. Their definitions are listed in Table 1 and more detailed information can refer to Refs.[20, 22-23].

Table 1 Air traffic complexity factors set

Factor	Definition
N	Total number of aircraft
$N_{\mathrm{des}}, N_{\mathrm{cl}}$	Number of descending / climbing aircraft
F_5, F_{15}, F_{30}	Future incoming flow in 5, 15, 30 min
Dens	Density of aircraft
hpro_1, hpro_2	Horizontal proximity between aircraft
vpro_1, vpro_2	Vertical proximity between aircraft
σ_{gs}^2	Variance of aircraft ground speeds
$\sigma_{\mathrm{gs}}/\bar{v}_{\mathrm{gs}}$	Ratio of σ_{gs} to average aircraft ground speed $\bar{v}_{\mathrm{gs}}$
avg_vs	Average absolute values of aircraft vertical speeds
inter_hori	Number of potential crossings of aircraft trajectories
inter_vert	Mixing degree of aircraft at different flight states(descending/level/climbing)
track_disorder	Variability in aircraft headings
speed_disorder	Variability in aircraft speeds
Div	Rate of divergence between aircraft pairs
Conv	Rate of convergence between aircraft pairs
sensi_d sensi_c	Sensitivity of distance change between diverging/converging aircraft with speed and heading modifications applied to them
insen_d	Div^2/sensi_d
insen_c	Conv^2/sensi_c

2.2 Label noise detection by confident learning

To handle the label noise problem in machine learning, there are two main solutions.

(1) Algorithm level: Construct a robust classifier to resist the impact of label noise.

(2) Data level: Detect and remove label noise to get a clean dataset for training.

We will start from the data level, which is most commonly used in applications because of its convenience and effectiveness.

Confident learning is an approach for characterizing, identifying and learning with noisy labels based on the principles of pruning noisy data, counting to estimate noise and ranking examples to train with confidence[24]. It uses probabilities and noisy labels to count examples in the unnormalized confident joint, estimate the joint distribution and prune noisy data. Only two inputs are needed: Out-of-sample predicted probabilities and array of noisy labels. This method requires no hyperparameters and will output ordered samples according to their label noise probabilities. The whole process is shown in Fig.2.

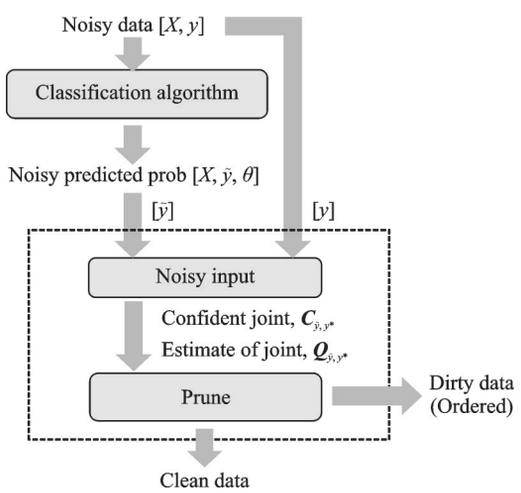

Fig.2 Label noise detection by confident learning

In Fig.2, $\hat{\boldsymbol{Q}}_{\tilde{y},y^*}$ is estimated by counting examples in the joint distribution and calibrating the estimated counts using the given count of noisy labels in each class, $\left|X_{\tilde{y}=i}\right|$, and then normalized. Counts are captured by the confident joint $\boldsymbol{C}_{\tilde{y},y^*} \in \mathbf{Z} \geqslant 0^{m\times m}$, which is the key structure of confident learning. $\boldsymbol{C}_{\tilde{y},y^*}$ is constructed as a confusion matrix $\boldsymbol{C}_{\text{confusion}}$ of given labels $\tilde{y}_k$ and predictions $\arg\max_{i\in 1,\cdots,m}\hat{p}(\tilde{y}=i; x_k, \theta)$. There seems to be dissimilarity problem of probabilities distributions. The problem is fixed by Eq.(1) so that $\boldsymbol{C}_{\tilde{y},y^*}$ is robust for any particular class with certain probabilities.

$$t_j = \frac{1}{\left|\boldsymbol{X}_{\tilde{y}=j}\right|}\sum_{x\in X_{\tilde{y}=j}}\tilde{p}\quad \tilde{y}=j;x,\theta \tag{1}$$

where $\boldsymbol{X}_{\tilde{y}=j}$ denotes a subset in $\boldsymbol{X}$ with noisy label y, $\tilde{p}(\tilde{y}=j; x, \theta)$ the predicted probability of label $\tilde{y}=j$ for $\boldsymbol{X}$ and model parameter θ, and the thresholds t_j are the expected self-confidence for each class.

The introduced label collision is handled by selecting $\tilde{y}^* \leftarrow \text{argmax}_{j\in 1,\cdots,m}\hat{p}_{x,\tilde{y}=j}$. Therefore, in the following formulas, the confident joint $\boldsymbol{C}_{\tilde{y},y^*}$ is defined as

$$\boldsymbol{C}_{\tilde{y},y^*}[i][j] := \left|\hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j}\right| \tag{2}$$

$$\hat{\boldsymbol{X}}_{\tilde{y}=i,y^*=j} = x\in \boldsymbol{X}_{\tilde{y}=i}: \hat{p}_{x,\tilde{y}=j}\geqslant t_j \tag{3}$$

$$j = \underset{k\in 1,\cdots,m;\hat{p}_{x,\tilde{y}=k}\geqslant t_k}{\arg\max}\ \hat{p}_{x,\tilde{y}=k} \tag{4}$$

where $j=\text{argmax}$ only matters when $\left|\left\{k\in 1,\cdots,m: \hat{p}(\tilde{y}=k; x\in X_{\tilde{y}=i}, \theta)\geqslant t_k\right\}\right|>1$, diagonal entries of $\boldsymbol{C}_{\tilde{y},y^*}$ count correct labels and non-diagonals capture asymmetric label error counts.

Given the confident joint $\boldsymbol{C}_{\tilde{y},y^*}$, $\hat{\boldsymbol{Q}}_{\tilde{y},y^*}$ is estimated as

$$T_{ij} = \boldsymbol{C}_{\tilde{y}=i,y^*=j}\cdot\left|\boldsymbol{X}_{\tilde{y}}=i\right| \Big/ \sum_{j\in 1,\cdots,m}\boldsymbol{C}_{\tilde{y}=i,y^*=j} \tag{5}$$

$$\hat{\boldsymbol{Q}}_{\tilde{y}=i,y^*=j} = T_{ij} \Big/ \sum_{i\in 1,\cdots,m;j\in 1,\cdots,m}(T_{ij}) \tag{6}$$

Following the estimation of the joint, pruning, ranking and other heuristics are applied for cleaning dataset. $\hat{\boldsymbol{Q}}_{\tilde{y},y^*}$ is used to estimate the number of label errors and remove errors by ranking over predicted probability. The prune method is based on the noisy rate, where $n\cdot\hat{\boldsymbol{Q}}_{\tilde{y}=i,y^*=j}$ examples were selected with max margin $\hat{p}_{x,\tilde{y}=j}-\hat{p}_{x,\tilde{y}=i}$ for each off-diagonal entry in $\boldsymbol{C}_{\tilde{y},y^*}$. Once label noise is found, we start to train our model with errors removed.

2.3 XGBoost classification algorithm

XGBoost is short for "extreme gradient boost-

ing", which is designed to be a scalable machine learning system for tree boosting[25]. The parallel tree boosting and regularization strategy enable it to run in a much faster way and achieve state-of-the-art results in many machine learning problems. As an ensemble method, the basic idea of XGBoost is to combine several weak models into a strong one, which can be presented as

$$\hat{y}_i = \phi(x_i) = \sum_{k=1}^{K} f_k(x_i) \tag{7}$$

where $f_k(\cdot)$ is a weak model and K the number of weak models.

As a tree boost, the core of XGBoost is the Newton boosting, which searches the optimal parameters by driving the objective function as Eq.(8) towards the minimum direction.

$$L(\phi) = \sum_{i=1}^{n} l(\hat{y}_i, y_i) + \sum_{k=1}^{K} \Omega(f_k) \tag{8}$$

$$\Omega(f_k) = \gamma T + \frac{1}{2}\alpha \| \omega \|^2 \tag{9}$$

where l is the loss function and Ω the regularized term. They measure the performance and control the complexity of the model.

The ensemble model works better in an addictive manner. f_t is added to improve the model and the new objective function is formed as

$$L^t = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t) \tag{10}$$

where $\hat{y}_i^{(t)}$ is the prediction of the ith sample and f_t the weaker model at the tth iteration.

Then, the second-order approximation is used to speed up the optimization procedure to obtain $g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)$, which changes the objective function into

$$L^t \simeq \sum_{i=1}^{n} [\, l(y_i, \hat{y}_i^{(t-1)}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)\,] + \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} \omega_j^2 \tag{11}$$

where g_i and h_i are the first and second order gradient statistics of the loss function. For a fixed tree structure, the optimal weight ω and the corresponding optimal splitting point can be found.

Besides the improvements in the regularized objective, several additional techniques are also used to promote the classification performance, such as overfitting prevention, computation enhancement and so on. More details can be found in Ref. [25].

Considering the mentioned advancements and excellent performance in applications, XGBoost is adopted for our air traffic complexity evaluation under label noise.

2.4 Integrated model based on confident learning and XGBoost

The label noise solution used in this paper is to filter out the noisy label samples, and then train the classification model on the clean dataset. There may be three remained problems:

(1) When detecting and removing noisy label samples, the right labeled samples which are difficult to distinguish may be wrongly deleted.

(2) After removing label noise samples massively, the training data will be severely reduced, which may cause an under-fitting problem.

(3) An imbalanced problem might be intensified for the original imbalance dataset, which is exactly our case. Some minority categories may have fewer samples and even disappear after removing step.

To deal with these problems, we design a novel framework including label noise removal strategy and XGBoost algorithm, as shown in Fig.3, where CL represents the confident learning method used to calculate the noise value according to label probability distributions. Firstly, we adopt several classifiers instead of a single one to acquire different label probability distributions of each sample, so that more general and various label noise information can be offered to the confident learning to detect more extensive label noise. Then several label noise sets are generated. Next step is to incorporate these label noise sets into an overall set that contains as much as label noise samples. Before that, we need to set two indicators NST and NV to reflect the noisy level. They are defined as follows

$$NST_i = \begin{cases} 0 & i \notin S_j \\ \sum_{j=1}^{m} NST_{ij} & i \in S_j \end{cases} \tag{12}$$

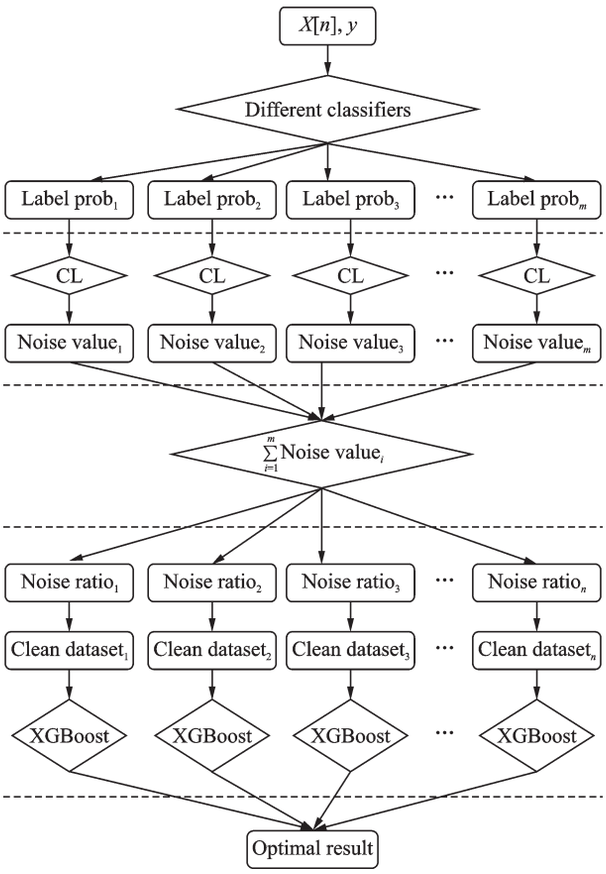

Fig.3 Framework of integrated model

$$NV_i = \begin{cases} 0 & i \notin S_j \\ \sum_{j=1}^{m} (Len_j - ID_{ij}) & i \in S_j \end{cases} \tag{13}$$

$$Len_j = len(S_j) \tag{14}$$

where j, m denote the jth classifier and the number of classifiers, respectively. S_j represent the label noise sets selected by confident learning method under corresponding label probability distribution, which are generated by the jth classifier. NST_{ij} represents the times that the ith sample is selected by S_j. ID_{ij} denotes the ith sample's sequence in S_j according to the noise probability. Len_j is the length of S_j. NV_i, calculated by ID_{ij} and Len_j, represents the noisy level of the ith sample.

In the overall label noise set, each sample corresponds to respective NST and NV. We remove different ratios of label noise samples from the dataset to get different cleanliness datasets for XGBoost algorithm, which are robust to weak label noise datasets. It is worth noting that the minority category should be kept from removing for the balance of datasets. Finally, by comparing the performance on different cleanliness datasets, we can find the optimal label noise removal ratio for XGBoost.

This section discusses the computational complexity. Our proposed integrated method mainly consists of three parts: (1) Getting label probability distribution from classification algorithms, (2) inputting the label possibility distribution into confident learning algorithm to detect label noise samples, and (3) adjusting removal ratio of different label noise samples to acquire optimal performance in XGBoost. Therefore, the computational complexity of our method could be divided into the classification algorithm complexity and the confident learning algorithm complexity. According to Eqs. (1) — (6) and detailed proof in Ref. [24], the computational complexity of confident learning is $O(c^2+nc)$, where c and n denote the number of classes and samples[24]. And for classification algorithms, XGBoost has the greatest computational complexity $O(Kdmn\log_n)$, where K, d, m are the number of trees, the depth of trees and features, respectively. To sum up, the computational complexity of our method is $O(c^2+nc+Kdmn\log_n)$.

3 Experiments and Results

3.1 Dataset and evaluation metrics

All the experiments are executed on the real air traffic operation data collected by automatic devices in Guangzhou region, China. Each record contains flight callsign, SSR code, longitude, latitude, altitude, speed, aircraft type, etc. The yellow part in Fig.4 is the airspace sector we focused on, which is located in the main air route from Guangzhou to Wuhan. From December 1st to December 15th in 2019, we collected 2 769 samples of this sector with each sample corresponding to a one-minute air traffic scenario. The dataset has 24 complexity factors as its features, shown in Table 1, and a complexity level (five ordinal levels) obtained from ATM experts as its label. A dataset with 200 samples is purposely selected as a test set in order to

maintain a unified baseline in experiments. The complexity labels of this dataset are thought to be clean and do not operate label noise removal process, as they are provided by several reliable ATM experts.

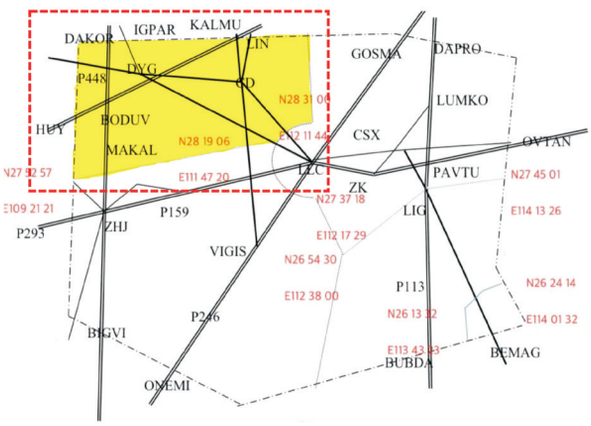

Fig.4 Target airspace sector structure

To verify the performance of the proposed method, we select accuracy, mean absolute error (MAE) and mean absolute error with ordinal penalty (MAE-ordinal) as the evaluation metrics, shown as

$$\text{accuracy}(\hat{Y}, Y) = \frac{\text{Counts}[\hat{y}_i = y_i]}{N} \tag{15}$$

$$\text{MAE}(\hat{Y}, Y) = \frac{1}{N}\sum_{i=1}^{N}|\hat{y}_i - y_i| \tag{16}$$

$$\text{MAE-ordinal}(\hat{Y}, Y) = \frac{1}{N}\sum_{i=1}^{N}\left(e^{|\hat{y}_i - y_i|} - 1\right) \tag{17}$$

where $\hat{Y} = \{\hat{y}_i | i = 1, 2, \cdots, N\}$ denotes the predicted value, $Y = \{y_i | i = 1, 2, \cdots, N\}$ the ground truth, and N the size of samples.

3.2 Effectiveness verification of label noise removal strategy

We have defined *NST* and *NV* to reflect the level of label noise. These label noise samples can be detected by confident learning when inputting sample's labels and class probability distributions. Considering the robust and diversity of label noise datasets, we carry out several label noise cleansing tests under different label probability distributions, which are generated by some classifiers such as support vector machine (SVM), random forest (RF), logistic regression (LR), neutral network (NN), and XGBoost (XGB). Then we integrate the filtered label noise sets to form a comprehensive one, shown in Table 2. In Table 2, every row represents a label noise sample and relevant noise information. Noise sample ID is the index of corresponding label noise samples in the original dataset.

Table 2 Noise level of each label noise sample

NV						*NST*	Noise sample ID
SVM	RF	LR	NN	XGB	Overall		
1 188	1 125	1 129	1 268	1 354	6 144	5	521
1 168	1 191	1 124	1 291	1 363	6 137	5	1 436
1 131	1 210	1 096	1 295	1 362	6 094	5	156
1 100	1 180	1 110	1 290	1 352	6 032	5	157
1 171	1 185	1 076	1 242	1 317	5 991	5	404
1 172	1 153	1 088	1 246	1 323	5 982	5	403
1 123	1 188	1 067	1 203	1 365	5 946	5	598
1 135	1 196	1 044	1 258	1 312	5 945	5	1 527
1 187	1 214	1 104	1 180	1 257	5 942	5	248
1 158	1 119	1 117	1 276	1 259	5 929	5	580

NST and *NV* of label noise samples are shown in Fig.5, which demonstrates that they have strong positive correlation. It means that the more frequently a sample is selected as label noise samples by confident learning, the bigger the noise value is. A sample with bigger noise value is more likely to be a noise. Therefore, in order to verify the effectiveness of label noise removal strategy, we firstly decide to delete label noise samples (about 621 samples) whose *NST* is equal to the number of classifiers, to obtain a clean dataset.

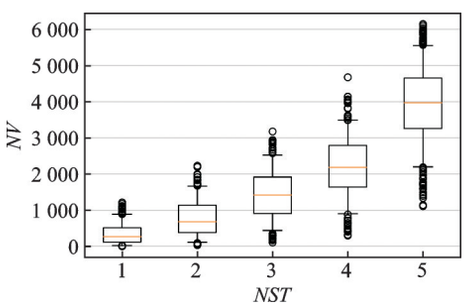

Fig.5 *NST* and *NV* of label noise samples

By inputting the original dataset and cleansed dataset into the classification algorithms, we can observe the effect of label noise removal strategy from

the results. Moreover, we also calculate another cleansed dataset by original confident learning method called orginal-CL, with the aim to compare the performance with our integrated method called rectified-CL. Above resluts are shown in Table 3, from which we can conclude that the presence of label noise actually has an impact on both evaluation metric accuracy and MAE. The performance of other classification algorithms is all improved after label noise removal strategy except LR. Especially for Adaboost ("Ada" for short in Table 3), its accuracy increases almost by 12%, and MAE drops from 0.475 to 0.300 in rectified-CL. These all show the significant influence of removing label noise samples. On the other hand, our rectified-CL results are generally better than the original-CL results. The optimal performance was obtained by XGBoost with accuracy up to 80.00% and MAE of 0.242.

Table 3 Performance comparison under different strategies

Strategy	Non-CL	Original-CL	Rectified-CL
LR	**72.13%(0.342)**	69.45%(0.364)	70.83%(0.351)
NN	68.03%(0.426)	68.89%(0.398)	**69.83%(0.360)**
RF	77.05%(0.305)	**78.05%(0.266)**	77.67%(0.273)
Ada	62.30%(0.475)	70.05%(0.362)	**74.17%(0.300)**
XGB	75.41%(0.292)	78.86%(0.256)	**80.00%(0.242)**

3.3 Label noise removal ratio

In this section, we will study the influence of different removal ratios in detail. Similar with the former section, we use the five classification algorithms, i.e. LR, NN, RF, Adaboost and XGBoost. The parameters of each algorithm are set as optimal values in different label noise removal ratios. The results are shown in Fig.6, and from Fig.6, we can get that:

(1) Different label noise removal ratios have different experimental results. The best result does not lie on the highest label noise removal ratio, but the middle. That means over cleansing may decrease the performance of classifiers, because many right samples may be wrongly removed and a smaller dataset may lead to an under-fitting problem.

(2) For the low label noise removal ratio (less than 30%), LR performs better than NN both in accuracy and MAE. When the label noise removal ratio becomes greater than 30%, the performance of LR is surpassed by NN. But they tend to be consistent as the removal ratio increases. This phenomenon reveals that NN is more easily to be affected by label noise samples in high label noise level, which is the truth in most machine learning problems. For example, compared with the linear model, Ref.[15] attributed the mediocre performance of NN to low intra-rater and inter-rater reliability in human experts, which is exactly the impact of label noise.

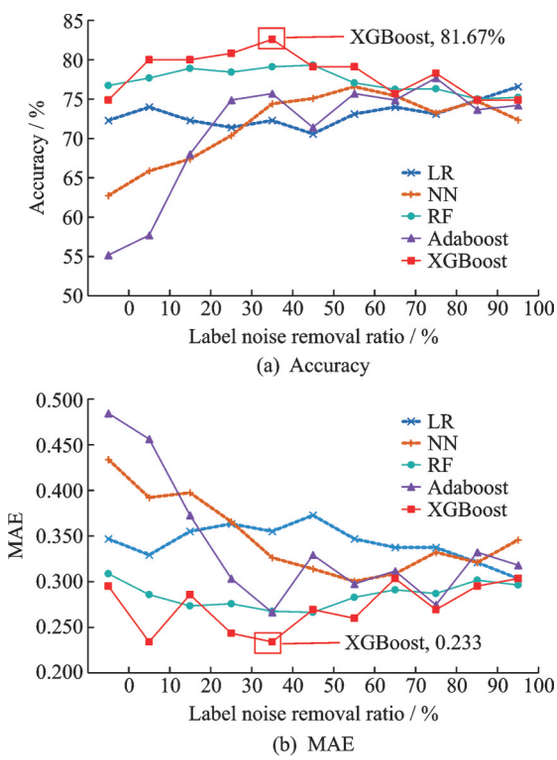

Fig.6 Performance on different label noise removal ratios

(3) Comparing the results of RF with that of Adaboost, we can find that the performance of Adaboost is extremely poor at first under a large number of label noise samples, but it rises rapidly when more label noise samples are removed. Their performance becomes similar when the label noise removal ratio exceeds 60%. The bagging algorithm can usually get better results than the boosting algorithm under label noise samples, because more

weights will be put on these misclassified samples in boosting learning to induce worse performance, while they are actually noisy samples.

(4) In general, when the label noise removal ratio is less than 60%, XGBoost and RF all show obvious and stable advantages on air traffic complexity evaluation. The optimal result with the accuracy of 81.67% and the MAE of 0.233 is achieved by the XGBoost algorithm with the label noise removal ratio of 40%. We can conclude that combining an excellent algorithm with appropriate label noise removal strategy may achieve better result.

In order to observe the ultimate performance of the classifiers, we calculate the optimal results in Table 4. We can find that XGBoost with a label noise removal ratio of 40% attains the greatest accuracy and MAE, nevertheless its optimal MAE-ordinal is achieved under label noise removal ratio of 10%. Similarly, performance of the other algorithms gets the optimum when they generally correspond to different label noise removal ratios. This phenomenon reminds us that the classifiers have different processing methods to deal with the label noise. That means it is almost impossible to get an identical label noise ratio suitable for all classifiers or evaluation metrics. Therefore, we should take the label noise removal ratio as an adjustable parameter in future air traffic complexity evaluation process to seek the best performance we expect.

Table 4 Optimal performance in different classifiers

Model	Accuracy	MAE	MAE-ordinal
CL-LR	75.83% [100%]	0.300 [100%]	0.342 [100%]
CL-NN	75.87% [60%]	0.297 [60%]	0.337 [60%]
CL-RF	78.50% [50%]	0.264 [50%]	0.299 [50%]
CL-Ada	76.92% [80%]	0.264 [40%]	0.274 [40%]
CL-XGB	**81.67% [40%]**	**0.233 [40%]**	**0.251 [10%]**

[•] denotes the label noise removal ratio.

4 Conclusions

In this paper, we firstly consider the label noise problem in air traffic complexity evaluation and propose a confident learning and XGBoost-based method to evaluate air traffic complexity under label noise. In the process of label noise cleansing, the label noise dataset is filtered when labels and their probability distributions are input. In order to contain more label noise information, we calculate several label probability distributions by using some classification algorithms and then incorporate them to form an overall label noise dataset. We define two indicators, NST and NV, to reflect the noisy level of each sample. Label noise samples are then removed by different ratios according to their noisy level to obtain datasets with certain cleanliness. Finally, we run classifiers on these datasets to get the best performance. The experimental results verify the effectiveness of the label noise removal strategy, and the accuracy of 81.67% (MAE of 0.233) is achieved by the XGBoost algorithm under label noise removal ratio of 40%.

The proposed method can be used for supporting airspace sector partition, dynamic airspace and air traffic flow management, etc. We will construct more complexity related features for describing air traffic complexity and carry out suitable features selection to eliminate redundant features to achieve better results in future study.

References

[1] BRÁZDILOVÁ S L, CÁSEK P, KUBALČÍK J. Air traffic complexity for a distributed air traffic management system[J]. Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering, 2011, 225(6): 665-674.

[2] SONG Z X, CHEN Y Z, LI Z L, et al. A review for workload measurement of air traffic controller based on air traffic complexity[C]//Proceedings of 2013 25th Chinese Control and Decision Conference (CCDC). Guiyang, China: IEEE, 2013: 2107-2112.

[3] ZHU X, CAO X B, CAI K Q. Measuring air traffic complexity based on small samples[J]. Chinese Journal of Aeronautics, 2017, 30(4): 1493-1505.

[4] PRANDINI M, PIRODDI L, PUECHMOREL S, et al. Toward air traffic complexity assessment in new generation air traffic management systems[J]. IEEE Transactions on Intelligent Transportation Systems, 2011, 12(3): 809-818.

[5] ZHU X, CAI K Q, CAO X B. A semi-supervised learning method for air traffic complexity evaluation[C]//Proceedings of 2017 Integrated Communica-

tions, Navigation and Surveillance Conference (ICNS). Herndon, VA, USA: [s.n.], 2017: 1A3-1-1A3-11.

[6] PRANDINI M, PUTTA V, HU J H. A probabilistic measure of air traffic complexity in 3-D airspace[J]. International Journal of Adaptive Control and Signal Processing, 2010, 24(10): 813-829.

[7] LEE K, FERON E, PRITCHETT A. Describing airspace complexity: Airspace response to disturbances[J]. Journal of Guidance, Control, and Dynamics, 2009, 32(1): 210-222.

[8] DELAHAYE D, PAIMBLANC P, PUECHMOREL S, et al. A new air traffic complexity metric based on dynamical system modelization[C]//Proceedings of the 21st Digital Avionics Systems Conference. Irvine, CA, USA: IEEE, 2002: 4A2-1-4A2-12.

[9] KOPARDEKAR P, MAGYARITS S. Measurement and prediction of dynamic density[C]//Proceedings of the 5th USA/Europe Air Traffic Management R & D Seminar. Budapest, Hungary: [s.n.], 2003: 10.

[10] GIANAZZA D. Forecasting workload and airspace configuration with neural networks and tree search methods[J]. Artificial Intelligence, 2010, 174(7): 530-549.

[11] XIAO M M, ZHANG J, CAI K Q, et al. ATCEM: A synthetic model for evaluating air traffic complexity[J]. Journal of Advanced Transportation, 2016, 50(3): 315-325.

[12] CAO X B, ZHU X, TIAN Z C, et al. A knowledge-transfer-based learning framework for airspace operation complexity evaluation[J]. Transportation Research Part C: Emerging Technologies, 2018, 95: 61-81.

[13] FRENAY B, VERLEYSEN M. Classification in the presence of label noise: A survey[J]. IEEE Transactions on Neural Networks and Learning Systems, 2014, 25(5): 845-869.

[14] ZHANG W N, WANG D, TAN X Y. Robust class-specific autoencoder for data cleaning and classification in the presence of label noise[J]. Neural Processing Letters, 2019, 50(2): 1845-1860.

[15] ANDRASI P, RADISIC T, NOVAK D, et al. Subjective air traffic complexity estimation using artificial neural networks[J]. Promet-Traffic & Transportation, 2019, 31(4): 377-386.

[16] JELENA D, LORENZ B, FRICKE H. Air traffic control complexity as workload driver[J]. Transportation Research Part C: Emerging Technologies, 2010, 18(6): 930-936.

[17] RADISIC T, NOVAK D, JURICIC B. Reduction of air traffic complexity using trajectory-based operations and validation of novel complexity indicators[J]. IEEE Transactions on Intelligent Transportation Systems, 2017, 18(11): 3038-3048.

[18] NAC Branch. Dynamic density—A review of proposed variables: ACT-540[R]. [S.l.]: NAS Advanced Concepts Branch, 2000.

[19] KOPARDEKAR P, SCHWARTZ A, MAGYARITS S, et al. Airspace complexity measurement: An air traffic control simulation analysis[J]. International Journal of Industrial Engineering—Theory Applications and Practice, 2009, 16(1): 61-70.

[20] DELAHAYE D, PUECHMOREL S. Air traffic complexity: Towards intrinsic metrics[C]//Proceedings of the 3rd USA/Europe Air Traffic Management R & D Seminar. Napoli, Italy: [s.n.], 2000: 11.

[21] LEE K, FERON E, PRITCHETT A. Air traffic complexity: An input-output approach[C]//Proceedings of 2007 American Control Conference. New York, NY, USA: IEEE, 2007: 474-479.

[22] GIANAZZA D, GUITTET K. Selection and evaluation of air traffic complexity metrics[C]//Proceedings of IEEE/AIAA 25th Digital Avionics Systems Conference. Portland, OR, USA: IEEE, 2006: 1-12.

[23] CHATTERJI G, SRIDHAR B. Measures for air traffic controller workload prediction[C]//Proceedings of the 1st AIAA, Aircraft, Technology Integration, and Operations Forum. Los Angeles, CA, USA: AIAA, 2001: 14.

[24] NORTHCUTT C G, JIANG L, CHUANG I L. Confident learning: Estimating uncertainty in dataset labels[C]//Proceedings of International Conference on Machine Learning (ICML). Vienna, Austria: [s.n.], 2020.

[25] CHEN T Q, GUESTRIN C. XGBoost: A scalable tree boosting system[C]//Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. San Francisco, California, USA: ACM, 2016: 785-794.

Acknowledgements This work was supported by the National Natural Science Foundation of China (No.61903187) and Nanjing University of Aeronautics and Astronautics Graduate Innovation Base (Laboratory) Open Fund (No. kfjj20190732).

Authors Mr. **ZHANG Minghua** received his B.S. degree

in air traffic management from Civil Aviation University of China, Tianjin, China, in 2018. He is currently a postgraduate student at the College of Civil Aviation, Nanjing University of Aeronautics and Astronautics, Nanjing, China. His research interests include air traffic complexity and machine learning.

Dr. **XIE Hua** received his B.S. and M.S. degrees in computer science and the Ph.D. degree in System Engineering from Nanjing University of Aeronautics and Astronautics (NUAA) in 1999, 2005 and 2015, respectively. He is currently a lecturer at College of Civil Aviation, NUAA. His research interests include air traffic flow management and security technology.

Author contributions Mr. **ZHANG Minghua** designed the study and wrote the manuscript. Dr. **XIE Hua** contributed to the background of the study and conducted the result analysis. Mr. **ZHANG Dongfang** and Mr. **GE Jiaming** complied the model and interpreted the results. Dr. **CHEN Haiyan** participated in the result analysis and modified the manuscript. All authors commented on the manuscript draft and approved the submission.

Competing interests The authors declare no competing interests.

(Production Editor: ZHANG Huangqun)

基于置信学习和XGBoost的空中交通复杂度含噪评估

张明华[1], 谢 华[1], 张东方[2], 葛家明[2], 陈海燕[2]

(1.南京航空航天大学民航学院,南京 211106,中国;
2.南京航空航天大学计算机科学与技术学院/人工智能学院,南京 211106,中国)

摘要:空中交通复杂度是空中交通运行的关键指标,在空中交通管理中发挥重要作用,例如空域重新配置、空中交通流量管理和空中交通管制员分配。近年来,许多机器学习技术通过构建从复杂度相关因素到空中交通复杂度标签的映射来评估空中交通复杂度。但是复杂度标签的质量问题常常被忽略,导致不佳的空中交通复杂度评估效果。本文针对空中交通复杂度样本中存在的标签噪声,提出了一种基于置信学习和XGBoost的方法来评估标签噪声背景下的空中交通复杂度。置信学习过程通过不同分类算法得到的标签概率分布过滤掉标签噪声样本,XGBoost用于在不同的标签噪声过滤比率数据集上训练健壮且高性能的空中交通复杂度评估模型。对来自中国广州空域扇区的真实数据集进行实验,结果表明:适当的标签噪声移除策略和XGBoost可以有效缓解标签噪声影响,从而使得空中交通复杂度评估模型取得更好的性能。

关键词:空中交通复杂度评估;标签噪声;置信学习;XGBoost