A review of research on co-training

Article *in* Concurrency and Computation Practice and Experience · March 2021

DOI: 10.1002/cpe.6276

7 authors, including:

Xin Ning
Institute of Semiconductors□Chinese Academy of Sciences
129 PUBLICATIONS **2,949** CITATIONS
SEE PROFILE

Weiwei Cai
Jiangnan University
76 PUBLICATIONS **2,172** CITATIONS
SEE PROFILE

Zhang Liping
Chinese Academy of Sciences
41 PUBLICATIONS **1,069** CITATIONS
SEE PROFILE

Lina Yu
Institute of Semiconductors, Chinese Academy of Sciences
61 PUBLICATIONS **680** CITATIONS
SEE PROFILE

Received: 28 September 2020 | Revised: 29 December 2020 | Accepted: 5 February 2021

DOI: 10.1002/cpe.6276

RESEARCH ARTICLE

WILEY

A review of research on co-training

Xin Ning[1,2,3] | **Xinran Wang**[4] | **Shaohui Xu**[2,3] | **Weiwei Cai**[5] | **Liping Zhang**[1,2] | **Lina Yu**[1] | **Wenfa Li**[6]

[1]Institute of Semiconductors, Chinese Academy of Sciences, Beijing, China

[2]Cognitive Computing Technology Joint Laboratory, Wave Group, Beijing, China

[3]Shenzhen Wave Kingdom Co., Ltd., Shenzhen, China

[4]Beijing University of Posts and Telecommunications, Beijing, China

[5]School of Logistics and Transportation, Central South University of Forestry and Technology, Changsha, China

[6]College of Robotics, Beijing Union University, Beijing, China

Correspondence
Lina Yu, Institute of Semiconductors, Chinese Academy of Sciences, Beijing, China.
Email: yulina@semi.ac.cn
Wenfa Li, College of Robotics, Beijing Union University, Beijing, China.
Email: 644085325@qq.com

Funding information
the National Natural Science Foundation of China, Grant/Award Numbers: 61901436, 61972040; the Premium Funding Project for Academic Human Resources Development in Beijing Union University, Grant/Award Number: BPHR2020AZ03

Summary

Co-training algorithm is one of the main methods of semi-supervised learning in machine learning, which explores the effective information in unlabeled data by multi-learner collaboration. Based on the development of co-training algorithm, the research work in recent years was further summarized in this article. In particular, three main steps of relevant co-training algorithms are introduced: view acquisition, learners' differentiation, and label confidence estimation. Finally, we summarized the problems existing in the current co-training methods, gave some suggestions for improvement, and looked forward to the future development direction of the co-training algorithm.

KEYWORDS

co-training algorithm, label confidence, machine learning, semi-supervised learning, unlabeled data

1 | INTRODUCTION

According to different learning styles, machine learning can be divided into supervised learning, unsupervised learning, and semi-supervised learning.[1,2] Among them, semi-supervised learning combines supervised learning and unsupervised learning, adopting the idea of self-training, that is, using the existing model to automatically label the unlabeled data, and adding self-labeled samples with high credibility to the training data set through the label confidence estimation, so as to further improve the generalization ability of the model.

As the basis of bifurcated method in semi-supervised learning, co-training is easy to implement and has good compatibility with most common machine learning algorithms. Therefore, it has attracted the interest of many researchers, and its development history is shown in Figure 1. Han and Han[3] conducted a statistical analysis of many key words in the research field of semi-supervised learning, and the results were shown in Figure 2. It can be seen that co-training algorithm has become a hot research topic in recent years.

Traditional co-training algorithms mainly apply some common models in machine learning, such as support vector machine (SVM), decision tree (DT), and so on. With the development of deep learning, researchers have gradually combined deep learning with co-raining, and improve some methods of co-training. The purpose of this article is to classify and summarize the methods in related fields, and compared with the previous co-training

Xin Ning, Xinran Wang, and Shaohui Xu contributed equally to this study.

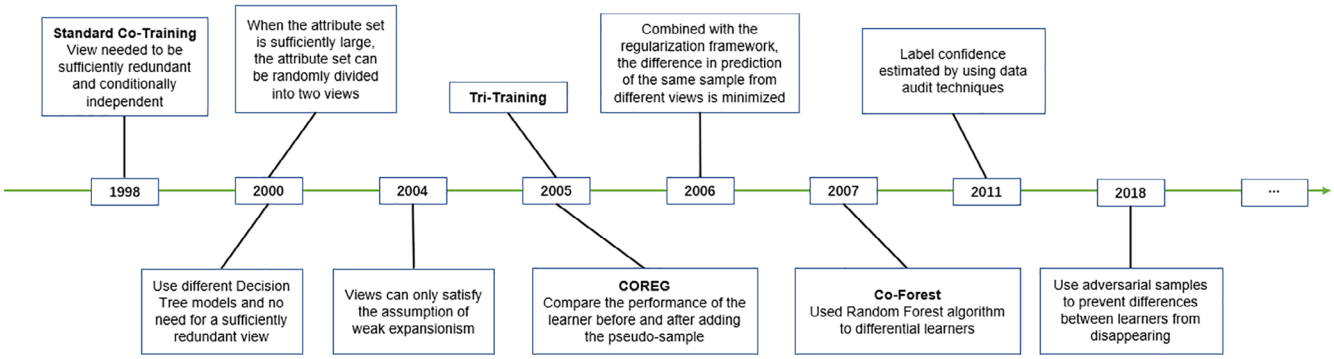


FIGURE 1 The development of co-training

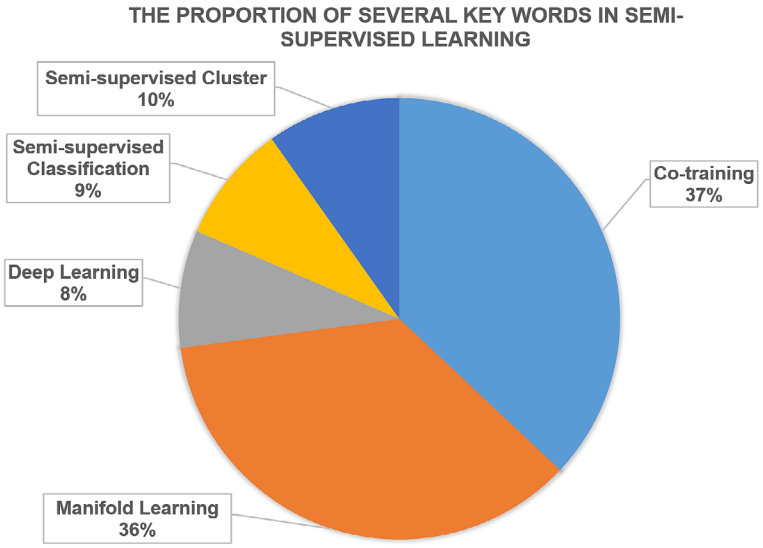


FIGURE 2 Keywords in semi-supervised learning

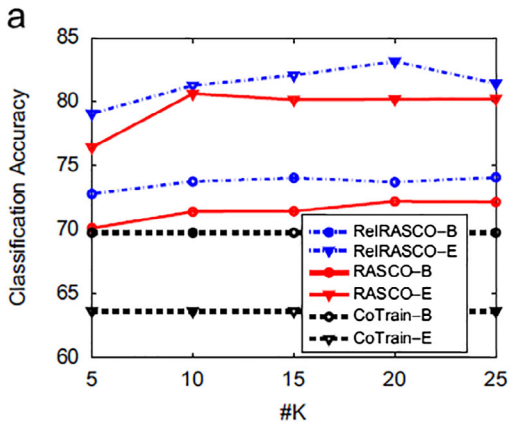
reviews, there are following differences: (1) Summarizes the innovation of related research in recent years, which starting from the main steps of co-training; (2) Some algorithms combining deep learning technology with co-training technology are introduced.

As a method of multi-view learning, co-training improves the generalization ability of the model through the cooperation among multiple learners (Figure 3). Take co-training under two views as an example, and the standard algorithm is shown in Figure 4: Given the labeled data $x_i (i = 1, 2, 3, \dots)$ and the unlabeled data $x_j (j = 1, 2, 3, \dots)$, the first step is to split the view from the labeled data x_i, so that the data representations x_i^1, x_i^2 under two different views can be obtained . Then, different learners C_1, C_2 are trained by using different view data, which can be utilized as the initial classifiers. Finally, the initial classifiers are used to estimate the label confidence of the unlabeled samples, and the trusted samples are added to the training dataset for iterative training to optimize the classifiers. Once all the unlabeled samples are self-labeled, the training ends.

The standard co-training algorithm uses multiple classifiers to explore useful information in unlabeled samples, but there are some problems: (1) In fact, many datasets do not have multiple views, and in the absence of professional knowledge, it is difficult to manually divide a single view of the current data into multiple views. (2) In the design of the initial learners, how to measure the difference between the two learners, and then to select the initial learners with large difference, there is no clear estimation method; (3) It is difficult to make further use of the differences between multiple learners, since after a certain stage of training, the differences between multiple learners tend to decrease; (4) Learners usually do not have high accuracy in the initial stage, and are easy to attach wrong labels to some unlabeled data, which leads to the continuous amplification of the noise in the data in the iterative training process, and the deterioration of model generalization ability when the noise is serious.

In this article, we will give an overview of the algorithm development of co-training in Section 2. Section 3 introduced the three main steps of co-training: view acquisition, learners' differentiation, and label confidence estimation, and gave a brief overview of the improvement and development in recent years. Section 4 classified and listed the applications of co-training in different research tasks in recent years. Section 5 summarized the existing researches and proposed the possible development direction of co-training in the future.

FIGURE 3 The results on "Audio Genre" dataset



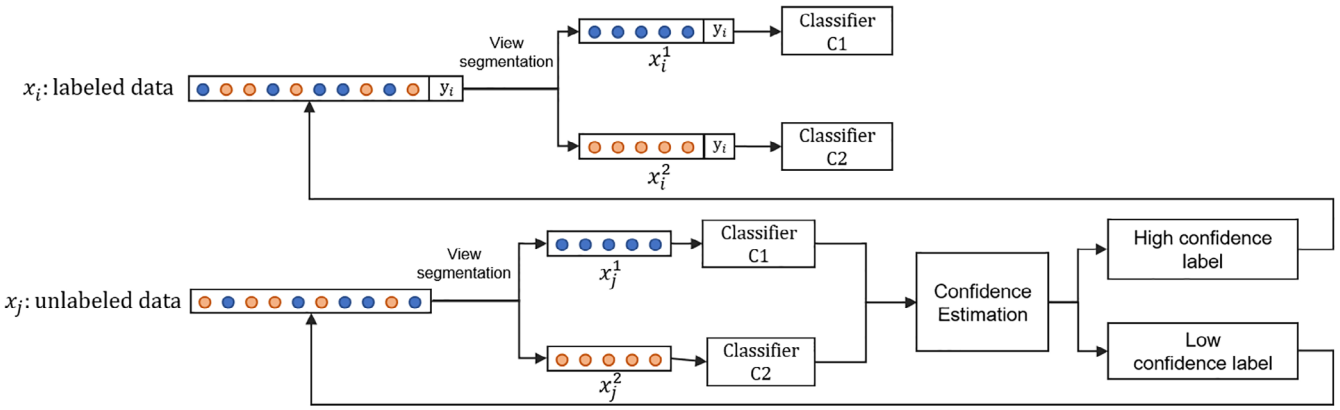


FIGURE 4 Schematic diagram of co-training algorithm

2 | DEVELOPMENT OF CO-TRAINING

In some application tasks, the dataset may contain a variety of attributes, and each attribute represents the characteristic expression of the current sample on different dimensions. The single attribute is called a "View" of the data.[1] According to the number of attributes used in the algorithm, the co-training algorithm can be divided into multi-view learning and single-view learning. Early co-training algorithms has high requirements for views, and most of them belong to multi-view co-training. Both single-view learning and multi-view learning is to make learners' differentiation, but their basic ideas are different.

Multi-view co-training utilizes multiple sets of attributes in the same dataset, such as different languages in multi-lingual data.[5] By training with the labeled data, each learner can preliminarily recognize the unique information under its own attribute and form a difference with other learners.

The original co-training algorithm was proposed by Blum and Mitchell,[6] who assumed that there are two naturally segmented views in the sample space that are sufficient and redundant and independent with conditions,[6] that is, data from any of them are sufficient to train a strong learner and the views are independent of each other. Subsequently, many researchers have explored the necessary conditions of co-training and demonstrated the requirements of co-training on views. Nigam et al.[7] compared the co-training with the EM model, and showed that the co-training had a better effect when the conditional independence hypothesis was established, and they also proved that artificial view segmentation could play a limited role when the natural segmentation was not established, but the effect was not as good as the natural segmentation. Balcan et al.[8] then proposed that co-training only needs to meet based on the assumption of weak expansionism, which further relaxed the requirements of views for co-training.

Different from the method of multi-view learning which takes advantage of the features of view redundancy and conditional independence, single-view learning differentiates the initial learners and mines more useful information from the data. These differences can be reflected in the data set used for training, the basic models of the learners and the optimization algorithms of the learners. Goldman and Zhou[9] used different decision tree (DT) algorithms to train two different learners from a single attribute, and realized the co-training algorithm on a single-view can be implemented through designing the differentiated basic model, furtherly relaxed the requirements of the co-training algorithm on the view. Subsequently, Zhou and Li[10] used Bootstrap sampling mechanism to generate three sub-data sets from the original data set in the tri-training algorithm

and used the same basic model to train a classifier on each data set generated. Ma and Wang[11] used genetic algorithm (GA) and particle swarm optimization (PSO) to optimize the parameters of two SVMs respectively and formed GA-SVM and PSO-SVM with differences.

3 | THE MAIN STEPS OF CO-TRAINING

By sorting and classifying many co-training algorithms, the main steps of the co-training algorithms are divided into view acquisition, learner differentiation, and label confidence estimation, as shown in Figure 5.

3.1 | View acquisition

Since many datasets do not have the properties that can be naturally splited, how to obtain multiple views from the original single-view has become a hot research topic. In most cases, multiple views obtained after manual splitting cannot guarantee the sufficiency and conditional independence of each view at the same time. Because, if the split views are independent from each other, the feature interdependence in each view is stronger, and the sufficiency of each view is lower. Conversely, if the split view has strong sufficiency, the independence between views is weak.[12] The independence and sufficiency of the split view cannot be maximized at the same time. In recent years, the research work of multiple views acquisition mainly focuses on the following aspects: Random subspace splitting algorithm, which selects the optimal case by dividing the original attribute set into several sub-attribute sets randomly and calculating each splitting situation; The view splitting algorithm based on the view sufficiency and independence. The former based on the priority to ensure the sufficiency of the separated sub-views as the principle of view segmentation, while the latter based on the priority to ensure the independence of the view; the automatic splitting algorithm, which completes the automatic partition of attribute set by adding related constraint items to the loss function; attribute splitting based on prior knowledge of professional domain.

3.1.1 | Random subspace co-training algorithm

Wang et al.[13] believed that two views in the standard co-training algorithm could not cover the global information in some cases, that is, the sufficiency of each view could not be guaranteed. For example, in a three-dimensional view, it is impossible to recover the whole object if only the front and top views of the object are obtained. Therefore, Wang et al. proposed a co-training algorithm called random subspace co-training (RASCO) to generalize two views to multiple views. Furthermore, the effect of the number of subspaces and the dimension on the classification performance is discussed. Relevant experiments show that the more random subspaces, the lower the error rate of classifiers are. The error rate is the lowest when the dimension of the random subspace is about half of the original sample space dimension.

However, when a lot of irrelevant features are included in the data, RASCO is often unable to select the optimal feature subspace, and even the features in one subspace may all be irrelevant. The low degree of feature correlation may reduce the accuracy of the classifier trained by the corresponding subspace. Therefore, Yaslan et al.[14] further improved RASCO algorithm and proposed Rel-RASCO algorithm. By screening out the features with high label correlation among all features, the features with low label correlation were removed to ensure the effectiveness of the random subspace, where the feature correlation was defined as the mutual information (MI) between the feature and the label. The experimental results on five different datasets are shown in Table 1. In addition, for the knn classifier, the overall accuracy results of CoTrain, Rel-RASCO,

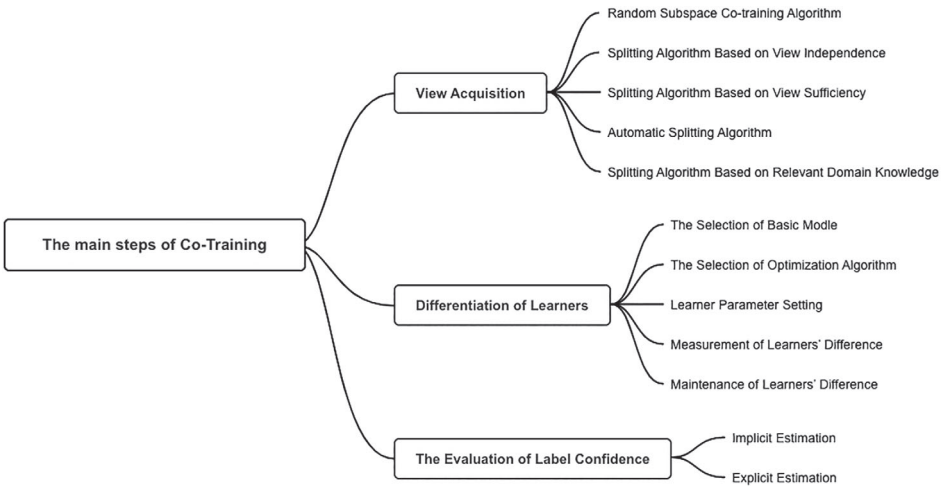


FIGURE 5 The main steps of co-training algorithm

TABLE 1 Datasets

Datasets	#features	#instances	#classes	#Avg. feature relevance
Audio Genre	50	500	5	1.14
OptDigits	64	5620	10	0.55
Classic-3	273	3000	3	0.05
Isolet	617	480	2	0.59
MFeat	649	2000	10	1.14

and RASCO relative to the different values of K on the "Audio Gnere" dataset are shown in Figure 3. It can be seen that CoTrain ensemble accuracies at the beginning is higher than accuracies at the end, which means that CoTrain does not benefit from the unlabeled data. Rel-RASCO outperforms both RASCO and CoTrain. Increasing the number of classifiers (K) increases both Rel-RASCO and RASCO's accuracies, however, the increase after $K = 10$ is not as significant as the increase when K increases from 5 to 10.

3.1.2 | Splitting algorithm based on view independence

Feger et al.[15] proposed maxInd algorithm based on graphs to split views, and conditional mutual information (CondMI) was used to measure the independence between two views and the independence of each pair of features in the same view, and mutual information was used as an indicator to measure the amount of information shared between features. The formula is shown in Equation (1), where H is the entropy, the function $f(x)$ is defined as the probability for a given state x of X and the function $g(y)$ is defined as the probability for a given state y of Y, the function $p(x, y)$ is defined as the probability for the combination of the states of X and Y.[15] As shown in Equation (2),CondMI can be obtained, by giving classes on the basis of mutual information. The algorithm can guarantee the maximum independence between two views. In the ideal case, the $CondMI = 0$ between two sets of attributes after splitting. However, as shown in Table 2, the experiment results of maxInd showed that better independence between views does not necessarily make co-training better. The experiment is based on the News2x2 dataset. The split features sets A and B are conditionally independent and will be referred to as the truly independent split. MaxInd sometimes do not outperform the random split, however in most of times the truly independent way of split does not have a good performance compare to other two ways. Thus, co-training is not only sensitive to the independence between different views but also sensitive to the dependence of the features within each view. View splitting not only needs to consider the independence between views, but also needs to consider the dependency between the features within views. How to balance the trade between the two is also important for co-training.

$$MI(X, Y) = H(X) - H(X|Y) = \sum_x f(x) log_2 \frac{1}{f(x)} - \sum_x \sum_y p(x, y) log_2 \frac{1}{f(x|y)} = \sum_x \sum_y p(x, y) log_2 \frac{p(x, y)}{f(x)g(y)} \tag{1}$$

$$CondMI(X, Y|z = c) = H(X|z = c) - H(X|Y, z = c) \tag{2}$$

Tang et al.[16] evaluated the mutual independence between the two features through CondMI and Chi-square conditional statistics (CHI), and further proposed feature subset selection methods PMID-MI and PMID-chi algorithms. Compared with random partition, Tang's algorithm is easier to make the conditional independence between views stronger. The experiment shows that the error rate of the two algorithms in co-training is lower than that of the random partition algorithm.

TABLE 2 Strength of the classifiers on News2x2 dataset

Split	Classifier	Strength of the classifier	
		$S(V_1)$	$S(V_2)$
truly independent	RBF NN	92.1	92.5
	SVM	92.6	91.3
	NB	91.0	83.4
random	RBF NN	94.4	94.9
	SVM	93.7	94.9
	NB	86.6	87.0
maxInd	RBF NN	94.5	94.2
	SVM	94.3	93.1
	NB	87.7	88.0

3.1.3 | Splitting algorithm based on view sufficiency

Sheng et al.[17] used the rough sets theory to complete view splitting. Attribute reduction is one of the important research contents in rough set theory, which can effectively reduce high-dimensional data to low-dimensional data without causing loss of classification information and can remove some redundant attributes. Based on the core attributes, Sheng et al. added the attributes with the greatest importance in turn until the optimal rough subspace is formed when the mutual information of the current attribute set is equal to the mutual information of original attribute set. Another rough subspace should avoid choosing the set of attributes in the optimal rough subspace to ensure the maximum independence between two rough subspace. This algorithm is based on the premise that the attribute set's sufficiency after splitting is equal to the original attribute set's sufficiency, and the independence between the views after splitting is maximized. They compared the algorithm with the standard co-training algorithm and the self-training algorithm on several UCI datasets with the initial percentage of labeled data was 10%. As shown in Table 3, the new algorithm has the lowest error rate compared to self-training and random co-training.

3.1.4 | Automatic splitting algorithm

Chen et al.[18,19] proposed an algorithm called PMC (pseudo multi-view co-training) for automatic feature decomposition of single view. They hoped that the algorithm could automatically split the attribute set, and each classifier would only use one of the two views after the split. This algorithm initializes two classifiers' weights u and v. And at least one of them is zero in the same dimension. So, there is a constraint condition $u_i v_i = 0$,and it is taken as the constraint of loss function. The two classifiers will split the original data under the condition of optimal loss function, and then two new views are obtained. However, the constraint condition is not convenient for optimization, the deformation of the condition can be obtained in Equation (3). By adding the constraint item of formula (3) to the loss function, the weight parameter was updated to complete the differentiation of the two classifiers. Chen et al.[18] extended the PMC algorithm to multiple classifications. The constraint item is shown in Equation (4), where K is the number of class categories. Chen et al. compared the experimental results of PMC algorithm on Caltech-256 dataset with Random Feature Split (RFS) and other algorithms. Experimental results are concluded in Table 4, which shows that the error rate of the proposed algorithm is lower than other algorithms. In addition, in terms of calculating resource consumption, it took about 12 hours for PMC to complete the whole training in terms of the total number of labeled data and unlabeled data is about 80,000.

$$\sum_{i=1}^{d} u_i^2 v_i^2 = 0 \tag{3}$$

$$\sum_{k=1}^{K} \sum_{i=1}^{d} (u_i^k)^2 (v_i^k)^2 \tag{4}$$

TABLE 3 Comparison of error rates among algorithms (the percentage of labeled data is 10%)

	Self-training		Random co-training		New Algorithm	
Datasets	**Begin**	**End**	**Begin**	**End**	**Begin**	**End**
TTT	0.3161	0.3162	0.3443	0.3769	0.3142	0.3001
Lymp	0.3524	0.3297	0.3218	0.2660	0.2531	0.2514
MR	0.019	0.019	0.0426	0.0244	0.0126	0.0043
Cancer	0.1022	0.0978	0.1073	0.1101	0.0951	0.0826
Iono	0.2090	0.2090	0.2510	0.2667	0.2135	0.2013
Chess	0.276	0.2274	0.3966	0.2898	0.2972	0.1795
Average	0.2125	0.1999	0.2439	0.2223	0.1976	0.1699

TABLE 4 Comparison of co-training with automatic feature split (PMC) to (1) baseline model with only labeled instances; (2) co-training with random feature split (RFS); (3) co-training with ICA and then random feature splitting (ICA-RFS), on the paired handwritten digits set

Test Err(%)	Baseline	RFS	ICA-RFS	PMC
Mean	18.64	13.78	12.22	3.99
STD	8.86	14.24	13.59	3.24

3.1.5 | Splitting algorithm based on relevant domain knowledge

The previous methods are based on certain algorithms to complete the views splitting. In the professional field, prior knowledge can also play a certain role in the splitting of views.

Different from the work of Chen's, Yang et al.[20] do not use PMC algorithm to split views on domain adaptation issues. Instead, they take advantage of common data distribution patterns of SSDA and divide SSDA into two sub-problems: semi-supervised learning (SSL) and unsupervised domain adaptation (UDA). Since SSL and UDA data distribution are significantly different, SSL data and UDA data are two views with strong independence. As shown in Figure 6, the algorithm uses the confidence threshold to screen the new labeled samples, that is, the unlabeled samples whose classification probability is greater than the threshold are regarded as the trusted samples. The experimental results are better than those of previous SOTA papers and more stable.

In the field of biomedical research, Zhang et al.[21] proposed a semi-supervised non-invasive blood glucose detection algorithm based on co-training according to the relevant theories of energy metabolism conservation method. The energy metabolism conservation method can calculate the metabolic heat exchange rate according to the local heat generation of human body, and deduce the blood glucose value from the metabolic heat exchange rate, blood flow rate, blood oxygen, and other physiological parameters. This algorithm can measure the blood glucose value from the relevant data of oxygen consumption and can also deduce the oxygen consumption according to the conservation of energy metabolism. Two sufficient views can be obtained according to the prior knowledge. Only 20%–30% of labeled blood glucose samples are needed for the model to achieve the purpose of supervised training, and the manifold hypothesis in blood glucose parameters is verified.

Among the many algorithms of view splitting, some algorithms cannot guarantee the sufficiency of view after splitting, so the advantages of multiple views complement each other to make up for this defect. In the practice situation, the data are relatively complex and the noise content is high, so the learner trained on the inadequate view is likely to regard the label with high confidence as low confidence, and the label with low confidence as high confidence. This may lead to the amplification of noise during the training process, and then cause the deterioration of the model.

In summary, random subspace co-training algorithms are more suitable for processing high dimensional data and does not need the prior knowledge and difficult mathematical design to find the most perfect division. However, this kind of algorithms need to calculate the optimal number of subspaces, the computational cost is usually larger. Splitting algorithms based on view independence or view sufficiency usually use knowledge of information theory and statistics to evaluate the view independence or the view sufficiency. However, the view independence and the view sufficiency usually cannot both be satisfied and its difficult to decide which should be satisfied first. The advantage of these two kinds of algorithm is that they provide us with methods to divide the original dataset into different views. The advantage of the automatic splitting algorithm is that the cost function and the optimization algorithm are used to automatically complete the splitting of views without considering the relationship between views. However, this kind of algorithm is likely to be limited to the local optimal case because of the optimization algorithm. By using prior knowledge to split views, which can be divided from a specific perspective in this field, and eliminates the computational cost of exploring better segmentation, but requires the algorithm designer to have a high level in relevant fields.

3.2 | Differentiation of learners

Differentiation of learners is mainly provided by the use of basic models, the selection of optimization algorithms and the setting of learner parameters.[22] As to implement the co-training of single view, how to measure the difference between the two learners and how to maintain the difference between the two learners are also discussed in this section.

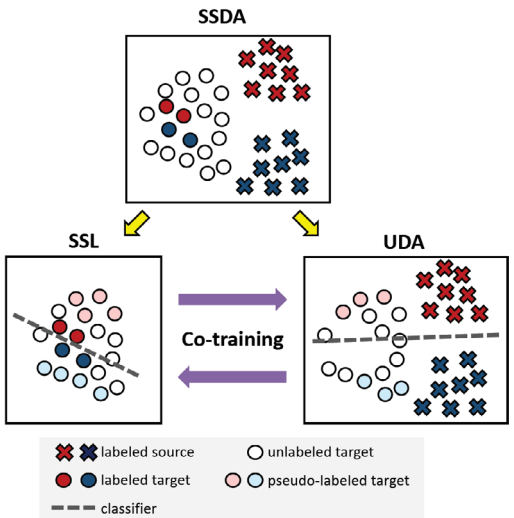


FIGURE 6 Schematic diagram of SSDA view segmentation

3.2.1 | The selection of basic model

Due to the differences in learning mechanisms of different models, more comprehensive data information can be obtained by selecting different basic learners for co-training. For example, Liu et al.[23] used Naive Bayes (NB), SVM, and Neural Network (NN) as the basic models of learning machines respectively to label the unlabeled data through the cooperation among three learners. First of all, the prediction probability of the two teacher-learner to the current unlabeled data should be consistent. Then, the prediction probability of the two teacher-learner should exceed the teacher confidence threshold. And the prediction probability of the student-learner to the current unlabeled data labels should be less than the student confidence threshold. The experimental results show that the "teacher-student" model proposed by Liu is superior to tri-training and self-training algorithms, and the use of fewer labeled samples can achieve the optimal results quickly.

Ju et al.[24] use convolutional neural network (CNN) and recurrent neural network (RNN) as the basic models to perform a differentiation of learners, solve the problem of hyperspectral image classification. And based on the diversity of class probability estimation (DCPE),[25] an modified diversity of class probability estimation (MDCPE) is proposed. MDCPE assumes that there are K-class $c_i(i = 1, 2, 3, \ldots, k)$ in the classification problem. The probability of x belonging to C_i for each sample is calculated as $P(y_i|x)$. The samples with the difference of prediction probability between the maximized classifier C_1 and C_2 for the same category are regarded as the trusted samples. The sample is provided by a classifier with a higher prediction probability to another classifier with a lower prediction probability, as the Equation (5). And vice versa, as shown in Equation (6). On this basis, combined with K-means algorithm, the pseudo-label dataset which is added to the training set is selected under the condition of ensuring class balance. In the experiment, Ju et al.[24] conducted experiments on three datasets SA, PU, PC, and compared overall accuracy (OA), average accuracy (AA), and Kappa coefficient (Kappa) in Table 5 which showed that MDCPE algorithm was superior to DCPE algorithm.

$$label(x) = argmax(P_2(c_i|x) - P_1(c_i|x)) \tag{5}$$

$$label(x) = argmax(P_1(c_i|x) - P_2(c_i|x)) \tag{6}$$

3.2.2 | The selection of optimization algorithm

Ma and Wang[11] established a semi-supervised regression model of co-training using SVM. The algorithm uses GA and PSO to optimize the parameters of two SVMs respectively, forming GA-SVM and PSO-SVM with differences, and made use of the advantages of SVM in solving few-shot samples and non-linear regression problems. The experimental results show that the semi-supervised regression model based on co-training of SVM can effectively utilize the available information in unlabeled samples and improve the accuracy of regression estimation. The algorithm expressed the confidence of pseudo label by calculating the difference of mean square error before and after the pseudo label is added to the sample. In the experiment, the authors compared GA-SVM model, PSO-SVM model, self-GA model (using GA-SVM selection parameters), self-PSO model (using PSO-SVM selection parameters) and semi-supervised co-regression model (semi-SVM). Experimental results show in Table 6 that in the absence of labeled data, a semi-supervised co-regression model based on GA and PSO compared with the results of SVM and a semi-supervised self-learning model has higher noise resistance which can effectively reduce the impact of noise in the pseudo-label.

TABLE 5 Classification results of different methods for the SA, PU, PC datasets

Datasets		RF	SVM	RNN	CNN	SSRN	DCPE-RNN-CNN	MDCPE-RNN-CNN
SA	OA	84.35	81.49	85.67	83.71	91.86	87.76	90.21
	AA	90.01	86.85	91.04	87.69	92.01	92.76	94.01
	Kappa	0.8248	0.7913	0.8402	0.8181	0.9093	0.8633	0.891
PU	OA	75.73	78.32	82.53	82.78	89.47	89.14	91.84
	AA	65.92	74.79	83.2	80.6	89.53	87.81	90.9
	Kappa	0.6541	0.709	0.7685	0.773	0.8585	0.8563	0.8917
PC	OA	95.3	96.12	96.65	92.64	96.61	96.52	97.81
	AA	86.7	87.46	89.72	77.69	94.13	87.04	92.81
	Kappa	0.9338	0.945	0.9524	0.8947	0.9517	0.9506	0.9688

TABLE 6 Mean square error results of five models in Boston Housing Datasets

model	MSE
GA-SVM	1.077418
PSO-SVM	1.354704
Self-GA	1.117736
Self-PSO	1.111669
Semi-SVM	1.015122

3.2.3 | Learner parameter setting

Using different parameter settings on the basic learner can also achieve the purpose of differentiating the learners. Zhou et al.[26] initialized the Minkowski distance measurement in KNN with a different parameter p. They used two regression models with different parameters for co-training, and estimated the label confidence according to the principle of reducing the error rate of the regression model on the labeled data by adding a new labeled example to the training set, and proposed a semi-supervised regression algorithm COREG based on co-training. Specifically, they used the mean square error (MSE) reduction value of different regressors on the labeled instances before and after the addition of the new labeled instances as the final estimation index. However, calculating MSE of all labeled samples is a large cost. In order to reduce the calculation amount, Zhou et al.[26] expressed the label confidence as the MSE of the K-nearest neighbor (KNN) samples, as shown in Equation (7), where h is the original regression learner's prediction, and h' is the new regression learner's prediction. After the training, the final decision is the average of the two KNN predictions. The experimental results in Table 7 show that COREG can effectively solve the semi-supervised regression problem compared with ARTRE or self-training algorithm (all the kNN regressors used in SELF, ARTRE, and LABELED employ 2nd-order Minkowski distance, and the k value is set to 3. The same pool, U', as that used by COREG is used in each iteration of SELF and ARTRE, and the maximum number of iterations is also set to 100). Compared with the standard co-training algorithm, COREG has no requirement on the redundancy of view, and can be applied in a wider situation. However, the KNN algorithm still has the disadvantage of large computational cost.

$$\Delta_{x_u} = \sum_{x_i \in \Omega} (y_i - h(x_i))^2 - (y_i - h'(x_i))^2 \tag{7}$$

3.2.4 | Measurement of learners' difference

By calculating the differences among the candidate learners, the basic learner can be formed by selecting the learners with large differences. However, how to define and calculate the differences among the learners is the main problem faced by this method. In order to solve this problem, Kuncheva et al.[27] proposed a method based on Q statistics to explicitly measure the difference between learning devices and defined the difference between learning devices as shown in Equation (8). For unlabeled samples, each classifier makes different prediction trends. In the human body recognition task, Tang et al.[28] first calculated the Q statistic between the two dichotomies by using the classifier and the classification sample relationship table in Table 8. And then used the Q statistic to measure the difference between the two classifiers. In Table 8, N_{11} represents the number of

TABLE 7 Improvement on average mean squared error

Data set	Attribute	Size	SELF	ARTRE	COREG
2-d Mexican Hat	1	5000	9.2%	12.8%	19.6%
3-d Mexican Hat	2	3000	3.9%	3.7%	5.7%
Friedman #1	5	5000	−1.8%	−4.0%	0.5%
Friedman #2	4	5000	−1.3%	−4.3%	2.1%
Friedman #3	4	3000	−0.9%	−3.6%	0.0%
Gabor	2	3000	4.0%	3.8%	9.0%
Multi	5	4000	−1.9%	−4.4%	1.4%
Plane	2	1000	−3.8%	−3.5%	−1.6%
Polynomial	1	3000	15.1%	17.4%	22.0%
SinC	1	3000	13.0%	16.4%	26.0%

TABLE 8 Classifier result difference matrix

	C_i correct(1)	C_i wrong(0)
C_j correct(1)	N_{11}	N_{10}
C_j wrong(0)	N_{01}	N_{00}

TABLE 9 Comparative results for MCM with the other semi-supervised learning method (5% labeled rate)

Data set	LABELED	MCM	SELF1	SELF2	FAKE-CO
JPCD+SRD	0.60	0.95	0.89	0.89	0.95
SRD	0.46	0.91	0.87	0.86	0.90
JPCD	0.78	0.96	0.95	0.91	0.94

samples which both two classifiers predicted correct, $N10$ represents the number of samples which C_i predicted wrong, C_j predicted correct number of samples, and so on. Tang et al. initialized a batch of SVM with different kernel functions, selected some SVMs with the greatest difference as the basic learners, and took cosine similarity measure as the basic rule for judging the similarity between samples according to the clustering hypothesis. The average prediction result of the learners was taken as the final prediction result by estimating the label confidence of the new labeled sample. The experiment in Table 9 shows that the MCM algorithm has higher accuracy compared with the co-training using random split view (FAKE-CO) and self-training algorithms (SELF1, SELF2).

$$Q_{ij}(h_i, h_j) = \frac{N_{11}N_{00} - N_{01}N_{10}}{N_{11}N_{00} + N_{01}N_{10}} \tag{8}$$

3.2.5 | Maintenance of learners' difference

With the iterative training of the model, different learners tend to be consistent. In order to maintain the differences between learners, based on the principle of compatibility in co-training, Qiao et al.[29] believed that different learners should have similar predictive results for the same sample, as shown in Equation (9). C_1, C_2 are the classifiers of the two views respectively, x_i is a sample of the overall sample space $\{x_1, x_2, x_3, \dots\}$, x_i^1, x_i^2 are samples of x_i in two views, respectively. Qiao et al.[29] calculated the JS divergence of the output value between the learners and added this item into the loss function. Considering the difference constraint between views, they constructed X' with the idea of adversarial learning[30,31] and added the term into the loss function to encourage the learners to learn the difference between views, as shown in Equation (10), where, X' is the adversarial sample space, and its relationship with X satisfies Equation (11). Qiao et al. carried out experiments on CIFAR-10 and other datasets with their proposed co-training algorithm based on deep learning. Experimental results in Table 10 show that the error rate of using eight views is the lowest

TABLE 10 Error rates on SVHN (1000 labeled) and CIFAR-10 (4000 labeled) benchmarks

Method	SVHN	CIFAR-10
GAN	8.11±1.30	18.63±2.32
Stochastic Transformations	-	11.29±0.24
Π Model	4.82±0.17	12.36±0.31
Temporal Ensembling	4.42±0.16	12.16±0.24
Mean Teacher	3.95±0.19	12.31±0.28
Bad GAN	4.25±0.03	14.41±0.30
VAT	3.86	10.55
Deep Co-Training with 2 Views	3.61±0.15	9.03±0.18
Deep Co-Training with 4 Views	3.38±0.05	8.54±0.12
Deep Co-Training with 8 Views	3.29±0.03	8.35±0.06

Note: They report the averages of the single model error rates without ensembling them for the fairness of comparisons. They use architectures that are similar to that of Π Model. "-" means that the original papers did not report the corresponding error rates. Their results are reported from 5 runs.

compared with other algorithms, and it is better to add two loss functions based on view consistency and difference on the basis of conventional loss function. Peng et al.[32] applied this method in the field of image segmentation and achieved good results.

$$f_1(v_1) = f_2(v_2), \forall x = (v_1, v_2) \in X \tag{9}$$

$$f_1(v_1) \neq f_2(v_2), \forall x = (v_1, v_2) \in X' \tag{10}$$

$$X' \bigcap X = \emptyset \tag{11}$$

Zhou et al.[10] proposed co-forest based on the random forest algorithm in ensemble learning. Because the random forest[33] injects a certain randomness into the learning process of trees, even if any two decision trees[34] with the same training set, there will be great differences. Zhou et al. adopted the label confidence estimation method of tri-training method: the voting method and verified the effectiveness of co-forest on the UCI dataset.

The design of the initial learners is often concerned with the effect of the model and the cost of computing resources. Co-training algorithms generally use multiple learners to train at the same time. If learners have a complex structure, it will lead to huge computational resource overhead and slow the formation of the model. Therefore, how to reduce the computational cost of current algorithms is also a future research direction.

3.3 | The estimation of label confidence

The estimation of label confidence is an important step in the incremental algorithm of self-training.[35] Its purpose is to prevent the unlabeled samples from being labeled with wrong labels, thus decreasing the ability of the learner. According to the estimation method of label confidence, we can divide the estimation of label confidence into explicit estimation and implicit estimation. Most of the algorithms in implicit estimation use the difference degree between the results of the learner to reflect the confidence of the current pseudo label. Algorithms in explicit estimation use exact number to display confidence of the label. Usually, these algorithms use the outputs of learners in probabilistic form, the accuracy difference of the model before and after using pseudo labeled sample, or similarity between the current unlabeled sample's pseudo label and surrounding labeled samples.

3.3.1 | Implicit estimation

The implicit estimation does not need to calculate the label confidence value, and the calculation cost is smaller. However, in the initial stage of learners, implicit estimation is highly dependent on learners, unlabeled data may be mislabeled, and that will degrade the performance of learners during iteration, so the accuracy of implicit estimation is lower than that of explicit estimation.

The tri-training algorithm proposed by Zhou et al.[10] solves the problem of label confidence estimation implicitly by using three classifiers to vote. The specific mechanism is as follows: Bootstrap sampling mechanism is used to generate three datasets from the original dataset. The three datasets are used to train the three different basic classifiers C_1, C_2, C_3, respectively. The unlabeled data with consistent classification results of C_2 and C_3 are labeled and added to the training set of C_1 and so on. This algorithm uses voting mechanism, which idea is minority obey majority, to implicitly estimate the label confidence, avoiding the more time-consuming ten-fold cross validation algorithm.[36] In the experiment, Zhou et al. used three J4.8 decision tree classifiers, three BP neural network classifiers and three Naive Bayesian classifiers as the basic learners in the tri-training algorithm and compared the performance with the standard co-training algorithm and Self-training algorithm,the result of tri-training algorithm can improve the performance of the classifier by using the unlabeled data and the classification error rate is lower. Wang et al.[37] proposed the TMNN algorithm by applying tri-training method in named entity recognition task. TMNN algorithm selects LSTM network, BLSTM network, and GRU network as three basic learners. In the experiment, the proportion of labeled data was 20%, and TMNN model had the best effect compared with the three single learners. Ge et al.[38] also adopted the voting mechanism of tri-training in the image classification task. Wang et al. used the summing of the results of multiple learners to estimate the confidence of labels,[13] which is also the voting idea in essence. Sheng et al.[17] used the difference matrix of classifier classification results to divide unlabeled data into three categories: rejected samples, pending samples, and confident samples. However, tri-training can still attach wrong labels to the unlabeled data when learners are not strong enough.

In order to avoid the influence of noise as much as possible, Zhou and Li[36] proposed an algorithm of co-forest, which used the random forest in ensemble learning[39,40] to solve how to pick out the unlabeled data with the highest confidence and how to determine the final hypothesis.[36] Random forest is composed of multiple decision trees, and decision trees can remove the feature of weak correlation by pruning process. Experimental studies have shown that pruning can reduce the impact of noise in data.[41] Although voting mechanism can count differences in learning classification results, but the defect there is only a factor to decide whether the label is confident and it ignored the probability of each classifier prediction results. If most learners' prediction confidence is poorer, voting mechanism is easy to attach wrong label to the unlabeled data.

3.3.2 | Explicit estimation

The earliest explicit estimation method is to use ten-fold cross validation in each training iteration to estimate the confidence of labels when two sufficiently redundant subsets of attributes do not exist.[42] However, the ten-fold cross validation method has the disadvantage of high computational cost. Zhou et al.[43] proposed democratic co-learning algorithm to improve the disadvantages of the voting mechanism. Based on the voting mechanism, the algorithm further compares the sum of the average confidence of the majority learners and the minority learners. If the sum of the confidence average of the majority learners is greater than the minority learners, the current unlabeled data can be labeled, otherwise it cannot be. Zhou et al. used the three basic models of C4.5 decision tree, naive Bayes, and neural network as the three basic learners of the democratic co-learning algorithm in the experiment and compared them with the co-training algorithms combined with the three models on DNA dataset. The result in Figure 7 shows that the accuracy of the democratic co-learning algorithm is higher than that of other algorithms throughout the iteration.

Label confidence can also be estimated based on clustering hypothesis or manifold hypothesis. The clustering hypothesis: (1) The data are from the cluster; 2) Similar data have similar labels. The confidence of each unlabeled sample can be measured by calculating the classification consistency of the samples closest to its K samples which are similar to the unlabeled sample. There are various methods to measure the classification consistency.

In the MCM algorithm, Tang et al.[28] measured the similarity of the two samples by cosine similarity and combined the confidence of the prediction class as the label confidence. But the algorithm based on the similarity of surrounding labels is easy to ignore the difference between the distribution of unlabeled data and labeled data in the data space. Gong et al.[44] combined semi-supervised fuzzy clustering algorithm (SMUC)[45] and weighted KNN algorithm to probe the overall data distribution, and used weighted KNN as the third classifier to evaluate the samples with inconsistent classification of the first two classifiers as the final label of the samples, so as to further prevent noise amplification. Yin et al.[45] and Guo et al.[46] also improved the defect of clustering hypothesis by removing the label that degrades the performance of the learner from the current training set after each iteration, thus timely avoiding the noise in the data distribution that further affects the performance of the learner. Jing et al.[47] compared their Co-KNN-SVM algorithm with MCM algorithm, and the accuracy of the classifier was improved. Zhou and Zhang[48] use the data auditing technique combined with Cut Edge Weight Statistic and select k neighbor as the segmentation criteria to build sample graph structure. Based on manifold hypothesis that is similar samples have similar labels, they put forward the COTRADE algorithm, which can explicitly estimate the label confidence. And they determine the number of pseudo-label data added to the training set in each iteration by minimizing the error rate. The label confidence is mainly calculated by Equation (12), where the pseudo-label sample is (z_p, y_p), C_p is a set of points adjacent to z_p in the graph structure. W_{pq} is the weight of the edge connecting z_p and z_q, which is the similarity of the two samples. The calculation method is shown in Equation (13), where d is the Euclidean distance. If z_p is equal to the label of z_q, $I_{pq} = 0$, otherwise, $I_{pq} = 1$. According to the above equation, the higher the label confidence, the smaller the J_p value. Zhou and Zhang compared COTRADE algorithm with standard co-training algorithm, self-training algorithm, and other algorithms. The experiment results in Table 11 showed that COTRADE achieved better results in course and ads12 datasets than other algorithms.

$$J_p = \sum_{z_q \in C_p} W_{pq} I_{pq} \tag{12}$$

$$W_{pq} = (1 + d(z_p, z_q))^{-1} \tag{13}$$

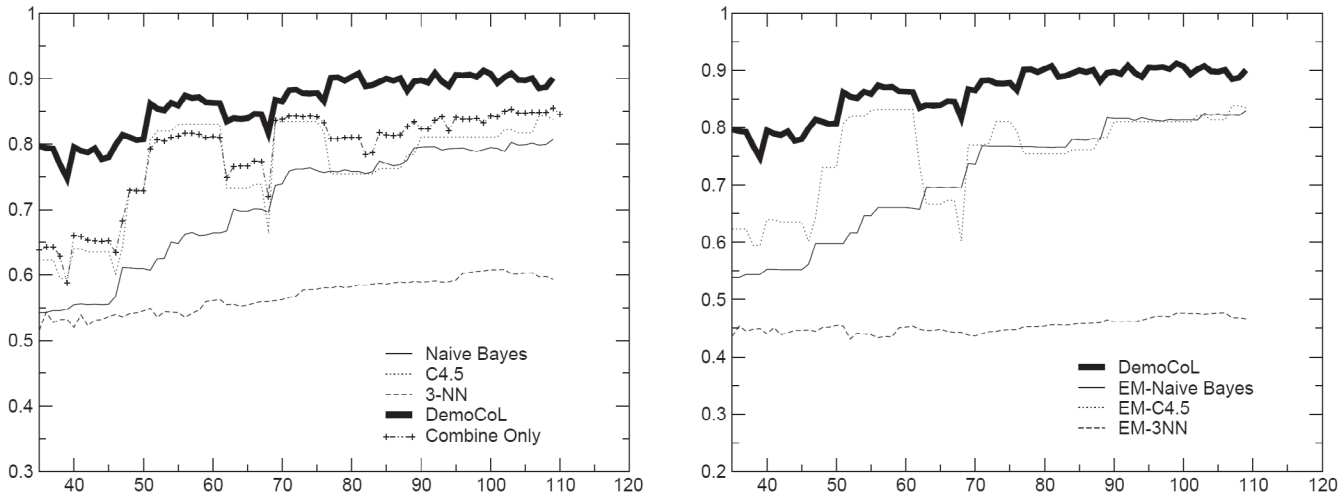


FIGURE 7 Results on DNA data. The *x*-axis is the number of labeled sample, the *y*-axis is the accuracy

TABLE 11 The WIN/TIE/LOSS counts for cotrade against STDCOTRAIN, SELFTRAIN, SETRED, and TRAINORG under different datasets and classifier inducers

Datasets	COTRADE against	Base learner (win/tie/loss [min. p-value, max. p-value, ave. p-value])		
		NAIVE BAYES	CART	LIBSVM
course	STDCOTRAIN	10/0/0 [2e-59, 3e-5, 9e-6]	10/0/0 [7e-40, 2e-26, 2e-27]	3/2/5 [5e-14, 7e-1, 1e-1]
	SELFTRAIN	10/0/0 [2e-64, 3e-34, 7e-35]	9/1/0 [4e-19, 6e-2, 1e-2]	10/0/0 [2e-37, 6e-7, 6e-8]
	SETRED	10/0/0 [1e-25, 4e-11, 4e-12]	6/4/0 [2e-7, 9e-2, 3e-2]	10/0/0 [8e-19, 2e-3, 2e-4]
	TRAINORG	10/0/0 [1e-64, 1e-25, 1e-26]	10/0/0 [4e-19, 3e-2, 7e-3]	10/0/0 [4e-19, 4e-5, 5e-6]
ads12	STDCOTRAIN	1/1/8 [3e-46, 8e-2, 1e-2]	10/0/0 [2e-38, 1e-5, 1e-6]	9/1/0 [2e-48, 6e-1, 6e-2]
	SELFTRAIN	3/2/5 [2e-47, 6e-1, 7e-2]	5/4/1 [6e-6, 9e-1, 3e-1]	9/1/0 [7e-34, 5e-1, 5e-2]
	SETRED	2/5/3 [1e-25, 5e-1, 1e-1]	3/7/0 [2e-3, 9e-1, 4e-1]	8/2/0 [7e-34, 5e-1, 5e-2]
	TRAINORG	10/0/0 [2e-49, 2e-4, 2e-5]	4/5/1 [6e-6, 9e-1, 3e-1]	4/6/0 [5e-3, 8e-1, 2e-1]
NG1	STDCOTRAIN	0/2/8 [1e-36, 8e-1, 1e-1]	8/0/2 [3e-6, 6e-3, 2e-3]	2/8/0 [5e-5, 8e-1,3e-1]
	SELFTRAIN	5/5/0 [4e-5, 8e-1, 2e-1]	1/7/2 [1e-3, 8e-1,3e-1]	7/1/2 [5e-9, 7e-1,7e-2]
	SETRED	6/1/3 [4e-5, 7e-2, 1e-2]	2/6/2 [2e-2, 9e-1,3e-1]	6/4/0 [2e-24, 3e-1,3e-2]
	TRAINORG	6/1/3 [4e-5, 7e-2, 1e-2]	2/8/0 [2e-2, 9e-1,4e-1]	9/1/0 [3e-27, 1e-1,1e-2]

The explicit estimation of label confidence can also work with active learning or reinforcement learning. For example, Gong et al.[49] defined the sample which has high ambiguity as a valuable sample or samples which has low label confidence. First, Naive Bayes was used to classify the unlabeled samples to obtain the probability belonging to each class, and then the variance of the probability belonging to different categories of the same sample was used to represent the ambiguity.[49] Liu et al.[50] also used the concept of ambiguity and gave priority to labeling unlabeled data with low ambiguity. Since iterative data selection steps in co-training can be regarded as a sequential decision problem, Wu et al.[51] combined co-training with reinforcement learning, and used the difference of learner accuracy before and after adding unlabeled data as a reward mechanism for Q learning to further select labels with higher confidence. The label confidence can also be estimate by setting the confidence threshold. Liu et al.[23] used three different learners and used threshold filtering method to label the unlabeled data. First of all, the prediction probability of the two teacher-learner to the current unlabeled data should be consistent; Second, the prediction probability of two teacher-learner should exceed the teacher-confidence threshold; the prediction probability of the student-learner to the current unlabeled data should be less than the student-confidence threshold. The experimental results show that the "teacher-student" model proposed by Liu et al. is superior to tri-training and self-training algorithms and achieves the optimal results quickly by using of fewer labeled samples. Tseng et al.[52] proposed the NDMTT algorithm by adding a confidence threshold mechanism based on tri-training. On the basis of the voting mechanism, the pseudo-labels of the unlabeled data are further screened, so that the accuracy of confidence estimation is further improved. The setting of the confidence threshold will be adjusted according to the accuracy of the model. In the initial stage of the experiment, the setting of the confidence threshold will be small to ensure the acquisition of more new samples. With the improvement of the accuracy of the model, the confidence threshold will be gradually increased.

The accuracy of explicit estimation is high, but the calculation is complex, and the calculation cost is large. The implicit estimation rule is simple, but its accuracy is usually low. Both the explicit estimation and the implicit estimation should pay attention to reducing the number of error labels to lower the label noise in the iterative process which may cause the collapse of the model.

4 | APPLICATION OF CO-TRAINING

With the continuous development of the theory of co-training, the idea of co-training is gradually introduced into different research scenes, which plays an important role in tasks such as data annotation, image segmentation/classification/recognition, and so on.

Xia et al.[53] propose the method of uncertainty aware multi-view co-training (UMCT), which uses pseudo-labeled data for medical image segmentation. Different views are generated by rotating or replacing 3D data, asymmetric cores are used to encourage the diversity of features in different subnets. In addition, uncertainty weighted tag fusion mechanism and Bayesian deep learning are used to estimate the reliability of each view prediction. In Pancreas and LiTS NIH liver Tumor datasets, the experimental results show the effectiveness of the proposed a semi-supervised learning method, and obtain the best performance of the Medical Segmentation Decathlon (MSD) challenge.In order to reduce the number of labeled samples in the synthetic aperture radar recognition task, Du et al.[54] propose a semi-supervised synthetic aperture radar target recognition method based on joint training. Which extracts Lincoln features from the training sample, divides it into two views according to different physical meanings, uses a joint training algorithm to extract the features of two subsets, iteratively trains two SVM classifiers. Using miniSAR real data to test the

method, it shows the advantages compared to other methods. Li et al.[55] propose a new semi-supervised automatic image annotation method based on co-training algorithm to alleviate the need for labeled samples during model training. First, construct two different classifiers from the labeled data, select the pseudo-labeled samples with high label confidence and combined them with the real labeled samples, and retrain the classifier with the faith-labeled samples until the stop condition is reached. Experiments on the LAPR TC-2 and NUS-WIDE datasets show that the accuracy, recall, F-measure, N+ and mAP are better than other methods.Joint training is based on the voting results of two algorithms to obtain labeling results, but when the two algorithms produce inconsistent results, the samples will not be labeled correctly. To solve this problem, Tseng et al.[52] proposed the novel decision module of tri-training (NDMTT) method to improve the automatic data annotation process based on co-training. By adding a third algorithm to assist in judging the credibility of pseudo-labeled samples, the effectiveness of labeled data is improved. Abdelgayed et al.[56] use joint training to process labeled and unlabeled samples for fault detection and classification. In order to extract the hidden features in the current and voltage waveforms, discrete wavelet transform is used, and the harmony search algorithm is used to identify the optimal parameters of the wavelet, which improves the accuracy of fault classification and improves the flexibility and adaptability of system processing.

In the identification task, Zhou et al.[57] in order to solve the limitations of oil well condition recognition and further improve the accuracy and practicability, propose a new method of rod pumping well condition recognition based on multi-view co-training and SVM Hesen regularization. Based on mechanism analysis, prior information and expert knowledge, the characteristics of dynamogram and electric power data are extracted, and a working condition recognition model based on SVM Hessian regularization multi-view co-training algorithm is established. The method is applied to the identification of 11 typical working conditions of 60 pumping wells with rod pump in a block of Shengli Oilfield. In addition, Duan et al.[58] uses joint training to improve the original multi-modal recognition algorithm, and applies the improved algorithm and original method to Audio-Visual Person identification tasks. It shows that the improved algorithm based on co-training has greatly improved the classification accuracy and convergence of the original method.

The theory of co-training algorithm and related algorithms are gradually being improved, and their applications are becoming more and more extensive. For a better and more systematic understanding of the idea of co-training, it is of great significance to summarize and form a unified framework. At present, few scholars have completed this work.

5 | SUMMARY AND PROSPECT

In this article, the innovation and development of co-training algorithm in recent years and some potential problems in co-training algorithm are summarized, starting from three key steps of co-training algorithm. Whether it is single-view learning or multi-view learning, co-training aims at enabling machines to think from multiple perspectives like humans. Therefore, how to divide data views effectively, how to design learners scientifically, and how to accurately estimate the confidence of labels are the essential problems faced by co-training algorithms. Although many researchers have carried out a lot of experiments in the direction of co-training, there is still a long way to go before the real industrial application of co-training algorithm. To sum up, this article has the following suggestions for future research directions of co-training: (1) Many new innovations in co-training algorithms are still limited to specific datasets in the literature. However, the practical situation is much more complex than the experimental dataset and these innovations need some practical or theoretical verification to prove their effectiveness. (2) Both single-view and multi-view co-training algorithm cannot avoid training multiple models at the same time, which directly leads to huge cost of computational resources. Therefore, how to optimize the existing algorithm to reduce its computational cost is also a research direction. (3) The proportion of unlabeled data and labeled data used by the co-training algorithm needs to be further explored. (4) For the regression task in machine learning, because its output is continuous value, the label confidence is difficult to evaluate. Therefore, at present, co-training algorithm to solve the regression problem is less, need to be further studied. (5) Most experiments in literatures have not been carried out on a unified dataset, and there is a lack of a common dataset for the whole co-training family.

ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China (Grant No. 61901436), the National Natural Science Foundation of China (No. 61972040), and the Premium Funding Project for Academic Human Resources Development in Beijing Union University (No. BPHR2020AZ03).

DATA AVAILABILITY STATEMENT

Data sharing not applicable to this article as no datasets were generated or analyzed during the current study.

REFERENCES

1. De Bie T, Cristianini N. Semi-supervised learning using semi-definite programming; 2006.
2. Shahshahani BM, Landgrebe DA. The effect of unlabeled samples in reducing the small sample size problem and mitigating the hughes phenomenon. *IEEE Trans Geosci Remote Sens*. 1994;32(5):1087-1095.
3. HAN S, Han Q. Review of semi-supervised learning research. *Comput Eng Appl*. 2020;6:19-27.

4. Zhou Z. Disagreement-based semi-supervised learning. *Acta Automat Sin*. 2013;11:39.
5. Wan X. Co-training for cross-lingual sentiment classification. Paper presented at: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP, Suntec, Singapore; 2009:235-243.
6. Blum A, Mitchell T. Combining labeled and unlabeled data with co-training. Paper presented at: Proceedings of the Eleventh Annual Conference on Computational Learning Theory, Madison Wisconsin; 1998:92-100.
7. Nigam K, Ghani R. Analyzing the effectiveness and applicability of co-training. Paper presented at: Proceedings of the 9th International Conference on Information and Knowledge Managemen, Funchal, Madeira, Portugal; 2000:86-93.
8. Balcan M-F, Blum A, Yang K. Co-training and expansion: towards bridging theory and practice. *Advances in Neural Information Processing Systems*; 2005:89-96.
9. Goldman S, Zhou Y. Enhancing supervised learning with unlabeled data. *International Conference on Machine Learning*. 2000:327-334.
10. Zhou Z-H, Li M. Tri-training: exploiting unlabeled data using three classifiers. *IEEE Trans Knowl Data Eng*. 2005;17(11):1529-1541.
11. Ma l, Wang X. Semi-supervised regression based on support vector machine co-training. *Comput Eng Appl*. 2011;3:177-180.
12. Qin H, Gong R, Liu X. Binary neural networks: a survey. *Pattern Recognit*. 2020;105:107281.
13. Wang J, Luo SW, Zeng XH. A random subspace method for co-training. Paper presented at: Proceedings of the 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), Hong Kong; 2008:195-200; IEEE.
14. Yaslan Y, Cataltepe Z. Co-training with relevant random subspaces. *Neurocomputing*. 2010;73(10-12):1652-1661.
15. Feger F, Koprinska I. Co-training using rbf nets and different feature splits. Paper presented at: Proceedings of the IEEE International Joint Conference on Neural Network Proceedings, Baltimore, MD; 2006:1878-1885.
16. Huanling T, Zhengkui L, Mingyu L, Jun W. An advanced co-training algorithm based on mutual independence and diversity measures. *J Comput Res Develop*. 2008;45(11):1874-1881.
17. Sheng X, Yue X. Novel co-training algorithm based on rough sets. *Appl Res Comput*. 2013;12:3546-3550.
18. Chen M, Weinberger KQ, Chen Y. Automatic feature decomposition for single view co-training. Paper presented at: Proceedings of the International Conference on Machine Learning, Bellevue, Washington; 2011.
19. Chen M, Weinberger KQ, Blitzer J. Co-Training for Domain Adaptation. *Advances in Neural Information Processing Systems 24*. 2011;24:2456-2464.
20. Yang L, Wang Y, Gao M, et al. Mico: mixup co-training for semi-supervised domain adaptation; 2020. arXiv preprint arXiv:2007.12684.
21. Zhang d, Chen Z, Liang Y, Wu Z, Zhu J, Zhong t. Application of co-training algorithm in noninvasive blood glucose detection. *Chinese J Med Phys*. 2018;35(11):61-66.
22. Motta D, Santos AÁ, Machado BA, et al. Optimization of convolutional neural network hyperparameters for automatic classification of adult mosquitoes. *Plos One*. 2020;15(7):e0234959.
23. Bhalgat Y, Liu Z, Gundecha P, Mahmud J, Misra A. Teacher-student learning paradigm for tri-training: an efficient method for unlabeled data exploitation; 2019. arXiv preprint arXiv:1909.11233.
24. Ju Y, Li L, Jiao L, Ren Z, Hou B, Yang S. Modified diversity of class probability estimation co-training for hyperspectral image classification; 2018. arXiv preprint arXiv:1809.01436.
25. Xu J, He H, Man H. Dcpe co-training for classification. *Neurocomputing*. 2012;86:75-85.
26. Zhou Z-H, Li M. Semi-supervised regression with co-training. *IJCAI*. 2005;5:908-913.
27. Kuncheva LI, Whitaker CJ. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Mach Learn*. 2003;51(2):181-207.
28. c. Tang, W.Wang, W. Li, G. Li, and F. Cao, "Multi-learner co-training model for human action recognition," *J Softw*, vol. 26, no. 011, pp. 2939–2950, 2015.
29. Qiao S, Shen W, Zhang Z, Wang B, Yuille A. Deep co-training for semi-supervised image recognition. Paper presented at: Proceedings of the European Conference on Computer Vision (ECCV), Munich Germany; 2018:135-152.
30. Goodfellow IJ, Shlens J, Szegedy C. Explaining and harnessing adversarial examples; 2014. arXiv preprint arXiv:1412.6572.
31. Xie C, Wang J, Zhang Z, Zhou Y, Xie L, Yuille A. Adversarial examples for semantic segmentation and object detection. Paper presented at: Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy; 2017:1369-1378.
32. Peng J, Estrada G, Pedersoli M, Desrosiers C. Deep co-training for semi-supervised image segmentation. *Pattern Recogn*. 2020;107:107269.
33. Breiman L. Random forests. *Mach Learn*. 2001;45(1):5-32.
34. Quinlan JR. Induction of decision trees. *Mach Learn*. 1986;1(1):81-106.
35. Wu Y. Research on semi-supervised learning based on collaborative training. *Modern Computers: Late Last Month*; Guangzhou: Sun Yat-sen University; 2012.
36. Li M, Zhou Z-H. Improve computer-aided diagnosis with machine learning techniques using undiagnosed samples. *IEEE Trans Syst Man Cybern Part A Syst Humans*. 2007;37(6):1088-1098.
37. WANG D, LI Y, Xiao Z. Named entity recognition based on tri-training of multiple neural network. *Intell Comput Appl*. 2020;10(2):123-127.
38. Ge M, Yu C, Zhou L, Ma Y. Deep learning image classification algorithm based on semi-supervised collaboration training. *Comput Simulat*. 2019;36(02):206-210.
39. Dietterich TG. *Ensemble Learning: The Handbook of Brain Theory and Neural Networks*. Vol 2. 55 Hayward St.CambridgeMAUnited States; MIT Press; 2002:110-125.
40. Chen Z, Ahn H. Item response theory based ensemble in machine learning. *Int J Autom Comput*. 2020;17:621-636.
41. Mingers J. An empirical comparison of pruning methods for decision tree induction. *Mach Learn*. 1989;4(2):227-243.
42. Lan X. The research on semi-supervised collaboration-training algorithm [Ph.D. dissertation]. Sichuan Normal University; 2011.
43. Zhou Y, Goldman S. Democratic co-learning. Paper presented at: Proceedings of the 16th IEEE International Conference on Tools with Artificial Intelligence, Boca Raton, FL; 2004:594-602.
44. Gong Y, Lv J. Co-training method combined with semi-supervised clustering and weighted k -nearest neighbor. *Computer Engineering and Applications*. 2019;55(22):114–118.
45. Yin X, Shu T, Huang Q. Semi-supervised fuzzy clustering with metric learning and entropy regularization. *Knowl Based Syst*. 2012;35:304-311.
46. X. Guo, W. Wang, "An improved co-training style algorithm: compatible co-training," *J Nanjing Univ (Natural Sci)*, vol. 52, 4, p. 662-671, Nanjing: Nanjing University; 2016.

47. Jiang Z, Jing C, Zan Y. Research on action recognition algorithm based on hybrid cooperative. *Computer Science*. 2017;44(7):275–278.
48. Zhang M-L, Zhou Z-H. Cotrade: confident co-training with data editing. *IEEE Trans Syst Man Cybern B Cybern*. 2011;41(6):1612-1626.
49. Gong Y. Co-training algorithm with combination of active learning and density peak clustering. *J Comput Appl*. 2019;39(8):2297. http://www.joca.cn/CN/abstract/article_23109.shtml.
50. Liu W, Qin X, Wei G. Service identification of wechat traffic based on fuzziness and semi-supervised self-paced co-trainging. *J Univ Sci Technol China*. 2020;1:29-38.
51. Wu J, Li L, Wang WY. Reinforced co-training; 2018. arXiv preprint arXiv:1804.06035.
52. Tseng CM, Huang TW, Liu TJ, Data labeling with novel decision module of tri-training. Paper presented at: Proceedings of the 2020 2nd International Conference on Computer Communication and the Internet (ICCCI), Nagoya, Japan: IEEE; 2020:82-87.
53. Xia Y, Liu F, Yang D, et al. 3d semi-supervised learning with uncertainty-aware multi-view co-training. Paper presented at: Proceedings of the IEEE Winter Conference on Applications of Computer Vision, The Westin Snowmass Resort in Snowmass Village, Colorado; 2020:3646-3655.
54. Du L, Wang Y, Xie W. A semi-supervised method for SAR target discrimination based on co-training. Paper presented at: Proceedings of the IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium; 2019:9482-9485.
55. Li Z, Lin L, Zhang C, Ma H, Zhao W. Automatic image annotation based on co-training. Paper presented at: Proceedings of the 2019 International Joint Conference on Neural Networks (IJCNN), Budapest, Hungary; 2019:1-8.
56. Abdelgayed TS, Morsi WG, Sidhu TS. Fault detection and classification based on co-training of semisupervised machine learning. *IEEE Trans Ind Electron*. 2017;65(2):1595-1605.
57. Zhou B, Wang Y, Liu W, Liu B. Identification of working condition from sucker-rod pumping wells based on multi-view co-training and hessian regularization of SVM. Paper presented at: Proceedings of the 2018 14th IEEE International Conference on Signal Processing (ICSP), The InterContinental Budapest Hotel in Budapest, Hungary; 2018:969-973.
58. Duan X, Thomsen NB, Tan ZH, Lindberg B, Jensen SH Weighted score based fast converging CO-training with application to audio-visual person identification. Paper presented at: Proceedings of the 2017 IEEE 29th International Conference on Tools with Artificial Intelligence (ICTAI), Boston, MA; 2017:610-617.

How to cite this article: Ning X, Wang X, Xu S, et al. A review of research on co-training. *Concurrency Computat Pract Exper*. 2021;e6276. https://doi.org/10.1002/cpe.6276