electronics

MDPI

Article

An Accelerator for Semi-Supervised Classification with Granulation Selection

Yunsheng Song [1,2], Jing Zhang [1,*], Xinyue Zhao [1] and Jie Wang [3]

[1] School of Information Science and Engineering, Shandong Agricultural University, Tai'an 271018, China; sys_sd@126.com (Y.S.); zhaoxinyue1006@163.com (X.Z.)
[2] Key Laboratory of Huang-Huai-Hai Smart Agricultural Technology, Ministry of Agriculture and Rural Affairs, Shandong Agricultural University, Tai'an 271018, China
[3] School of Information, Shanxi University of Finance and Economics, Taiyuan 030006, China; 20191031@sxufe.edu.cn
* Correspondence: zhangjing_12138@163.com

Abstract: Semi-supervised classification is one of the core methods to deal with incomplete tag information without manual intervention, which has been widely used in various real problems for its excellent performance. However, the existing algorithms need to store all the unlabeled instances and repeatedly use them in the process of iteration. Thus, the large population size may result in slow execution speed and large memory requirements. Many efforts have been devoted to solving this problem, but mainly focused on supervised classification. Now, we propose an approach to decrease the size of the unlabeled instance set for semi-supervised classification algorithms. In this algorithm, we first divide the unlabeled instance set into several subsets with the information granulation mechanism, then sort the divided subsets according to the contribution to the classifier. Following this order, the subsets that take great classification performance are saved. The proposed algorithm is compared with the state-of-the-art algorithms on 12 real datasets, and experiment results show it could get a similar prediction ability but have the lowest instance storage ratio.

Keywords: semi-supervised classification; co-training method; instance selection; granular computing; information granulation

Citation: Song, Y.; Zhang, J.; Zhao, X.; Wang, J. An Accelerator for Semi-Supervised Classification with Granulation Selection. *Electronics* **2023**, *12*, 2239. https://doi.org/10.3390/electronics12102239

Academic Editors: Chao Zhang, Wentao Li, Huiyan Zhang and Tao Zhan

Received: 16 April 2023
Revised: 11 May 2023
Accepted: 12 May 2023
Published: 15 May 2023

1. Introduction

Co-training is a semi-supervised learning technique in which two classifiers are trained on separate, complementary views of the same data, with the idea that the two views contain different but complementary information [1–5]. In the context of co-training, a view refers to a different representation of the same data. For example, if the data is a set of documents, one view could be the text of the documents, while the other view could be the meta-data associated with the documents, such as the author or the date of publication. The basic idea behind co-training is that the two classifiers learn from each other and the labeled data so that they become more accurate over time. In each iteration of co-training, the classifiers make predictions on the unlabeled data, and the most confident predictions are used to label more data. This newly labeled data is then used to retrain both classifiers, and the process repeats. Research on co-training has shown that it can be an effective technique for semi-supervised learning, especially in domains where the two views of the data are indeed complementary.

It is also worth mentioning that co-training has been applied to a variety of domains, including natural language processing (NLP), computer vision, and bioinformatics. In the field of NLP, co-training has been used to improve the performance of sentiment analysis, text classification, and topic modeling [6]. In computer vision, co-training has been used for image classification and object detection [7]. In bioinformatics, co-training has been used for protein function prediction and gene expression analysis [8]. One of the strengths of

co-training is its ability to handle large and complex datasets, where traditional supervised learning methods may struggle. For instance, in NLP, co-training has been shown to be effective when dealing with imbalanced datasets, where the number of positive instances is much smaller than the number of negative instances. In such scenarios, co-training can effectively leverage the information contained in the unlabeled data to improve the performance of the classifier. Another area of application for co-training is in data privacy, where it is often the case that only a limited amount of labeled data is available for training machine learning models. In these scenarios, co-training can effectively leverage the information contained in the unlabeled data to improve the performance of the classifier, without compromising privacy or security [9].

In recent years, several variations and extensions of co-training have been proposed to address its limitations and improve its performance. For instance, some researchers have proposed using multiple views of the data rather than just two to capture more information and make the semi-supervised learning process more robust [10]. Another line of research has focused on developing new co-training algorithms that are able to handle noisy or conflicting views of the data [11]. These algorithms aim to identify and discard unreliable predictions made by one of the classifiers so that the other classifier can make better predictions in the absence of high-quality supervision. Additionally, there has been a growing interest in using deep learning models for co-training. For instance, one approach is to use generative models, such as Generative Adversarial Networks (GANs), to generate synthetic samples that can be used to augment the labeled data [12]. By using these synthetic samples in co-training, it is possible to effectively increase the size of the labeled data, leading to improved performance. Meanwhile, co-training can handle high-dimensional, complex data representations with deep learning models. For instance, some researchers have proposed using deep neural networks as the classifiers in co-training and have shown that this can lead to improved performance in various applications, including image classification, sentiment analysis, and document classification [13]. Overall, the field of co-training and semi-supervised learning is rapidly evolving, and there is a wealth of ongoing research aimed at improving the performance and robustness of these algorithms. As such, it is an exciting and promising area of study for anyone interested in machine learning and data science.

Although co-training plays an important role in the semi-supervised classification task, large-scale data poses a huge challenge to the efficiency of its modeling [14]. Existing co-training-based semi-supervised classification algorithms usually need to traverse all unlabeled samples multiple times to find high-confidence elements or valuable classification information, but large-scale unlabeled instances make it difficult to achieve efficient modeling. Some literature proposes using different subsets of unlabeled samples after division to improve the efficiency of the algorithm; it does not consider the differences in the contribution of different unlabeled samples to the algorithm. However, it takes a great challenge for traditional semi-supervised classification algorithms based on co-training to handle large-scale data in terms of compatibility, effectiveness, and timeliness. Instance selection as an important data reduction method can solve the large-scale classification problem by reducing the labeled instances depending on enough label information to achieve the aim [15,16]. Therefore, traditional instance selection methods cannot be applied to the semi-supervised classification problem because there exists a small number of labeled instances with little labeled information. Moreover, each instance is seen as a basic processing unit to judge whether it is selected or not [17]. It is difficult to follow this approach to dealing with large-scale unlabeled instances, and there is a need to solve this problem from a new perspective.

Granular computing is a methodology for processing and analyzing complex data by partitioning it into smaller, more manageable pieces [18–22]. These smaller pieces, or granules, can then be further analyzed and processed to provide insights into the original data. The goal of granular computing is to simplify complex problems by reducing their complexity to more manageable pieces. This approach has been applied to a variety of

problems in machine learning, including clustering, classification, and feature selection. Meanwhile, granular computing and co-training are both techniques that can be used to improve classification accuracy. Granular computing can be used to reduce the complexity of the data by partitioning it into smaller, more manageable pieces [23]. Once the data has been partitioned into granules, co-training can be used to train multiple models on each granule. This approach can be particularly effective in semi-supervised learning applications where labeled data is limited. Nevertheless, the contribution of each kind of information granularity with a large difference to the classifier has not received sufficient attention, so its efficiency has dropped dramatically, and information could be redundant [24,25].

For the problem based on the above analysis, this paper has proposed an effective instance selection for a co-training-based semi-supervised classification task using the granulation mechanism, which deals with the large data using the information particles as the basic processing unit rather than each instance and considers the different contribution of granularity to the classifier. The contribution of this paper is as follows:

- Proposing a progressive instance selection mechanism to reduce unlabeled instances by the significant variation in classification accuracy.
- Giving a novel unlabeled information granulation mechanism based on the extent to which the unlabeled instance improves the performance of the classifier, and it avoids the influence of human subjective factors.
- Adaptive determining in which unlabeled information granulation is ultimately saved according to its contribution to the classification performance.
- Verifying the proposed method could largely reduce the unlabeled data size and keep the original classification performance by the experiment result on the real datasets.

The rest of this paper is listed as follows. Section 2 introduces related work about co-training-based semi-supervised classification algorithms. Section 3 analyzes the effect of unlabeled instances on the classifier and has proposed an effective instance selection for co-training-based semi-supervised classification. Section 4 verifies the effectiveness of the proposed method. Section 5 concludes this paper.

2. Related Work

A co-training-based semi-supervised classification algorithm needs to cooperate with different classifiers from multiple perspectives at the same time to realize the utilization of unlabeled data, and it has become the focus of research with its higher effectiveness [3,26]. According to the different learning strategies, the existing co-training algorithms are mainly divided into two categories: the ones based on the sample set augmentation and the ones based on regularization.

Co-training algorithms based on sample set augmentation, which use classifiers from different perspectives to select high-confidence unlabeled samples and corresponding prediction labels from the unlabeled sample set, alternately assign the newly added samples to different classifiers for retraining and finally repeat the above process until the prediction results converge. In such algorithms, how to efficiently select labeled samples with high confidence is the bottleneck that restricts the efficiency of the algorithm. Paper [27] divides the sample space into a set of equivalence classes and uses cross-validation to determine how to label unlabeled samples. Paper [28] uses voting to select unlabeled samples with high confidence; In order to improve the robustness of the collaborative training algorithm, and papers [29,30] use filtering to screen the newly added unlabeled samples instead of using them all [31].

Co-training algorithms based on regularization use the information provided by different perspective classifiers as the regularization term of the learning object, and transform the semi-supervised multi-view learning problem into an optimization problem [32,33]. In order to improve the training efficiency of such algorithms, Sun et al. [34] propose a sequential training method that uses the union of different unlabeled sample subsets and labeled sample set L on the basis of dividing the unlabeled sample set into ten subsets

of equal size, the union of the first unlabeled sample subset and set L is first used for modeling, and then the next unlabeled sample set, some elements of the utilized unlabeled sample set participate in the modeling, and the union modeling of set L. Finally, repeat the previous step until all unlabeled sample subsets are utilized. Existing difference-based semi-supervised classification algorithms need to traverse all unlabeled samples multiple times to find high-confidence elements or valuable classification information, but the massive scale of unlabeled data makes it difficult to achieve efficient modeling. Although some literature proposes to use different subsets of unlabeled samples after division to improve the efficiency of the algorithm, it does not consider the differences in the contribution of different unlabeled samples to the algorithm.

In conclusion, the existing large-scale co-training-based supervised classification algorithms mainly improve the training efficiency from the view of optimization design. However, the time complexion is difficult to reduce and obtain a greater improvement for the large problem, and it still suffers from the large training burden of using the whole of the unlabeled instances to participate in the training process.

3. Main Content

For the given training set T, which is the union of the labeled instance set $L = \{x_1, x_2, \cdots, x_l\}$ and the unlabeled instance set $U = \{x_{l+1}, x_{l+2}, \cdots, x_{l+u}\}$, where x_i denotes the training instance, l and u are the number of labeled instances and unlabeled instances, and $i = 1, 2, \cdots, l + u$. Semi-supervised classification algorithms simultaneously use the labeled instance set L, and unlabeled instance set U to train a classifier $f(x)$ with good performance. A co-training-based semi-supervised classification algorithm uses the idea of compatible complementarity of multiple views to learn the final classifier. It assumes that the data has multiple sufficient and conditional independence views, and the classifier trained on one view can offer supplemental information to the classifiers on the other view. The supplemental information is achieved by selecting the most trusted unlabeled instances and pseudo-labels. Nevertheless, several iterations are required, and each iteration must scan the whole of the unlabeled instances set to the most trusted instances. The large-scale unlabeled instance carries difficulty in efficiently learning the final classifier.

Instance selection, as one of the most important data preprocessing technologies to reduce dataset size, is widely used for classification problems, as is the fact that the contribution of training instances with the different locations in the space to learn a classifier varies greatly. Numerous studies have shown that the training instances can be divided into critical instances and non-critical instances, where critical instances mainly define the class boundary and separate the instances of the same label from the ones from other labels [16]. Meanwhile, the number of critical instances is significantly smaller than that of non-critical instances in most real-world datasets. Therefore, the process requires an effective way to reduce the training set to a relatively small subset by selecting critical instances and preserving the original data information. Compared with the performance on the original training set, the efficiency of training the classifier on the reduced set can be significantly improved on the reduced subset.

Traditional instance selection methods for supervised classification tasks start with each instance as the most basic processing unit, critical instances are selected by the contribution of each labeled instance to the classifier. The contribution of an instance to the classifier is usually measured by its location in the input space and the difference with its nearest neighbors on the label. Although the instance selection is very efficient for supervised classification tasks, it is difficult to apply directly to semi-supervised classification tasks because of its limitations. Different from supervised classification, there exists a large number of unlabeled instances and few labeled instances for the semi-supervised classification tasks. Only labeled instances take labeled information to the learner, and this information is vital to learn a classifier with good performance, so it cannot reduce labeled instances. Otherwise, the generation ability of the obtained classifier could significantly decrease. On other hand, traditional instance selection needs the labeled information of

each instance to execute data reduction, while this condition is not met for the unlabeled instances. Moreover, the way of treating each instance as the basic process unit is undesirable for large-scale problems because it is very time-consuming.

To overcome this difficulty, we have proposed a novel instance selection with the granulation mechanism. This proposed method consists of two key processes: unlabeled information granulation and granulation selection.

3.1. Unlabeled Information Granulation

Unlabeled instance selection aims to reduce the unlabeled instance set U, and the original unlabeled information remains relatively unchanged. Unlabeled information is expressed by the contribution of unlabeled instances to learn the classifier for semi-supervised learning, and it is closely related to the feature of semi-supervised classification algorithms. For co-training-based semi-supervised classification algorithms, the contribution of the unlabeled instance to learn the final classifier mainly depends on the determination of predictive label consistency of the classifiers trained with different views, as well as its location in the input space. The unlabeled instances located in different regions in the input space have different contributions to the classifier [35]. Figure 1 shows a 2-dimension binary classification problem to learn a linear classifier, where two labeled instances from different classes are represented by blue circle and yellow circle. The unlabeled instances nearby the decision boundary have much more of a contribution than the ones far from the decision boundary.

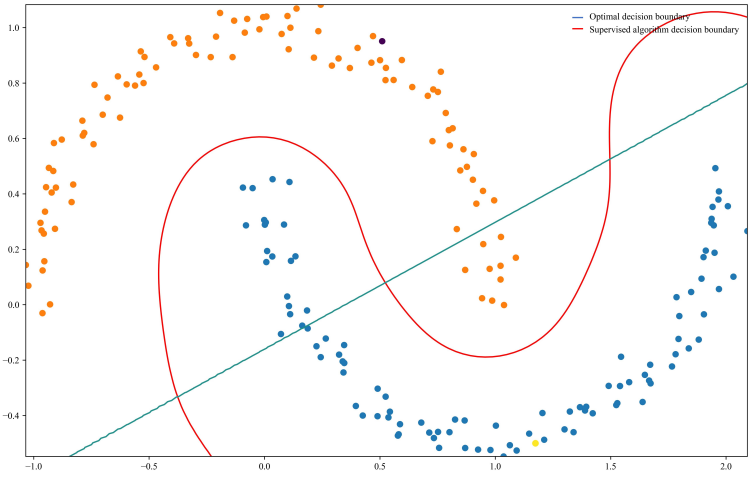

Figure 1. An example of a binary semi-supervised classification problem.

This difference in the contribution of unlabeled instances to the classifier yields the possibility of executing an instance selection. Compared with the abundant labeled information of the labeled instances, unlabeled instances bring a limited classification contribution to the learner. Due to the presence of a large number of unlabeled instances with limited classification information, it is difficult to select critical unlabeled instances with their contribution one by one. Furthermore, semi-supervised classification should not reduce unlabeled instances one by one from an execution efficiency perspective. This process is a disaster, especially for classification algorithms with high time complexity. Therefore, we adopt the idea of granular computing to divide the unlabeled instance set U into m disjoint subsets U_i, $U = \bigcup \sum_{i=1}^{m} U_i$. All the instances in the same subset U_i are considered as a basic information granularity to participate in the learning process. In this way, it can greatly improve learning efficiency by only processing a small number of units. Meanwhile, the contribution of a subset U_i is easy to obtain compared with the single unlabeled instance.

Data partition, as one of the important data granulation techniques, plays a crucial role in granular computing. There are three key factors to performing data partition for the co-training-based semi-supervised classification tasks.

- Divided unlabeled instance subsets have the unbalanced information for the final classifier to obtain a relatively small number of aim subsets to achieve data reduction.
- Number of divided subsets is determined by the characteristics of unlabeled instances and the contribution to the classifier rather than the subjective prior determination.
- Data partition should use the contribution of the unlabeled instances to the semi-supervised classifier and close together with the distinguishing feature of co-training.

Therefore, we utilize a similar framework as a Tri-Training [36] method to perform data partition. Each initial decision tree classifier $f_r(x)$ is independently trained on the different set B_r sampled from the labeled set L using Bootstrap sampling method, where r = 1, 2, 3. Owning to the feature of Bootstrap sampling, the sample set B_r has a large difference, as well as the classifier $f_r(x)$ on it. Then it iteratively retrains each classifier with the enlarged labeled set L, which is created by introducing several confident unlabeled instances and their pseudo-label until none of the classifiers changes. The confident unlabeled instance and its pseudo-label obtained by each classifier are provided by the remaining two classifiers. Specifically, if the two classifiers have the same prediction for the same unlabeled instances, these instances are considered to have high labeling confidence and are added to the labeled training set of the third classifier. In this way, we can estimate the frequency $fre(x_i)$ at which each unlabeled instance x_i is selected as a confident element during this process. Finally, the unlabeled set U is divided into several subsets according to the condition that all the unlabeled instances x_i in the same subset have the same frequency. The pseudocode of the proposed method is presented in Algorithm 1.

A decision tree (DT) is chosen as the basic classifier for the Tri-training algorithm for its unique advantages in Algorithm 1, that is, learning features, high efficiency, and instability. Both the conditional probability distribution information for the class and the local geometry information in the input space are used to learn the classifier of DT, and this kind of information is very comprehensive. Furthermore, the time complexion of DT is approximately linear of time complexion to efficiently process large-scale data. Finally, the instability of DT is sensitive to data change for its instability, this is constructive to instance selection [37].

The measurement $fre(x_i)$ is the frequency at which each unlabeled instance x_i is selected as the confident instance for three classifiers in the whole training process, and it reflects each unlabeled instance x_i to learn the final classifier. The large value of frequency $fre(x_i)$ means the unlabeled instance x_i is always chosen and has a large contribution to the final classifier. A different value of $fre(x_i)$ also indicates different degrees of effect on classification performance. Different from previous methods to evaluate the contribution with a real number, the measurement metrics takes a limited integer value. It is constructive to divide the unlabeled set U into several subsets according to the possible value of the measurement $fre(x_i)$. Moreover, the number $n = \max_{x_i \in U} fre(x_i) - \min_{x_i \in U} fre(x_i)$ of discrete values of the measurement $fre(x_i)$ is not subjectively predetermined; it depends on the effect of unlabeled instances on the classifier.

Algorithm 1 Unlabeled information granulation algorithm

Input : Training set $T = L \bigcup U$, where the labeled set $L = \{x_1, x_2, \cdots, x_l\}$ and the unlabeled instance set $U = \{x_{l+1}, x_{l+2}, \cdots, x_{l+u}\}$, decision-tree classification algorithm.

Output: The divided unlabeled instance subsets U_h, $U = \bigcup_{h=1}^{n} U_h$.

Initialization : $fre(x_i) = 0$ for $x_i \in U$, $L_1 = L_2 = L_3 = L$, $Update = 1$;

foreach $r \in \{1, 2, 3\}$ **do**

  Train the decision tree classifier $f_r(x)$ on the set B_r sampled from the set L using Bootstrap sampling method;

end

while *Update= 1* **do**

  foreach $r \in \{1, 2, 3\}$ **do**

    $Update_r = 0$;

    foreach $x_i \in U$ **do**

      if $f_j(x_i) = f_k(x_i)$ $(j, k \neq r)$ **then**

        $fre(x_i) = fre(x_i) + 1$, $L_r = L_r \bigcup \{(x_i, f_j(x_i))\}$, $Update_r = 1$;

      end

    end

  end

  foreach $r \in \{1, 2, 3\}$ **do**

    if $Update_r = 1$ **then**

      Re-train the decision tree classifier $f_r(x)$ on the new set L_r, $Update = 1$;

    end

  end

end

$n = Maxf - Minf + 1$, where $Maxf = \max_{x_i \in U} fre(x_i)$ and $Minf = \min_{x_i \in U} fre(x_i)$;

foreach $h \in \{1, 2, \cdots, n\}$ **do**

  $U_h = \{x_i \in U : fre(x_i) = h - 1 + Minf\}$;

end

Return *The divided unlabeled instance subset* U_h;

3.2. Granulation Selection

After the unlabeled data granulation with data partition, divided subsets of unlabeled instances must be finally chosen to train the semi-supervised classifier. It is undesirable to randomly select several divided subsets as the final result for different contributions to the classifier. According to the same value of $f(x_i)$ of the instances $x_i \in U_h$, the order of contribution from smallest to largest is $U_1, U_2, \cdots, U_m$, where m is the number of the divided unlabeled instance subsets. To keep the full information of the unlabeled set U with the smaller number of divided unlabeled instance subsets as much as possible, we adopt the way of one subset by one subset in reverse order. In this way, it is constructive to search the smaller number of subsets with much more auxiliary classification information to the learner in the following process. Moreover, this method relies solely on the value $f(x_i)$ over the divided subset to select results without any limitation to the classifier.

Let $Acc(U_g)$ be the measurement of classification performance for the classifier trained on the set U_h and the labeled set L, the change on the classification performance $\Delta(U_{g-1}) = Acc(U_g \bigcup U_{g-1}) - Acc(U_{g-1})$ between U_g and U_{g-1}, $g = 2, 3, \cdots, m$. $Acc(U_g)$ evaluates the effect of the set U_h to the semi-supervised learner classification performance, and $\Delta(U_g - 1)$ indicates the effect of adding set U_{g-1} to set U_g. If the value of $\Delta(U_{g-1})$ is small relative to $Acc(U_g)$, then merging set U_{g-1} with set U_g is difficult to train a strong semi-supervised classifier. Therefore, the following condition (1) is set to judge whether it merges set U_{g-1} with set U_g or not.

$$\frac{Acc(U_g \bigcup U_{g-1}) - Acc(U_{g-1})}{Acc(U_g)} \geq \alpha, \tag{1}$$

where $\alpha \in (0,1)$, $g = 2,3,4,\cdots,m$. Many papers suggest that the critical value $\alpha = 0.05$ to obtain a significant change in performance [38]. Above all, the pseudocode of the granulation selection is presented in Algorithm 2.

In Algorithm 2, the early stopping condition is used to prevent performing too many iterations. The classifier trained on set $L \bigcup U_g \bigcup U_{g-1}$ may improve the classification performance of the one on the set $L \bigcup U_g$ because it adds more unlabeled sample information from the set U_{g-1}. Moreover, the unsupervised information of the set U_g that is constructive to improve the classification performance could be more than the set U_{g-1}, where $g = 2,3,\cdots,m$. Therefore, the subset U_j is difficult to satisfy condition (1) if the previous subset U_i cannot meet, where $1 \leq j < i \leq m$. In this way, this selection process can be terminated early and obtain a lower number of divided unlabeled instance subsets.

Algorithm 2 Granulation selection algorithm

Input : Training set $T = L \bigcup U$, where the labeled set $L = \{x_1, x_2, \cdots, x_l\}$ and the unlabeled instance set $U = \{x_{l+1}, x_{l+2}, \cdots, x_{l+u}\}$, decision-tree classification algorithm, and semi-supervised classification algorithm f, critical value $\alpha = 0.05$.

Output: The reduced set U_s of the set U.

Run Algorithm 1 with decision tree to get m divided subsets U_h, $U = \bigcup_{h=1}^{m} U_h$;
Train semi-supervised classifier with the set U_m to obtain Acc^*, $U_s = U_m$;
foreach $g \in \{m-1, \cdots, 2\}$ **do**
    $\tilde{U}_g = U_s \bigcup U_g$;
    Train semi-supervised classifier with the set $\tilde{U}_g$ to obtain $Acc(\tilde{U}_g)$;
    if $\frac{Acc(\tilde{U}_g) - Acc^*}{Acc^*} \geq \alpha$ **then**
        $U_s = \tilde{U}_g$;
        $Acc^* = Acc(\tilde{U}_g)$;
    else
        Get out of the loop
    end
end
Return *The set* U_s;

3.3. Complexity Analysis

Complexity analysis is very important for evaluating the classifier, and it starts with two main steps of the proposed method. The first step includes the frequency in which an unlabeled instance is selected as a trust element and the frequency discretization, where the former mainly depends on the time complexion of the basic classifier and the latter is linear with the number of unlabeled instances. A decision tree is selected as the basic classifier in this method of the approximately linear time complexity $O(dml\log(l))$ with the size l of labeled instances and m features. Meanwhile, the efficiency of a decision tree is very high to predict the label for the unlabeled instances with the time complexity $O(d)$, where d is the depth of the tree. Thus, it can efficiently process big data, which has massive unlabeled data, to offer the pseudo-labels of the linear time complexion that is linearly related to the size of the data. Therefore, the time complexity of the first step is $O(dml\log(l) + du)$. The time complexity of the second step relies on the complexity of the adopted semi-supervised algorithm and the number of iterations n. In each iteration, the semi-supervised classifier is trained using the subset of unlabeled set U rather than all the instances, and its time complexity is proportional to the size of the labeled instance subset. Meanwhile, the early

stopping condition is constructive in reducing the number of iterations. In conclusion, the time complexity of the proposed method is approximately linear with the number and the dimension of labeled instances and unlabeled instances, and it is proportional to the time complexity of the adopted semi-supervised classifier that is used to get the classification accuracy.

4. Experiments

To verify the effectiveness and efficiency of the proposed algorithm for real problems, extensive experiments on real datasets have been implemented against the typical method under differently labeled rations.

4.1. Experiment Setup

Twelve large datasets of different types are randomly selected to evaluate the performance of the algorithms from the KEEL-dataset repository [31] and LIBSVM-dataset repository [39], where each data has larger than 4000 instances. The basic information of the selected datasets is listed in Table 1.

Table 1. Summary of twelve datasets.

Dataset	Size	Features	Classes
combined	98,528	101	3
connect-4	67,557	126	3
covtype	581,012	54	7
letter	20,000	16	26
optdigits	5620	65	10
pendigits	10,992	17	10
phoneme	5404	5	2
ring	7400	21	2
seismic	98,528	51	3
texture	5500	41	11
usps	9298	257	10
winequality	4898	11	7

Further, a typical image dataset of five generic categories called NORB [40] is selected to test the performance of the proposed method for high-dimensional datasets. The following Figure 2 shows examples of the training image and test image of the dataset.

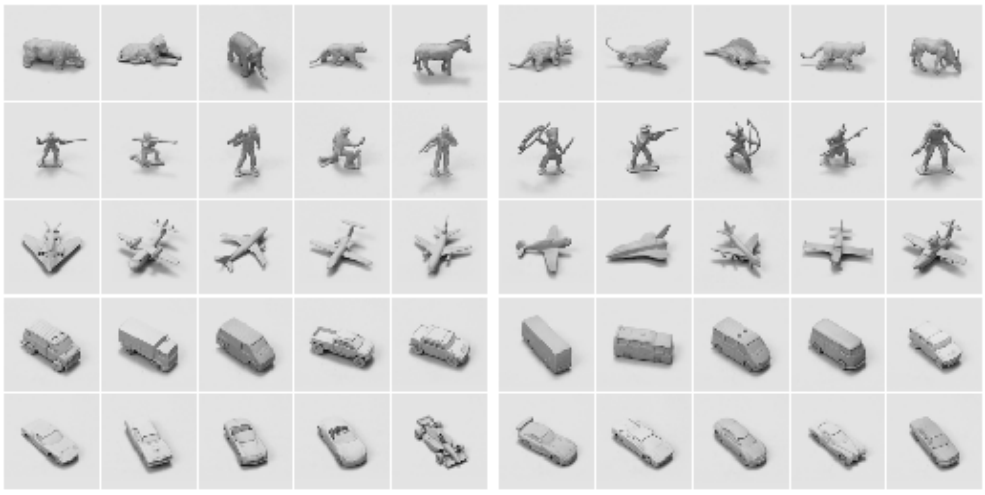

Figure 2. The examples of NORB dataset.

For each dataset, about 3/4 of the data is selected as the training set and the rest as the test set, where each training set is the union of the labeled subset L and the unlabeled

subset U. To effectively verify the generalization performance of the proposed algorithm for real data, the proportion of labeled instances (PL) to the total training instances has different values. According to the suggestion of the paper [36], the value of PL includes 20%, 40%, 60%, and 80%.

The most commonly used method for evaluating semi-supervised classification algorithms is the performance measurement of algorithms on the datasets. Classification accuracy (Acc) and Cohen's kappa (Kappa) [41] on the test set are used to measure the generation ability of algorithms, and executing time (ET) in seconds of learning the classifier is applied to estimate training efficiency. Besides the above two performance indicators, the number of the selected unlabeled instances to learn the semi-supervised classifier is also an important measure of the performance of instance selection. Therefore, the proportion of the selected unlabeled instances (PS) to the total unlabeled instances is adopted to eliminate the impact of dataset size.

Tri-training (Tri) [36] is selected as the representative co-training semi-supervised classification algorithm for its good performance, and the improved Tri-training algorithm trained on the result based on the proposed instance selection is denoted as ISTri. To verify whether there is a significant difference in the performance between Tri and ISTri on different data, the Wilcoxon signed rank test [42] is selected for its weaker data distribution assumptions and good statistical performance on the real datasets, where the null hypothesis is that the proposed algorithm performance is significantly different from each of the other algorithms on the same multiple datasets. The p-value of the test is computed to judge whether the null hypothesis is rejected or not under the given significant level α. If the p-value is larger than α, then the null hypothesis should be accepted. Otherwise, the null hypothesis is rejected, and there exists a significant difference between the proposed algorithm and another algorithm. The significant level $\alpha = 0.05$. All the experiment is executed in Python 3.10 on Windows 10 on a PC of Intel(R) Xeon(R) Silver 4280 CPU (2.10 GHz) and 160 GB RAM.

4.2. Experimental Analysis

This section concretely compares the performance of the proposed algorithm with the Tri algorithm from the perspective of classification performance, execution time, and proportion of the selected unlabeled instances on twelve medium-dimensional datasets, as well as a typical high-dimensional dataset. Furthermore, the effect of the proportion of the selected unlabeled instances on the algorithm's performance is also studied.

4.2.1. Classification Performance

Classification accuracy and Cohen's Kappa are two common measurements to evaluate the classification performance of the classifier, where the latter is an important complement to the former for the class-imbalance problem.

Table 2 lists the classification accuracy on the selected data sets, where the mean and median of classification accuracy on all the datasets are at the back of this table. Meanwhile, the p-value of the Wilcoxon signed rank test between the two algorithms is also listed in the last row of Table 2. Figure 3 intuitively shows the comparison of classification accuracy of the two algorithms on different datasets under different label proportions.

The following comparative analysis is done from a single perspective and a holistic perspective. It can be found that the ISTri algorithm obtains very similar classification accuracy with the Tri algorithm on each dataset under the same labeled rate PL = 0.2, 0.4, 0.6, and 0.8 from Figure 2. To compare the classification accuracy of different algorithms from a holistic perspective, descriptive statistical analysis is made. The means of the classification accuracy of the ISTri algorithm on all the datasets under different PLs are 0.851, 0.873, 0.889, and 0.894, and the ones of the Tri algorithm are 0.857, 0.874, 0.886, and 0.893. This numeric result also corroborates the absolute difference between the two algorithms on the mean of classification accuracy under the same PL value is very small. Meanwhile, the medians of the classification accuracy of the ISTri algorithm on all the datasets under different PLs

are 0.890, 0.920, 0.933, and 0.947, and the ones of the Tri algorithm are 0.896, 0.919, 0.930, and 0.942. Therefore, the absolute difference between the two algorithms on the median of classification under the same PL value is also very small.

Table 2. Classification accuracy of two algorithms on the selected datasets.

Data	PL = 0.2		PL = 0.4		PL = 0.6		PL = 0.8	
	Tri	**ISTri**	**Tri**	**ISTri**	**Tri**	**ISTri**	**Tri**	**ISTri**
combined	0.815	0.821	0.823	0.824	0.831	0.830	0.834	0.837
connect-4	0.792	0.801	0.823	0.822	0.829	0.837	0.845	0.848
covtype	0.896	0.894	0.923	0.924	0.938	0.939	0.947	0.948
letter	0.895	0.886	0.932	0.927	0.951	0.947	0.960	0.959
optdigits	0.967	0.951	0.977	0.975	0.984	0.984	0.980	0.974
pendigits	0.978	0.976	0.988	0.986	0.990	0.990	0.992	0.990
phoneme	0.859	0.845	0.871	0.877	0.898	0.903	0.900	0.905
ring	0.917	0.905	0.915	0.917	0.922	0.928	0.938	0.945
seismic	0.729	0.731	0.731	0.734	0.741	0.744	0.744	0.746
texture	0.934	0.918	0.966	0.961	0.971	0.973	0.972	0.973
usps	0.930	0.921	0.951	0.947	0.949	0.949	0.959	0.963
winequality-white	0.567	0.565	0.585	0.583	0.624	0.638	0.641	0.642
Mean	0.857	0.851	0.874	0.873	0.886	0.889	0.893	0.894
Median	0.896	0.890	0.919	0.920	0.930	0.933	0.942	0.947
p-value	0.064		0.519		0.058		0.129	

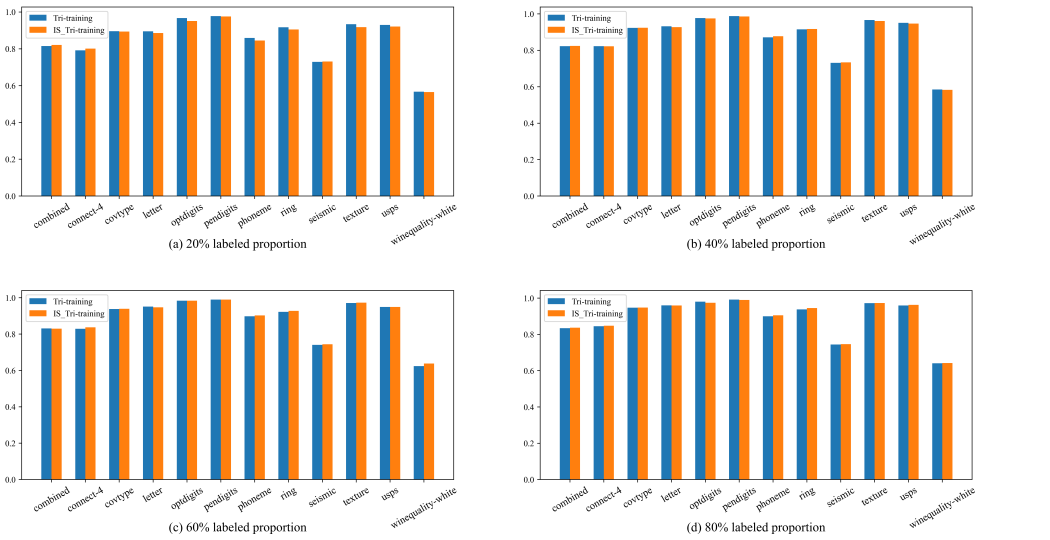

Figure 3. The comparison of classification accuracy between two algorithms on the selected datasets.

Finally, the Wilcoxon signed rank test between two algorithms classification accuracy is made to avoid the effect of the subjective judgment. *p*-values of this test under different PL are 0.060, 0.720, 0.375, and 0.206; these values are all larger than the given significant level of 0.05. Thus, there exists no significant difference in the classification accuracy between two algorithms.

Besides classification accuracy, Cohen's kappa is also used to evaluate the classification performance of the learner, which can solve the problem that accuracy does not compensate for hits that can be attributed to mere chance. Similar to the result of Table 2, Table 3 lists Kappa of two algorithms under different labeled proportions, as well as the descriptive statistics and *p*-value of the Wilcoxon signed-rank test. Figure 4 shows the comparison of the Kappa of the two algorithms.

Figure 4 shows the ISTri algorithm also obtains quite a similar Kappa as the Tri algorithm on each dataset under the same labeled rate PL = 0.2, 0.4, 0.6, and 0.8. A

descriptive statistical analysis of kappa is made to compare the classification accuracy of different algorithms from a holistic perspective. The means of the Kappa of the ISTri algorithm on all the datasets under different PL are 0.739, 0.777, 0.806, and 0.814, and the ones of the Tri algorithm are 0.744, 0.775, 0.798, and 0.810. The absolute difference between the two algorithms on the mean of Kappa under the same PL value is very small from this numeric result. Meanwhile, the medians of the classification accuracy of the ISTri algorithm on all the datasets under different PL are 0.818, 0.855, 0.878, and 0.904, and the ones of the Tri algorithm are 0.833, 0.852, 0.872, and 0.895. Thus, the absolute difference between the two algorithms on the median of classification under the same PL value is also very small. Therefore, there exists a small difference between these two algorithms about Kappa from the above descriptive statistical results.

To make a more objective comparative evaluation, Wilcoxon signed rank test between two algorithms classification accuracy is made. *p*-values of this test under different PLs are 0.168, 0.519, 0.028, and 0.041, where the first two values are both larger than the given significant level of 0.05 and the latter two are smaller than 0.05. So, there exists no significant difference in the classification accuracy between two algorithms under PL = 0.2 and 0.4, while it exists no significant difference in the classification accuracy between two algorithms under PL = 0.6 and 0.8. Combing with the medians of Kappa on all the datasets, the ISTri algorithm achieves a better performance than the Tri algorithm under PL = 0.6 and 0.8.

The reason for the fact that the ISTri algorithm gets no significant difference classification in accuracy with the Tri algorithm corroborates the effectiveness and availability of the proposed unlabeled instance selection. It chooses the unlabeled instance subset by selecting the frequently confident ones identified by two other classifiers, where these selected unlabeled instances take much more ancillary information to the classifier than others. In other words, the proposed instance selection method obtains enough classification information as all the unlabeled instances so that the ISTri algorithm gets a similar classification performance as Tri algorithm.

Table 3. Kappa of two algorithms on the selected datasets.

Data	PL = 0.2		PL = 0.4		PL = 0.6		PL = 0.8	
	Tri	**ISTri**	**Tri**	**ISTri**	**Tri**	**ISTri**	**Tri**	**ISTri**
combined	0.708	0.717	0.720	0.722	0.733	0.740	0.737	0.743
connect-4	0.371	0.415	0.478	0.486	0.519	0.550	0.555	0.568
covtype	0.831	0.827	0.875	0.876	0.900	0.902	0.915	0.916
letter	0.891	0.881	0.929	0.924	0.949	0.944	0.958	0.958
optdigits	0.963	0.945	0.975	0.972	0.982	0.982	0.978	0.972
pendigits	0.976	0.973	0.987	0.985	0.989	0.989	0.991	0.989
phoneme	0.649	0.613	0.694	0.705	0.752	0.765	0.758	0.771
ring	0.834	0.810	0.830	0.835	0.844	0.855	0.876	0.891
seismic	0.566	0.570	0.571	0.577	0.588	0.591	0.590	0.594
texture	0.926	0.908	0.962	0.956	0.967	0.970	0.968	0.970
usps	0.922	0.912	0.945	0.940	0.943	0.943	0.954	0.959
winequality-white	0.294	0.294	0.339	0.344	0.408	0.437	0.439	0.443
Mean	0.744	0.739	0.775	0.777	0.798	0.806	0.810	0.814
Median	0.833	0.818	0.852	0.855	0.872	0.878	0.895	0.904
p-value	0.168		0.519		0.028		0.041	

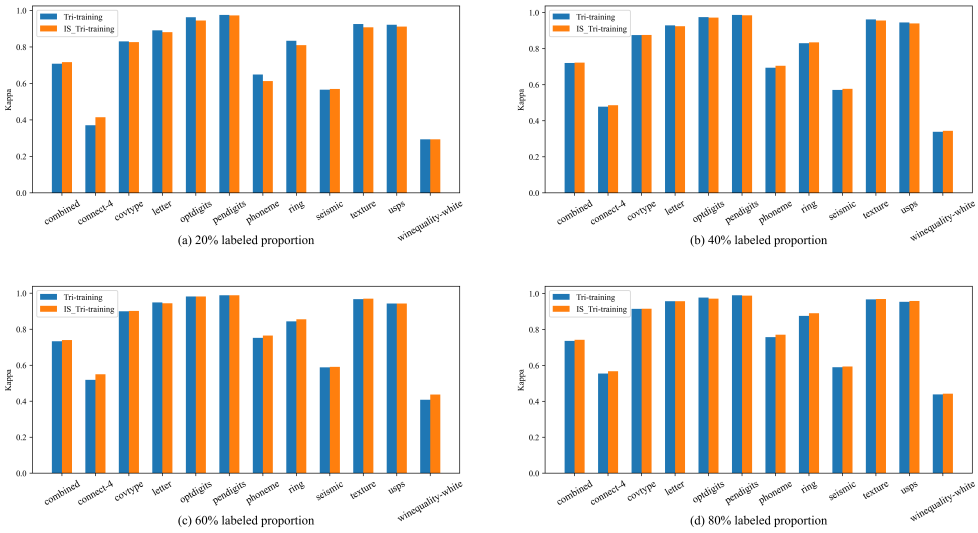

Figure 4. The comparison of Kappa between two algorithms on the selected datasets.

4.2.2. Selection Rate

The proportion of the selected instances to the original instances is an indicator to evaluate the reduction performance of instance selection. In our proposed method, the selected unlabeled instances and labeled instances reconstitute the training set that is used to efficiently learn the classifier. Therefore, the reduction performance of unlabeled instance selection affects the training efficiency of the classifier. Table 4 lists the PS of the proposed method under different labeled proportions.

It can be found that the value of PS on each dataset is significantly smaller than one under different PLs from Table 4. This result indicates the proposed method does obviously reduce the unlabeled instances, and the reformed training set with the selected unlabeled instances and labeled instances is smaller than the original set. Own to the different characteristics of the dataset, the values of PS on different datasets have significant differences. Especially for the winequality–white dataset, the proposed method finally saves a storage rate of up to 25% unlabeled instances. To carefully evaluate the selection proportion of the proposed algorithm from a global perspective, descriptive statistics are also computed on all the datasets. The means of PS on all the datasets under different PL are 0.539, 0.598, 0.618, and 0.648, and the medians are 0.568, 0.627, 0.661, and 0.693. The proposed instance selection method can reduce at least 30% of unlabeled instances from the average state.

There exist two reasons for the higher reduction ratio. Firstly, all the unlabeled instances have different levels of contributions to learning the classifier, and the number of unlabeled instances with large contributions is less than the ones with small contributions. On the other hand, our method aims to select the unlabeled instances with high confidence which have high contributions to the classifier. Thus, this experiment result has confirmed the effectiveness of the proposed algorithm.

Table 4. PS of ISTri algorithm on the selected dataset.

Data	PL = 0.2	PL = 0.4	PL = 0.6	PL = 0.8
combined	0.486	0.522	0.491	0.517
connect-4	0.585	0.641	0.629	0.672
covtype	0.670	0.728	0.761	0.808
letter	0.513	0.603	0.671	0.696
optdigits	0.531	0.603	0.651	0.718
pendigits	0.733	0.808	0.847	0.857
phoneme	0.590	0.653	0.682	0.693
ring	0.552	0.614	0.618	0.668
seismic	0.395	0.425	0.406	0.433
texture	0.686	0.743	0.745	0.779
usps	0.591	0.645	0.708	0.693
winequality–white	0.131	0.191	0.212	0.241
Mean	0.539	0.598	0.618	0.648
Median	0.568	0.627	0.661	0.693

4.2.3. Training Efficiency

Training efficiency is also an important indicator to evaluate the performance of the classification algorithms, where the execution time (in seconds) on the selected data is the common metric to measure training efficiency. Table 5 lists the execution time of two methods under different PLs on the selected datasets, and the simple statistical result is also listed in the bottom two rows of this table.

Table 5. ET of two algorithms on the selected datasets.

Data	PL = 0.2		PL = 0.4		PL = 0.6		PL = 0.8	
	Tri	ISTri	Tri	ISTri	Tri	ISTri	Tri	ISTri
combined	643.075	322.514	588.245	388.624	1328.337	795.159	1365.614	943.540
connect-4	11.757	5.905	14.787	8.466	18.317	10.916	21.232	13.493
covtype	768.997	395.715	1050.317	605.054	1336.295	815.328	1645.704	1036.018
letter	19.530	8.666	24.332	13.058	28.442	16.382	32.392	19.873
optdigits	6.763	3.204	7.974	4.502	9.265	5.615	10.407	6.582
pendigits	10.211	5.318	12.743	7.457	14.877	9.088	17.419	10.739
phoneme	5.076	2.556	6.293	3.506	7.417	4.386	8.888	5.325
ring	17.100	7.655	22.455	12.564	28.235	16.736	34.692	21.797
seismic	395.678	149.781	548.484	287.580	571.574	367.503	1038.199	511.116
texture	9.762	5.089	12.413	7.262	15.022	9.156	17.441	11.058
usps	45.562	24.265	59.183	38.108	73.571	50.979	89.867	66.263
winequality–white	6.142	2.468	7.805	3.846	9.200	5.121	10.698	6.351
Mean	161.638	77.761	196.253	115.002	286.713	175.531	357.713	221.013
Median	14.428	6.780	18.621	10.515	23.276	13.649	26.812	16.683

Table 5 shows that the ISTri algorithm has much less execution time on each dataset under the same value of PL. Meanwhile, the means of ET of the ISTri algorithm on all the datasets under different PL are 77.761, 115.005, 175.531, and 221.013, while the ones of the Tri algorithm are 161.638, 196.253, 286.73, and 221.013. Additionally, the medians of ET of the ISTri algorithm on all the datasets under different PL are 6.780, 10.515, 13.649, and 16.683, while the ones of the Tri algorithm are 14.428, 18.621, 23.276, and 26.812. This descriptive statistical result also corroborates the ISTri algorithm being able to obtain much less execution time than the Tri algorithm. The execution time of algorithms is affected by the dataset size, and its value is positively correlated with the amount of data.

To effectively compare the training efficiency of the algorithm, a speedup ratio named SR = ET(Tri)/ET(ISTri) is defined, where ET(Tri) and ET(ISTri) are the execution time of the Tri algorithm and IStri algorithm on the same dataset. This new relative indicator can eliminate the effect of data volume on the algorithm performance, and it evaluates the

difference between the two algorithms' performance from a relative perspective. Table 6 lists the SR between two algorithms under different labeled proportions.

Table 6. SR between two algorithms on the selected datasets.

Data	PL = 0.2	PL = 0.4	PL = 0.6	PL = 0.8
combined	1.994	1.764	1.671	1.447
connect-4	1.991	1.747	1.678	1.574
covtype	1.943	1.736	1.639	1.588
letter	2.254	1.863	1.736	1.630
optdigits	2.111	1.771	1.650	1.581
pendigits	1.920	1.709	1.637	1.622
phoneme	1.986	1.795	1.691	1.669
ring	2.234	1.787	1.687	1.592
seismic	2.642	1.907	1.555	2.031
texture	1.918	1.709	1.641	1.577
usps	1.878	1.553	1.443	1.356
winequality-white	2.489	2.029	1.797	1.684
Mean	2.113	1.760	1.652	1.613
Median	1.993	1.759	1.660	1.590

The result of Table 6 shows that the value of SR on each data is significantly greater than one on each dataset under different PLs, and it confirms that the proposed algorithm obtains higher training efficiency than the original algorithm. Especially, the ISTri algorithm obtains a training efficiency of more than two times higher than the Tri algorithm on dataset letter, optdigits, ring, seismic, and winequality–white under PL = 0.2. ISTri algorithm also obtains nearly 1.5 times higher training efficiency than the Tri algorithm on most datasets when PL = 0.2, 0.4, and 0.6. Simple statistical result lists that the means of SR on all the dataset under different PLs are 2.113, 1.760, 1.652, and 1.613, and the medians are 1.993, 1.759, 1.660, and 1.590. Therefore, the ISTri algorithm achieves a training efficiency of more than 0.5 times higher than the original algorithm from a global perspective.

The reason for the higher training efficiency of the ISTri algorithm is that it uses the reduced unlabeled instance subset rather than the original unlabeled instance set to learn the classifier. As we all know, the training time of the classifier is negatively correlated with the training set size. The larger the training set, the longer the training time. For the semi-supervised classification tasks, unlabeled instances make up a large proportion of the training set. Moreover, the proposed instance selection method can effectively and efficiently compress unlabeled instances while retaining most of the information valid for the classifier, and this result can be verified by the low proportion of the selected unlabeled instances.

4.2.4. High-Dimensional Problem

The proposed method has obtained a good performance on twelve medium-dimensional datasets in the previous experiment. In this section, a high-dimensional representative image classification dataset called NORB is selected, in which each image is converted to a 2047-dimensional vector by the package SciPy. Table 7 lists classification accuracy, Kappa, execution time, and selection ratio under different labeled ratios.

Table 7. The performance of two algorithms on NORB.

PL	Acc		Kappa		ET		PS-ISTri
	Tri	ISTri	Tri	ISTri	Tri	ISTri	
0.2	0.980	0.971	0.974	0.968	481.778	271.731	0.326
0.4	0.987	0.982	0.984	0.979	621.516	514.607	0.424
0.6	0.994	0.992	0.992	0.990	795.907	686.806	0.499
0.8	0.996	0.995	0.994	0.993	936.799	836.467	0.514

As the result in Table 7 shows, the ISTri algorithm efficiently and effectively processes the high-dimensional problems and achieves comparable results to the Tri algorithm. The absolute difference in Acc between two algorithms under different values of PL are 0.009, 0.005, 0.002, and 0.001, as well as the one on Kappa, are 0.006, 0.005, 0.002, and 0.001. In the worst case, the largest difference between Acc and Kappa is 0.009 and 0.006, and this difference is very small relative to the overall performance of the algorithm. Therefore, there exists a negligible difference between Acc and Kappa under different values of PL. The execution time of the ISTri algorithm is much less than the Tri algorithm under the same PL. The ratio SR between them is 1.773, 1.208, 1.159, and 1.120; all the values are larger than one. Therefore, the ISTri algorithm obtains higher training efficiency than the Tri algorithm. The last column of Table 7 also lists the unlabeled selection proportion; the value of PS is 0.326, 0.424, 0.499, and 0.514; the values are significantly smaller than one. To sum up, the proposed instance selection method greatly reduces the size of unlabeled instances while it can preserve the classification information to learn the classifier. Meanwhile, the execution time of the IStri algorithm is much less than the Tri algorithm under each value of PL, and the ratio SR between them is also larger than 1. Moreover, the selection ratio of unlabeled instances under different PLs is 0.326, 0.424, 0.499, and 0.514, which shows that the ISTri algorithm uses fewer unlabeled instances to constitute the training set. This result demonstrates that the ISTri algorithm has a higher training efficiency than the Tri algorithm.

4.2.5. Effect of Labeled Proportion

The proposition of the labeled instances to all the training instances plays an important role in the performance of the learner for the semi-supervised classification tasks. Therefore, we study the effect of PL on the classifier from three metrics: classification performance, selection rate, and training efficiency. Figures 5–7 separately show the effect of PL on three metrics. Moreover, the Friedman test is used to compare whether there exists a significant difference in each metric under the different values of PL or not, where the null hypothesis is that there does not exist a significant difference against the alternative that there is a significant difference.

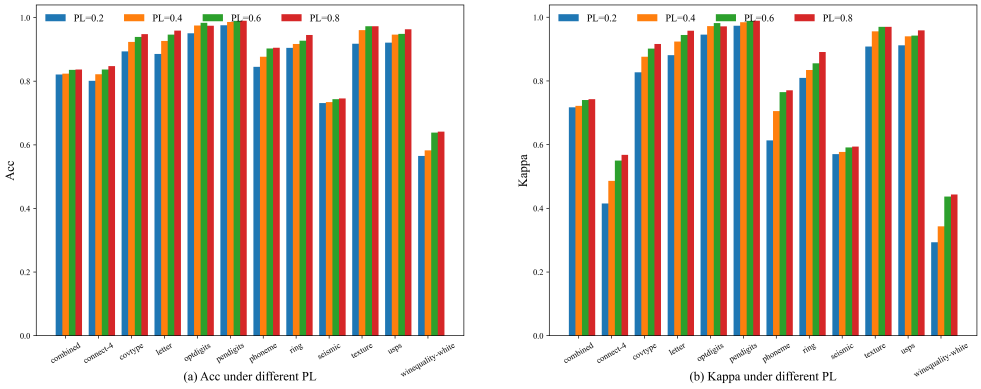

Figure 5. Classification performance of ISTri algorithm under different PL.

Figure 5 shows the change in classification performance of the proposed method under different PLs, where the left fig describes classification accuracy and the right fig for Kappa. There exists a noticeable difference in the value of Acc on almost all datasets except data combine, pendgitits, and seismic from Figure 5a. The value of Kappa also has a significant change on each dataset under different PL, especially for dataset connext-4, phoneme, winequality–white from Figure 5b. Meanwhile, the p-values of the Friedman test on Acc and Kappa are 4.02×10^{-7} and 5.49×10^{-7}, both smaller than the given significant level of 0.05. So, PL affects the classification performance of the proposed method. Moreover, Figure 5 also shows the value of Acc is positively correlated to PL on these datasets, i.e., its value significantly increases by the increasing PL, as well as this similar result for Kappa. The descriptive statistics of Acc and Kappa over all the datasets under different values

of PLs also verify this result from Tables 3 and 4. The labeled instances take much more valuable label information that is critical to learn the classifier than unlabeled instances, so PS plays a key role in the classification performance of the classifier for semi-supervised classification problems. This fact explains why the classification performance of the ISTri algorithm is positively correlated with PS. Nevertheless, the ISTri algorithm still obtains no significant difference from the Tri algorithm.

Figure 6 shows the change of the metric PS under different PLs. The value of PS fluctuates greatly on each dataset, and this result is also proved by the numeric results in Table 4. The p-value of the Friedman test on PS is 1.38×10^{-6}. smaller than the given significant level of 0.05. Therefore, there exists a significant difference in PS under different values of PL. Similar to the performance of Acc and Kappa under different PLs, the value of PS is also positively correlated with PL. The unlabeled instances selection of the ISTri algorithm mainly depends on the agreement on the pseudo-labels offered by the classifiers on the labeled instance subsets, where the parameter PL controls the number of labeled instances. The classification ability of multi-view classifiers trained on the labeled instance subsets increasingly improves as the enlarging value of PL so that the likelihood that predictive labels for each unlabeled instance are the same could increase obviously. In this way, the final selection of unlabeled data increases significantly.

The change in speedup ratio (SR) under different PLs is shown in Figure 7, where the baseline SR = 1 is also plotted on it. It can be found that all the value of SR on each dataset under different values of PL is larger than one. The metric SR has significantly different values on each dataset under different PLs, and it can be validated by the result of Table 6. The p-value of the Friedman test on SR is 3.73×10^{-7}, smaller than the given significant level of 0.05. Therefore, there exists a significant difference in SR under different values of PL. Meanwhile, SR is negatively correlated with PL on each dataset from Figure 7. SR evaluates the ratio of the execution time between the ISTri algorithm and the Tri algorithm, and the main difference between them is the number of unlabeled instances that are used to learn the classifier. The selected number of unlabeled instances continues to increase with the increasing value of PL for the ISTri algorithm, and it also induces its execution time to get longer. This result explains the reason that SR is negatively correlated with PL.

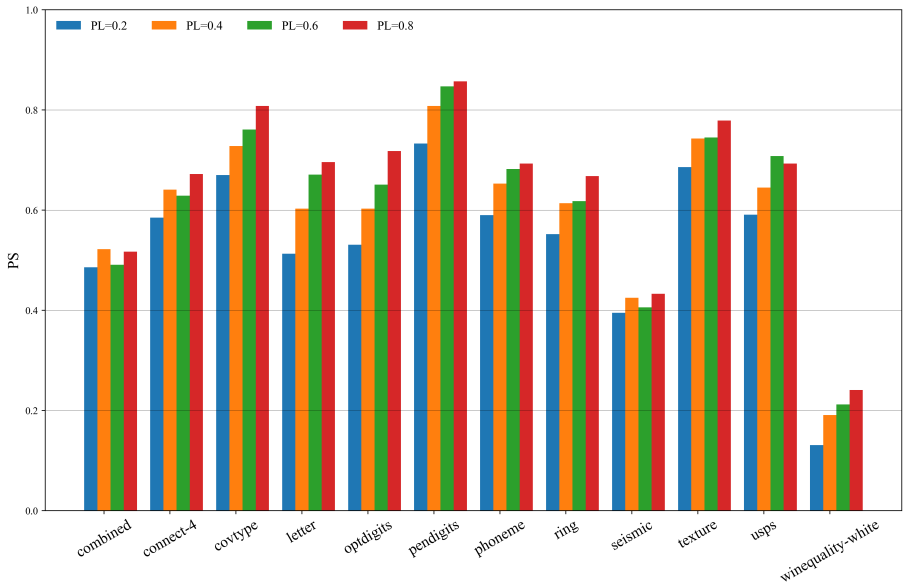

Figure 6. The change of PS under different PL.

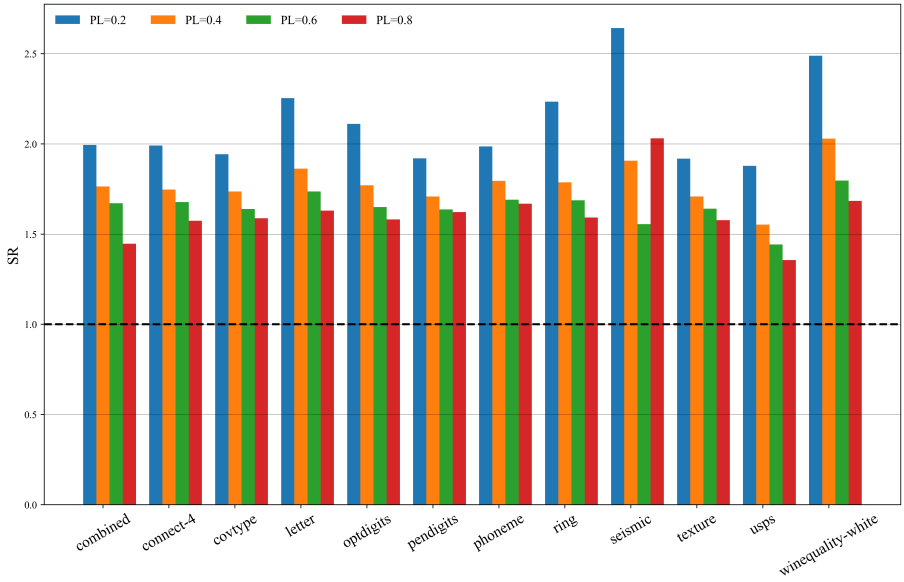

Figure 7. The change of SR under different PL.

5. Conclusions

For the problem of massive unlabeled instances bringing a great challenge to efficiently train co-training-based semi-supervised classification algorithms, this paper has developed an unlabeled instance selection algorithm based on the granulation mechanism. Different from the previous approaches from the view of algorithm optimization, it takes advantage of data reduction to avoid the difficulty of using domain knowledge to improve the efficiency of algorithms. The proposed method treats the unlabeled instances with the same frequency at which the trust instance is selected as the basic information granulation rather than each unlabeled instance; it is constructive to significantly improve execution efficiency. The selection of each unlabeled instance subset into the training set depends on its contribution to the current classification performance; this operation is guaranteed to have strong adaptability for different datasets and algorithms. The advantage of the proposed method is verified by the experiment results on the medium-dimensional and high-dimensional datasets. Especially it has a comparable classification performance with the typical algorithm, while it has high execution efficiency and fewer unlabeled instances within the training set. The proposed method can be widely used for driverless car obstacle recognition, mobile phone face recognition, temperature monitoring in greenhouses, and other large-scale application scenarios. Finally, this paper provides a potentially effective solution to improve the training efficiency of other kinds of semi-supervised classification algorithms. Future research work will explore the application of proposed algorithms in practical systems such as text classification, image classification, and pattern recognition.

Author Contributions: Writing the original draft and data preparation, Y.S.; writing the review and editing, J.Z.; oversight and leadership responsibility for the research activity planning and execution, X.Z.; implementation of the computer code and supporting algorithms, J.W. All authors have read and agreed to the published version of the manuscript.

Funding: National Natural Science Foundation of China: 62006145; Shandong Provincial Natural Science Foundation, China: ZR2020MF146.

Data Availability Statement: The selected datasets in this paper are public, and they can be freely downloaded at LIBSVM-dataset repository (https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/, 12 April 2023), KEEL-dataset repository (https://sci2s.ugr.es/keel/datasets.php, 12 April 2023) and NORB (https://cs.nyu.edu/~yann/research/norb/, 12 April 2023).

Acknowledgments: This paper was completed by Key Laboratory of Huang-Huai-Hai Smart Agricultural Technology of Ministry of Agriculture and Rural Affairs, Shandong Agricultural University. We thank the school for its support and help.

Conflicts of Interest: This paper represents the opinions of the authors and does not mean to represent the position or opinions of the Shandong Agricultural University.

References

1. Blum, A.; Mitchell, T. Combining labeled and unlabeled data with co-training. In Proceedings of the Eleventh Annual Conference on Computational Learning Theory, Madison, MI, USA, 24–26 July 1998; pp. 92–100.
2. Prasetio, B.H.; Tamura, H.; Tanno, K. Semi-supervised deep time-delay embedded clustering for stress speech analysis. *Electronics* **2019**, *8*, 1263. [CrossRef]
3. Ning, X.; Cai, W.; Zhang, L.; Yu, L. A review of research on co-training. *Concurr. Comput. Pract. Exp.* **2021**, *21*, e6276. [CrossRef]
4. Ng, K.W.; Furqan, M.S.; Gao, Y.; Ngiam, K.Y.; Khoo, E.T. HoloVein—Mixed-reality venipuncture aid via convolutional neural networks and semi-supervised learning. *Electronics* **2023**, *12*, 292. [CrossRef]
5. Li, L.; Zhang, W.; Zhang, X.; Emam, M.; Jing, W. Semi-supervised remote sensing image semantic segmentation method based on deep learning. *Electronics* **2023**, *12*, 348. [CrossRef]
6. Lang, H.; Agrawal, M.N.; Kim, Y.; Sontag, D. Co-training improves prompt-based learning for large language models. In Proceedings of the 39th International Conference on Machine Learning, Baltimore, MD, USA, 17–23 July 2022; pp. 11985–12003.
7. Fan, J.; Gao, B.; Jin, H.; Jiang, L. Ucc: Uncertainty guided cross-head co-training for semi-supervised semantic segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 9947–9956.
8. Sheikh Hassani, M.; Green, J.R. Multi-view Co-training for microRNA prediction. *Sci. Rep.* **2019**, *9*, 10931. [CrossRef]
9. Wang, H.; Shen, H.; Li, F.; Wu, Y.; Li, M.; Shi, Z.; Deng, F. Novel PV power hybrid prediction model based on FL Co-Training method. *Electronics* **2023**, *12*, 730. [CrossRef]
10. Sun, S.; Jin, F. Robust co-training. *Int. J. Pattern Recognit. Artif. Intell.* **2011**, *25*, 1113–1126. [CrossRef]
11. Dong, Y.; Jiang, L.; Li, C. Improving data and model quality in crowdsourcing using co-training-based noise correction. *Inf. Sci.* **2022**, *583*, 174–188. [CrossRef]
12. Cui, K.; Huang, J.; Luo, Z.; Zhang, G.; Zhan, F.; Lu, S. GenCo: Generative co-training for generative adversarial networks with limited data. In Proceedings of the AAAI Conference on Artificial Intelligence, Vancouver, BC, Canada, 22 February–1 March 2022; Volume 36, pp. 499–507.
13. Han, T.; Xie, W.; Zisserman, A. Self-supervised co-training for video representation learning. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 5679–5690.
14. Li, B.; Wang, J.; Yang, Z.; Yi, J.; Nie, F. Fast semi-supervised self-training algorithm based on data editing. *Inf. Sci.* **2023**, *626*, 293–314. [CrossRef]
15. Li, Y.; Maguire, L. Selecting critical patterns based on local geometrical and statistical information. *IEEE Trans. Pattern Anal. Mach. Intell.* **2010**, *33*, 1189–1201.
16. Garcia, S.; Derrac, J.; Cano, J.; Herrera, F. Prototype selection for nearest neighbor classification: Taxonomy and empirical study. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *34*, 417–435. [CrossRef] [PubMed]
17. Li, Y.; Liang, D. Safe semi-supervised learning: a brief introduction. *Front. Comput. Sci.* **2019**, *13*, 669–676. [CrossRef]
18. Liang, J.; Qian, Y.; Li, D.; Qinghua, H. Theory and method of granular computing for big data mining. *Sci. China Inf. Sci.* **2015**, *45*, 188–198.
19. Yao, Y. Three-way granular computing, rough sets, and formal concept analysis. *Int. J. Approx. Reason.* **2020**, *116*, 106–125. [CrossRef]
20. Zhang, Z.; Gao, J.; Gao, Y.; Yu, W. Two-sided matching decision making with multi-granular hesitant fuzzy linguistic term sets and incomplete criteria weight information. *Expert Syst. Appl.* **2021**, *168*, 114311. [CrossRef]
21. Chu, X.; Sun, B.; Chu, X.; Wu, J.; Han, K.; Zhang, Y.; Huang, Q. Multi-granularity dominance rough concept attribute reduction over hybrid information systems and its application in clinical decision-making. *Inf. Sci.* **2022**, *597*, 274–299. [CrossRef]
22. Sangaiah, A.K.; Javadpour, A.; Ja'fari, F.; Pinto, P.; Zhang, W.; Balasubramanian, S. A hybrid heuristics artificial intelligence feature selection for intrusion detection classifiers in cloud of things. *Clust. Comput.* **2023**, *26*, 599–612. [CrossRef]
23. Song, Y.; Zhang, J.; Zhang, C. A survey of large-scale graph-based semi-supervised classification algorithms. *Int. J. Cogn. Comput. Eng.* **2015**, *45*, 1355–1369. [CrossRef]
24. Zheng, W.; Qian, F.; Zhao, S.; Zhang, Y. M-GWNN: Multi-granularity graph wavelet neural networks for semi-supervised node classification. *Neurocomputing* **2021**, *453*, 524–537. [CrossRef]
25. Zhu, P.; Zhang, W.; Wang, Y.; Hu, Q. Multi-granularity inter-class correlation based contrastive learning for open set recognition. *Int. J. Softw. Inf.* **2022**, *12*, 157–175. [CrossRef]
26. Zhao, J.; Xie, X.; Xu, X.; Sun, S. Multi-view learning overview: Recent progress and new challenges. *Inf. Fusion* **2017**, *38*, 43–54. [CrossRef]

27. Zhou, Y.; Goldman, S. Democratic co-learning. In Proceedings of the 16th IEEE International Conference on Tools with Artificial Intelligence, Boca Raton, FL, USA, 15–17 November 2004; pp. 594–602.
28. Li, M.; Zhou, Z. Improve computer-aided diagnosis with machine learning techniques using undiagnosed samples. *IEEE Trans. Syst. Man Cybern.-Part A Syst. Hum.* **2007**, *37*, 1088–1098. [CrossRef]
29. Xu, X.; Li, W.; Xu, D.; Tsang, I.W. Co-labeling for multi-view weakly labeled learning. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *38*, 1113–1125. [CrossRef]
30. Ma, F.; Meng, D.; Xie, Q.; Li, Z.; Dong, X. Self-paced co-training. In Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; pp. 2275–2284.
31. Derrac, J.; Garcia, S.; Sanchez, L.; Herrera, F. Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. *J. Mult. Valued Log. Soft Comput.* **2015**, *17*, 255–287.
32. Ye, H.; Zhan, D.; Miao, Y.; Jiang, Y.; Zhou, Z. Rank consistency based multi-view learning: A privacy-preserving approach. In Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, Melbourne, Australia, 19–23 October 2015; pp. 991–1000.
33. Tang, J.; Tian, Y.; Zhang, P.; Liu, X. Multiview privileged support vector machines. *IEEE Trans. Neural Netw. Learn. Syst.* **2017**, *29*, 3463–3477. [PubMed]
34. Sun, S.; Shawe-Taylor, J. Sparse semi-supervised learning using conjugate functions. *J. Mach. Learn. Res.* **2010**, *11*, 2423–2455.
35. Van Engelen, J.E.; Hoos, H.H. A survey on semi-supervised learning. *Mach. Learn.* **2020**, *109*, 373–440. [CrossRef]
36. Zhou, Z.; Li, M. Tri-training: Exploiting unlabeled data using three classifiers. *IEEE Trans. Knowl. Data Eng.* **2005**, *17*, 1529–1541. [CrossRef]
37. Breiman, L. Heuristics of instability and stabilization in model selection. *Ann. Stat.* **1996**, *24*, 2350–2383. [CrossRef]
38. Song, Y.; Liang, J.; Lu, J.; Zhao, X. An efficient instance selection algorithm for k nearest neighbor regression. *Neurocomputing* **2017**, *251*, 26–34. [CrossRef]
39. Chang, C.C.; Lin, C.J. LIBSVM: A library for support vector machines. *Acm Trans. Intell. Syst. Technol.* **2011**, *2*, 1–27. [CrossRef]
40. LeCun, Y.; Huang, F.J.; Bottou, L. Learning methods for generic object recognition with invariance to pose and lighting. In Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Washington, DC, USA, 27 June–2 July 2004; Volume 2.
41. Ben-David, A. A lot of randomness is hiding in accuracy. *Eng. Appl. Artif. Intell.* **2007**, *20*, 875–885. [CrossRef]
42. Demšar, J. Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.* **2006**, *7*, 1–30.