Soft Computing (2020) 24:15595–15610
https://doi.org/10.1007/s00500-020-04887-8

METHODOLOGIES AND APPLICATION

Improving self-training with density peaks of data and cut edge weight statistic

Danni Wei[1] · Youlong Yang[1] · Haiquan Qiu[1]

Published online: 4 April 2020

Abstract
Semi-supervised classification has become an active topic recently, and a number of algorithms, such as self-training, have been proposed to improve the performance of supervised classification using unlabeled data. Considering the influence of spatial distribution of data set and mislabeled samples on the classification performance of self-training method, an improved self-training algorithm based on density peaks and cut edge weight statistic is proposed in this paper. Firstly, the representative unlabeled samples are selected for labels prediction by space structure, which is discovered by clustering method based on density peaks. Secondly, cut edge weight is used as statistics to make hypothesis testing for identifying whether samples are labeled correctly. Thirdly, the labeled data set is gradually enlarged with correctly labeled samples. The above steps are iterated until all unlabeled samples are labeled. The framework of improved self-training method not only makes full use of space structure information, but also solves the problem that some samples may be classified incorrectly. Thus, the classification accuracy of algorithm is improved in a great measure. Extensive experiments on benchmark data sets clearly illustrate the effectiveness of proposed algorithm.

Keywords Semi-supervised classification · Self-training · Density peaks · Cut edge weight · Hypothesis testing

1 Introduction

Classification is an active research problem in the field of machine learning (Domingos 2012; Jm and Mitchell 2015). It has been widely used in many areas, including document classification (Manevitz and Yousef 2002), biological medicine (Zeng et al. 2016) and face recognition (Cao et al. 2011; Su et al. 2009). Traditional classification paradigm only relies heavily on labeled data to achieve an efficient classifier. However, a large amount of labeled instances are often difficult, expensive or time-consuming to obtain, as they require a lot of manpower and material resources. Meanwhile, unlabeled instances may be easy to obtain. Therefore, a great quantity of unlabeled data and few labeled data often appear in many practical applications. In this scenario, traditional supervised classification often fails to learn an appropriate classifier. Nevertheless, semi-supervised classification (SSC) (Zhu and Goldberg 2009; Sakai et al. 2017) addresses this problem. SSC researches on training better classifiers by using the abundant unlabeled data, together with few labeled data. Various SSC methods have been proposed. The common methods are as follows:

Generative method (Narayanaswamy et al. 2017; Nigam et al. 2000). This method assumes that all data, including labeled data and unlabeled data, are generated by the same model. The assumption allows us to utilize techniques, such as the expectation maximization (EM) algorithm, to estimate model parameters and labels of unlabeled data. The main difference of generative method is hypothesis model, and different hypothesis models will produce different methods. In addition, EM algorithm has a strong dependence on the selection of initial value. The common solution is that using multiple initial values for repeated calculation to select the best value, or the optimal solution of parameters can be obtained by optimization algorithm. These methods reduce

Communicated by V. Loia.

✉ Danni Wei
danni0820@163.com

✉ Youlong Yang
ylyang@mail.xidian.edu.cn

Haiquan Qiu
qiuhaiquan0902@163.com

[1] School of Mathematics and Statistics, Xidian University, Xi'an 710071, People's Republic of China

the sensitivity of initial value selection, but increase the computational complexity.

Graph-based method (Wang et al. 2013; Liu et al. 2014). This method maps all samples into a weighted graph. The nodes of graph are labeled or unlabeled samples, and the undirected edges between two vertices $x_i$and x_j represent the similarity of the two samples. Then, semi-supervised learning corresponds to the process of label propagation on the graph. A graph corresponds to a matrix, and the semi-supervised learning could be inferred based on matrix. Thus, graph-based method is easy to explore the properties of algorithm through the analysis of matrix operation, but it has some obvious defects. For example, in the storage aspect, if the number of samples is m, the size of matrix in algorithm is $O(m^2)$. It is difficult to directly deal with large-scale data.

Semi-supervised support vector machine (S3VM) (Chen et al. 2014). S3VM attempts to find a partition hyperplane that separates the two classes of labeled data and traversed a low-density region. The most prominent of S3VM is transductive support vector machine (TSVM). Similar to the standard SVM, TSVM is also proposed for two classification problems. TSVM tries to consider various label assignments to unlabeled samples. Each unlabeled sample is labeled positive or negative. Then, in all of these results, TSVM attempts to find a partition hyperplane that maximized the margin in all samples. Once the partition hyperplane is determined, the final label assignment of unlabeled data is the predicted result. Obviously, it is an exhausting process to attempt various assignments in the case of massive unlabeled samples. And more efficient optimization strategies must be considered. Therefore, one of the key research problems of S3VM is how to design efficient optimization strategies.

Co-training method (Zhou and Li 2010; Zhang et al. 2014). Co-training is one of an important disagreement-based method, and it is originally designed for multi-view data. Co-training assumes that the data have two or more sufficient and conditionally independent views. Each view trains a classifier individually based on labeled data. Then, classifiers learned from each other to well predict labels of unlabeled data. Two processes continue until the classifiers are no longer changed or a preset number of iterations is reached. In co-training algorithm, if sufficient and conditionally independent features can be divided, the performance of trained classifier will be improved greatly. However, the conditional independence of features is often difficult to satisfy in practical applications, so the classification accuracy cannot be enhanced by co-training method.

Self-training method (Yun et al. 2012; Tanha et al. 2017). As its name implies, it attempts to iteratively enlarge the labeled training sets. First, an initial classifier is trained with the small amount of labeled data. Second, unlabeled samples, which are selected with the highest confidence, are added incrementally into the labeled set with their predicted labels. The classifier is retrained. The procedures are repeated until convergence.

Among the above SSC methods, one of the most effective and concise methods is self-training. The self-training does not require explicit feature segmentation or specific assumptions. It has been widely used in many practical applications, such as text classification (Pavlinek and Podgorelec 2017), semantic segmentation (Zou et al. 2018)and sentiment classification (Zhang and He 2013). However, the effect of self-training is limited by the number of initial labeled data and their distribution. If the amount of initial labeled data is quite small and the distribution of entire data sets is not represented, the performance of initial classifier trained by the initial labeled set may be poor. Therefore, it is easy to misclassify unlabeled samples. The updated classifier will be worse along with mislabeled samples which are added directly into next iteration. Iterating in accordance with that will lead to errors accumulation. Consequently, the performance of trained classifier is extremely poor. Thus, identification of mislabeled data and distribution of entire data set play an important role in the self-training algorithm.

In order to solve the influence of the distribution of data set and mislabeled samples on the performance of classifier simultaneously, in this paper, a self-training algorithm based on density peaks and cut edge weight statistic (ST-DP-CEWS) is proposed. In ST-DP-CEWS, the underlying space structure of entire data set is found by density peak clustering method. In the process of discovering space structure, the representative unlabeled samples are selected previously to predict labels. Then, the cut edge weight statistic (CEWS) is used to determine whether the predicted labels are correct. Density peak clustering method and CEWS take full advantage of space structure and information of unlabeled samples. They also deal with the problem that some samples may be misclassify. Thereby, the error accumulation in iterative process is decreased and the performance of trained classifier is increased effectively. The proposed framework has two main advantages: (a) It is not limited by the number of initial labeled data and distribution of entire data space. (b) It identifies the mislabeled instances during self-training process without prior conditions. The experimental results on thirteen benchmark data sets clearly demonstrate the efficiency of the proposed algorithm, which is superior to some previous works.

The remainder of the paper is organized as follows: Sect. 2 reviews the related works. In Sect. 3, the proposed framework and algorithm are described. Section 4 provides the experimental study on twelve benchmark data sets. The conclusion of this paper and some future plans are discussed in Sect. 5.

2 Related works

In SSC, a sample is described by a d-dimensional vector of attributes plus class label as follows:

$$x_i = (x_i^1, x_i^2, \ldots, x_i^d, y_i) \quad (1)$$

where x_i is the ith instance and $i = 1, 2, \ldots, m$. y_i indicates the label of x_i and $y_i \in \{\omega_1, \omega_2, \ldots, \omega_r\}$. r is the number of class. L is the labeled set with y_i known, and U is the unlabeled set with y_i unknown. Particularly, the number of data in U is much larger than that in L for a typical SSC problem. $L \cup U$ forms the training set T_{R}. In addition, there are some unseen data which have the same characteristics as x_i with y unknown to form the testing set T_{s}. The purpose of SSC is to learn a better classifier C by using T_{R} instead of L only to predict the class labels of unlabeled set U or test set T_{s}.

2.1 Previous algorithms to improve self-training

Self-training is an iterative and autonomous learning process. The iterative procedure needs to constantly strengthen the requirements for new sample selection. This process needs to be done with cautious, as inappropriate operations will make the unlabeled samples to be labeled incorrectly. Thus, it will not get a proper classifier with good performance. In order to improve the performance of self-training algorithm and increase the classification accuracy, many algorithms have been researched. The details are as follows:

Zhou and Li proposed the Tri-training algorithm (Li and Guo 2012), and it is a co-training style SSC algorithm. In contrast to previous algorithm that uses two classifiers, it generates three classifiers from initial labeled data set and then they are refined by unlabeled data set in the Tri-training process. In detail, an unlabeled data point is labeled for a classifier if the other two classifiers agree on the labeling under certain conditions. Tri-training does not require sufficient and conditionally independent views, and it is applied to common data sets. But the performance of Tri-training algorithm is usually not stable because the unlabeled samples may often be wrongly labeled during training process. Moreover, the important information of unlabeled samples is not utilized.

As unlabeled data may contain crucial information about the data space, Gan et al. proposed to improve self-training algorithm with fuzzy c-means clustering (ST-FCM) (Gan et al. 2013). The fuzzy c-means (FCM) clustering is integrated into self-training as a helping strategy. It is employed to reveal the underlying space structure by using labeled set and unlabeled set. In the process of iteration, FCM generates the membership degree of each unlabeled sample to different classes. The unlabeled sample that has higher degree is labeled by the classifier trained using labeled data. Nevertheless, ST-FCM algorithm is not appropriate for the non-Gaussian distribution of data sets, which appear quite often in real application. And ST-FCM algorithm may not find the real decision boundary. The reason may be FCM algorithm cannot discover the space structure of non-Gaussian distribution of data.

In order not to be limited by the distribution of initial labeled samples and entire data space, a self-training SSC algorithm based on density peaks of data (ST-DP) (Wu and Shang 2018) was proposed. It improves the ST-FCM algorithm. In ST-DP, the underlying data space structure is discovered by clustering based on density peaks of data (Rodriguez and Laio 2014a). Then, structure is integrated into self-training process to train a better classifier. Clustering based on density peaks of data could discover the underlying structure of data sets, whether they are spherical distributed or non-spherical distributed. However, ST-DP cannot work on the data set with large amounts of strongly overlapping data.

Then, Wu et al. proposed a self-training SSC algorithm based on density peaks of data and differential evolution (ST-DP-DE)(Wu et al. 2018). ST-DP-DE utilizes differential evolution (DE) algorithm to optimize the position of newly labeled data (Di et al. 2017) during the self-training process. This optimization process is incorporated into the framework proposed in ST-DP. Although ST-DP-DE solves the problem of samples overlapping to some extent, it does not fundamentally deal with the defect of ST-DP. Moreover, the optimization algorithm introduces too much computation. The main reason is that overlapping samples are easy to be misclassified in self-training process, and DE cannot solve the trouble of wrong labels.

These previous algorithms have shown the promising prospect, but there are several issues to be considered. These methods may work ineffectively in some circumstances. The four self-training techniques update the classifier by adding unlabeled samples with their predicted labels to labeled data set. However, the predicted labels may be incorrect if the number of initial labeled samples is very small. The performance of trained classifier will be decreased by mislabeled data. Therefore, how to identify the mislabeled samples plays an important role in self-training algorithm.

2.2 Works related to mislabeled data

There are two main methods to identify the mislabeled samples (Triguero et al. 2014): One is based on the nearest neighbor rule, and the other is based on the classifier.

Methods based on the nearest neighbor include edited nearest neighbor (ENN), all KNN (ALLKNN), modified edited nearest neighbor (MENN) and nearest centroid neighbor edition (NCNEdit). ENN method depends only on the distances from the sample to be classified. For each data

sample, it is mislabeled if its label does not agree with the majority of its k-nearest neighbors. ALLKNN is an extension of ENN. In this algorithm, the NN rule varies the number of neighbors from 1 to k; therefore, the NN rule is applied k times. For each example x_t, its label removed at once if it is misclassified by all the NN rule. MENN is a modified technique of ENN. Each data point x_t is mislabeled if its label does not agree with all of its $k+l$-nearest neighbors, where l is the number of samples in L' which were the same distance as the last neighbor of x_t. NCNEdit has a slight modification to ENN. For a given data x_t, the concept of its neighbors is defined considering not only the proximity of x_t, but also their symmetrical distribution around x_t. Particularly, it takes count into the k-nearest centroid neighbors (k NCNs). These k neighbors are searched for through the iteration. The first neighbor of x_t is its nearest neighbor x_r^1. The ith neighbor, x_r^i, $i = 2, \ldots, k$ is such that the centroid of this and previously selected neighbors, $x_r^1, \ldots, x_r^i$, is the closest to x_t. The main thoughts of method based on classifier are as follows: The existing labeled set is divided into n subsets in each iteration. For each of these n parts, a learning algorithm, such as C4.5, is trained on the other $n-1$ parts, resulting in n different classifiers. Then, unlabeled samples are classified by these classifiers. The final labels of unlabeled samples are decided by consensus or majority voting schemes.

Note that the nearest neighbor rule needs to set the distance measure and the value of k in advance. The classifier-based identification method has a high demand for the division of samples and the selection of learning algorithm. Inappropriate selection of these parameters will lead to errors of identification and affect the final classification effect. In addition, these two methods do not utilize the valuable information of unlabeled samples, which will reduce the accuracy rate of identification.

In recent, cut edge weight statistic (CEWS) was proposed to identify mislabeled instances (Muhlenbach et al. 2004). For a given sample, CEWS utilized its sum of cut edge weights as a statistic for hypothesis testing to determine whether the label of the data was correct or not. The CEWS was originally proposed to calculate the separability index in supervised learning (Zighed et al. 2002). Moreover, CEWS does not need to set any parameters in advance, but also can make full use of the information of unlabeled samples. Thus, in this paper, CEWS is integrated into the self-training process to identify the mislabeled samples. CEWS is introduced briefly in next section.

3 Self-training based on density peaks and cut edge weight statistic

To improve the performance of self-training algorithm, two respects including the distribution of entire data set and the identification of mislabeled samples are considered. In this paper, an improved self-training method based on density peaks of data and cut edge weight statistic (ST-DP-CEWS) is proposed and is described in this section. In ST-DP-CEWS, first, the density peaks clustering method is used to discover the underlying space structure of data set, including labeled samples and unlabeled samples. Then, the CEWS technique is used to identify the mislabeled data. The space structure and the process of identification are integrated into each iteration of self-training framework. The proposed algorithm not only decreases the negative impact of mislabeled data on trained classifier, but also takes into account the structure of data space.

3.1 Clustering using density peaks for finding structure of data set

Clustering is a typical unsupervised learning method without labels for analyzing unlabeled data (Jain 2010). The clustering skill helps discover the underlying structure of data space. Recently, a density peaks clustering algorithm was achieved to detect non-spherical clusters (Rodriguez and Laio 2014b). And the correct number of clusters was found automatically. Density peaks clustering algorithm is an important clustering method and can discover the underlying structure of data space of any data set, no matter it is spherical distribution or non-spherical. For each sample x_i, this clustering algorithm computed two quantities: its local density and its distance from points of higher local density. The local density ρ_i of each data point x_i is defined as:

$$\rho_i = \sum_j \chi(d_{ij} - d_c), \chi(x) = \begin{cases} 1, & x < 0 \\ 0, & \text{others} \end{cases} \tag{2}$$

where $\chi(x)$ is the indicator function of x, and d_{ij} is the distance between data points x_i and x_j. d_c is a cutoff distance without a fixed value, and its value is related to data set (Wang and Xu 2017). The second measure δ_i is the minimum distance between x_i and any other data point with higher local density:

$$\delta_i = \begin{cases} \min_j(d_{ij}), & \rho_i < \rho_j \\ \max_j(d_{ij}), & \forall j, \rho_i \geq \rho_j \end{cases} \tag{3}$$

Note that the process of how to find underlying structure of data space by density peaks clustering has introduced in great detail in (Wu and Shang 2018). Here, we just provide a brief introduction. In the process of calculating these two quantities ρ_i and δ_i of all samples, the real structure of entire data space can be discovered by pointing each point x_i to its corresponding sample x_j, which is the nearest point with higher local density of x_i. The samples that like x_j being

pointed to are defined "next" samples; meanwhile, samples that like x_i are referred to as the "previous" sample of x_j.

3.2 Cut edges weight statistic to identify mislabeled data

Cut edge weight statistic (CEWS) is a technology of identifying and handling mislabeled instances. Its main idea is as follows: To begin with, create a relative neighborhood graph in data set and assign weights to each edge. Then, calculate the sum of cut edges weight for each data point. Finally, two-side hypothesis test is applied for identifying the mislabeled instances. The details are described below.

For expressing the proximity between examples, a relative neighborhood graph is used in data set. There exists an edge between vertices x_i and x_j if the distance between the two vertices satisfies the following condition:

$$d(x_i, x_j) \leq \max(d(x_i, x_m), d(x_j, x_m)), \ \forall m \neq i, j \tag{4}$$

where $d(x_i, x_j)$ denotes the distance between vertices x_i and x_j. Different from the KNN editing technology, for a given data point x_i, its neighbors can be automatically found through the above definition without setting the number of neighbors in advance. Two samples connected by edges are neighbors of each other. A cut edge is defined as the edge connecting two samples with different labels. Intuitively, most examples in a neighborhood possess the same label. The label of an example is supposed to be wrong if it has many cut edges. Thus, cut edge plays an important role in identifying mislabeled examples. In addition, for different samples, they might have the same number of cut edges, and the importance of every cut edge is different. Therefore, the weight is introduced to each edge. We define w_{ij} as the weight of edge connecting x_i and x_j. There are two methods based on distances or ranks of the neighbors to define w_{ij}:

$$w_{ij} = (1 + d_{ij})^{-1} \quad or \quad w_{ij} = \frac{1}{r_j} \tag{5}$$

where d_{ij} is the distance of x_i and x_j, and r_j is the rank of the vertex x_j among the neighbors of the vertex x_i.

In order to test whether a sample x_i is mislabeled, the sum of cut edge weights J_i running from this point is calculated. J_i is defined as follows:

$$J_i = \sum\nolimits_{j=1}^{n_i} w_{ij} I_i(j), \quad I_i(j) = \begin{cases} 1, & y_i \neq y_j \\ 0, & y_i = y_j \end{cases} \tag{6}$$

where n_i denotes the number of samples belonging to the neighborhood of x_i. y_i is the class label of x_i.

For x_i, it may be considered to be mislabeled supposing that its J_i value is particularly large. Thus, J_i is selected as a statistic for hypothesis testing. Firstly, null hypothesis is defined as:

H_0: the samples in the graph are labeled independently of each other on the basis of the same probability distribution π_y, where π_y denotes the probability of class y, $y \in \{\omega_1, \omega_2, \ldots, \omega_r\}$.

H_0 specifies a case that for any sample x_i, the probability of examples in its neighborhood possessing labels other than y_i is expected to be no more than $1 - \pi_{y_i}$ under H_0. Hence, an example x_i is considered as a correctly labeled example if the value of J_i is significantly smaller than its expectation under H_0. And x_i is considered as a wrongly labeled example if the value of J_i is abnormally greater than its expectation under H_0.

To test H_0 with statistic J_i, two-side test is used if we are interested in the significantly smaller value and abnormally greater value. The distribution to statistic J_i under H_0 should be studied firstly. $I_i(j)$ are independent and identically distributed Bernoulli random variables with parameter $1 - \pi_{y_i}$. Here, π_{y_i} is the global proportion of class y_i in the training set. As a result, the expectation μ_0 and variance σ^2 of J_i under H_0 are as follows:

$$\mu_0 = (1 - \pi_{y_i}) \sum\nolimits_{j=1}^{n_i} w_{ij} \tag{7}$$

$$\sigma^2 = \pi_{y_i}(1 - \pi_{y_i}) \sum\nolimits_{j=1}^{n_i} w_{ij}^2 \tag{8}$$

Under null hypothesis H_0, J_i is submitted to normal distribution $J_i \sim N(\mu_0, \sigma^2)$, and the test statistics is selected as follows:

$$u = \frac{J_i - \mu_0}{\sigma} \tag{9}$$

and thus, $u \sim N(0, 1)$. Given the significance level α, the rejection domain is concluded as

$$W = \{|u| \geq u_{1-\alpha/2}\} \tag{10}$$

Thereby, the rejection domain of J_i is

$$W = [-\infty, \mu_0 - \sigma \cdot u_{1-\alpha/2}] \cup [\mu_0 + \sigma \cdot u_{1-\alpha/2}, +\infty] \tag{11}$$

For sample x_i, if the value of J_i is significantly less than the expectation under H_0, that is, the value of J_i is in the left rejection domain, x_i is labeled correctly. Otherwise, it may be labeled wrongly. Detailed identification steps by CEWS are as follows: a) Create a relative neighborhood graph in data set S. Initialize correctly labeled data set $T = \{\emptyset\}$ and wrongly labeled data set $T' = \{\emptyset\}$. b) Calculate the weight of each edge. For each sample x_i, calculate J_i, the expectation μ_i and variance δ_i^2 of J_i under H_0. c) Given the significance level α,

for x_i, the rejection domain can be obtained. d) If the value of J_i is on the left rejection domain, its label is corrected. Otherwise, it is a suspect sample. The label of this sample should be relabeled or be predicted in next iteration. The pseudo-code of CEWS to identify mislabeled data samples in data set S is in Algorithm 1.

Algorithm 1 CEWS

Input:
Data set S;
Output:
Correctly labeled set T, Mislabeled set T';
1: Generate a relative neighbourhood graph in S according to formula (4);
2: $T = \{\emptyset\}$, $T' = \{\emptyset\}$;
3: Calculate the weight of each edge according to formula (5);
4: **for** each data x_i in S **do**
5: Calculate J_i, expectation μ_i and variance δ_i^2 of J_i under H_0 according to formulas (6), (7) and (8);
6: Given significant level α, calculate the rejection domain W according to formulas (9), (10) and (11);
7: **if** J_i in the left rejection region **then**
8: $T \leftarrow T \cup x_i$;
9: **else**
10: **if** the neighbours of x_i are all in T **then**
11: relabel x_i;
12: $T \leftarrow T \cup x_i$;
13: **else**
14: $T' \leftarrow T' \cup x_i$;
15: **end if**
16: **end if**
17: **end for**
18: Return T, T'

3.3 Proposed framework and algorithm

Self-training is a process of autonomous learning. In each iteration of self-training algorithm, it is easy to misclassify the unlabeled samples. These errors will participate in the next iteration, which will affect the trained classifier, and reduce the performance of algorithm. It is obvious that identification of mislabeled samples plays an important role in self-training process, especially in the early iterations. The method of CEWS finds the neighbors of a sample according to Eq. (4). It can avoid the negative impact caused by improper selection of parameter. Therefore, in order to improve the performance of self-training algorithm, CEWS is integrated into ST-DP for identifying the wrong labels. In this paper, ST-DP-CEWS is proposed to train a better classifier.

In ST-DP-CEWS, the space structure of data set is discovered by density clustering firstly. The representative unlabeled samples are selected previously for labels prediction by utilizing the structure information in the iterative process. In this way, the accuracy of the prediction labels is improved. Then, CEWS is used to determine whether prediction labels are correct. The labeled set is gradually enlarged by correctly labeled samples for next iteration training. In this way, the disturbance degree of wrongly labeled samples to algorithm performance is decreased. The above procedures are repeated until unlabeled samples are completely labeled. Figure 1 describes the proposed algorithm scheme.

Step 1. Discover the underlying space structure of entire data set by finding density peaks of samples. Making each sample x_i points to its unique nearest sample x_j with higher ρ_i. Then, the underlying structure of entire data space is used in Step 2 and Step 3, which are two similar steps to train a classifier.

Step 2. (a) A initial classifier is trained on the labeled set L with SVM or KNN as the base classifier.

(b) Select a subset L' from U, where each data sample x_k is the "next" sample of L according to the space structure. These samples are labeled by the trained classifier.

(c) Mislabeled samples in L' are identified by CEWS algorithm. The correctly labeled samples are obtained. Then, update L and U. After that, update classifier.

(d) Repeat steps from (a) to (c) until all the "next" samples of L are labeled.

Step 3. (a)Select a subset L' from U, where each data sample x_k is the "previous" sample of updated L. These samples are labeled by classifier.

(b) Mislabeled samples in L' are identified by CEWS algorithm. The correctly labeled samples are obtained. Then, update L and U. After that, update classifier.

(c) Repeat steps from (a) and (b) until all the "previous" samples of L are labeled.

Obviously, Step 3 is similar to Step 2, except replacing "next" in Step 2 with "previous." The specific algorithm pseudo-code of proposed algorithm is described in Algorithm 2.

4 Experimental results and analysis

Twelve benchmark classification data sets are selected to validate the effectiveness of proposed algorithm. These data sets are from the University of California Irvine (UCI) (Asuncion A 2007) and KEEL repositories (Alcalá-Fdez et al. 2011). The more information on these data sets is shown in Table 1. The samples with missing value are deleted from data sets Mammographic, Cleveland and Dermatology. The rest of the data sets do nothing.

4.1 Implementation of experiment

In order to illustrate the effectiveness of proposed algorithm ST-DP-CEWS, four algorithms: Tri-training (Li and Guo 2012), ST-FCM (Gan et al. 2013), ST-DP (Wu and Shang

Algorithm 2 ST-DP-CEWS

Input:
Labeled set L, unlabeled set U;
Output:
A trained classifier C;
for each sample x_i in $L \cup U$ **do**
  calculate ρ_i and δ_i according to formulas (2) and (3);
end for
Discover the "next" and "previous" sample of each instance in L and U;
Train a classifier C on L;
$N = \{x_j\}$, where x_j is "next" samples of L;
while $N \neq \{\emptyset\}$ **do**
  Classify samples in N by C;
  $S \leftarrow L \cup N$;
  Divide N into correctly labeled set T and incorrectly labeled set T' by calling Algorithm 1;
  $L \leftarrow L \cup T$;
  U $\leftarrow$ U\T;
  Update the classifier C with L;
  $N = \{x_j\}$, where x_j is "next" samples of L;
end while
$P = \{x_k\}$, where x_k is "previous" samples of L;
while $P \neq \{\emptyset\}$ **do**
  Classify samples in P by C;
  $S \leftarrow L \cup P$;
  Divide P into correctly labeled set T and incorrectly labeled set T' by calling Algorithm 1;
  $L \leftarrow L \cup T$;
  U $\leftarrow$ U\T;
  Update the classifier C with L;
  $P = \{x_j\}$, where x_j is "previous" samples of L;
end while
Update the classifier C with L;
Return the classifier C.

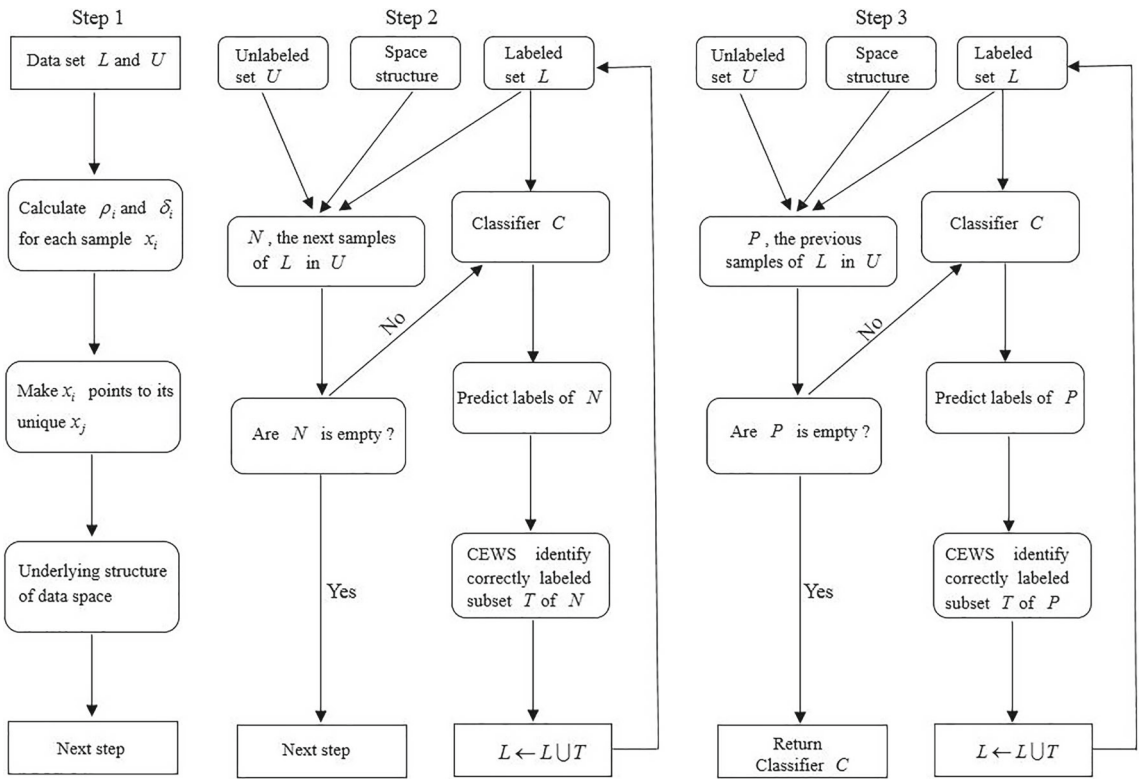

Fig. 1 Proposed algorithm scheme

Table 1 Experimental data sets

Data sets	Examples	Features	Classes
Haberman	306	3	2
Banknote authentic	1372	4	2
Mammographic	830	5	2
Bupa	345	6	2
Ecoli	336	7	8
Glass	214	9	6
Cleveland	297	13	5
Heart	270	13	2
Climate model simulation crashes	540	18	2
Wdbc	569	30	2
Dermatology	358	33	6
Ionosphere	351	34	2

Table 2 Parameters for all algorithms used in experiments

Algorithm	Parameters
KNN	$K = 3$
SVM	LIBSVM: all the parameters are set as default values (Chang and Lin 2011)
Tri-training	Base classifier: SVM or KNN
ST-FCM	Threshold ε_1=1
ST-DP	$P_a = 2$
ST-DP-DE	$P_a = 2$; the parameters of function *DE-POAC*(L',L) are same as (Wu et al. 2018; Di et al. 2017).
ST-DP-CEWS	$P_a = 2$; significant level $\alpha = 0.05$

2018) and ST-DP-DE (Wu et al. 2018), are chosen to compare with ST-DP-CEWS. The parameters of these algorithms are shown in Table 2.

In the experimental part, the tenfold cross-validation strategy is adopted to determine the experimental results. Each of these ten parts is selected as test set T_s, and the remaining nine folds are selected as the training set T_R . When each experiment is performed, 10% samples of T_R are selected as initial labeled set L by randomized strategy. The rest samples of T_R are as unlabeled set U. Therefore, each data set is divided into three parts: L, U and T_s. To ensure the accuracy of experiment, tenfold cross-validation experiment is conducted ten times. The average of ten experimental results is the final result. In each tenfold cross-validation experiment, accuracy rate (AR), mean accuracy rate (MAR) and standard deviation of AR (SD-AR) are selected as the comparison basis of algorithm performance. They are, respectively, computed as:

$$\mathrm{AR}=\frac{1}{N_{T_s}}\sum_{i=1}^{N_{T_s}}\psi(\omega, f(x_i)) \tag{12}$$

where

$$\psi(\omega, f(x_i)) = \begin{cases} 1, & \omega = f(x_i) \\ 0, & \text{else} \end{cases}.$$

$$\mathrm{MAR} = \frac{1}{n}\sum_{k=1}^{n} AR_k. \tag{13}$$

$$\mathrm{SD-AR} = \sqrt{\frac{1}{n}\sum_{k=1}^{n}(AR_k - MAR)^2}. \tag{14}$$

Here, $f(x_i)$ is the predicted label of algorithm to x_i, and ω is the original label. ψ is an indicator function to determine whether prediction label and original label are the same. N_{T_s} is the number of samples in test set T_s. n is the repeated times of computing AR, and AR_k is result of of the k-th calculating AR . MAR represents the classification performance of algorithms, and SD-AR represents the robustness of algorithms.

4.2 Analysis of experimental results

The classification results of tenfold cross-validation on data sets *Glass* and *Ionosphere* are shown in Tables 3, 4, 5 and 6. As shown in these tables, Tri-training, ST-FCM, ST-DP or ST-DP-DE may have worse classification accuracy than KNN or SVM. The reason for this circumstance might be that

Table 3 Experimental results of tenfold cross-validation on data set *Glass* with KNN as base classifier (MAR ± SD-AR, %)

Tenfold cross-validation	Base classifier: KNN					
	KNN only	Tri-training	ST-FCM	ST-DP	ST-DP-DE	ST-DP-CEWS
#1	57.14	61.90	61.90	52.38	52.38	61.90
#2	57.14	52.38	50.00	66.67	47.62	76.19
#3	63.64	57.14	66.67	52.38	76.19	57.14
#4	57.14	57.14	59.09	61.90	54.55	72.73
#5	63.64	57.14	66.67	71.43	59.09	72.73
#6	52.38	38.10	54.55	63.64	68.18	71.43
#7	45.45	59.09	52.38	57.14	61.90	90.91
#8	54.55	50.00	61.18	59.09	71.43	71.43
#9	52.38	59.09	61.90	59.09	68.18	71.43
#10	52.38	54.55	33.33	63.64	57.14	63.64
MAR	55.58	54.65	57.47	60.74	61.67	**70.95**
SD-AR	5.21	6.42	9.99	5.71	8.68	8.71

Bold indicates the highest value of *MAR* and the best performance of corresponding algorithm

Table 4 Experimental results of tenfold cross-validation on data set *Glass* with SVM as base classifier (MAR ± SD-AR, %)

Tenfold cross-validation	Base classifier: SVM					
	SVM only	Tri-training	ST-FCM	ST-DP	ST-DP-DE	ST-DP-CEWS
#1	52.38	61.90	45.45	52.38	33.33	57.14
#2	36.36	36.36	57.14	57.14	38.10	36.36
#3	31.82	47.62	33.33	59.09	59.09	52.38
#4	57.14	42.86	47.62	42.86	42.86	42.86
#5	38.10	40.91	38.10	28.57	50.00	72.73
#6	42.86	33.33	38.10	45.45	57.14	47.62
#7	47.62	31.82	59.09	50.00	59.09	57.14
#8	54.55	47.62	47.62	52.38	50.00	59.09
#9	57.14	45.45	50.00	57.14	42.86	52.38
#10	40.91	38.10	59.09	45.45	61.90	59.09
MAR	45.89	42.6	47.55	49.05	49.44	**53.68**
SD-AR	8.70	8.34	8.66	8.59	9.36	9.50

Bold indicates the highest value of *MAR* and the best performance of corresponding algorithm

some unlabeled data samples are misclassified and directly used in the next iteration. As a result, Tri-training, ST-FCM, ST-DP and ST-DP-DE fail to improve the performance of classification. However, ST-DP-CEWS has better classification accuracy than the base classifier or the other four methods. We believe the reason is that ST-DP-CEWS utilizes CEWS to identify the mislabeled data and utilizes distribution of entire data during the self-training process.

The comparative experimental results of the remaining ten data sets are shown in Tables 7 and 8. As shown in these two tables, when KNN is used as base classifier, the proposed algorithm may be less effective than other algorithms in some data sets. In order to analyze reason, four of the data sets are visualized. As shown in Fig. 2, the data with different classes can be distinguished by the distribution of four data sets of Haberman, Banknote authentication, Wdbc and Heart. By contrast, the data sets of Haberman, Banknote authentication and Wdbc all have an area, in which various classes of samples are densely distributed. Samples in this area are difficult to classify KNN classifier and hard to distinguish whether they are labeled incorrectly. The distribution of Heart is obviously different from the other three data sets. Its distribution is relatively balance. In addition, note that the classification accuracy of proposed algorithm in Cleveland and Climate... is not increased when SVM is base classifier. The reason may be that most attributes values in data sets are close to 0. For the same feature, the difference of each sample is tiny, resulting in the tiny difference between samples. Hence, it is hard to find the decision boundary, which will affect the classification effect.

Table 5 Experimental results of tenfold cross-validation on data set *Ionosphere* with KNN as base classifier (MAR ± SD-AR, %)

Tenfold cross-validation	Base classifier : KNN					
	KNN only	Tri-training	ST-FCM	ST-DP	ST-DP-DE	ST-DP-CEWS
#1	77.11	78.31	71.08	54.22	72.29	78.31
#2	78.31	77.11	72.29	71.08	72.29	77.11
#3	73.49	78.31	81.93	77.11	77.11	74.70
#4	67.47	73.49	72.29	75.90	81.93	72.29
#5	73.49	68.67	69.88	79.52	84.34	81.93
#6	73.49	66.27	74.70	71.08	73.49	75.90
#7	72.29	62.65	74.70	69.88	77.11	75.90
#8	73.49	69.88	71.08	84.34	71.08	71.08
#9	80.72	74.70	77.11	75.90	73.49	81.93
#10	80.72	75.90	73.49	85.54	69.88	77.11
MAR	75.06	72.53	73.86	74.46	75.30	**76.63**
SD-AR	3.92	5.13	3.37	8.43	4.52	3.37

Bold indicates the highest value of *MAR* and the best performance of corresponding algorithm

Table 6 Experimental results of tenfold cross-validation on data set *Ionosphere* with SVM as base classifier (MAR ± SD-AR, %)

Tenfold cross-validation	Base classifier: SVM					
	SVM only	Tri-training	ST-FCM	ST-DP	ST-DP-DE	ST-DP-CEWS
#1	68.57	80.00	85.71	82.86	77.14	82.86
#2	74.29	88.89	82.86	82.86	80.00	91.43
#3	74.29	65.71	77.14	85.71	88.89	91.43
#4	86.11	94.29	74.29	88.57	88.57	94.29
#5	85.71	74.29	88.89	85.71	85.71	74.29
#6	85.71	82.86	77.14	80.00	80.00	91.43
#7	80.00	71.43	82.86	77.14	80.00	88.57
#8	88.57	85.71	74.29	80.00	94.29	91.67
#9	77.14	68.57	85.71	71.43	77.14	85.71
#10	68.57	85.71	77.14	83.33	80.00	82.86
MAR	78.90	79.75	80.60	81.76	83.17	**87.45**
SD-AR	7.06	8.93	4.97	4.66	5.52	5.76

Bold indicates the highest value of *MAR* and the best performance of corresponding algorithm

Table 7 Experimental results of tenfold cross-validation on different data sets with KNN as base classifier (MAR ± SD-AR, %)

Data sets	Algorithms with base classifier: KNN					
	KNN only	Tri-training	ST-FCM	ST-DP	ST-DP-DE	ST-DP-CEWS
Haberman	71.88 ± 7.06	71.28 ± 6.78	**74.45 ± 6.68**	72.65 ± 9.70	73.45 ± 8.26	73.84 ± 7.39
Banknote...	97.96 ± 1.87	98.76 ± 1.09	98.32 ± 0.98	**99.49 ± 0.66**	97.30 ± 2.23	96.76 ± 2.45
Mammographic	72.89 ± 5.78	73.73 ± 3.80	74.30 ± 4.94	74.49 ± 5.01	74.80 ± 3.48	**77.59 ± 5.71**
Bupa	56.52 ± 8.54	54.75 ± 10.56	54.17 ± 10.57	57.41 ± 9.56	59.70 ± 5.46	**62.89 ± 5.67**
Ecoli	63.68 ± 6.49	71.84 ± 5.37	64.3 ± 7.52	70.29 ± 7.19	69.46 ± 8.12	**77.66 ± 5.06**
Cleveland	47.86 ± 8.5	46.11 ± 7.53	48.18 ± 9.03	47.52 ± 7.94	48.16 ± 8.65	**51.17 ± 6.45**
Heart	61.48 ± 6.46	62.59 ± 8.19	61.85 ± 7.78	62.96 ± 8.45	64.81 ± 7.99	**68.15 ± 6.67**
Climate...	90.93 ± 3.15	91.11 ± 3.69	91.30 ± 3.71	91.30 ± 4.15	91.48 ± 4.70	**91.48 ± 3.90**
Wdbc	89.28 ± 3.27	89.63 ± 3.87	92.27 ± 2.39	92.79 ± 1.65	**92.79 ± 3.37**	87.35 ± 7.00
Dermatology	54.44 ± 9.98	55.57 ± 6.16	55.37 ± 10.06	73.54 ± 7.17	71.81 ± 8.40	**77.86 ± 11.68**

Bold indicates the highest value of *MAR* and the best performance of corresponding algorithm

Table 8 Experimental results of tenfold cross-validation on different data sets with SVM as base classifier (MAR ± SD-AR, %)

Data sets	Algorithms with base classifier: SVM					
	SVM only	Tri-training	ST-FCM	ST-DP	ST-DP-DE	ST-DP-CEWS
Haberman	72.19 ± 5.09	73.88 ± 5.68	72.6 ± 7.81	73.53 ± 3.97	73.47 ± 8.90	**75.16 ± 7.08**
Banknote...	94.09 ± 3.99	94.31 ± 5.02	95.04 ± 3.64	97.02 ± 3.79	96.06 ± 3.66	**98.03 ± 1.30**
Mammographic	80.84 ± 5.24	80.96 ± 2.99	80.72 ± 6.17	80.36 ± 5.39	81.08 ± 5.39	**81.81 ± 2.50**
Bupa	58.76 ± 10.14	57.75 ± 11.57	59.76 ± 9.05	59.97 ± 7.96	60.18 ± 8.55	**68.7 ± 8.18**
Ecoli	42.51 ± 8.80	42.54 ± 5.86	42.55 ± 8.95	42.55 ± 9.04	42.58 ± 5.26	**42.66 ± 9.81**
Cleveland	53.86 ± 7.96	53.90 ± 8.01	53.90 ± 7.09	53.84 ± 6.77	53.90 ± 6.89	**53.93 ± 7.17**
Heart	66.67 ± 9.66	71.11 ± 7.37	62.59 ± 8.68	64.07 ± 8.12	64.81 ± 6.68	**78.15 ± 5.35**
Climate...	91.48 ± 4.4	91.48 ± 3.90	91.67 ± 4.70	91.67 ± 3.01	91.85 ± 3.12	**91.85 ± 3.43**
Wdbc	92.79 ± 3.12	92.27 ± 4.45	92.8 ± 3.87	94.02 ± 2.39	94.02 ± 1.62	**95.25 ± 2.10**
Dermatology	57.26 ± 8.16	58.67 ± 8.24	59.21 ± 7.2	65.9 ± 8.58	69.82 ± 5.32	**85.45 ± 9.64**

Bold indicates the highest value of *MAR* and the best performance of corresponding algorithm

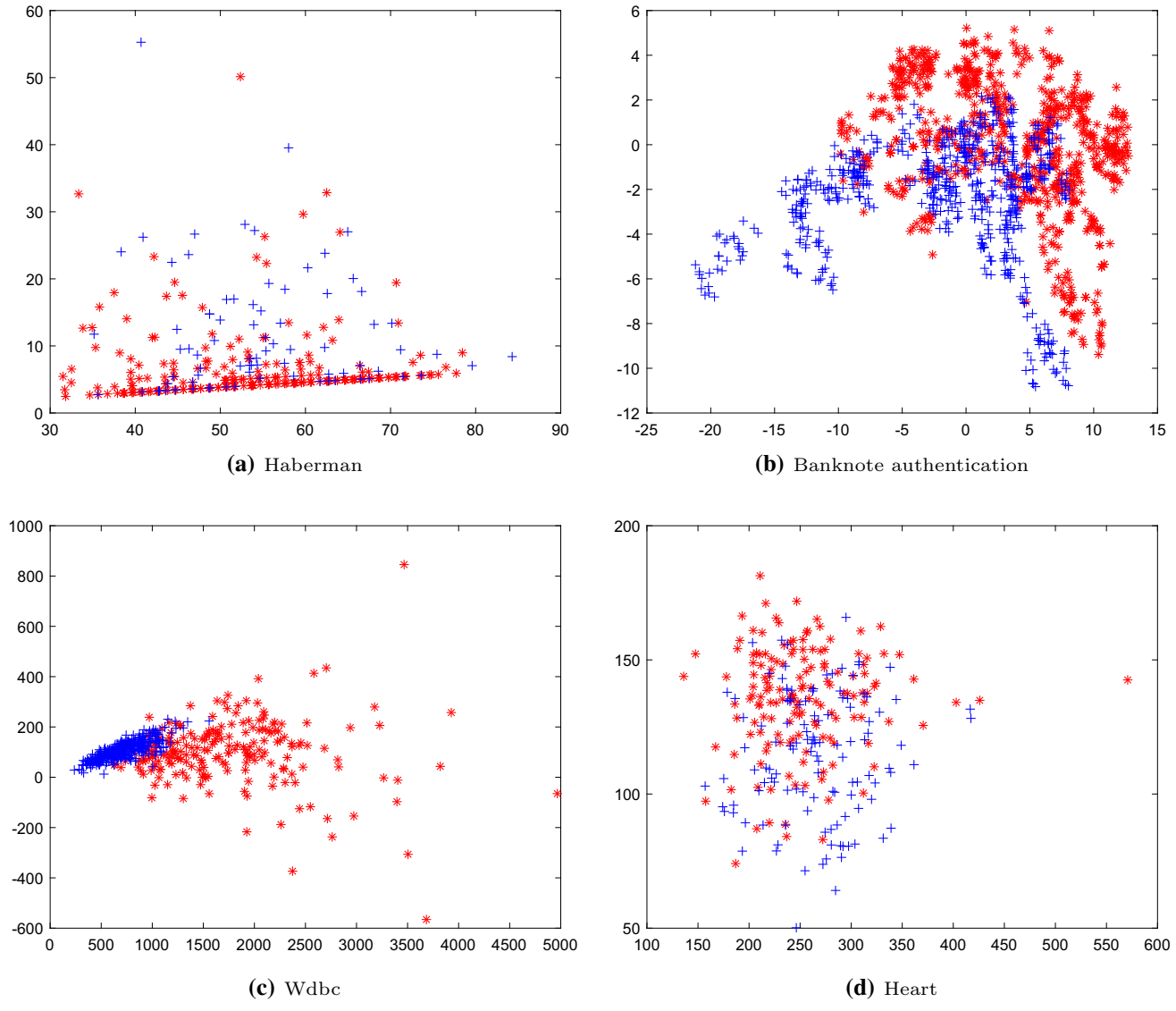

Fig. 2 The distribution of four data sets

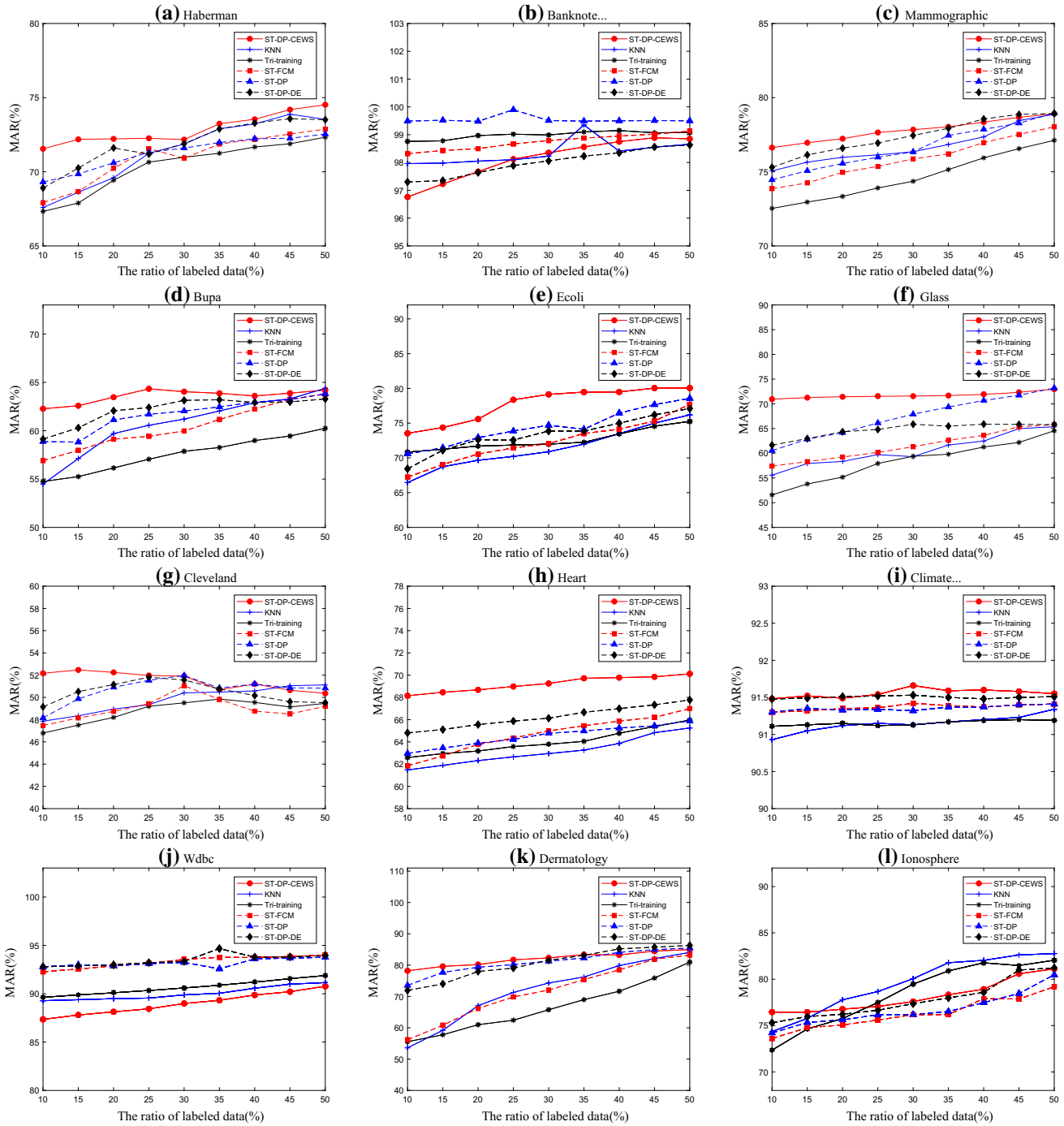

Fig. 3 The relationship between MAR and the ratio of labeled data with KNN as base classifier

In summary, we can draw a conclusion from the experimental results that the performance of proposed algorithm ST-DP-CEWS is better than other algorithms. The main reason is that ST-DP-CEWS utilizes CEWS to identify the mislabeled samples. Furthermore, the space structure information of entire data set and the valuable information of unlabeled samples are used to full advantage in ST-DP-CEWS.

4.3 Impact of labeled data ratio

The performance of proposed algorithm with respect to the labeled data ratio is discussed. Figures 3 and 4, respectively, show the trend of *MAR* of different algorithms with the ratio of initial labeled data. The labeled samples ratio increases from 10% to 50%.

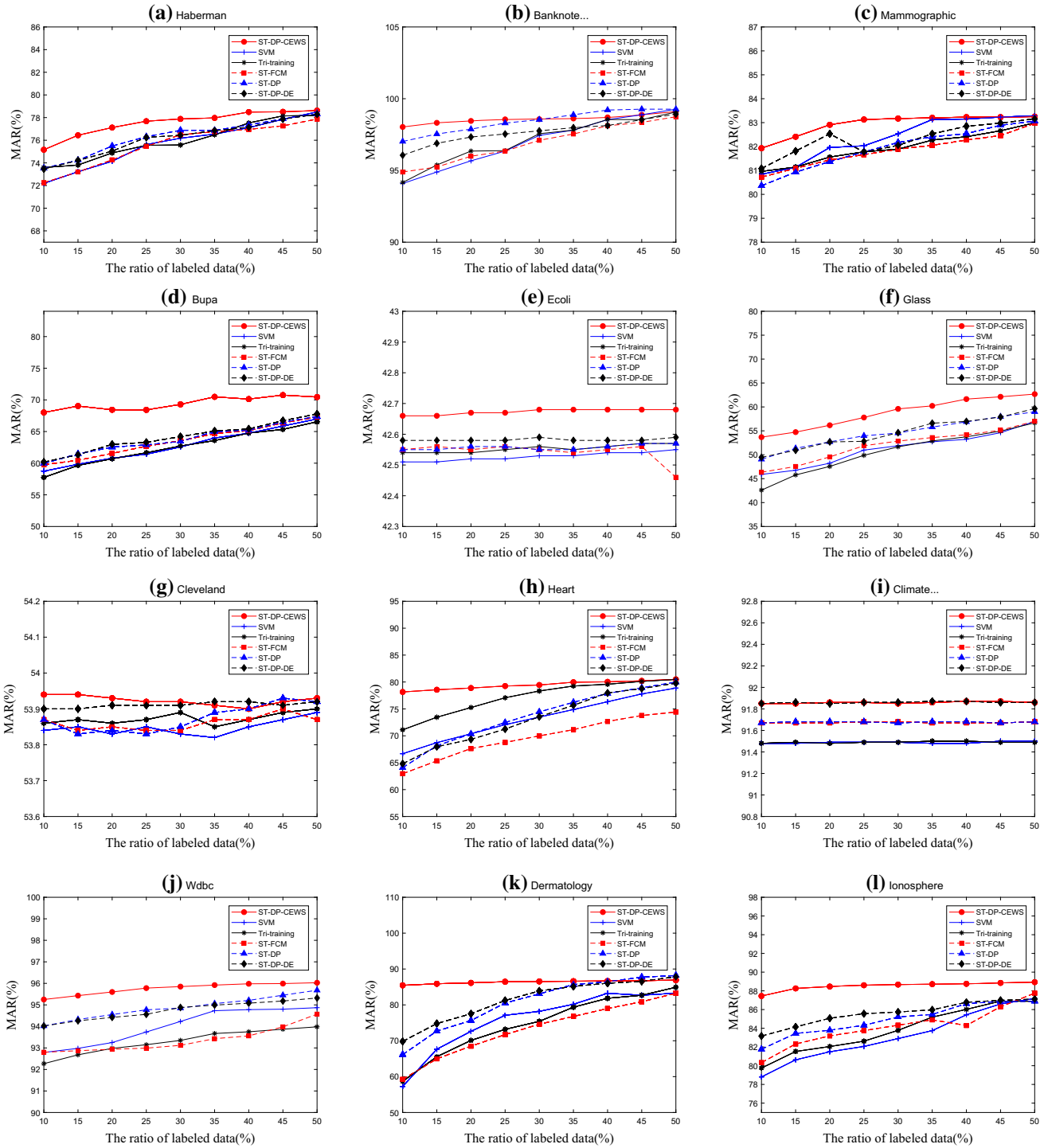

Fig. 4 The relationship between MAR and the ratio of labeled data with SVM as base classifier

It can be seen from Figures 3 and 4 that the performance of ST-DP-CEWS is better than other algorithms on the whole. When the ratio of labeled samples is very little, the classification accuracy of ST-DP-CEWS is significantly higher than other algorithms. The reason is that ST-DP-CEWS utilizes CEWS to identify mislabeled samples during labeled set which is gradually expended. The error accumulation caused by mislabeled samples is reduced in iterative process. Thereby, the classification accuracy is improved. Figures 3 and 4 also show that the classification accuracy of ST-DP-CEWS is close to other algorithms as the ratio of labeled data gradually increases. This is because when the number of

Table 9 Result of the Wilcoxon test with KNN as base classifier

Algorithm versus ST-DP-CEWS	R^+	R^-	P value
KNN	76.0	2.0	0.00297
Tri-training	76.0	2.0	0.00297
ST-FCM	73.0	5.0	0.006272
ST-DP	73.0	5.0	0.006272
ST-DP-DE	64.0	2.0	0.004623

Table 10 Result of the Wilcoxon test with SVM as base classifier

Algorithms versus ST-DP-CEWS	R^+	R^-	P value
SVM	77.5	0.5	0.001944
Tri-training	77.5	0.5	0.001944
ST-FCM	77.5	0.5	0.001944
ST-DP	77.0	1.0	0.002813
ST-DP-DE	65.0	1.0	0.003775

labeled samples is sufficient, the average classification accuracy of these algorithms is relatively stable. ST-DP-CEWS is proposed based on the situation that there are very few labeled samples. ST-DP-CEWS is mainly used in semi-supervised classification, and it is more suitable for classification in the environment with low ratio of labeled samples.

4.4 Comparison with previous algorithms

To validate the effectiveness of our self-training algorithm, ST-DP-CEWS is compared with Tri-training, ST-FCM, ST-DP and ST-DP-DE. The Wilcoxon's test and the Friedman's test, executed by KEEL software Zhou and Li (2010), are used to detect the statistical differences of the compared methods based on the PR values (Wang et al. 2015). Tables 9 and 10 collect the Wilcoxon's test results, and Table 11 collects the average rankings of algorithms by Friedman's test. As shown in Tables 9 and 10, ST-DP-CEWS achieves higher R^+ values than R^- values in all cases. Moreover, the P value is less than 0.05 which means that ST-DP-CEWS has more reliable performance. In addition, according to Friedman test in Table 11, ST-DP-CEWS exhibits the best ranking.

Furthermore, to verify whether the accuracy improvement of the proposed ST-DP-CEWS algorithm is statistical significant, the classification results of KNN, SVM, Tri-training, ST-FCM, ST-DP, ST-DP-DE and ST-DP-CEWS algorithms are determined by the Friedman's test (Fan et al. 2018; Zhang and Hong 2019). The Friedman's test (Derrac et al. 2011) is a multiple comparisons test that aims to detect significant differences between the results of two or more algorithms.

Table 11 Average rankings of algorithms by Friedman test

Algorithms	Ranking with KNN	Ranking with SVM
KNN/SVM	4.8333	5.2083
Tri-training	4.75	4.4583
ST-FCM	4.0417	4.1667
ST-DP	3.1667	3.625
ST-DP-DE	2.6667	2.5
ST-DP-CEWS	1.5417	1.0417

Table 12 Friedman test for ST-DP-CEWS against compared other algorithms

ST-DP-CEWS versus compared algorithms	Significant level $\alpha = 0.05$
(KNN as base classifier)	$H_0 : e_1 = e_2 = e_3 = e_4 = e_5 = e_6$
KNN	
Tri-training	$F = 28.3706$
ST-FCM	$p = 7.84478\text{e}{-09}$
ST-DP	
ST-DP-DE	
(SVM as base classifier)	$H_0 : e_1 = e_2 = e_3 = e_4 = e_5 = e_6$
SVM	
Tri-training	$F = 38.8799$
ST-FCM	$p = 7.8448\text{e}{-09}$
ST-DP	
ST-DP-DE	

The statistic F of Friedman test is shown as follows:

$$F = \frac{12N}{q(q+1)}\left[\sum_{j=1}^{q}\text{Rank}_j^2 - \frac{q(q+1)^2}{4}\right] \tag{15}$$

where N is the total number of data sets; q is the number of compared algorithms. $Rank_j$ is the average rank sum received from each classification value for each algorithm. The null hypothesis for Friedman test is that equality of classification errors among compared algorithms. The alternative hypothesis is defined as the negation of the null hypothesis. The test results are shown in Table 12, at the 0.05 significance level in one-tail test. Clearly, the proposed ST-DP-CEW self-training algorithm is significant superior to other algorithms.

5 Conclusion

In this paper, an improved self-training method based on density peaks and cut edge weight statistic is proposed. First, clustering based on density is used to discover the underlying space structure of data set. The representative unlabeled samples are selected priority to be classified by utilizing

information of space structure. Second, The CEWS method is used to identify the mislabeled data. The incorrectly labeled samples are added to enlarge the labeled set. The above procedures are repeated until unlabeled samples are labeled completely. In proposed algorithm, the valuable information of unlabeled samples is excavated in the course of discovering space structure. And in the process of self-training, mislabeled samples are handled by CEWS, which reduces the negative impact of wrong labels on the performance of algorithm. The experimental results in this paper show the superiority of ST-DP-CEWS over the compared algorithms.

Certainly, the proposed algorithm may have a weakness. In the identification process, the predicted labels are either correct or incorrect. In essence, the identification method adopts one-zero sampling. There are two types of errors during the mislabeled data identifying. Type 1, a correctly labeled sample is regarded as mislabeled data and relabel. Type 2, a mislabeled sample is regarded as correctly labeled data and retained. Two types of errors may harm the classification performance. Therefore, in the future plan, the probability that may be mislabeled is assigned to each sample. The samples, which have higher probability, will be regarded as mislabeled data. We will further study how to use the probability concept to improve the performance of self-training.

Acknowledgements This research was supported by National Natural Science Foundation of China (No. 61573266).

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflicts of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

Informed consent Informed consent was obtained from all individual participants included in the study.

References

Alcalá-Fdez J, Fernández A, Luengo J, Derrac J, García S (2011) KEEL data-mining software tool: data set repository, integration of algorithms and experimental analysis framework. Mult Valued Log Soft Comput 17(2–3):255–287

Asuncion A, Newman D (2007) UCI machine learning repository. Available at http://archive.ics.uci.edu/ml/datasets.php

Cao Y, He H, Huang H (2011) Lift: a new framework of learning from testing data for face recognition. Neurocomputing 74(6):916–926

Chang C, Lin C (2011) LIBSVM: a library for support vector machines. ACM TIST 2(3):27:1–27:27

Chen W, Shao Y, Hong N (2014) Laplacian smooth twin support vector machine for semi-supervised classification. Int J Mach Learn Cybern 5(3):459–468

Derrac J, García S, Molina D, Herrera F (2011) A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm Evol Comput 1(1):3–18

Di W, Xin L, Wang G, Shang M, Yan H (2017) A highly-accurate framework for self-labeled semi-supervised classification in industrial applications. IEEE Trans Ind Inform PP(99):1

Domingos PM (2012) A few useful things to know about machine learning. Commun ACM 55(10):78–87

Fan GF, Peng LL, Hong WC (2018) Short term load forecasting based on phase space reconstruction algorithm and bi-square kernel regression model. Appl Energy 224:13–33

Gan H, Sang N, Huang R, Tong X, Dan Z (2013) Using clustering analysis to improve semi-supervised classification. Neurocomputing 101:290–298

Jm I, Mitchell TM (2015) Machine learning: trends, perspectives, and prospects. Science 349(6245):255–260

Jain AK (2010) Data clustering: 50 years beyond k-means. Pattern Recogn Lett 31(8):651–666

Li Y, Guo M (2012) A new relational tri-training system with adaptive data editing for inductive logic programming. Knowl Based Syst 35(none):173–185

Liu X, Pan S et al (2014) Graph-based semi-supervised learning by mixed label propagation with a soft constraint. Inf Sci 277:327–337

Manevitz LM, Yousef M (2002) One-class svms for document classification. J Mach Learn Res 2(1):139–154

Muhlenbach F, Lallich S, Zighed DA (2004) Identifying and handling mislabelled instances. J Intell Inf Syst Integr Artif Intell Database Technol 22(1):89–109

Narayanaswamy S, Paige B, van de Meent J, Desmaison A, Goodman ND, Kohli P, Wood FD, Torr PHS (2017) Learning disentangled representations with semi-supervised deep generative models. In: Advances in neural information processing systems 30: annual conference on neural information processing systems 2017, 4–9 December 2017. Long Beach, pp 5925–5935

Nigam K, Mccallum AK, Thrun S, Mitchell T (2000) Text classification from labeled and unlabeled documents using em. Mach Learn 39(2–3):103–134

Pavlinek M, Podgorelec V (2017) Text classification method based on self-training and LDA topic models. Expert Syst Appl 80:83–93

Rodriguez A, Laio A (2014a) Clustering by fast search and find of density peaks. Science 344(6191):1492–1496

Rodriguez A, Laio A (2014b) Clustering by fast search and find of density peaks. Science 344(6191):1492–1496

Sakai T, du Plessis MC, Niu G, Sugiyama M (2017) Semi-supervised classification based on classification from positive and unlabeled data. In: Proceedings of the 34th international conference on machine learning, ICML 2017, Sydney, NSW, Australia, 6–11 August 2017, pp 2998–3006

Su Y, Shan S, Chen X, Gao W (2009) Hierarchical ensemble of global and local classifiers for face recognition. IEEE Trans Image Process 18(8):1885–1896

Tanha J, van Someren M, Afsarmanesh H (2017) Semi-supervised self-training for decision tree classifiers. Int J Mach Learn Cybern 8(1):355–370

Triguero I, Sáez JA et al (2014) On the characterization of noise filters for self-training semi-supervised in nearest neighbor classification. Neurocomputing 132:30–41

Wang J, Jebara T, Chang SF (2013) Semi-supervised learning using greedy max-cut. J Mach Learn Res 14(1):771–800

Wang XF, Xu Y (2017) Fast clustering using adaptive density peak detection. Stat Methods Med Res 26(6):2800–2811

Wang Y, Li H, Yen GG, Song W (2015) MOMMOP: multiobjective optimization for locating multiple optimal solutions of multimodal optimization problems. IEEE Trans Cybern 45(4):830–843

Wu D, Shang EA (2018) Self-training semi-supervised classification based on density peaks of data. Neurocomputing 275:180–191

Wu D, Shang M, Wang G, Li L (2018) A self-training semi-supervised classification algorithm based on density peaks of data and differential evolution. In: 15th IEEE international conference on networking, sensing and control, ICNSC 2018, Zhuhai, China, March 27-29, 2018, pp 1–6

Yun J, Yong M, Li Z (2012) A modified self-training semi-supervised SVM algorithm. In: 2012 international conference on communication systems and network technologies. IEEE, pp 224–228

Zeng N, Wang Z, Zhang H, Liu W, Alsaadi FE (2016) Deep belief networks for quantitative analysis of a gold immunochromatographic strip. Cognit Comput 8(4):684–692

Zhang P, He Z (2013) A weakly supervised approach to chinese sentiment classification using partitioned self-training. J Inf Sci 39(6):815–831

Zhang Y, Wen J, Wang X, Jiang Z (2014) Semi-supervised learning combining co-training with active learning. Expert Syst Appl 41(5):2372–2378

Zhang Z, Hong WC (2019) Electric load forecasting by complete ensemble empirical mode decomposition adaptive noise and support vector regression with quantum-based dragonfly algorithm. Nonlinear Dyn 98(2):1107–1136

Zhou ZH, Li M (2010) Semi-supervised learning by disagreement. Knowl Inf Syst 24(3):415–439

Zhu X, Goldberg AB (2009) Introduction to semi-supervised learning. Synth Lect Artif Intell Mach Learn 3(1):130

Zighed DA, Lallich S, Muhlenbach F (2002) Separability index in supervised learning. In: Proceedings of the 6th European conference on principles of data mining and knowledge discovery

Zou Y, Yu Z, Kumar BVKV, Wang J (2018) Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In: Computer vision—ECCV 2018—15th European conference, Munich, Germany, September 8–14, 2018, proceedings, part III, pp 297–313