Journal of Machine Learning Research 21 (2020) 1-38 Submitted 11/18; Revised 2/20; Published 4/20

Self-paced Multi-view Co-training

Fan Ma FAN.MA@STUDENT.UTS.EDU.AU
Centre for Artificial Intelligence, University of Technology Sydney
15 Broadway, Ultimo NSW 2007, Australia
School of Mathematics and Statistics and Ministry of Education Key Lab of Intelligent Networks and Network Security, Xian Jiaotong University
Xi'an, Shaan'xi Province, P. R. China

Deyu Meng* DYMENG@MAIL.XJTU.EDU.CN
School of Mathematics and Statistics and Ministry of Education Key Lab of Intelligent Networks and Network Security, Xian Jiaotong University
Xi'an, Shaan'xi Province, P. R. China
Macau Institute of Systems Engineering, Macau University of Science and Technology
Taipa, Macau, P. R. China

Xuanyi Dong XUANYI.DONG@STUDENT.UTS.EDU.AU
Centre for Artificial Intelligence, University of Technology Sydney
15 Broadway, Ultimo NSW 2007, Australia

Yi Yang YI.YANG@UTS.EDU.AU
Centre for Artificial Intelligence, University of Technology Sydney
15 Broadway, Ultimo NSW 2007, Australia

Editor: Samuel Kaski

Abstract

Co-training is a well-known semi-supervised learning approach which trains classifiers on two or more different views and exchanges pseudo labels of unlabeled instances in an iterative way. During the co-training process, pseudo labels of unlabeled instances are very likely to be false especially in the initial training, while the standard co-training algorithm adopts a "draw without replacement" strategy and does not remove these wrongly labeled instances from training stages. Besides, most of the traditional co-training approaches are implemented for two-view cases, and their extensions in multi-view scenarios are not intuitive. These issues not only degenerate their performance as well as available application range but also hamper their fundamental theory. Moreover, there is no optimization model to explain the objective a co-training process manages to optimize. To address these issues, in this study we design a unified self-paced multi-view co-training (SPamCo) framework which draws unlabeled instances with replacement. Two specified co-regularization terms are formulated to develop different strategies for selecting pseudo-labeled instances during training. Both forms share the same optimization strategy which is consistent with the iteration process in co-training and can be naturally extended to multi-view scenarios. A distributed optimization strategy is also introduced to train the classifier of each view in parallel to further improve the efficiency of the algorithm. Furthermore, the SPamCo algorithm is proved to be PAC learnable, supporting its theoretical soundness. Experiments conducted on synthetic, text categorization, person re-identification, image recognition and object detection data sets substantiate the superiority of the proposed method.

*. Corresponding author

Keywords: Co-training, self-paced learning, multi-view learning, semi-supervised learning, ϵ-expansion theory, probably approximately correct learnable

1. Introduction

Semi-supervised learning (SSL) aims to implement learning on both labeled and unlabeled data through fully considering the supervised knowledge delivered by labeled data and potential data structure underlying unlabeled ones. Co-training (Blum and Mitchell, 1998) is one of the most classical and well-known SSL approaches that trains classifiers on two or more views and exchanges labels of unlabeled instances in an iterative way. In recent years, co-training has been attracting much attention attributed to both of its wide applications (Nigam and Ghani, 2000; Wan, 2009; Kumar and Iii, 2011; Zhu et al., 2012; Do et al., 2016) and rational theoretical supports (Blum and Mitchell, 1998; Balcan et al., 2004; Balcan and Blum, 2010; Wang and Zhou, 2007, 2010, 2013, 2017).

Blum and Mitchell (1998) originally designed the co-training scheme and proved its correctness under the assumption that instances of different views are independent given that the class label and classifier of each view make useful predictions on unlabeled instances. Later, Balcan et al. (2004) reduced the strong theoretical requirements that the co-training algorithm would be useful when there exist confident predictions on unlabeled instances in each view. However, these assumptions require a strong pre-assumption that the pseudo labels of unlabeled instances selected in each iteration are of a high confidence extent. Based on such high-confidence assumptions, most of the current co-training style algorithms (see Section 2.1 for more details) put pseudo-labeled instances into the training set with their pseudo labels fixed during the whole learning process. Nevertheless, in most real cases such an assumption is too subjective to be satisfied, especially in the early learning stage of a co-training algorithm. The learned classifiers might not be able to distinguish certain instances confidently nor precisely pseudo-annotate them with an expected accuracy requirement. This not only inclines to degenerate the performance of co-training since the wrongly pseudo-labeled instances involved in training have no chance to be rectified in the latter training process, but also might make the underlying assumption under the theoretical support of co-training incorrect.

Another issue in most of the current co-training style methods is on the absence of an optimization model that can measure the performance and explain the intrinsic iterative mechanism under the co-training implementation. The performance measure is generally one of the necessary elements for a machine learning method. Some recent works jointly optimize an objective function based on the same assumption as with the co-training style algorithms that predictions of different views on instances should be consistent (Sindhwani et al., 2005b; Li et al., 2012). Those co-regularization approaches (see Section 2.2 for more details) encode relations of predictions from different views into a co-regularization term and turn multi-view SSL into a new convex optimization problem. However, the new objective function is often hard to be optimized, and its solution is generally different from the co-training process, which makes it unclear how the regularization term impacts the learning process. Thus, it is meaningful to explore whether there exists such an optimization model, which can finely interpret the co-training implementation during the process of solving this model. Such a model also should help reveal more insights underlying co-

training. Besides, most of the existing co-training regimes are mainly implemented in two-view cases. When more views are available, these methods are not easy to be extended. A reasonable performance measure or an objective function is necessary to inspire a sound learning manner on training classifiers in general multi-view scenarios.

To address the aforementioned issues, the self-paced multi-view co-training (SPamCo) is proposed in this study. The basic idea of SPamCo was first introduced in Ma et al. (2017) which presented the SPaCo (self-paced co-training) method. The method contains a specified objective function in which the optimization process complies with the learning procedure of conventional co-training in two-view cases. In this study, we have made a substantial improvement to the prior work. Specifically, this paper proposes a general framework for multi-view co-training, which allows rich variations for practical realizations. The SPaCo algorithm in Ma et al. (2017) is only a specific hard implementation scheme contained in this framework only usable for two-view cases, by properly setting the forms of self-paced regularizer and the weight co-regularizer. While in such a general framework, more implementation paradigms can be conducted. Not only the soft weighting scheme can be easily built, which is expected to more faithfully reflect sample importance in the learning process, but also multi-view co-training on more than three views can be naturally attained in a sound modeling manner. Together with other essential ameliorations, like the implementation scheme from serial to parallel (as introduced in Section 4), the theoretical rationality from two-view conditions to general multiple view premises (as introduced in Section 5), experimental evaluations from two-view toy experiments to multi-view text classification, image recognition, image retrieval, and object detection problems (as introduced in Section 6), this paper substantially enhances the previous SPaCo strategy to be a potentially useful regime to wider range of practical scenarios.

In summary, this work makes the following contributions:

- A unified self-paced multi-view co-training (SPamCo) framework is presented, which is formulated as a concise optimization model and can be easily applied to multiple tasks with more than two views. Specifically, two forms of SPamCo models have been introduced, including those with hard and soft co-regularization terms, respectively. The SPamCo method with the hard co-regularization term, whose two-view case accords with the scheme proposed in Ma et al. (2017), follows the binary sample selection manner of conventional co-training style algorithms. Different from the traditional co-training algorithms, the soft co-regularization term imposes continuous weights on samples for cross-view sample training. By using this more elaborate learning fashion, the prediction consistency among different views is reflected more faithfully and considered comprehensively, and thus tends to lead a better generalization performance than the hard version on multi-view co-training.

- The solutions to SPamCo share a similar iterative process with the conventional co-training algorithms. The difference lies in the strategy of selecting pseudo-labeled instances. Instead of keeping the unlabeled samples in the training set unchanged and selecting examples based on predictions of the current view, our model draws the unlabeled instances with replacement and considers predictions from all views to select pseudo examples. Furthermore, we introduce a distributed training strategy to speed up the learning procedure by using an average sample weight from all views

$\mathcal{D}$	Training set	$\mathcal{L}$	Labeled set
$\mathcal{U}$	Unlabeled set	X	Instance input space
X^+	Instance positive region	S	Confident set
N_l	Number of labeled examples	N_u	Number of unlabeled examples
M	Number of available views	K	Number of classes
y	Groundtruth label	$\tilde{y}$	Pseudo label
ℓ	Loss function	$\mathcal{R}$	Regularization term
θ	Model parameters	v	sample weight
λ	self-paced hyperparameter	γ	co-regularization hyperparameter

Table 1: Notation Table.

for each unlabeled instance to exchange information from the current view to other views. In this way, classifiers of all views are trained in parallel when multi-views or multi-models are available.

- The effectiveness of the proposed SPamCo algorithms under multi-view cases is analyzed based on the ϵ-expansion theory previously used in co-training analysis (Balcan et al., 2004). The result can easily degenerate to the two-view cases as proved in Balcan et al. (2004). The rationality of the proposed method can thus also be explained in the conventional co-training framework. We additionally analyze the proposed model from the perspective of a robust loss of self-paced learning regime, which provides a natural explanation for the effectiveness of such a co-training strategy.

- The superiority of the proposed algorithms is comprehensively substantiated on multiple types of pattern recognition and computer vision tasks including multi-view text classification, person re-identification, image recognition, and object detection.

The rest of the paper is organized as follows. We first briefly introduce related works in Section 2. Then in Section 3, we present the proposed SPamCo framework and propose its different variations with hard/soft co-regularization terms. After that, the SPamCo algorithms with both serial and parallel schemes are designed in Section 4. In Section 5, we provide theoretical analysis to support the rationality of the proposed algorithms based on ϵ-expansion theory. Experimental results are provided in Section 6, and then we conclude with a brief discussion in Section 7. The utilized notations in this paper are listed in Table 1 for easy reference of readers.

2. Related Work

Blum and Mitchell (1998) introduced co-training algorithm which trains classifiers for different views and exchanges predictions of high-confidence unlabeled data to augment the training set of each view in every iteration. Afterward, multiple advancements have been developed, which can be roughly categorized into two paradigms: co-training and co-regularization style algorithms. In this section, both types of algorithms are discussed at first and then their theoretical supports are presented. We further review the self-paced learning framework employed in our proposed model for selecting pseudo labeled instances.

2.1. Co-Training Style Algorithms

Co-training style algorithms follow the iterative learning process of co-training but formulate different schemes to select unlabeled instances. Goldman and Zhou (2000) adopted two distinct algorithms on one view of data when no redundant views are available. Nigam and Ghani (2000) analyzed the effectiveness of co-training algorithms and proposed Co-EM to operate on all unlabeled samples at each iteration. Brefeld and Scheffer (2004) further improved Co-EM by replacing naive Bayes classifier with SVM. To increase the reliability of selected unlabeled instances, Zhang and Zhou (2011) constructed a neighbour graph to improve the confidence of selected unlabeled instances. Recently, Xu et al. (2016) used predictions from all views in every iteration with different strategies and formed a pseudo-label vector for obtaining a robust prediction. Zhou (2019) introduced the abductive learning which pseudo labels can be corrected by logical reasoning. All these methods manage to improve the quantity of right predicted unlabeled instances and boost the performance of the original classifier.

As compared to conventional co-training methods, the proposed SPamCo method has mainly two ameliorative aspects. Firstly, instead of fixing pseudo-labels for high-confident instances selected into the training process, our method can continuously update the labels of all instances through being compensated from predictions of all views. Secondly, instead of the "draw without replacement" learning manner, our method employs a "draw with replacement" training mode, which allows some meaningless or even wrongly labeled instances selected in the early training stage possibly to be removed from training in the latter training process. Both of these modifications incline to help increase the robustness of the co-training calculation.

2.2. Co-Regularization Style Algorithms

Co-regularization style algorithms assume each unlabeled instance from all views with the same label, so the learned predictions of multi-views on unlabeled data should be consistent. Based on this assumption, co-regularization style algorithms directly encode parameters of classifiers and predictions on unlabeled data of different views into one optimization problem and simultaneously calculate all these variables through solving this problem.

Suppose we have M views of training set $\mathcal{D} = \{\mathcal{D}^{(j)}|j = 1, \ldots, M\}$, each training set consists of labeled set $\mathcal{L}^{(j)} = \{(x_i^{(j)}, y_i)\}_{i=1}^{N_l}$ and unlabeled set $\mathcal{U}^{(j)} = \{x_i^{(j)}\}_{i=N_l+1}^{N_l+N_u}$, where N_l and N_u are numbers of labeled and unlabeled instances, respectively. $x_i^{(j)} \in \mathcal{X}^{(j)}$ from all views (i.e., for all js) share the label $y_i \in \mathcal{Y} = \{1, 2, ..., K\}$ for all $i = 1, \cdots, N_l$. Let $\mathcal{L} = \{\mathcal{L}^{(1)}, \ldots, \mathcal{L}^{(M)}\}$ and $\mathcal{U} = \{\mathcal{U}^{(1)}, \ldots, \mathcal{U}^{(M)}\}$. The general optimization problem of co-regularization style algorithms can be written as:

$$\min_{\theta} \sum_{j}^{M} \sum_{i=1}^{N_l} \ell(y_i, g(\mathbf{x}_i^{(j)}; \theta^{(j)})) + \gamma R(\Theta, \mathcal{U}, \mathcal{L}),$$

where ℓ is a pre-defined loss function (e.g., a cross-entropy loss), and $g(x_i^{(j)}; \theta^{(j)})$ represents the prediction label with input $x_i^{(j)}$. $\Theta = \{\theta^{(j)}|j = 1, \ldots, M\}$ are the model parameters to be learned. Since only a small portion of training data is labeled (i.e., $N_l \ll N_u$), the

regularization term R is important to leverage unlabeled instances from all views along this line of algorithms. Various regularization terms have been designed to better mine the information from unlabeled instances. Typical ones are introduced as follows:

Sindhwani et al. (2005b) first introduced co-regularized least squares trying to minimize the difference between two predictions on both labeled and unlabeled instances. Sindhwani and Rosenberg (2008) further proposed the Co-MR method, which deduces a co-regularization kernel by exploiting two Reproducing Kernel Hilbert Spaces defined over the same input space. Yu et al. (2011) proposed an improved version of co-training called Bayesian co-training with the Bayesian undirected graphical model. Li et al. (2012) later designed two-view TSVM by enforcing consensus predictions between two views. Recently, Ye et al. (2015) designed a new rank constrained regularizer, which assumes predictions for unlabeled data under different views consistent with each other and enforces an affixed rank constraint on the optimization function of each view.

Most conventional co-regularization methods need to be essentially integrated with a certain learning regime, like SVM, as their base classifier, and specifically design an algorithm to attain its solution. For different co-regularization methods, their algorithms are generally different and need to use different optimization techniques. This makes them not very easy to reformulate their implementation schemes and implement their methods with variations of classifiers. Comparatively, the proposed method is with a more general realization form, and the implementation is easy to replace with different base classifiers. Besides, co-regularization approaches tend to make similar emphasis on all instances in all views (i.e., the loss of each instance is intrinsically imposed with consistent sample weight 1). This inclines to make the method more easily overfit to "bad" views with noisy instance representations. The SPamCo method alleviates this issue since it is able to automatically undermine those noisy samples on certain views through imposing small or even 0 weights on them.

2.3. Co-Training Theory Development

The rationality of co-training is supported by a series of related theoretical analyses. Blum and Mitchell (1998) showed that the class on two views is learnable in the PAC model with classification noise when the features of two views are independent given the class. To further relax the assumption for co-training, Abney (2002) provided a weaker view-independence condition that assures the success of co-training. Afterward, Balcan et al. (2004) introduced the ϵ-expansion assumption, which is a confident assumption on pseudo-labeled positive instances, further relaxing the condition for guaranteeing the effectiveness of co-training strategy. Wang and Zhou (2007) proved that co-training algorithm would succeed when there is only one view available if disagreement exists among different classifiers. Later, Wang and Zhou (2010) made a new analysis of co-training from the standpoint of label propagation. One important assumption in co-training is that the condition of high confidence of pseudo-labels of unlabeled instances should be fulfilled, and Wang and Zhou (2013) relaxed this condition by introducing diversity between two views.

Despite providing theories to support the rationality of co-training methods, most of these theories are conducted under two-view cases and include some subjective assumptions like independence between classifiers of different views or high confidence extents of pseudo-

labels of unlabeled instances obtained by the algorithm. These assumptions, however, are not only hard to be justified in real applications, but also not very intuitive to be easily understood, which might possibly keep it from being more extensively used in practice.

2.4. Self-Paced Learning

Bengio et al. (2009) proposed a learning paradigm called *curriculum learning*(CL), in which a model is learned by gradually including instances from easy to complex in training so as to increase the entropy of training instances. Afterward, self-paced learning (Kumar et al., 2010) is proposed to embed curriculum design as a regularization term into the learning objective. Due to its generality, the SPL approach has been widely applied to various tasks, such as object tracking (Supancic and Ramanan, 2013), multimedia event detection (Jiang et al., 2014a,b), image classification (Jiang et al., 2015), person re-identification (Wu et al., 2019), and object detection (Dong et al., 2017, 2018). The SPL model considers a weighted loss term for all samples and a general self-paced regularizer with respect to instance weights, expressed as:

$$\min_{\theta,\mathbf{v}\in[0,1]^n} \sum_{i=1}^{n} (v_i \ell(y_i, g(\mathbf{x}_i;\theta)) + f(v_i,\lambda)),$$

where λ is the age parameter for controlling the learning pace, $\mathbf{v} = \{v_1, v_2, \cdots, v_n\}$ contains all weights imposed on data, and $f(v,\lambda)$ represents the self-paced regularizer (SP-regularizer briefly). The form of this regularizer naturally leads to the "easy-to-hard" learning manner of SPL by its following definition (Jiang et al., 2014a; Zhao et al., 2015; Meng et al., 2017).

Definition 1 (SP-regularizer) *Suppose that v is a weight variable, ℓ is the loss, and λ is the age parameter. $f(v,\lambda)$ is called a self-paced regularizer, if*

1. *$f(v,\lambda)$ is convex with respect to $v \in [0,1]$;*
2. *$v^*(\ell,\lambda)$ is monotonically decreasing with respect to ℓ, and it holds that $\lim_{\ell\to 0} v^*(\ell,\lambda) = 1$, $\lim_{\ell\to\infty} v^*(\ell,\lambda) = 0$;*
3. *$v^*(\ell,\lambda)$ is monotonically increasing with respect to λ, and it holds that $\lim_{\lambda\to\infty} v^*(\ell,\lambda) \leq 1$, $\lim_{\lambda\to 0} v^*(\ell,\lambda) = 0$;*

where

$$v^*(\ell,\lambda) = \arg\min_{v\in[0,1]} v\ell + f(v,\lambda),$$

The three conditions in Definition 1 provide basic principles for constructing a SP-regularizer. Condition 2 indicates that the model inclines to select easy samples (with smaller losses) in favor of complex samples (with larger losses). Condition 3 states that when the model " age" (controlled by the age parameter λ) becomes larger, it tends to incorporate more, probably complex, samples to train a "mature" model. The convexity in Condition 1 ensures the soundness of this regularizer for optimization. Multiple forms of SP-regularizers based on this definition have been designed recently, such as the hard, linear and mixture SP-regularizers proposed in (Kumar et al., 2010), (Jiang et al., 2014a), and (Zhao et al., 2015), respectively. By using the alternative optimization strategy to

iteratively update $\mathbf{v}$ and $\mathbf{w}$ in the SPL regime with gradually increasing age parameter λ, more instances can be automatically included into training from easy to complex in a purely self-paced way.

SPL has recently witnessed increased focus in various applications. Jiang et al. (2015) proposed a more effective self-paced curriculum learning (SPCL) regime by embedding useful loss prior knowledge into the model and analyzed that this regime is analogous to rational instructor-student-collaborative learning mode of human teaching. Meng et al. (2017) proved that the optimization problem of SPL solved by the alternative optimization algorithm is equivalent to a robust loss minimization problem solved by a majorization-minimization algorithm. This work reveals an understanding of why SPL can conduct robust learning which are critical in various applications (Xie et al., 2017; Yong et al., 2017). Instead obtaining sample weights from losses, Shu et al. (2019) showed that sample weights can be learned from another network. Multiple literatures (Zhang et al., 2015a; Zhao et al., 2015; Pi et al., 2016) also showed that SPL worked well when dealing with real data.

3. Self-paced Multi-view Co-training

In this section, we introduce the general optimization problem of the proposed self-paced multi-view co-training (SPamCo) algorithm, and then introduce two of its realization forms. Each of them provides a particular characteristic of correlating different views.

3.1. The SPamCo Model

The general SPL framework introduces a weight for each training instance to decide its learning order. If we attach weight to every unlabeled instance, the status of this instance being selected for training can then be determined by the attached weight. Considering that both co-training style and co-regularization style algorithms assume the same class of an unlabeled data from all views and the prediction of an instance is associated with its weight in the SPL framework, we are able to fulfill such class consistency assumption by regularizing the weight vector implicitly. We can then present the SPamCo optimization problem as follows:

$$\min_{\Theta,\mathbf{V},\tilde{Y}} E = \sum_{j=1}^{M}\left(\sum_{i=1}^{N_l}\ell_i^{(j)} + \sum_{i=N_l+1}^{N_l+N_u}\left(v_i^{(j)}\ell_i^{(j)} + f(v_i^{(j)},\lambda^{(j)})\right)\right) + \mathcal{R}(\mathbf{V}) + \mathcal{R}(\mathbf{\Theta}), \tag{1}$$

where

$$\ell_i^{(j)} = \begin{cases} \ell\big(y_i, g(\mathbf{x}_i^{(j)};\theta^{(j)})\big), i = 1,\cdots,N_l, \\ \ell\big(\tilde{y}_i, g(\mathbf{x}_i^{(j)};\theta^{(j)})\big), i = N_l+1,\cdots,N_l+N_u, \end{cases}$$

where y_i denotes pre-annotated labels on the supervised samples, and $\tilde{y}_i$ represents those to be learned on the unsupervised ones, $\tilde{Y}$ denotes the set of all $\tilde{y}_i$s, $\mathbf{V} \in \mathbf{R}^{N_u \times M}$ contains all weights of unlabeled instances from all views, and the element of i^{th} row and j^{th} column in $\mathbf{V}$ is denoted by $\mathbf{V}_{i,j} = v_i^{(j)} \in [0,1]$ which corresponds to the weight imposed on $x_i^{(j)}$, $\mathbf{\Theta} = \{\theta^{(1)}, \theta^{(2)}, \cdots, \theta^{(M)}\}$ are the classifier model parameters. $\mathcal{R}(\mathbf{\Theta})$ is the regularization term on model parameters, which is a general term used in machine learning models. We

employ the commonly used L_2 regularization to penalize the weights in the present paper, and more choices, like the L_1 or more general L_p forms (Meng et al., 2012), can also be easily replaced based on certain practical scenarios. $\mathcal{R}(\mathbf{V})$ is the specific co-regularizer imposed on the sample weights of unlabeled data.

Note that only unlabeled instances are attached with weights as the labeled ones have been properly annotated and should be fully used. When there are enough labeled examples, and some labels are outliers or noises, we can also attach weights to the labeled ones to make the model robust to the noise. In the present paper, we assume that labeled examples are clean to instruct training. The following SP-regularization term proposed by (Jiang et al., 2014a) is used in our model due to its simplicity:

$$f(v_i^{(j)}, \lambda^{(j)}) = -\lambda^{(j)} v_i^{(j)}.$$

where λ_j controls how many unlabeled examples would be selected for training in each iteration. When λ_j is small, only most confident examples with small losses will be considered. As λ_j grows, more unlabeled examples will be gradually put into the training. Kumar et al. (2010) increased the parameter by multiplying a scale variable in each iteration. Zhang et al. (2015b) adjusted λ based on the specific portion of training examples. These strategies help the model augment more examples into the training.

As there always exist evident imbalance among different classes, we expect that the samples with non-zero weights (i.e., the selected pseudo-labeled samples for training) should be adaptively selected into different classes. We thus specify a different age parameter for each class to add unlabeled instances as the way in the co-training algorithm. The corresponding SP-regularization term can be written as follows:

$$f(v_i^{(j)}, \lambda^{(j)}) = -\lambda_c^{(j)} v_i^{(j)},$$

where c is the pseudo label of instance $x_i^{(j)}$, and $\lambda^{(j)} = \{\lambda_c^{(j)} | c = 1, \ldots, K\}$, K is the total number of instance classes. As there are M views, we have to set the MK values of the age parameters in different views, which would be hard to be tuned during training. Instead of directly setting λ_c^j, we simply specify the number of selected examples as in the co-training algorithm (Blum and Mitchell, 1998). λ_c^j can then be calculated based on this predefined value. The details of calculating λ_c^j are discussed in Section 4.1 (Agument λ). As the informative knowledge of different views are imperceptible, we share the number of unlabeled examples to be selected for all views. For different classes, the selected number is in proportion to the class distribution which can be simply deducted from the labeled examples. It thus makes tuning λ_c^j as a relatively much easier task by just setting the number of selected instances with only one value.

The last term $\mathcal{R}(\mathbf{V})$ in Eq.(1) is to encode the intrinsic correlation among weights of different views and compensate each other by combining knowledge from all views. Without this term, the above Eq.(1) will degenerate into the traditional self-training semi-supervised problem in each view since all views can be calculated separately with no influence to and from other views. For that reason, we call $\mathcal{R}(\mathbf{V})$ as the co-regularization term since it plays a critical role in our algorithm for multi-view training.

Both co-training and co-regularization style algorithms assume that all views share consensus predictions while utilizing this assumption in different ways. For the co-training style

algorithms, confident prediction on an unlabeled instance from one view can be trusted for the other views, while the co-regularization style algorithms enforce the consistent predictions of disparate views by adding consistent cost into the regularization term. We formulate the co-regularization term for each mechanism and introduce two types of co-regularization terms, including hard and soft regularization terms, and explain how these terms correlate different views.

3.2. SPamCo With Hard Co-Regularization Term

For co-training style algorithms, the unlabeled instances with high prediction probability of one class in one view would be added into the training pool of the other views. In our SPamCo framework, the weight of an unlabeled example in one view would be 1 if the classifier of this view predicts its corresponding instance with high confidence. To force the algorithm into selecting this instance to others views, we ought to encourage its weight in other views also being 1. The co-regularization term for implementing this can thus be written as follows:

$$\mathcal{R}_h(\mathbf{V}) = -\gamma \sum_{p<q} (\mathbf{v}^{(p)})^T \mathbf{v}^{(q)}, \tag{2}$$

where $p, q \in \{1, \ldots, M\}$, and $\mathbf{v}^{(p)} = \mathbf{V}_{*p}$ contains all weights of unlabeled instances in the p^{th} view. γ is the co-regularization parameter that controls how strongly the regularization is penalized.

The inner product form of the co-regularization term encodes the relationship of "instance easiness degree" between two views and encourages unlabeled instances of both views being selected at the same time. This co-regularization term also follows the basic strategy of co-training that most confident pseudo-labeled instances selected from one view can be used by the other views. Suppose we are minimizing Eq.(1) using the regularization term in Eq.(2) with all other parameters fixed except the weight vectors of j^{th} view, by calculating the derivative of Eq.(1) with respect to $v_i^{(j)}$, we have

$$\frac{\partial E}{\partial v_i^{(j)}} = \ell_i^{(j)} - \lambda_c^{(j)} - \gamma \sum_{q \neq j} v_i^{(q)}. \tag{3}$$

Then we can get the closed-form updating equation for $v_i^{(j)}$ as follows:

$$v_i^{(j)*} = \begin{cases} 1, \ell_i^{(j)} < \lambda_c^{(j)} + \gamma \sum_{q \neq j} v_i^{(q)}, \\ 0, otherwise. \end{cases} \tag{4}$$

From Eq.(4), we can observe that an instance with its loss value less than $\lambda_c^{(j)} + \gamma \sum_{q \neq j} v_i^{(q)}$ would be selected into training in the next iteration. This indicates that the confident instances of one view (with relatively smaller loss value $\ell_i^{(j)}$ in the classifier of j^{th} view) and instances selected by other views (with $v_i^{(q)} = 1$, meaning that the instance has been taken as confident ones and selected in previous training process), are prone to be selected than those with $v_i^{(q)} = 0$. Note for an unlabeled instance, its weight can be only 0

or 1 and is related with all other views. Thus we call the regularization term in Eq. (2) as the hard co-regularization term.

The parameter γ controls the association degree between different views. If γ is set sufficiently large with the quantity of added unlabeled instances fixed, all instances selected from other views will be chosen by the classifier of the current view. It is then equivalent to conventional co-training style algorithms in which the classifier of one view first picks instances and then puts them all into the training pool of other views. However, if predictions from one view are not reliable, we can set a small γ to combine predictions from all views to improve the robustness of predicted results on unlabeled instances.

3.3. SPamCo With Soft Co-Regularization Term

By introducing the inner-product-form co-regularizer term, the correlation information of sample confidence from different views is finely encoded in the SPamCo model. The proposed model can select unsupervised samples in one iteration and replace them with other instances. It makes the model choose the confident pseudo-labeled samples for the next iteration of training. However, the weights on the unlabeled instances can only be 0 or 1, meaning that they can only be roughly selected or removed. Compared with such a hard learning manner, the soft one should be more expected since it tends to more faithfully and comprehensively reflect the correlation information among different views. To this aim, we further design the following soft co-regularization term:

$$\mathcal{R}_s(\mathbf{V}) = \gamma \sum_{p<q} (\mathbf{v}^{(p)} - \mathbf{v}^{(q)})^T (\mathbf{v}^{(p)} - \mathbf{v}^{(q)}). \tag{5}$$

As compared to the hard co-regularization term, the meaning of this regularizer should be more evident: it is the square of the difference between weight vectors from any two views, and tends to enforce similar importance weights, as well as selected pseudo-labeled instances for further training, among different views. This form is similar to the form in co-regularization style algorithms, while instead of forcing the same predictions from different views, we require that the confidence level of an unlabeled instance should be similar in disparate views. As the confidence level of an instance is intrinsically related to its prediction, the proposed co-regularization term implicitly correlates predictions of all views. In addition, pseudo-labeled instances with high confidence would also be trained to further boost model performance, which can be easily observed from the following solution forms.

By taking the derivative with $v_i^{(j)}$, we can get:

$$\frac{\partial E}{\partial v_i^{(j)}} = \ell_i^{(j)} - \lambda_c^{(j)} + \gamma\big((M-1)v_i^{(j)} - \sum_{q \neq j} v_i^{(q)}\big). \tag{6}$$

Then we can obtain the closed-form updating equation for $v_i^{(j)}$ as follows:

$$v_i^{(j)*} = \begin{cases} 0, \ell_i^{(j)} \geq \lambda_c^{(j)} + \gamma \sum\limits_{p \neq j} v_i^{(p)}, \\ 1, \ell_i^{(j)} \leq \lambda_c^{(j)} + \gamma \sum\limits_{p \neq j} (v_i^{(p)} - 1), \\ \dfrac{1}{M-1}(\sum\limits_{p \neq j} v_i^{(p)} + \dfrac{\lambda_c^{(j)} - \ell_i^{(j)}}{\gamma}), otherwise. \end{cases} \tag{7}$$

It can be seen that for each $x_i^{(j)}$, $v_i^{(j)}$ is also calculated as 0 when the $\ell_i^{(j)}$ is larger than the sum of $\lambda_c^{(j)} + \gamma \sum_{q \neq j} v_i^{(q)}$, similar as the 0-weight case in hard SPamCo model. Otherwise, as $\ell_i^{(j)}$ linearly decreases to $\lambda_c^{(j)} + \gamma \sum_{p \neq j} (v_i^{(p)} - 1)$, $v_i^{(j)}$ would linearly increase to 1. This means the sample weight is possible to be soft values in [0,1] beyond only 0 or 1. We thus call the term in Eq. (5) as the soft co-regularization term. Only for those pseudo-labeled instances with sufficient confidence, $v_i^{(j)}$ will be 1, i.e., the instance will be used in the next training process. There are two possible types of such confident instances: the instance with large $v_i^{(p)}$ for all other views, and that with relatively smaller prediction loss value $\ell_i^{(j)}$ in the current view. Both correspond to the confident instances complying with our intuition.

The parameter γ is very similar to that in the hard SPamCo model. A relatively larger γ would make most of the weights of unlabeled instances tend to be 1 and a smaller one would make these weights as 0. The difference is that it leads to a soft weight updating scheme in soft SPamCo cases and thus tends to get a more faithful evaluation of instances' importance weights.

3.4. Remark

Our proposed SPamCo model introduces an importance weight for each pseudo-labeled instance of each view to reflect its confidence degree (with pseudo-label annotated in the training process) for training and to facilitate selection of confident ones for the next training process. The predictions among different views are correlated by imposing a co-regularization term on weight terms. Hard and soft co-regularization terms are developed for this task. For the hard co-regularization term, the designed inner product form between any of two views encourages instances selected from one view to more possibly be put into the training pool of the other views. It follows the learning procedure in co-training style algorithms, but has a specific objective function for this procedure. The soft co-regularization term is similar to the form adopted by most co-regularization style algorithms while the difference between predictions is encoded by the difference between weights from all views. This form not only forces the consensus predictions implicitly but also uses unlabeled data with confident predictions for further training.

4. Optimization Strategy

In the previous section, we propose the SPamCo model with hard and soft co-regularization terms, respectively. The alternative optimization strategy (AOS) can then be readily employed to solve both models. In this section, we first introduce the traditional optimization strategy in which each view is updated in a serial way. Then to speed up the learning process, we introduce the parallel amelioration of our algorithm.

4.1. Alternative Optimization Algorithm

The inputs to our model include the labeled set $\mathcal{L}^{(j)} = \{(x_i^{(j)}, y_i)\}_{i=1}^{N_l}$ and the unlabeled set $\mathcal{U}^{(j)} = \{x_i^{(j)}\}_{i=1}^{N_u}$. Then the detailed optimization steps for solving the proposed SPamCo model can be provided as follows.

Initiation: The first step is to initialize the parameters in the proposed model. The importance weight parameter matrix $\mathbf{V} \in \mathcal{R}^{N_u \times M}$ can be easily initiated as a zero matrix. Classifiers in all views are firstly trained based on labeled set, and predictions are made on unlabeled set. Labels of all unlabeled instances are set based on the average predictions from classifiers in all views. Age parameter $\lambda_c^{(j)}$ in each view is initialized with a small value to allow the most confident unlabeled instances of each class in all views being selected. The strategy of tuning $\lambda_c^{(j)}$ will be discussed in the following contents. The $\mathbf{V}$ is then updated based on the rule in Eq.(4) or Eq.(7) for picking confident unlabeled instances for each view. After that, for each iteration round, we repeat the following steps to update each view.

Update $\mathbf{v}^{(j)}$: For the current j^{th} view, the weight vector $\mathbf{v}^{(j)}$ is updated for preparing training samples. By taking derivatives with each $v_i^{(j)}$, we can easily get the selected pseudo-labeled into the training process (i.e., obtain their weights). As discussed in Section 3, the solution for updating $v_i^{(j)}$ given hard and soft co-regularization terms are presented in Eq.(4) and Eq.(7), respectively.

Update $\theta^{(j)}$: The training pool in the current view now contains labeled and newly selected pseudo-labeled instances. The problem of updating parameters $\theta^{(j)}$ now becomes the following sub-optimization problem:

$$\min_{\theta^{(j)}} \sum_{i=1}^{N_l} \ell_i^{(j)} + \sum_{i=N_l+1}^{N_l+N_u} v_i^{(j)} \ell_i^{(j)} + \mathcal{R}(\theta^{(j)}), \tag{8}$$

This is a standard objective function of supervised learning and can be easily solved by off-the-shelf toolkits. For instance, if a neural network is adopted and the cross-entropy loss is used for image classification tasks, the parameter $\theta^{(j)}$ is simply optimized using the SGD algorithm. Our proposed model has no limitation on the base classifiers which makes it applicable for general applications.

Update $\tilde{Y}$: The newly learned classifier is expected to perform gradually better since more confident data are expected to be used for training. It is then reasonable to make use of the updated predictions on the unlabeled set to update their pseudo-labels. It can be

Algorithm 1 Serial SPamCo Algorithm

Input: Labeled set $\mathcal{L}$ and unlabeled set $\mathcal{U}$, co-regularization parameter γ, and iteration rounds T.
Output: $\mathbf{\Theta} = \{\theta^{(j)} | j = 1, \ldots, M\}$.
Initialize weight matrix $\mathbf{V}$, age parameter λ, and current training round $t = 1$.
Update $\mathbf{\Theta}$
Update $\mathbf{V}$
while $t < T$ || no available data **do**
  for $vid \leftarrow 1$ to M **do**
    Update $\mathbf{v}^{(vid)}$: prepare training pool for current view
    Update $\theta^{(vid)}$: learn a new classifier based on added instances
    Update $\tilde{Y}$: renew predictions on all unlabeled instances
    Augment λ: allow more instances being picked
    Update $\mathbf{v}^{(vid)}$: select confident instances for other views
  end for
end while
Return $\mathbf{\Theta}$

easily done by solving the following minimization sub-problem:

$$\min_{\tilde{y}_i} \sum_{j=1}^{M} v_i^{(j)} \ell(\tilde{y}_i, g(\mathbf{x}_i^{(j)}; \theta^{(j)})). \tag{9}$$

It is easy to prove that the global optimum of the above problem can be obtained by setting the pseudo-label $\tilde{y}_i$ as the weighted average predictions directly. Note that in this manner, some of the wrongly pseudo-labeled instances are possible to be rectified.

Augment λ and Update $\mathbf{v}^{(j)}$: Once pseudo-labels of unlabeled data are refreshed, $\lambda = \{\lambda_c^{(j)} | c \in [K], j \in [M]\}$ is enlarged to allow more instances with lager loss values, i.e., the unlabeled instances with lower confidences, into the training pool in the next iteration. Specifically, at each iteration, we increase the number of selected unlabeled instances in the same way employed by co-training algorithms. Suppose that we increase the number of unlabeled samples by 5 for each class in the current iteration. We first calculate losses of all unlabeled examples by Eq.(4) and Eq.(7), and then sort the losses for each class in the ascending order. We then set λ_c^j as the value of the top 6^{th} loss for the c^{th} class under hard and soft regularization term settings, respectively.

We then update $\mathbf{v}^{(j)}$ to pick the specific number of unlabeled instances for the next iteration. There are chances that instances selected for previous training (i.e., weight equals 1 in the previous iteration) may not be selected (i.e., the weight is updated as 0) if their loss values increase to an evident large value. That is, our algorithm possesses the capability of "draw with replacement" instead of "draw without replacement" manner as most current co-training approaches do.

The iteration will be terminated when all unlabeled instances have been involved in training or the preset largest iteration number is reached. Algorithm 1 presents the entire optimization procedure. It is easy to see that the training steps of Algorithm 1 are very

Algorithm 2 Parallel SPamCo Algorithm

Input: Labeled set $\mathcal{L}$ and unlabeled set $\mathcal{U}$, co-regularization parameter γ, and iteration rounds T.
Output: $\mathbf{\Theta} = \{\theta^{(j)} | j = 1, \ldots, M\}$.
Initialize weight matrix $\mathbf{V}$, age parameter λ, and current training round $t = 1$.
Update $\mathbf{\Theta}$
Update $\mathbf{V}$:
while $t < T$ || no available data **do**
  Update $\mathbf{V}$: prepare training data for all views
  Update $\mathbf{\Theta}$: train classifiers for all views in a distributed way
  Update $\tilde{Y}$: renew predictions on all unlabeled instances
  Augment λ: allow more instances being picked
end while
Return $\mathbf{\Theta}$

similar to the standard co-training method proposed in (Blum and Mitchell, 1998). Specifically, it also iteratively trains classifiers on different views by exchanging labels of unlabeled instances in an iterative way. This shows that the proposed algorithm is closely related to other co-training approaches. Yet beyond others, the proposed algorithm complies with an optimization implementation on an underlying self-paced learning model. This model makes the co-training process capable of being easily executed in multi-view scenarios (more than 3 views) under sound objective guidance, and tends to provide some novel insightful understandings on the intrinsic effectiveness mechanism under the co-training approach.

4.2. Training Model in Parallel

The problem of the above training strategy lies in its training speed. Since the parameters of all views need to be updated one by one serially, the training time will increase especially in the cases that many views are available for the problem or multi-modal information is expected to be employed. The training time becomes critical when deep neural networks are adopted for each view. The parallel training manner should be not only necessary but also a must. For this reason, we develop a parallel learning strategy for the proposed SPamCo model, as summarized in Algorithm 2.

To guarantee a feasible parallel model of our algorithm, we need to avoid further using those up-to-date weights calculated in the current iteration since these values are temporarily restored in other machines and we need to reduce the costs of communication of different machines. The updating rule for importance weight vectors is thus simplified in each view based on weights of all views learned from the previous iteration. If a hard co-regularization term is adopted, $v_i^{(j)}$ is determined by its loss and weights from all other views, and the solution in Eq.(4) is modified as follows:

$$v_{i,t}^{(j)*} = \begin{cases} 1, \ell_i^{(j)} < \lambda_c^{(j)} + \gamma \bar{v}_{i,t-1}, \\ 0, otherwise. \end{cases} \tag{10}$$

where t denotes the current training round, and $\bar{v}_{i,t-1} = \frac{1}{M}\sum_j v_{i,t-1}^{(j)}$ is the average weight for x_i in the previous $(t-1)^{th}$ training round. Similarly, given the soft co-regularization term, we can rewrite the updating rule for $v_i^{(j)}$ as below:

$$v_{i,t}^{(j)*} = \begin{cases} 0, \ell_i^{(j)} \geq \lambda_c^{(j)} + \gamma\bar{v}_{i,t-1} \\ 1, \ell_i^{(j)} \leq \lambda_c^{(j)} + \gamma(\bar{v}_{i,t-1} - 1) \\ \bar{v}_{i,t-1} + \dfrac{\lambda_c^{(j)} - \ell_i^{(j)}}{\gamma}, otherwise. \end{cases} \tag{11}$$

The updating rule for the sample weight in each view is now correlated with the average instance weight, and the classifier of each view can thus be optimized in a distributed way. The training of classifiers in all views can be deployed on several threads or machines, and the bottleneck of training time in one iteration depends on the classifier with the longest training time among all views. Such a parallel learning manner can also be easily executed in distributed machines when multiple deep neural networks are employed. It is useful if we employ multi-classifiers for each view to further improve the probability of selecting correct pseudo-labeled instances.

5. Rationality Exploration

In the previous section, we have introduced the unified self-paced multi-view co-training model for multi-view semi-supervised learning and introduce the optimization strategy for solving it. However, rationality is not discussed despite the optimization strategy is very similar to the current co-training learning process. In this section, we will analyze the effectiveness of our proposed framework from two aspects, including traditional co-training theoretical support and self-paced learning explanation.

5.1. Multi-view Expansion Theory

Similar to the theoretical support for traditional co-training methods, we want to prove that the proposed SPamCo algorithm is a PAC learning algorithm (Valiant, 1984) under certain assumptions about the data. Since traditional investigations mainly focus on the rationality for data with only two views, it is then critical to guarantee the effectiveness of the learning algorithm when applied to the case with more available views. To make this feasible, we define a more general version of ϵ-expansion condition as used in Balcan et al. (2004) and prove its effectiveness when being applied to multi-view data.

Let D be the distribution over an instance space $X = X^1 \times \cdots \times X^M$, and X^+ and X^- denote the positive and negative regions of X, respectively (for simplicity we assume we are doing binary classification). Let D^+ and D^- denote the margin distributions of D over X^+ and X^-, respectively. Following the definition in (Balcan et al., 2004), we denote $\mathbf{S} = \{\mathbf{S}^{(j)} | i = 1, \ldots, M\}$ as confident sets in each view ($\mathbf{S}^j \subseteq X^{j+}$), and then $Pr(|\bigvee_{j\in[M]} \mathbf{S}^{(j)}|) = Pr(\mathbf{S}^{(1)} \vee \cdots \vee \mathbf{S}^{(M)})$ denotes the probability mass on instance for which we are confident about at least one view. The multi-view ϵ-expansion condition is defined as follows:

Definition 2 *D^+ is ϵ-expanding if the following inequality holds:*

$$Pr\Big(\Big|\bigoplus_{j\in[M]}\mathbf{S}^{(j)}\Big|\Big)\geq\epsilon\ \min\Bigg(Pr\Big(\Big|\bigvee_{j\in[M]}^{\geq 2}\mathbf{S}^{(j)}\Big|\Big),Pr\Big(\Big|\bigwedge_{j\in[M]}\bar{\mathbf{S}}^{(j)}\Big|\Big)\Bigg),$$

where $Pr\big(|\bigoplus_{j\in[M]}\mathbf{S}^{(j)}|\big)$ denotes the probability mass on instances for which we are confident about only one view, $Pr\big(|\bigvee_{j\in[M]}^{\geq 2}\mathbf{S}^{(j)}|\big)$ denotes the probability mass on instances being confident at least two views, and $Pr\big(|\bigwedge_{j\in[M]}\bar{\mathbf{S}}^{(j)}|\big)$ denotes the probability of instances which none of views are confident about. $Pr(|\bigvee_{j\in[M]}\mathbf{S}^{(j)}|)=Pr\big(|\bigvee_{j\in[M]}^{\geq 2}\mathbf{S}^{(j)}|\big)+Pr\big(|\bigoplus_{j\in[M]}\mathbf{S}^{(j)}|\big)$ and $Pr(|\bigvee_{j\in[M]}\mathbf{S}^{(j)}|)+Pr\big(|\bigwedge_{j\in[M]}\bar{\mathbf{S}}^{(j)}|\big)=1$.

Definition 2 is a more general version of that provided in (Balcan et al., 2004). If there are only two views ($M=2$), the proposed definition degenerates to the original one. Based on this multi-view ϵ-expansion condition, we can prove the following two lemmas.

Lemma 1 *Suppose $Pr\big(|\bigvee_{j\in[M]}^{\geq 2}\mathbf{S}^{(j)}|\big)\leq Pr\big(|\bigwedge_{j\in[M]}\bar{\mathbf{S}}^{(j)}|\big)$ and $Pr\big(\mathbf{T}^{(j)}\big||\bigvee_{j\in[M]}\mathbf{S}^{(j)}|\big)\geq 1-\epsilon^{(j)}$ for every $\epsilon^{(j)}\leq\frac{\epsilon}{8}$, and then $Pr\big(|\bigvee_{j\in[M]}^{\geq 2}\mathbf{T}^{(j)}|\big)\geq(1+\frac{\epsilon}{2})Pr\big(|\bigvee_{j\in[M]}^{\geq 2}\mathbf{S}^{(j)}|\big)$* where $\mathbf{T}^{(j)}=\mathbf{S}^{(j)}_{t+1}$ denotes the updated confident region of i^{th} view.

Lemma 2 *Suppose $Pr\big(|\bigvee_{j\in[M]}^{\geq 2}\mathbf{S}^{(j)}|\big)>Pr\big(|\bigwedge_{j\in[M]}\bar{\mathbf{S}}_i|\big)$ and let $\alpha=1-Pr\big(|\bigvee_{j\in[M]}^{\geq 2}\boldsymbol{S}^{(j)}|\big)$, if $Pr\big(\mathbf{T}^{(j)}\big||\bigvee_{j\in[M]}\mathbf{S}^{(j)}|\big)>1-\alpha\epsilon^{(j)}$ for every $\epsilon^{(j)}<\frac{\epsilon}{8}$, and then $Pr\big(|\bigvee_{j\in[M]}^{\geq 2}\mathbf{T}^{(j)}|\big)\geq(1+\frac{\alpha\epsilon}{8})Pr\big(|\bigvee_{j\in[M]}^{\geq 2}\mathbf{S}^{(j)}|\big)$.*

Based on the above two lemmas, we can then prove the following theorem for the proposed SPamCo algorithm.

Theorem 1 *Let ϵ_{fin} and δ_{fin} be the desired accuracy and confidence parameters. Suppose that the multi-view ϵ-expanding condition is satisfied in each training round, and our algorithm trains classifier in each view with accuracy and confidence parameters set to $\frac{\epsilon\cdot\epsilon_{fin}}{8}$ and $\frac{\delta_{fin}}{N}$, respectively. After running the SPamCo for $N=O(\frac{1}{\epsilon}log\frac{1}{\epsilon_{fin}\cdot\rho_{init}})$ rounds, we can then achieve the error rate as follows:*

$$Pr\big(E_{(\mathbf{x},y)\sim D}(\ell(y,g(\mathbf{x},\boldsymbol{\Theta}))<\epsilon_{fin}\big)\geq 1-\delta_{fin}\tag{12}$$

As a result, the rationality of our proposed algorithm can also be supported in terms of traditional PAC theory. And to the best of our knowledge, this is the first time that the expansion theory is analyzed for general multi-view semi-supervised learning.

5.2. Explanation by Self-Paced Learning Robustness Theory

Meng et al. (2017) proved that the optimization problem of SPL is closely related to a robust loss minimization problem. Such an understanding can be adopted in this study to present a new understanding of the effectiveness insight underlying this co-training strategy. Specifically, in the SPamCo model, there is a separate SPL objective function for each view, which implicitly corresponds to a robust loss function for training the classifier of each view on pseudo-labeled samples. However, such robust losses for different views are not independent while closely related to each other since a sample should be synchronously labeled correct or wrong for any view of data representation. Thus in the SPamCo model, the co-training curriculum regularization actually encodes such a relationship among robustness of different views. That is, through consistently exchanging pseudo-labels selected in different views, the robust loss functions of all views are enforced to be related by the regularization term. This guarantees a sound learning manner for the co-training process. Note that such an explanation for the effectiveness of the SPamCo algorithm can be easily understood and requires no subjective assumptions on pseudo-label confidences or two-view independence. It is thus expected to facilitate a better extension of co-training paradigms to general users.

6. Experimental Results

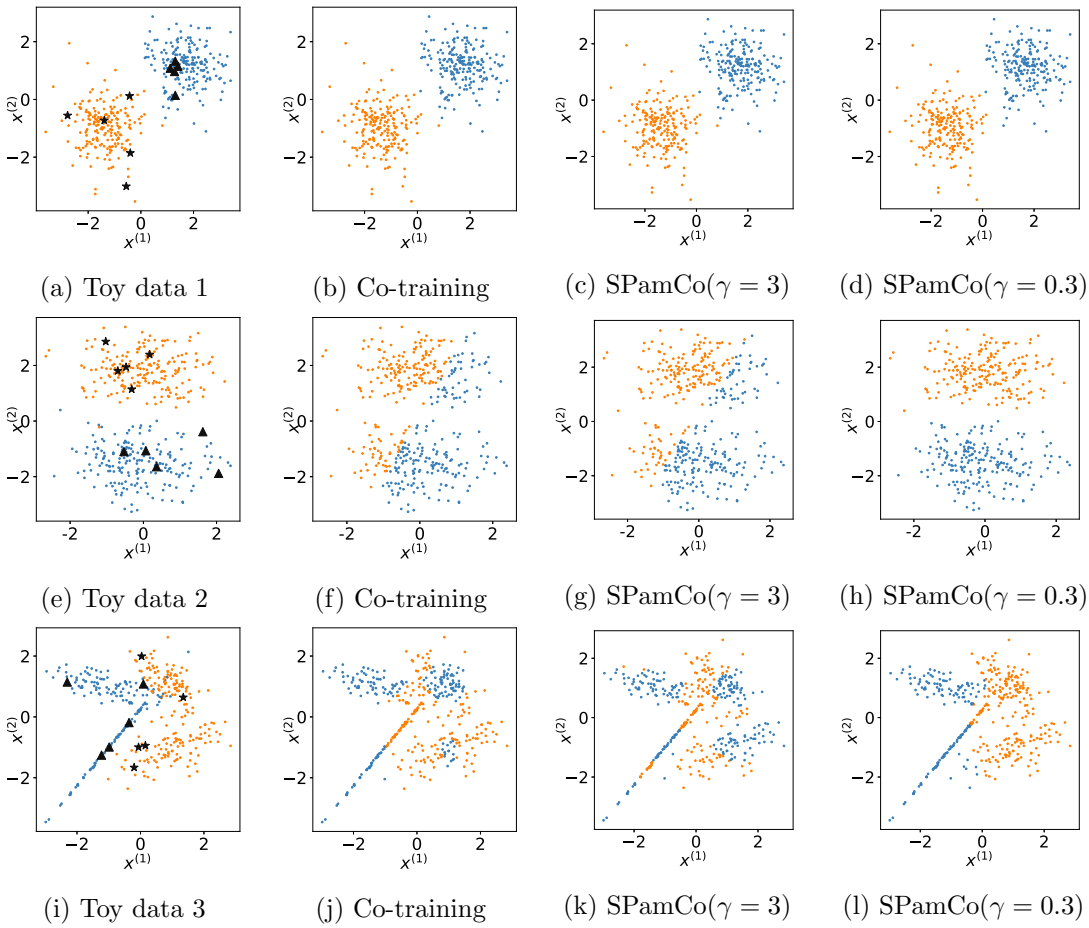
To validate the performance of the proposed method, we conduct five series of experiments on different tasks. First, we compare our proposed SPamCo with classical co-training on 3 toy instances. The progress of how each view selects pseudo-labeled examples in a "draw with replacement" manner is also visualized. We also conduct experiments on multi-view text classification, person re-identification, image recognition and object detection tasks.

6.1. Toy Data Experiments

First of all, we display some 2D toy classification tasks to visualize the co-training results in Figure 1. For each of these 2D problems, we assume that one view only contains one single feature. The traditional co-training algorithm iteratively trains the classifier of each view and adds most confident unlabeled samples into the training pool of the other view. In SPamCo, we use the hard co-regularization term with $\gamma = 3$ and 0.3, respectively, and follow the training process as described in Section 4. All instances are generated using scikit-learn Python module (Pedregosa et al., 2011)[1].

The first example shown in Figure 1(a) is a two-Gaussian case where the two view features of an instance are its two coordinates $x^{(1)}$ and $x^{(2)}$, respectively. It can be obviously obtained that each view is used to train the classifier for finely separating all instances. SVM with linear kernel function is employed as a base classifier in this case with hinge-loss as its loss function. The canonical co-training handles this problem very well since every single view is sufficient to train a classifier and both views are conditionally independent. Our SPamCo algorithm can also solve this case with γ set to different values.

1. More details about our algorithm codes and datasets can be seen in https://github.com/Flowerfan/SPamCo.



(a) Toy data 1 (b) Co-training (c) SPamCo($\gamma = 3$) (d) SPamCo($\gamma = 0.3$)

(e) Toy data 2 (f) Co-training (g) SPamCo($\gamma = 3$) (h) SPamCo($\gamma = 0.3$)

(i) Toy data 3 (j) Co-training (k) SPamCo($\gamma = 3$) (l) SPamCo($\gamma = 0.3$)

Figure 1: Toy problems for co-training. The first column is toy data generated by different Gaussian distributions, (a) and (e) are two-Gaussian data in which each distribution corresponds to one class and (i) is four-Gaussian data in which two distributions correspond to one class. The canonical co-training results on toy data are shown in the second column. Last two columns are results of SPamCo with different γ. The blue and yellow dots denote the instances from two classes, and black triangles and stars are labeled points.

For the second toy data depicted in Figure 1(e), only one view feature $x^{(2)}$ can be used to get the correct classifier while $x^{(1)}$ is irrelevant to the classification task. In this case, the traditional co-training fails to separate two clusters since wrong pseudo-labeled instances are selected in the earlier training stage by using the $x^{(1)}$ feature. The SPamCo with a large γ, approximately degenerated into the traditional co-training algorithms (as introduced in Section 3.2), also encounters such issue, while with a relatively small γ, this phenomenon can be relieved since both predictions are considered when adding pseudo-labeled instances, and wrongly labeled ones would be removed in the latter training process even when they are wrongly picked into training pool in the earlier iterations attributed to the "draw with replacement" property of our method. We visualize the process of how the classifier in each

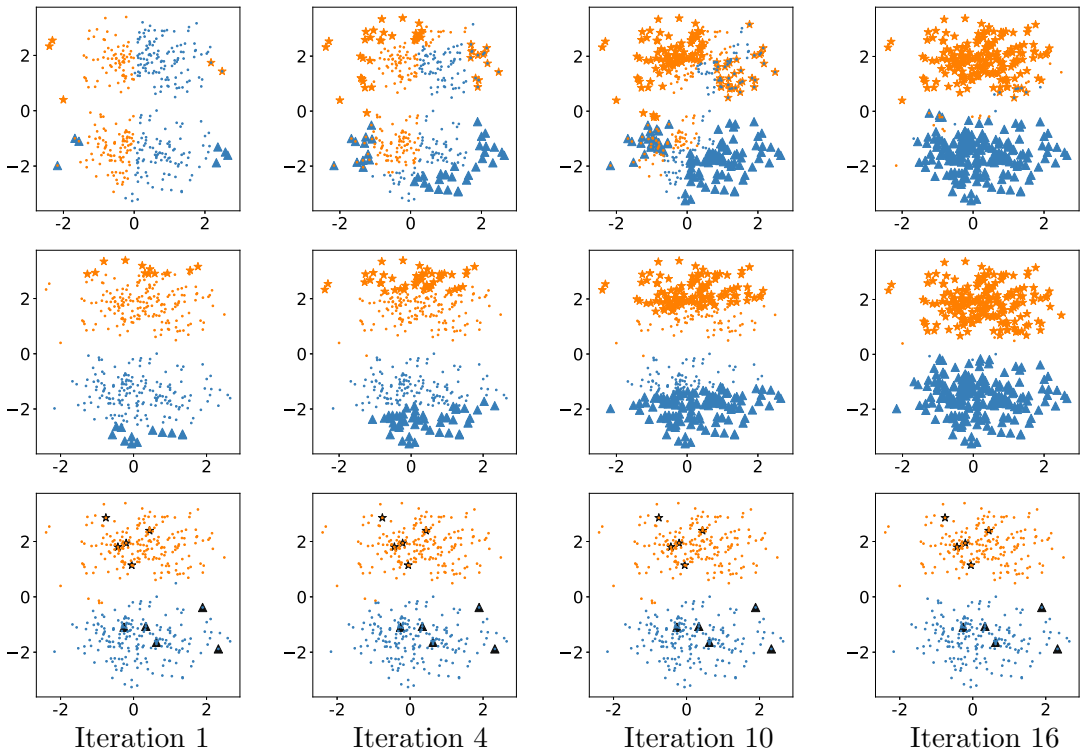


Figure 2: Visualized illustrations over the selected unlabeled examples during iterations of our method. Yellow and blue dots denote the predictions on unlabeled examples, respectively. Yellow stars are the selected pseudo-labeled examples of the first class, and blue triangles denote the pseudo-labeled examples of the second class. The first row presents the view using features along the vertical axis, and the second row represents the view using features along the horizontal axis. The third row is the fused predictions from both the first and the second views. Black triangles and stars denote the labeled points.

view selects unlabeled examples with $\gamma = 0.3$ in Figure 2. Predictions from four iterations are presented in the figure. We can obtain the view using the feature $x^{(1)}$ which fails to give right predictions and select some wrongly pseudo-labeled examples during iteration 4 to 10. However, these wrongly labeled examples are rectified in the 16 round. Moreover, some examples selected in early iterations are removed in later training data. This validates that although the first view is bad for generating a good classifier, we can relieve its influence by setting the γ to a small value. By allowing more unlabeled examples into the training, the boundary of each class is also updated and these correct pseudo-labeled examples contribute to the improvement of the classification ability.

Both of the above cases are linearly separable ones. The third experiment is a more intricate one in which the classification boundary is nonlinear. As shown in Figure 1(i), each class of the data is related to a two-Gaussian distribution. We also change the linear kernel function with radial basis function for producing a nonlinear decision surface. The traditional co-training and SPamCo with a large γ both fail to get the right classifier. The

Language	#docs	(%)	#dim	c	#l	#u	#t
English	18,758	16.78	21,531	6	84	2,916	18,674
French	26,648	23.45	24,893	6	84	2,916	26,564
German	12,342	26.80	11,547	6	84	2,916	12,258
Italian	29,953	21.51	34,279	6	84	2,916	29,869
Spanish	24,039	11.46	15,506	6	84	2,916	23,855

Table 2: Reuters multilingual data set summarization. #dim is the dimension of corresponding language, #docs, #c, #l, #u, and #t are the numbers of documents, categories, labeled instances, unlabeled instances and test instances, respectively.

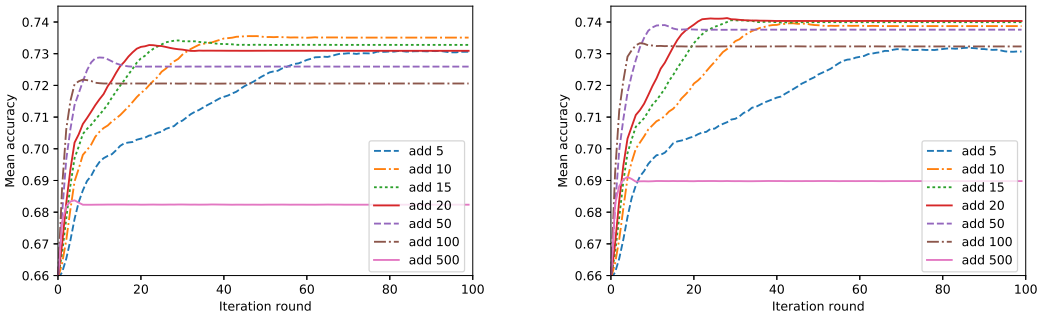


Figure 3: Convergence tendency of accuracy for SPamCo with hard and soft regularization terms under different λ updating strategy, and λ is adjusted by the number of samples added in each iteration. The left figure is the trend of the mean accuracy on the test set over iteration rounds for SPamCo with hard regularization term, and the right figure is the result for SPamCo with soft regularization term.

	SVM	TSVM	Co-LapSVM	Co-Label	SPamCo(hard)	SPamCo(soft)
Accuracy	66.79±1.11	69.34±1.22	69.34±0.82	72.45±1.12	73.28±1.23	**73.83±0.99**

Table 3: Results for Reuters with different semi-supervised learning algorithms. Mean accuracy with deviation for all competing methods are presented.

SPamCo with a smaller γ, however, can learn a good decision boundary in this case, showing its capability in recovering the non-linear structure under an appropriate γ.

In summary, these toy problems indicate that our SPamCo method with a relatively large γ possess similar characteristics compared to the traditional co-training algorithm, and SPamCo with a proper small γ performs better than, or at least as well as the traditional co-training model. Therefore, SPamCo model can be viewed as a more adaptive co-training model for various multi-view data structures.

6.2. Multi-view Text Categorization Experiments

We also evaluate our SPamCo model for multi-view semi-supervised learning on the Reuters multilingual data set in Amini et al. (2009), which is from Reuters RCV1 and RCV2 collections. This data set contains newswire articles written in 5 languages, including *English, French, German, Italian* and *Spanish*, so there are 5 views in total. Each language is categorized into 6 classes: *C15 (Performance), CCAT (Corporate/Industrial), E21 (Government Finance), ECAT (Economics), GCAT (Government Social), M11 (Equity Markets).* All documents in the data set are represented as a bag of words, using a TFIDF-based weighting scheme. And each document in one language is translated into other four languages using the statistical machine translation system PORTAGE. The processed data set can be directly downloaded from the UCI website.

To compare with other multi-view semi-supervised algorithms, we follow the experiment setting as described in Xu et al. (2016). For each class of each language, 14 and 486 documents are selected as labeled and unlabeled training instances, respectively. Thus a total number of 84 and 2916 documents are used as the labeled and unlabeled data for each language. The rest of all the instances are used as test data. Detailed information of this data set is listed in Table 2. For each view, SVM with a linear kernel is employed as a base classifier and one-versus-all strategy is employed for the multi-class task. The corresponding loss function in our model is thus the sum of k hinge loss function values. All experiments are repeated for five times with random data partitions.

We first analyze the converge rate for SPamCo with hard and soft regularization terms under different λ tuning strategies. Since λ is hard to be tuned for choosing unlabeled samples in each iteration, we specify the value for λ by controlling the number of unlabeled samples after every update. The mean accuracy on the test set with two settings is displayed in Figure 3. We employ seven λ tuning strategies by setting the increment of selected unlabeled instances as 5, 10, 15, 20, 50, 100 and 500 respectively for each class in every iteration, and γ is set as 0.3 in this experiment. Results of 100 iterations under these settings are presented for better comparison.

From Figure 3, it can be seen that our SPamCo algorithm with both hard and regularization terms under all λ settings converges and improves the performance of initialized model which are only trained on the labeled set. The increment of the selected unlabeled instance is in direct ratio with the converging rate of the proposed model but may degenerate its performance. Adding more unlabeled data with pseudo-labels into the training pool in one iteration would also introduce more noise data which may degenerate the model performance. Besides that, SPamCo with soft regularization term is less sensitive to the increment of selected unlabeled instances than that with hard regularization term.

We also compare our proposed method with other competing semi-supervised learning methods. For single view semi-supervised learning algorithms, features from all views are combined for training in SVM and TSVM (Collobert et al., 2006). Two multi-view learning methods, including CoLapSVM (Sindhwani et al., 2005a) and Co-label (Xu et al., 2016), are also compared in this experiment. The Co-LapSVM is a typical co-regularization style algorithm which introduces a prediction consistency regularization term of multi-views. For the Co-Label method, it uses predictions from all views in every iteration with different strategies and forms a pseudo-label vector for obtaining robust predictions. For our SPamCo

	Resnet50 & Densenet121			Resnet101 & Densenet121		
	View1	View2	Final	View1	View2	Final
Base	40.5±1.57	38.5±1.20	47.7±0.78	44.5±1.06	38.5±1.20	49.8±0.85
SelfTrain	59.2±0.70	61.7±1.14	67.7±0.72	62.7±0.50	61.7±1.14	69.3±0.42
Cotrain	59.3±0.50	61.9±0.80	67.0±0.33	62.5±0.15	62.2±0.65	68.5±0.29
Cotrain(Rep)	60.1±0.72	62.5±0.77	67.7±0.42	63.1±0.64	63.2±0.52	69.3±0.39
SPamCo(hard)	61.4±0.44	63.8±0.39	68.9±0.37	63.7±0.43	64.4±0.61	70.3±0.30
SPamCo(soft)	**61.7±0.21**	**64.7±0.66**	**69.5±0.33**	**64.6±0.90**	**64.8±0.31**	**70.9±0.35**

Table 4: Mean average precision (MAP) comparison of all competing methods on Market-1501 data set with two views. The first line is the supervised learning result using only labeled data. Self iterative training and co-training results are presented in the second and third lines, respectively. The "Rep" denotes that the co-training algorithm is trained with the replacement strategy. The last two lines show the results of our proposed SPamCo model with hard and soft regularization terms.

method, both hard and soft regularization terms are employed with γ fixed as 0.3 and the increment quantified in each iteration is set to 15. The means and the standard deviations of accuracy of all five languages for different methods on Reuters data set are presented in Table 3.

From the table, we can observe that our SPamCo method with both hard and soft regularization terms perform better than other methods. And SPamCo with soft regularization term achieves relatively higher mean accuracy with lower deviation than that with hard one. This demonstrates that it should be beneficial to select confident unlabeled instances during training with the soft regularization term.

6.3. Person Re-identification Experiments

The person re-identification task is usually viewed as an image retrieval problem, aiming to match pedestrians from the gallery (Zheng et al., 2016). Specifically, given a person-of-interest (query), the person re-identification method aims to determine whether the person has been observed by cameras.

Experiments are conducted on Market-1501 dataset for this task. This dataset contains 32,668 detected bounding boxes with persons of 1,501 identities (Zheng et al., 2015). Images of each identity are captured by six cameras at most, and two at least. According to the data set setting, the training set contains 12936 cropped images of 751 identities and testing set contains 19,732 cropped images of 750 identities. They are directly detected by Deformable Part Model (DPM) instead of hand-drawn bounding boxes, which is closer to the realistic setting. Each identity may have multiple images under each camera. We use the provided fixed train and test sets, under both the single-query and multi-query evaluation settings.

In this experiment, 20% instances of training data are chosen as the labeled set, and the rest of the data are treated as unlabeled. Since images for different classes are unbalanced, we randomly select 20% labeled instances for each class to make sure that the training set contains images of every class. The experiment is repeated for ten times, and the average performance in test data is reported as the final result.

	Resnet50 & Densenet121			Resnet101 & Densenet121		
	View1	View2	Final	View1	View2	Final
Base	63.4±2.06	61.9±1.65	70.1±0.99	66.7±1.04	61.9±1.65	71.8±0.65
SelfTrain	79.5±0.77	81.7±0.59	85.1±0.43	81.5±0.59	82.2±0.45	86.0±0.32
Cotrain	79.5±0.41	81.7±0.45	84.6±0.32	81.4±0.37	81.8±0.56	85.6±0.42
Cotrain(Rep)	79.9±0.50	82.3±0.37	85.1±0.40	81.7±0.40	82.7±0.38	86.0±0.43
SPamCo(hard)	80.6±0.56	83.2±0.57	85.7±0.45	82.5±0.54	83.5±0.24	86.6±0.31
SPamCo(soft)	**81.0±0.57**	**83.8±0.58**	**86.3±0.27**	**82.6±0.87**	**83.6±0.37**	**86.9±0.35**

Table 5: Rank 1 accuracy of all competing methods on Market-1501 data set with two views. The first line is the supervised learning result using only labeled data. Self iterative training and co-training results are presented in the second and third lines, respectively. The "Rep" denotes that the co-training algorithm is trained with the replacement strategy. The last two lines show the results of our proposed SPamCo model with hard and soft regularization term.

Three state-of-art deep network structures, including ResNet-50, ResNet-101 (He et al., 2016) and DenseNet-121 (Huang et al., 2017), are used to get 3-view features for the Market-1501 data set. All these models are pre-trained with ImageNet data sets, and input images are resized to 256×128 for ResNet50 and Resnet-101, 224×224 for DenseNet-101, respectively. In the training phase, images are randomly horizontal flipped and cropped for data augmentation. The cross entropy loss function is used in this experiment, and thus the re-ID task can be well handled using the SPamCo algorithm.

For two-view experiments, two combinations, ResNet-50 with DenseNet-121 and ResNet-101 with DenseNet-121, respectively, are adopted. The base algorithm uses only labeled data in this experiment. Self-train algorithm iteratively trains each classifier and adds unlabeled instances in its own view while the co-training algorithm exchanges their selected unlabeled data for training. To make a fair comparison, we also trained the co-training algorithm with the "draw with replacement" strategy. Specifically, instead of fixing the pseudo-labeled examples in the training pool, the selected unlabeled examples are re-selected from all unlabeled examples in each iteration. For the SPamCo method, hard and soft regularization terms are both implemented with the same γ set as 0.3. The number of added unlabeled samples is proportional to the number of labeled samples. We set this proportion to 0.5 in algorithms for fair comparison. The maximum iteration round is set as 5 so that all unlabeled instances get their chance to be selected. Both mean average precision (MAP) and rank-1 accuracy measures are employed for performance evaluation. All trials were repeated for 10 times and the means and standard deviations are shown in Tables 4 and 5, in terms of both measures. Compared with the traditional co-training, the co-training with the "draw with replacement" strategy performs better. However, it is still inferior to the SPamCo method. This can be explained by that the selected pseudo-labeled examples are more confident in our method as they are selected based on all views rather than on the predictions from the single view in the co-training algorithm.

The triple view experiment combines all three networks as 3-view features for learning. The traditional co-training is not included since it can only deal with two views. All other

	Mean average precision				Rank-1 accuracy			
	Res50	Den121	Res101	Final	Res50	Den121	Res101	Final
Base	40.5±1.57	38.5±1.20	44.5±1.06	52.3±0.73	63.4±2.06	61.9±1.65	66.7±1.04	73.8±0.69
Selftrain	59.2±0.70	61.7±1.14	62.7±0.50	70.8±0.37	79.5±0.77	81.7±0.59	81.5±0.59	86.7±0.41
SPamCo(hard)	61.2±0.61	63.8±0.48	63.7±0.47	71.3±0.32	80.6±0.78	83.2±0.64	82.3±0.62	87.0±0.60
SPamCo(Fhard)	54.7±0.83	56.6±0.59	56.8±0.43	64.8±0.52	75.5±0.65	78.2±0.34	77.2±0.59	83.1±0.38
SPamCo(Phard)	61.4±0.74	63.9±0.81	63.7±0.72	71.2±0.52	80.6±0.56	83.0±0.81	82.2±0.70	86.8±0.57
SPamCo(soft)	61.6±0.75	**64.5±0.72**	**64.3±0.43**	**71.8±0.45**	81.1±0.88	**83.6±0.56**	**82.8±0.59**	**87.4±0.47**
SPamCo(Fsoft)	57.3±1.02	59.9±0.82	59.4±0.61	67.3±0.63	77.7±0.50	80.6±0.47	79.1±0.55	84.6±0.50
SPamCo(Psoft)	**61.7±0.61**	64.4±0.75	**64.3±0.38**	71.7±0.44	**81.3±0.45**	**83.6±0.62**	82.7±0.43	87.3±0.27

Table 6: MAP and rank-1 accuracy of all competing methods on Market-1501 data set with triple-views. The first line is the supervised learning result using only labeled data. SelfTrain result is presented in the second line. Phard and Psoft indicate that parallel training strategy is employed compared to serial training strategy. Fhard and Fsoft denote that the model does not update labels of unlabeled examples during iterations. Last six lines show the results of SPamCo method with hard and soft regularization term under different training strategies

.

settings, including initialized parameters and training strategy, are also the same with two-view experiments. The results of all competing methods are compared in Table 6.

From Tables 4, 5 and 6, it is seen that MAP and rank-1 accuracy of all methods are improved compared to the base algorithm, in which only labeled samples are involved into training. Although multi-view features are generated by employing multi-models, the integrated results are evidently better than results using only any single model. We also fix the labels (as predicted in the first iteration) and only learn sample weights during iterations. It means that Eq. (9) is removed when optimizing the whole model. We can obtain that the model without updating predictions on unlabeled examples achieves much lower performance. It indicates that updating labels of unlabeled examples is necessary for generating better predictions. This can be easily explained by the fact that the pseudo-labels roughly annotated on the unsupervised instances inevitably contain many false ones, and naturally degenerate the classification capacity. While by allowing the pseudo-labels capable of being ameliorated during the training process, the wrongly annotated labels can thus to be possibly rectified. Besides, when fixing the labels of unlabeled examples, the model with soft regularization term performs better than the model with hard regularization term. It shows that soft sample weights make the model relatively more robust to unexpected noises. Besides, SPamCo in three-view combination with serial or parallel training strategy performs better than that in two-view settings. This can be explained by the fact that different network structures learn their own representations which together build up a better representation for original images from multiple aspects. However, the performance of traditional co-training performs worse compared with the self-train algorithm. Our proposed method performs better than both co-training and self-train algorithms with both hard or soft regularization terms. This can be explained by the mechanism that the proposed model considers to add pseudo-labeled instances from predictions of all views and some wrongly labeled samples involved into training in an early stage can be removed or rectified in the later iterations. Note that the best rank 1 accuracy and MAP results under every single and combined view are achieved by our SPamCo model with soft regularization term.

Method	CIFAR-10 (2000 examples)	CIFAR-10 (4000 examples)
LadderNetwork (Rasmus et al., 2015)	–	20.40±0.47
ImprovedGAN (Salimans et al., 2016)	19.61±2.09	18.63±2.32
TripleGAN (Chongxuan et al., 2017)	–	16.99±1.62
GoodBadGAN (Dai et al., 2017)	–	14.14±0.30
Temporal Ensembling (Laine and Aila, 2016)	15.64±0.39	12.16±0.24
Mean Teacher (Tarvainen and Valpola, 2017)	15.73±0.31	12.31±0.28
SNTG (Luo et al., 2018)	13.64±0.32	9.89±0.34
ICT (Verma et al., 2019)	**9.26±0.09**	7.66±0.17
SPamCo(Phard)	12.23±0.43	7.28±0.28
SPamCo(Psoft)	11.97±0.49	**7.05±0.24**

Table 7: Performance comparison of all competing methods on CIFAR-10 with different labeled examples (2000 and 4000). The mean error rates (%) and standard deviation are presented. The best performance is marked in bold.

This indicates that some unlabeled instances may harm the model performance while the SPamCo algorithm with soft model finely relieves this negative effect.

6.4. Image Recognition Experiments

To compare with more latest methods using deep learning models, we conduct experiments on the image recognition task. The CIFAR-10 dataset is employed. The dataset consists of 60000 32x32 colour images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images. In this experiment, 2000 and 4000 training samples are randomly selected to be taken as supervised data, respectively, and other rest training ones are taken as unsupervised instances. In both cases, the same 10000 test images are used for evaluation.

The following methods represent the latest state-of-the-art handling this problem. The Ladder network (Rasmus et al., 2015) constrained the predictions of unlabeled examples with different perturbations. Some recent works (Salimans et al., 2016; Chongxuan et al., 2017; Dai et al., 2017) used generative adversarial networks (GAN) to generate samples for auxiliary training. The samples generated by GAN can be viewed as another kind of "data augmentation" to "tell" the decision boundary where to lie. The temporal ensembling method (Laine and Aila, 2016) maintained an exponentially moving average (EMA) of predictions over epochs. Instead of averaging predictions every epoch, the mean teacher algorithm (Tarvainen and Valpola, 2017) updated the targets more frequently by average model parameters. Later Luo et al. (2018) proposed the smooth neighbors on the teacher graph (SNTG) based on previous methods which considered the connections between data points to induce smoothness on the data manifold. Verma et al. (2019) introduced a co-regularization style algorithm[2], called ICT, which encourages the prediction at an interpolation of unlabeled points to be consistent with the interpolation of the predictions at those

2. https://github.com/vikasverma1077/ICT

Steps	ER (Shake)	ER (WRN)	ER (Fuse)	DR	CE
Iteration 0	32.77	29.62	28.43	27.43	1.10
Iteration 1	24.52	25.40	23.20	19.15	0.81
Iteration 2	22.26	22.15	20.96	14.92	0.64
Iteration 3	19.05	19.14	18.33	11.39	0.48
Iteration 4	15.62	16.90	15.50	9.15	0.38
Iteration 5	12.38	13.04	12.21	6.37	0.32

Table 8: Results of SPamCo on the test data of CIFAR-10 during model iterations with 2000 examples labeled. We use ER to denote the error rate (%) and DR to denote the difference rate between predictions from two views. Shake and WRN represent the network names of two views. We report the cross entropy (CE) loss between the predictions from two views in the last column.

points. We report the error rate of these algorithms on the CIFAR-10 dataset in Table 7 for comprehensive performance comparison.

We evaluate our model on CIFAR-10 dataset with two models employed as two views: the Wide Resnet (Zagoruyko and Komodakis, 2016) and ShakeDrop (Yamada et al., 2018). We set γ to 0.3, and iteration steps to 5 and 4 for the experiment with 2000 and 4000 labeled instances, respectively. The model in each view is trained for 300 epochs in all iterations, and the learning rate is 0.1 in the beginning and is reduced 10 times after training of 100 epochs. In each iteration, the number of selected unlabeled examples increase by the number of training examples in the last iteration. We employ the random erasing technique in Zhong et al. (2020) in the data augmentation to increase the diversity of samples from different views.

Table 7 summarizes the error rates obtained by all competing methods. It can be observed that our method with both hard and soft regularization terms outperform other algorithms with only 4000 labeled examples. The SPamCo model with soft co-regularization term achieves 7.05% error rate, lower than that of the ICT method by 8%. It thus shows that the SPamCo method also works well integrated with deep learning models on the standard image recognition task.

We further present the error rate of each model in different iterations on the test set in Table 8. The algorithm is performed once with the hard co-regularization term for this experiment. As more unlabeled examples are pseudo-labeled and selected for updating classifiers, the error rate on the test data decreases. We also report the diversity degree among different models using difference rate (DR) and cross entropy (CE) loss. From Table 8, we can see that the different models indeed introduce diverse predictions. The diversity between classifiers help different views exchange information on unlabeled examples, and the model can thus add confident pseudo-labeled examples into the training to improve the model performance.

6.5. Object Detection Experiment

We also conduct experiments on the object detection task, which is one of the most fundamental problems in computer vision. Instead of simply classifying images into a single class,

	aero	bike	bird	boat	botl	bus	car	cat	chair	cow	table	dog	hors	mbik	pers	plnt	shp	sofa	train	tv	mean
Zhang et al. (2017)	47.4	22.3	35.3	23.2	13.0	50.4	48.0	41.8	1.8	28.9	27.8	37.7	41.6	43.8	20.0	12.0	27.8	22.9	48.9	31.6	31.3
Wang et al. (2014)	48.9	42.3	26.1	11.3	11.9	41.3	40.9	34.7	10.8	34.7	18.8	34.4	35.4	52.7	19.1	17.4	35.9	33.3	34.8	46.5	31.6
Kantorov et al. (2016)	**57.1**	52.0	31.5	7.6	11.5	55.0	53.1	34.1	1.7	33.1	**49.2**	42.0	47.3	56.6	15.3	12.8	24.8	48.9	44.4	47.8	36.3
Bilen and Vedaldi (2016)	46.4	58.3	35.5	25.9	14.0	66.7	53.0	39.2	8.9	41.8	26.6	38.6	44.7	59.0	10.8	17.3	40.7	49.6	56.9	50.8	39.3
Li et al. (2016)	54.5	47.4	**41.3**	20.8	17.7	51.9	**63.5**	46.1	**21.8**	**57.1**	22.1	34.4	**50.5**	61.8	16.2	**29.9**	40.7	15.9	55.3	40.2	39.5
Diba et al. (2017)	49.5	60.6	38.6	**29.2**	16.2	**70.8**	56.9	42.5	10.9	44.1	29.9	**42.2**	47.9	**64.1**	13.8	23.5	**45.9**	**54.1**	60.8	**54.5**	**42.8**
Vgg16-FRCNN	35.8	57.5	24.3	19.8	19.6	41.1	53.8	46.7	19.8	19.0	25.5	14.9	45.4	53.5	33.3	14.3	31.8	47.5	57.9	44.9	35.3
Res50-RFCN	41.0	51.6	28.6	16.9	23.5	49.5	46.7	47.4	14.6	24.1	23.7	16.4	41.9	53.8	25.7	14.4	28.4	33.7	57.2	47.4	34.3
Res101-RFCN	40.2	56.8	37.5	20.4	22.6	47.2	54.1	52.1	19.9	26.8	17.3	14.3	44.4	56.8	29.9	17.7	29.6	46.7	61.3	43.6	36.9
Final	42.4	**61.3**	39.4	23.5	**25.1**	50.1	57.3	**55.2**	18.8	26.4	22.4	17.0	48.2	56.3	**34.8**	19.2	30.6	49.0	**61.3**	51.0	39.5

Table 9: Performance comparison in average precision (AP) of all competing methods on the PASCAL VOC 2007 test set. The five compared methods make use of full image-level labels for training. Our method (the last four rows) requires only approximately four strong annotated images per class. Results on each class are shown in one column. We use Fast RCNN with VGG16 and RFCN with ResNet 50 and 101 as our base detectors to get 3-view features for the task.

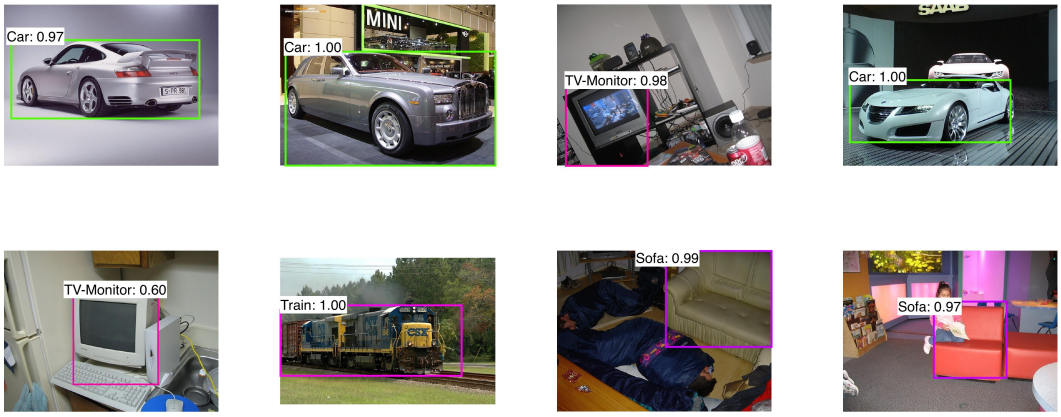
all objects in one image with their position are required to be predicted in this task. It is often expensive and time-consuming to obtain amounts of labeled objects, and thus how to use the collected small amount of labeled data together with large amounts of unlabeled instances in object detection is important.

Object detection methods can be divided into proposal based and proposal free types. Proposal based methods first determine bounding boxes of objects in each image and then make predictions on these given bounding boxes, while proposal free methods predict object bounding box and its class at the same time. In this experiment, every bounding box instead of every image is viewed as a training instance, and thus proposal based methods are employed for simplicity. Two proposal based objected detection models, Fast RCNN (Girshick, 2015) and R-FCN (Dai et al., 2016), are adopted as base detectors, and VGG (Simonyan and Zisserman, 2014), ResNet (He et al., 2016) are the backbone networks for the detectors. Three combinations, Fast RCNN with VGG, R-FCN with ResNet50 and ResNet101, are treated as three separate views for each image. In the meanwhile, selective search (Uijlings et al., 2013), an unsupervised method, is used to generate proposals for both training and test images.

We evaluate our method on PASCAL VOC 2007 detection data set (Everingham et al., 2010), which is one of the most widely used benchmarks in the object detection task. This data set contains 10022 images annotated with bounding boxes for 20 object categories. It is officially split into 2501 training, 2510 validation, and 5011 testing images.

For each class, we randomly label 4 images, which contain at least one bounding box belonging to the given class. It results in a total of 60 initial annotated images, and all the object bounding boxes in these 60 images are annotated. There are in fact an average of 4.2 images per class since some images have multiple bounding boxes.

For our proposed SPamCo method, classification and localization loss are both employed for selecting unlabeled boxes during training. In the training phase, 2000 proposals for each image are generated using the selective search method and all images are randomly flipped for data augmentation. γ is set to 0.3 for leveraging predictions from all views. The maximum iteration round is set to 5 and training epochs in each round is set to 9. We



(a) Iteration 1 (b) Iteration 2 (c) Iteration 3 (d) Iteration 4

Figure 4: Typical selected pseudo-labeled instances during training, where the bounding boxes with different colors indicate the generated pseudo-boxes by our method for different classes.

empirically use the learning rate 0.001 for the first eight epochs and reduce it to 0.0001 for the last epochs. The momentum and weight decay are set as 0.9 and 0.0005, respectively.

Since there is rare work that only uses such few instances for object detection, we compare our proposed approach with recent state-of-art semi-supervised object detection methods which use full image-level labels from training. Li et al. (2016) decomposed this task into several steps to improve the detection accuracy. Wang et al. (2014) used the typical probabilistic latent semantic analysis to learn categories of images. Zhang et al. (2017) simply used self-paced curriculum learning to detect objects from easy to hard. Kantorov et al. (2016) introduced context-aware guidance models to improve the localization. Bilen and Vedaldi (2016) proposed a weakly supervised detection network using selective search to generate proposals and train image-level classification based on regional features. Diba et al. (2017) employed location, segmentation and multi-instance network to solve this problem.

Table 9 summarizes the average precision (AP) of all competing methods on the PASCAL VOC 2007 test set. The competing methods usually use full image-level labels. In contrast, we use the same set of images but with much fewer annotations: totally 60 annotated images and the others are free-labeled. Our proposed SPamCo method achieves 39.5% mAP, a competitive performance compared to state-of-art weakly supervised object detection algorithms. And results in some specific classes, e.g., bike, bottle and persons, even achieve the best performance.

We also display some pseudo-labeled images obtained by our method over each iteration in Figure 4. It is seen that the detector tends to choose images with relatively high classification confidence aggregated over the bounding boxes. After the detector is updated, it can gradually label objects in a more complicated situation. For instance, A rotated TV-Monitor is selected with higher confidence in iteration round 3 compared to the TV-Monitor instance selected in the first iteration round, and Sofa overlapped with the person

is also selected with higher confidence in last iteration round while the detector in other three iterations fails to detect it.

7. Conclusion and Future Work

In this paper, we have proposed a unified self-paced multi-view co-training(SPamCo) model, which iteratively trains the classifier of each view and adds unlabeled instances into training with a "draw with replacement" learning manner. Two co-regularization terms, including hard and soft co-regularization terms, are introduced to define different strategies for unlabeled data. Our proposed model with hard co-regularization term follows traditional co-training style algorithms which pick confident instances from one view and then puts them into the training pool of other views. The soft co-regularization term implicitly enforces identical predictions for unlabeled instances which are often employed in conventional co-regularization style algorithms. Both co-regularization terms can be easily extended to multi-view cases with more than 3 views. We present two optimization strategies, including the serial and parallel training regimes, for solving the proposed model. The rationality of our proposed SPamCo model is theoretically analyzed by PAC learning theory and SPL robustness explanation.

From the experiment, we obtain that the diversity between different views may result in prediction biases. The diverse predictions may contribute to learning among views (i.e., the results shown in Table 7), but there is also the chance that some views are bad and thus hurting the performance. We can reduce the influence from bad views by tuning the γ to a small value as we have done in the experiment. We can also make the model robust to the bad view by directly imposing weights on views and learning it by the similar self-paced strategy. The weight can be learned from the weights on unlabeled examples among views. It is worth further developing such strategies to leverage different views. Besides, when the supervised samples contain certain outliers or heavy noises, it should be better to also impose weights to labeled instances to further suppress the influence of these noisy ones. This is also a meaningful research issue in our future investigations. The optimization theory of the proposed method (Hestenes, 1975) is also a meaningful research direction worthy of being further investigated. Furthermore, we will make more investigations to strengthen the theoretical results of our algorithms from the benefit of its better label correction capability in our future research.

Acknowledgments

We thank Ms. Ying Feng for her thorough polishing of this paper. This research was supported by National Key R&D Program of China (2018YFB1004300), the China NSFC projects under contracts 61721002, 11690011, 61603292, MoE-CMCC "Artifical Intelligence" Project No. MCM20190701, and ARC DP200100938.

Appendix A. Proof of Lemma 1

$$\begin{aligned} Pr(|\bigvee_{j\in[M]}^{\geq 2}\mathbf{T}^{(j)}|) &\geq Pr_{p\neq q}\Big(\mathbf{T}^{(p)}\wedge\mathbf{T}^{(q)}\Big) \\ &\geq Pr_{p\neq q}\Big(\mathbf{T}^{(p)}\wedge\mathbf{T}^{(q)}\Big||\bigvee_{j\in[M]}\mathbf{S}^{(j)}|\Big)Pr\Big(|\bigvee_{j\in[M]}\mathbf{S}^{(j)}|\Big) \\ &\geq (1-\epsilon_p-\epsilon_q)Pr\Big(|\bigvee_{j\in[M]}\mathbf{S}^{(j)}|\Big) \\ &\geq (1-\frac{\epsilon}{4})(1+\epsilon)Pr\Big(|\bigvee_{j\in[M]}^{\geq 2}\mathbf{S}^{(j)}|\Big) \\ &\geq (1+\frac{\epsilon}{2})Pr\Big(|\bigvee_{j\in[M]}^{\geq 2}\mathbf{S}^{(j)}|\Big). \end{aligned} \tag{13}$$

Appendix B. Proof of Lemma 2

$$\begin{aligned} \alpha &= Pr\Big(|\bigwedge_{j\in[M]}\bar{\mathbf{S}}_i|\Big)+Pr\Big(|\bigoplus_{j\in[M]}\mathbf{S}^{(j)}|\Big) \\ &\geq (1+\epsilon)Pr\Big(|\bigwedge_{j\in[M]}\bar{\mathbf{S}}_i|\Big) \\ &\geq (1+\epsilon)(1-Pr\Big(|\bigvee_{j\in[M]}\mathbf{S}^{(j)}|\Big)). \end{aligned} \tag{14}$$

From Eq.(14) we can get $Pr\Big(|\bigvee_{j\in[M]}\mathbf{S}^{(j)}|\Big)\geq 1-\frac{\alpha}{1+\epsilon}$. Thus

$$\begin{aligned} Pr(|\bigvee_{j\in[M]}^{\geq 2}\mathbf{T}^{(j)}|) &\geq Pr_{p\neq q}\Big(\mathbf{T}^{(p)}\wedge\mathbf{T}^{(q)}\Big||\bigvee_{j\in[M]}\mathbf{S}^{(j)}|\Big)Pr\Big(|\bigvee_{j\in[M]}\mathbf{S}^{(j)}|\Big) \\ &\geq (1-\frac{\alpha\epsilon}{4})(1-\frac{\alpha}{1+\epsilon}) \\ &\geq (1-\alpha)(1+\frac{\alpha\epsilon}{8}) \\ &\geq (1+\frac{\alpha\epsilon}{8})Pr\Big(|\bigvee_{j\in[M]}^{\geq 2}\mathbf{S}^{(j)}|\Big). \end{aligned} \tag{15}$$

Appendix C. Proof of Theorem 1

For $i\geq 1$, assume that $S_i^{(j)}\subseteq\mathcal{X}^{(j)+}$ is the confident set in each view after step $i-1$ of self-paced co-training. Define $p_i=Pr(|\bigvee_{j\in[M]}^{\geq 2}\mathbf{S}_i^{(j)}|)$, $q_i=Pr(|\bigwedge_{j\in[M]}\bar{\mathbf{S}}_i|)$, and $\alpha_i=1-p_i$,

with all probabilities with respect to D^+. We try to bound $Pr(|\bigvee_{j\in[M]} \mathbf{S}_n^{(j)}|)$ after N rounds of iteration.

After each training round, we get that with probability $1-\frac{\delta_{fin}}{N}$, we have:

$$Pr\big(\mathbf{S}_{i+1}^{(j)}\big|\big|\bigvee_{j\in[M]} \mathbf{S}_i^{(j)}|\big) \geq 1-\frac{\epsilon_{fin}\cdot\epsilon}{8}. \tag{16}$$

Then after first iteration, with probability $1-\frac{\delta_{fin}}{N}$, we can get:

$$p_1 = Pr(|\bigvee_{j\in[M]}^{\geq 2} \mathbf{S}_1^{(j)}|) \geq (1-\frac{\epsilon}{4})Pr(|\bigvee_{j\in[M]} \mathbf{S}_0^{(j)}|) \geq (1-\frac{\epsilon}{4})\rho_{init}. \tag{17}$$

Now we consider that for $i\geq 1$, If $p_i \leq q_i$, we can obtain that with probability $1-\frac{\delta_{fin}}{N}$, we have $Pr(|\bigvee_{j\in[M]}^{\geq 2} \mathbf{S}_{i+1}^{(j)}|) \geq (1+\frac{\epsilon}{2})Pr(|\bigvee_{j\in[M]}^{\geq 2} \mathbf{S}_i^{(j)}|)$ using Lemma 1. Similarly, if $p_i > q_i$, with probability $1-\frac{\delta_{fin}}{N}$, we have $Pr(|\bigwedge_{j\in[M]}^{\geq 2} \mathbf{S}_{i+1}^{(j)}|) \geq (1+\frac{\alpha_i\epsilon}{8})Pr(|\bigwedge_{j\in[M]}^{\geq 2} \mathbf{S}_i^{(j)}|)$ using Lemma 2. And with probability at least $1-\delta_{fin}$, learning algorithm $A^{(j)}$ of each view will success after N rounds of training.

From above observations, we have $p_{i+1} = (1+\frac{\epsilon}{16})^i(1-\frac{\epsilon}{4})\rho_{init}$ as long as $p_i \leq \frac{1}{2}$. Then the required training rounds for $p_{N_1} > \frac{1}{2}$ can be calculated by solving the following inequality:

$$(1+\frac{\epsilon}{16})^{N_1}(1-\frac{\epsilon}{4})\rho_{init} > \frac{1}{2}. \tag{18}$$

From Eq.(18), we can easily get that $N_1 > \frac{\log\frac{2}{4-\epsilon}+\log\rho_{init}}{\log(1+\frac{\epsilon}{16})}$. Since $\frac{\log\frac{2}{4-\epsilon}+\log\rho_{init}}{\log(1+\frac{\epsilon}{16})} < \frac{32}{\epsilon}log\frac{1}{\rho_{init}}$, we have $p_{N_1} > \frac{1}{2}$ after iterations $N_1 = O(\frac{1}{\epsilon}log\frac{1}{\rho_{init}})$. At this point, we compare the relationship between α_i and α_{i+1}. From Lemma 2, we can get:

$$\begin{aligned} 1-\alpha_{i+1} &\geq (1+\frac{\alpha_i\epsilon}{8})(1-\alpha_i) \\ 1+\frac{\alpha_i\epsilon}{8}-\frac{\epsilon}{8} &\geq \frac{\alpha_{i+1}}{\alpha_i} \\ 1-\frac{\epsilon}{16} &\geq \frac{\alpha_{i+1}}{\alpha_i}. \end{aligned} \tag{19}$$

Given $p_{N_1} > \frac{1}{2}$, after N_2 iterations, we have $\frac{\alpha_{N_1+N_2}}{\alpha_{N_1}} \leq (1-\frac{\epsilon}{16})^{N_2}$. Due to $\alpha_{N_1} = 1-p_{N_1} < \frac{1}{2}$, we can then make $\alpha_{N_1+N_2} \leq \epsilon_{fin}$ by calculating the required training rounds through solving the following inequality:

$$\frac{1}{2}(1-\frac{\epsilon}{16})^{N_2} \leq \epsilon_{fin}. \tag{20}$$

By solving Eq.(20), we can get that after iterations $N_2 = O(\frac{1}{\epsilon}log\frac{1}{\epsilon_{fin}})$, we have $p_{N_1+N_2} < 1-\epsilon_{fin}$. Therefore, after a total of $O(\frac{1}{\epsilon}log\frac{1}{\epsilon_{fin}\cdot\rho_{init}})$ rounds, we can have a predictor of desired accuracy with the desired confidence.

References

Steven Abney. Bootstrapping. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 360–367. Association for Computational Linguistics, 2002.

Massih Amini, Nicolas Usunier, and Cyril Goutte. Learning from multiple partially observed views - an application to multilingual text categorization. In *Advances in Neural Information Processing Systems 22*, pages 28–36. Curran Associates, Inc., 2009.

Maria-Florina Balcan and Avrim Blum. A discriminative model for semi-supervised learning. *J. ACM*, 57:19:1–19:46, 2010.

Maria-Florina Balcan, Avrim Blum, and Ke Yang. Co-training and expansion: Towards bridging theory and practice. In *Advances in neural information processing systems*, pages 89–96, 2004.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48. ACM, 2009.

Hakan Bilen and Andrea Vedaldi. Weakly supervised deep detection networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2846–2854, 2016.

Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100. ACM, 1998.

Ulf Brefeld and Tobias Scheffer. Co-em support vector learning. In *Proceedings of the twenty-first international conference on Machine learning*, page 16. ACM, 2004.

LI Chongxuan, Taufik Xu, Jun Zhu, and Bo Zhang. Triple generative adversarial nets. In *Advances in neural information processing systems*, pages 4088–4098, 2017.

Ronan Collobert, Fabian Sinz, Jason Weston, and Léon Bottou. Large scale transductive svms. *Journal of Machine Learning Research*, 7(Aug):1687–1712, 2006.

Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-fcn: Object detection via region-based fully convolutional networks. In *Advances in neural information processing systems*, pages 379–387, 2016.

Zihang Dai, Zhilin Yang, Fan Yang, William W Cohen, and Ruslan R Salakhutdinov. Good semi-supervised learning that requires a bad gan. In *Advances in neural information processing systems*, pages 6510–6520, 2017.

Ali Diba, Vivek Sharma, Ali Pazandeh, Hamed Pirsiavash, and Luc Van Gool. Weakly supervised cascaded convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 914–922, 2017.

Quynh Ngoc Thi Do, Steven Bethard, and Marie-Francine Moens. Facing the most difficult case of semantic role labeling: A collaboration of word embeddings and co-training. In *COLING*, 2016.

Xuanyi Dong, Deyu Meng, Fan Ma, and Yi Yang. A dual-network progressive approach to weakly supervised object detection. In *Proceedings of the 2017 ACM on Multimedia Conference*, MM '17, pages 279–287, New York, NY, USA, 2017. ACM. ISBN 978-1-4503-4906-2.

Xuanyi Dong, Liang Zheng, Fan Ma, Yi Yang, and Deyu Meng. Few-example object detection with model communication. *IEEE transactions on pattern analysis and machine intelligence*, 41(7):1641–1654, 2018.

Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.

Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

Sally Goldman and Yan Zhou. Enhancing supervised learning with unlabeled data. In *ICML*, pages 327–334, 2000.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

M. R. Hestenes. *Optimization theory: the finite dimensional case*. Wiley, 1975.

Gao Huang, Zhuang Liu, and Kilian Q. Weinberger. Densely connected convolutional networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2261–2269, 2017.

Lu Jiang, Deyu Meng, Teruko Mitamura, and Alexander G Hauptmann. Easy samples first: Self-paced reranking for zero-example multimedia search. In *Proceedings of the 22nd ACM international conference on Multimedia*, pages 547–556. ACM, 2014a.

Lu Jiang, Deyu Meng, Shoou-I Yu, Zhenzhong Lan, Shiguang Shan, and Alexander Hauptmann. Self-paced learning with diversity. In *Advances in Neural Information Processing Systems*, pages 2078–2086, 2014b.

Lu Jiang, Deyu Meng, Qian Zhao, Shiguang Shan, and Alexander G Hauptmann. Self-paced curriculum learning. In *AAAI*, volume 2, pages 2694–2700, 2015.

Vadim Kantorov, Maxime Oquab, Minsu Cho, and Ivan Laptev. Contextlocnet: Context-aware deep network models for weakly supervised localization. In *European Conference on Computer Vision*, pages 350–365. Springer, 2016.

Abhishek Kumar and Hal Daume Iii. A co-training approach for multi-view spectral clustering. In *International Conference on International Conference on Machine Learning*, pages 393–400, 2011.

M Pawan Kumar, Benjamin Packer, and Daphne Koller. Self-paced learning for latent variable models. In *Advances in Neural Information Processing Systems*, pages 1189–1197, 2010.

Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. *arXiv preprint arXiv:1610.02242*, 2016.

Dong Li, Jia-Bin Huang, Yali Li, Shengjin Wang, and Ming-Hsuan Yang. Weakly supervised object localization with progressive domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3512–3520, 2016.

Guangxia Li, Kuiyu Chang, and Steven C. H. Hoi. Multiview semi-supervised learning with consensus. *IEEE Transactions on Knowledge and Data Engineering*, 24(11):2040–2051, 2012.

Yucen Luo, Jun Zhu, Mengxi Li, Yong Ren, and Bo Zhang. Smooth neighbors on teacher graphs for semi-supervised learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8896–8905, 2018.

Fan Ma, Deyu Meng, Qi Xie, Zina Li, and Xuanyi Dong. Self-paced co-training. In *ICML*, 2017.

Deyu Meng, Qian Zhao, and Zongben Xu. Improve robustness of sparse pca by l1-norm maximization. *Pattern Recognition*, 45(1):487–497, 2012.

Deyu Meng, Qian Zhao, and Lu Jiang. A theoretical understanding of self-paced learning. *Information Sciences*, 414:319–328, 2017.

Kamal Nigam and Rayid Ghani. Analyzing the effectiveness and applicability of co-training. In *Proceedings of the ninth international conference on Information and knowledge management*, pages 86–93. ACM, 2000.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Te Pi, Xi Li, Zhongfei Zhang, Deyu Meng, Fei Wu, Jun Xiao, and Yueting Zhuang. Self-paced boost learning for classification. In *IJCAI*, 2016.

Antti Rasmus, Mathias Berglund, Mikko Honkala, Harri Valpola, and Tapani Raiko. Semi-supervised learning with ladder networks. In *Advances in neural information processing systems*, pages 3546–3554, 2015.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in neural information processing systems*, pages 2234–2242, 2016.

Jun Shu, Qi Xie, Lixuan Yi, Qian Zhao, Sanping Zhou, Zongben Xu, and Deyu Meng. Meta-weight-net: Learning an explicit mapping for sample weighting. In *Advances in Neural Information Processing Systems*, pages 1917–1928, 2019.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Vikas Sindhwani and David S Rosenberg. An rkhs for multi-view learning and manifold co-regularization. In *Proceedings of the 25th international conference on Machine learning*, pages 976–983. ACM, 2008.

Vikas Sindhwani, Partha Niyogi, and Mikhail Belkin. Beyond the point cloud: from transductive to semi-supervised learning. In *International Conference on Machine Learning*, pages 824–831, 2005a.

Vikas Sindhwani, Partha Niyogi, and Mikhail Belkin. A co-regularization approach to semi-supervised learning with multiple views. In *Proceedings of ICML workshop on learning with multiple views*, pages 74–79. Citeseer, 2005b.

James S Supancic and Deva Ramanan. Self-paced learning for long-term tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2379–2386, 2013.

Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in neural information processing systems*, pages 1195–1204, 2017.

Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2): 154–171, 2013.

L. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

Vikas Verma, Alex Lamb, Juho Kannala, Yoshua Bengio, and David Lopez-Paz. Interpolation consistency training for semi-supervised learning. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, IJCAI'19, pages 3635–3641. AAAI Press, 2019. ISBN 978-0-9992411-4-1. URL `http://dl.acm.org/citation.cfm?id=3367471.3367546`.

Xiaojun Wan. Co-training for cross-lingual sentiment classification. In *ACL/IJCNLP*, 2009.

Chong Wang, Weiqiang Ren, Kaiqi Huang, and Tieniu Tan. Weakly supervised object localization with latent category learning. In *European Conference on Computer Vision*, pages 431–445. Springer, 2014.

Wei Wang and Zhi-Hua Zhou. Analyzing co-training style algorithms. In *European Conference on Machine Learning*, pages 454–465. Springer, 2007.

Wei Wang and Zhi-Hua Zhou. A new analysis of co-training. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 1135–1142, 2010.

Wei Wang and Zhi-Hua Zhou. Co-training with insufficient views. In *ACML*, pages 467–482, 2013.

Wei Wang and Zhi-Hua Zhou. Theoretical foundation of co-training and disagreement-based algorithms. *CoRR*, abs/1708.04403, 2017. URL `http://arxiv.org/abs/1708.04403`.

Yu Wu, Yutian Lin, Xuanyi Dong, Yan Yan, Wei Bian, and Yi Yang. Progressive learning for person re-identification with one example. *IEEE Transactions on Image Processing*, 28(6):2872–2881, 2019.

Qi Xie, Qian Zhao, Deyu Meng, and Zongben Xu. Kronecker-basis-representation based tensor sparsity and its applications to tensor recovery. *IEEE transactions on pattern analysis and machine intelligence*, 40(8):1888–1902, 2017.

Xinxing Xu, Wen Li, Dong Xu, and Ivor W Tsang. Co-labeling for multi-view weakly labeled learning. *IEEE transactions on pattern analysis and machine intelligence*, 38(6): 1113–1125, 2016.

Yoshihiro Yamada, Masakazu Iwamura, Takuya Akiba, and Koichi Kise. Shakedrop regularization for deep residual learning. *arXiv preprint arXiv:1802.02375*, 2018.

Han-Jia Ye, De-Chuan Zhan, Yuan Miao, Yuan Jiang, and Zhi-Hua Zhou. Rank consistency based multi-view learning: A privacy-preserving approach. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 991–1000. ACM, 2015.

Hongwei Yong, Deyu Meng, Wangmeng Zuo, and Lei Zhang. Robust online matrix factorization for dynamic background subtraction. *IEEE transactions on pattern analysis and machine intelligence*, 40(7):1726–1740, 2017.

Shipeng Yu, Balaji Krishnapuram, Rmer Rosales, and R. Bharat Rao. Bayesian co-training. *Journal of Machine Learning Research*, 12(3):2649–2680, 2011.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

Dingwen Zhang, Deyu Meng, and Junwei Han. Co-saliency detection via a self-paced multiple-instance learning framework. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2015a.

Dingwen Zhang, Deyu Meng, Chao Li, Lu Jiang, Qian Zhao, and Junwei Han. A self-paced multiple-instance learning framework for co-saliency detection. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015b.

Dingwen Zhang, Deyu Meng, Long Zhao, and Junwei Han. Bridging saliency detection to weakly supervised object detection based on self-paced curriculum learning. *arXiv preprint arXiv:1703.01290*, 2017.

Min-Ling Zhang and Zhi-Hua Zhou. Cotrade: confident co-training with data editing. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 41(6):1612–1626, 2011.

Qian Zhao, Deyu Meng, Lu Jiang, Qi Xie, Zongben Xu, and Alexander G Hauptmann. Self-paced learning for matrix factorization. In *AAAI*, pages 3196–3202, 2015.

L. Zheng, Y. Yang, and A. G. Hauptmann. Person Re-identification: Past, Present and Future. *ArXiv e-prints*, October 2016.

Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. Scalable person re-identification: A benchmark. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1116–1124, 2015.

Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2020.

Zhi-Hua Zhou. Abductive learning: towards bridging machine learning and logical reasoning. *Science China Information Sciences*, 62(7):76101, 2019.

Xiaojin Zhu, Bryan R. Gibson, and Timothy T. Rogers. Co-training as a human collaboration policy. In *AAAI Conference on Artificial Intelligence, AAAI 2011, San Francisco, California, Usa, August*, 2012.