Improving the co-training algorithm to enhance semi-supervised learning results

School of Computer Science and Electronic Engineering
University of Essex,
Essex, United Kingdom
ragokh@essex.ac.uk

Executive Dean (Science and Health) - Professor (R)
School of Computer Science and Electronic Engineering
University of Essex
Essex, United Kingdom
mfasli@essex.ac.uk

***Abstract*—The co-training algorithm is one of the most common methods of semi-supervised learning in machine learning, which allows multiple learners to collaborate to discover the best information in unlabelled data. Co-training works well if the two views satisfy the sufficiency and independence assumptions. As a result of these assumptions of co-training and advancements in data classification algorithms, the performance of the underlying model could be improved. Specifically, view division, correlation between features in each view, domain knowledge, and label confidence estimation are introduced as key steps in improving co-training algorithms in this paper. Furthermore, we discuss the problems with the co-training methods currently being used, suggest some improvements, and speculate at how the algorithm could be improved going forward.**

***Index Terms*—Semi-supervised learning, text analysis, domain knowledge, co-training, classification algorithms, label prediction, independence, sufficiency, conditional independence.**

I. Introduction

Machine learning algorithms can be categorized into three types based on the way they learn - supervised learning, unsupervised learning, and semi-supervised learning as reviewed in [1]. Based on pre-classified patterns, supervised learning creates a knowledge base that assists in identifying new patterns. Mapping input features to an output called a class is the main task in this type of learning. In unsupervised learning, systems can learn to represent particular input patterns in ways that reflect their statistical structure. Combining supervised and unsupervised learning, semi-supervised learning uses the concept of self-training to label the unlabelled data based on the existing model. Using elements of both supervised and unsupervised learning, semi-supervised learning falls in between. Using this method, only a limited set of data is available to train the system, and it is thus only partially trained. Therefore such methods are useful when the availability of labelled data is limited. During partial training, the machine generates pseudo data, which later on the method combines with labeled data to make predictions.

Semi-supervised learning includes semi-supervised classification, semi-supervised clustering, and wrapper method as described in [2], [3] . The large amount of test data is classified with less training data in semi-supervised classification. Using semi supervised classification reduces the use of training data and makes it less time consuming and more cost effective. Using side information and pairwise constraints, semi-supervised clustering helps cluster data patterns with both labeled and unlabelled data. In addition to labelled data, wrapper methods allow retraining classifiers on pseudo-labelled data. Using a wrapper procedure, unlabelled data is pseudo-labelled, and a purely supervised learning algorithm constructs an inductive classifier unaware of the distinction.

Multiple supervised classifiers are trained together in co-training. The co-training process involves iteratively training two or more supervised classifiers on labelled data, adding their most confident predictions each time. In order for co-training to succeed, the base learners' predictions should not be too closely correlated. In that case, they cannot provide useful information to each other. A co-training method provides the basis for semi-supervised learning and is easy to implement. A common method of learning from multiple perspectives was co-training as proposed by [4]. Two data views that are sufficiently redundant and conditionally independent are alternately trained to maximize mutual agreement. This has sparked much research interest in recent years, which can be seen in the survey conducted by [3]. Many studies have been conducted in several fields successfully using this algorithm and its variants. In [5], unlabelled data are given changeable probabilistic labels for generalized expectation maximization (EM). Active learning and co-training were combined in [6] for robust semi-supervised learning. These algorithms and their variants were applied to natural language processing in [7], [8]. In co-training, label propagation was combined over two views and semi-supervised learning based on graphs and disagreements was integrated shown by [9]. Finally, an approach combining deep learning and co-training was demonstrated by [10].

Co-training algorithms use multiple classifiers to examine information in unlabelled samples, but there are some problems with them:

- It is difficult to manually divide a single view of the current data into multiple views in the absence of professional knowledge.
- The design of the initial learners requires measuring the difference between two learners, then selecting those with a large difference;
- The differences between multiple learners tend to decrease after a certain stage of training, which makes it difficult to utilize them further.

- There is a tendency for learners to attach incorrect labels to some unlabelled data in the initial stages, which causes the model generalization ability to deteriorate when the noise in the data becomes greater during iterative training.

The purpose of this paper is to determine the most important aspects and scenarios in which co-training performs exceptionally well in order to identify potential benefits. Furthermore, it shows how to circumvent the drawbacks it has, in order to still improve the accuracy of the prediction while working around them. In Section 2, you will find a brief overview of how the algorithm for co-training has been developed. A brief overview of the improvements and developments in recent years has been presented in section 3 of this paper along with the drawbacks encountered when using the co-training approach. A discussion of the impact of the extensions to co-training is presented in Section 4. The final section of this paper is dedicated to summarising our findings and lessons learned making suggestion for the next steps for the development of co-training.

II. Overview of the Co-training Algorithm

As described in [4], co-training is based on the following assumptions:

- it is possible to divide the features into two categories;
- the sub-feature sets are sufficient for a good classification algorithm to be trained;
- there is a conditionally independent relationship between the two sets given that the class is the same.

In the first and second case, the views are believed to be sufficient; each view is sufficient to predict the class perfectly. This assumption is known as the sufficiency assumption. According to the third assumption, the two views must be conditionally independent. This assumption is known as the independence assumption. The sufficiency and independence assumptions guarantee co-training's success if both assumptions are satisfied. A standard algorithm for co-training under two views is illustrated in Figure 1 using co-training as an example. The whole dataset is represented as View. The first step is to split the View using the labeled data, so that View1 and View2 can be obtained. Next, different base classifiers C1, C2 are trained using different view data, which can be used as initial classifiers. A final step involves estimating the confidence level of the unlabelled samples with the initial classifiers, and then adding the trusted samples to the training dataset for iterative training. After all the unlabelled samples have been self-labeled, the training is complete.

Initially two separate classifiers are trained with the labelled data, on the two sub-feature sets respectively. Each classifier then classifies the unlabelled data, and 'teaches' the other classifier with the few unlabelled examples (and the predicted labels) they regard most confident. Each classifier is retrained with the additional training examples given by the other classifier, and the process repeats. In co-training, unlabelled data helps by reducing the version space size. In other words, the two classifiers (or hypotheses) must agree on the much larger unlabelled data as well as the labelled data.

III. How does Co-training work?

There are many factors that affect how well the co-training algorithm works. As we proceed through this section, we are going to discuss the most important ones and explain why they are important to the algorithm's performance.

A. Creation of the views

Blum's co-training algorithm expresses that each instance can be described using two sets of features. Classifiers are trained by sampling unlabelled data and beginning with a weak predictor in order to learn from it. It assumes that both feature sets are conditionally independent and that the target function over each of these feature sets will predict the same label. As a result of these assumptions, the paper emphasizes that the algorithm would perform exceptionally well if the instance is able to be split into two disjoint views. Feature splits and the views are a key factor which determines the performance of the co-training.

Many datasets do not naturally partition into two sets. It can be challenging to determine whether the algorithm is successful when applied to multiple views generated from the original view. Even after manual splitting, multiple views are not always sufficient and conditionally independent at the same time. Independent views have a higher level of feature interdependence, and their sufficiency is lower when they are independent. A strong sufficiency of the split view, on the other hand, results in a weak degree of independence between the views. There is no way to maximize both sufficiency and independence in the split view simultaneously.

Researchers proposed two techniques for splitting views based on the independency and sufficiency assumption, and both proved more effective than the others. They are:

- **Random splitting of the views** A standard co-training algorithm may not be able to cover global information in some cases because two views will not be sufficient. Despite being the simplest and most straightforward method of splitting views into two, random splits are an excellent way to see how the sufficiency and independence assumptions interact with co-training.
  As shown in Figure 2 , the view is randomly split in such a way that View1 gets 75% of the feature subset and View2 gets 25%. The random subspace co-training algorithm (RASCO) was proposed by [11] for generalizing two views to multiple views. The number of subspaces and dimension also affect the classification performance. In this paper the authors show by relevant experiments that classifiers are more accurate when there are more random subspaces.
- **Automatic splitting of the views** A model could be trained based on a set of n random features and used to automatically split the views. In order to achieve the most accurate results, two distinct sets of m features are created as seen in Figure 3. They could be referred to as base-views. One can then start adding features to these two base views either at random, one at a time or in random

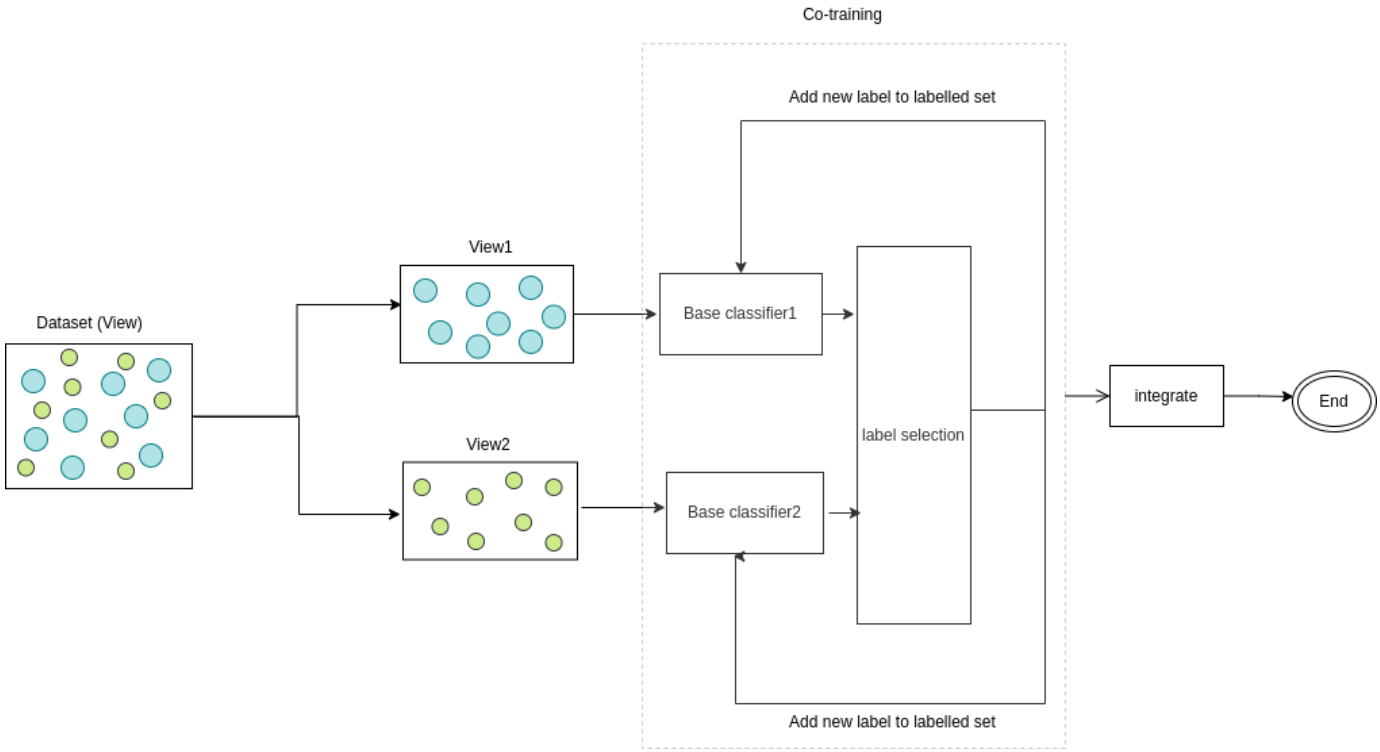

Fig. 1: Working of the standard co-training algorithm

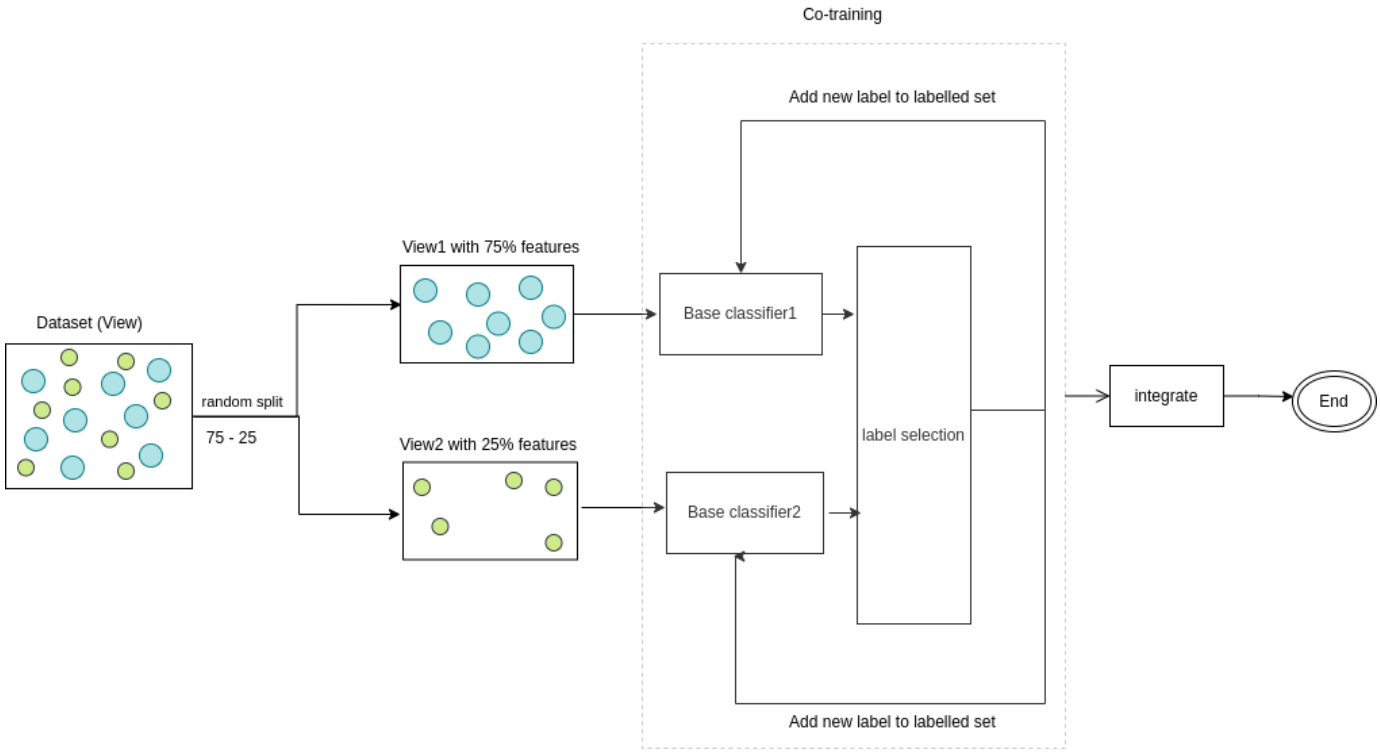

Fig. 2: Random splitting of the views

subsets.The ones that maintain decent quality for both models are then selected. One then continues iterating until one or both views have been assigned to all the features. For automatic feature decomposition of single

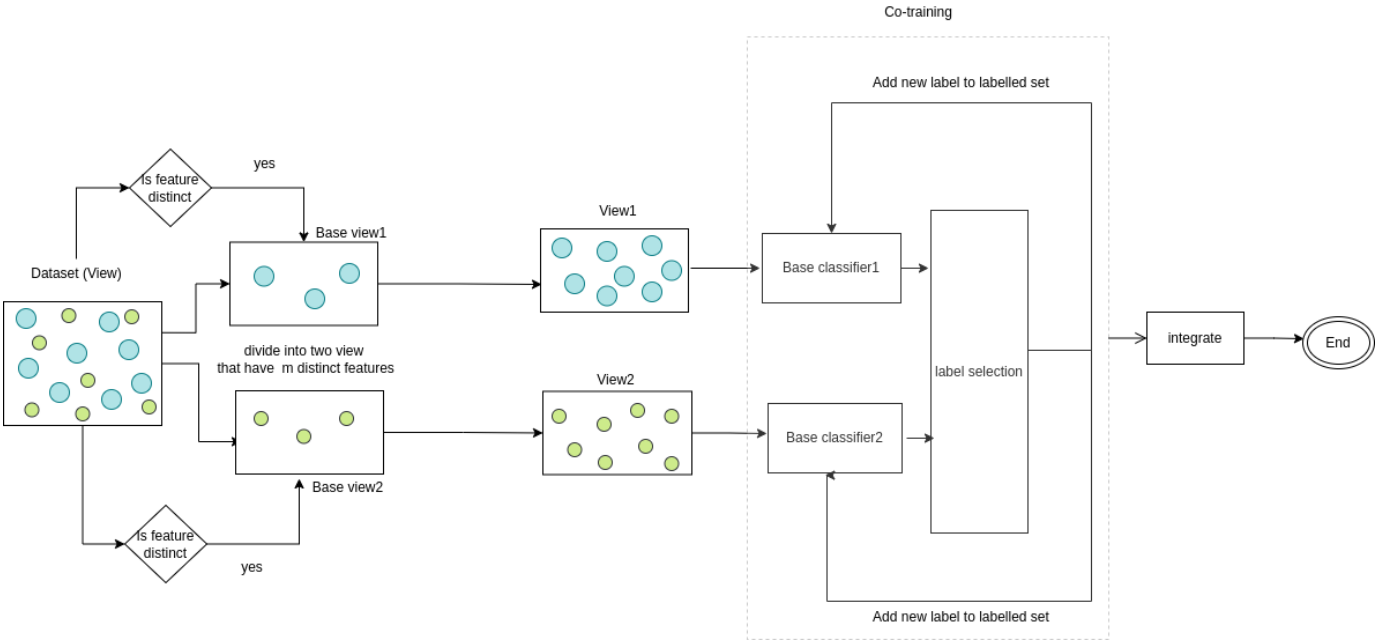

Fig. 3: Automatic splitting of the views

views, [12] proposed an algorithm called PMC (pseudo multi-view co-training). According to their findings, the algorithm splits the attribute set automatically, and each classifier uses only one view after the split. As part of this algorithm, the weights u and v of two classifiers are initialized. At least one of the weights in the same dimension is zero. As a result, there is a constraint condition $u_i v_i = 0$, which is the constraint for the loss function. Therefore, the original data will be split between the two classifiers under the condition of optimal loss function, and two new views will be generated.

B. *View feature correlation*

In reality, there are few dual views that meet the standard co-training requirement of being redundant and conditionally independent. Multi-view learning approaches use subspace-based approaches which consider correlation between views, as correlation is a linear distance between random variables. According to [13], features with a high correlation are also more linearly dependent, and thus they almost equally affect the dependent variable. On the other hand, mutual information measures the distance between two probability distributions in order to determine the degree of dependence between the variables. Most of the time, a truly independent way of splitting views may not perform well. The problem with Blum's co-training is that it is sensitive to both the independence between views and the correlation between features within each view. The separation of views needs to consider not only their independence, but also their dependencies within them. Managing the trade between the two is also crucial for co-training. Taking advantage of feature correlation and conditional mutual information, on the other hand, can enable automatic splitting of views to comply with the independence and sufficiency assumptions for co-training. MaxInd algorithms based on graphs were proposed by [14]. Furthermore, conditional mutual information (CondMI), [15], was used to assess the individual independence of features within each view. In order to determine how much information is shared between features, mutual information was used as an indicator.

Using co-training, the aim is to improve classification performance and reveal its intrinsic mechanisms without sacrificing relevant features with conflicting attributes. Optimising relevant features and limiting irrelevant features are both difficult in real-world application scenarios. A single view can be partitioned based on feature attributes as seen in Figure 4.

The main steps to achieve this are as follows:

- Determine which features are most important in the dataset, rank them, and then select a number of these features for the dual views.
- Use correlation coefficients to determine which features are correlated, process the remaining features based on the correlation coefficients, and then add them to the dual views as other features.
- The two views should be separated as independently as possible based on the correlations and differences between the features using a correlation matrix as a simple heuristic for the feature partitioning.
- The mutual information shared between features is calculated using the formula in 1 where H is the entropy, and X and Y are two data samples with distribution functions $p_x(x)$ and $p_y(y)$.

$$\mathrm{I}(\mathrm{X}:\mathrm{Y}) = \mathrm{H}(\mathrm{X}) + \mathrm{H}(\mathrm{Y}) - \mathrm{H}(\mathrm{X},\mathrm{Y}) \tag{1}$$

- Given mutual information, CondMI can be obtained using

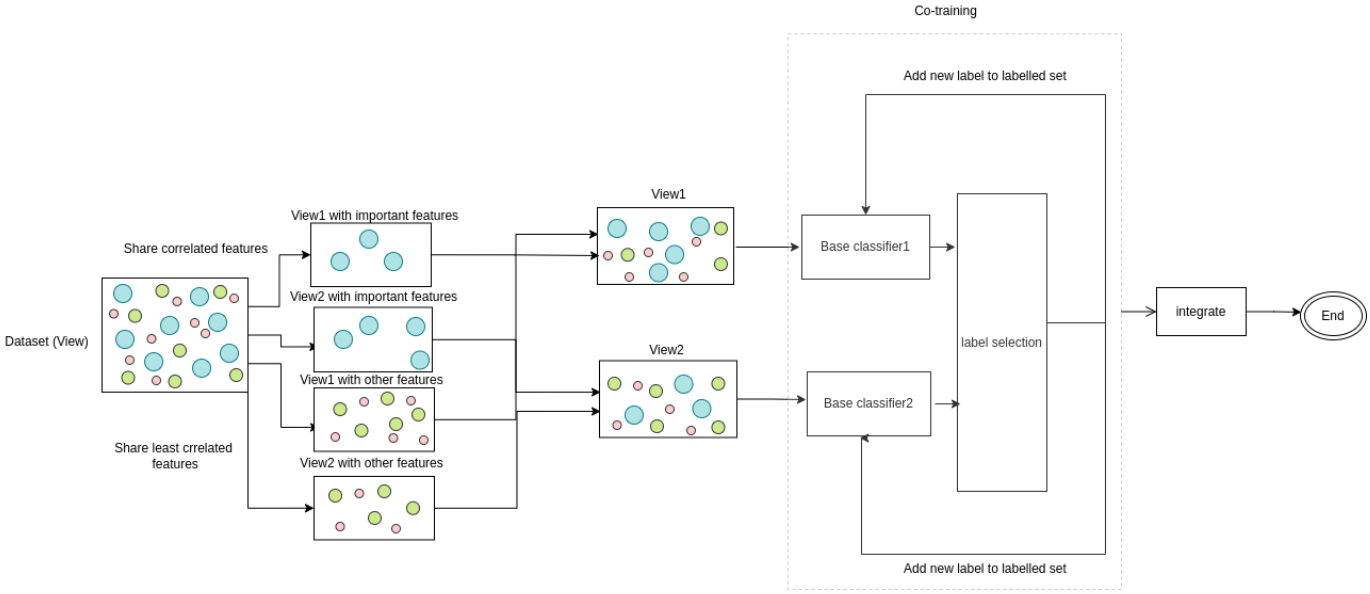

Fig. 4: Splitting of the views based on correlation and conditional mutual information.

2. Here H is the entropy, two data samples X and Y with distribution functions $p_x(x)$ and $p_y(y)$ and Z is a random variable.

$$\mathrm{CondMI}\,(\mathrm{X};\mathrm{Y}|\mathrm{Z}) = \mathrm{H}\,(\mathrm{X}|\mathrm{Z}) + \mathrm{H}\,(\mathrm{X}) - \mathrm{H}\,(\mathrm{X}|\mathrm{Y},\mathrm{Z}) \quad (2)$$

It is possible to ensure the maximum degree of independence between two views by splitting the views and selecting the features in this way. Both these methods have been separately used to enhance the efficiency of the standard co-training algorithm. In [16], the split criteria are based on the confidence and diversity of views, and are compared to random and natural splits. However, in [17], the views are split with conditional mutual information and minimizing their dependency on each other. When the two algorithms are co-trained, their error rates are lower than those of the random partition algorithm. To examine the predictions of co-training on unlabelled training examples, [18] integrates canonical correlation analysis (CCA). The original multiview data is represented in a low-dimensional and closely correlated manner using CCA. Unlabelled examples and original labeled examples are compared based on this representation.

C. *Influence of Domain Knowledge*

The split of views was achieved using certain algorithms in previous methods. Domain knowledge can also influence the splitting of views in the real world. By matching an example in the dataset through a dictionary to extract features, [19] demonstrated that the dictionary-based method could be combined with machine learning. Domain adaptation leverages abundant labelled training data from an unlabelled domain to reduce the effort of collecting labelled training data in the target domain as reviewed in [20]. Many real-world applications have benefited from domain adaptation, including sentiment analysis [21], text classification [22], [23], and visual object recognition [24], [25].

By using common distribution patterns of domain adaptation, one can explore whether a feature subset exists, irrespective of the distribution differences between the source and target domains, that can reduce the distribution difference measured on that subset. It would be possible to extract a set of features as a context-based feature with lexicons that provide a consistent representation of domain knowledge.

The mismatch between the distributions can be reduced by selecting a set of informative features that share similar properties between source and target domains. To prevent classifiers from being influenced by noisy features in the source domain, it is necessary to remove them from the source domain. In addition, supervised training and the manifold hypothesis can be achieved with only a small number of labelled samples.

Depending on the availability of labelled target data, we can divide domain adaptation into two categories: supervised and unsupervised as explained in [26]. For supervised domain adaptation, some labeled target data is used for training; for unsupervised domain adaptation, unlabelled target data is used for training. The algorithm screens the new labelled samples according to the confidence threshold, as shown in Figure 5. Domain knowledge can therefore be used to divide a single view into two sufficient views. The authors of [27] define the boundaries between typical words and biomedical terminology, assigning them based on domain knowledge. Based on domain knowledge, the researchers in [28] scale down the original set of features and then exploit side information from the partial view to enhance the complete model. [29] proposed an unsupervised domain adaptation method that selects a set of informative features. Co-training performance was enhanced by eliminating the influence of noise during the training

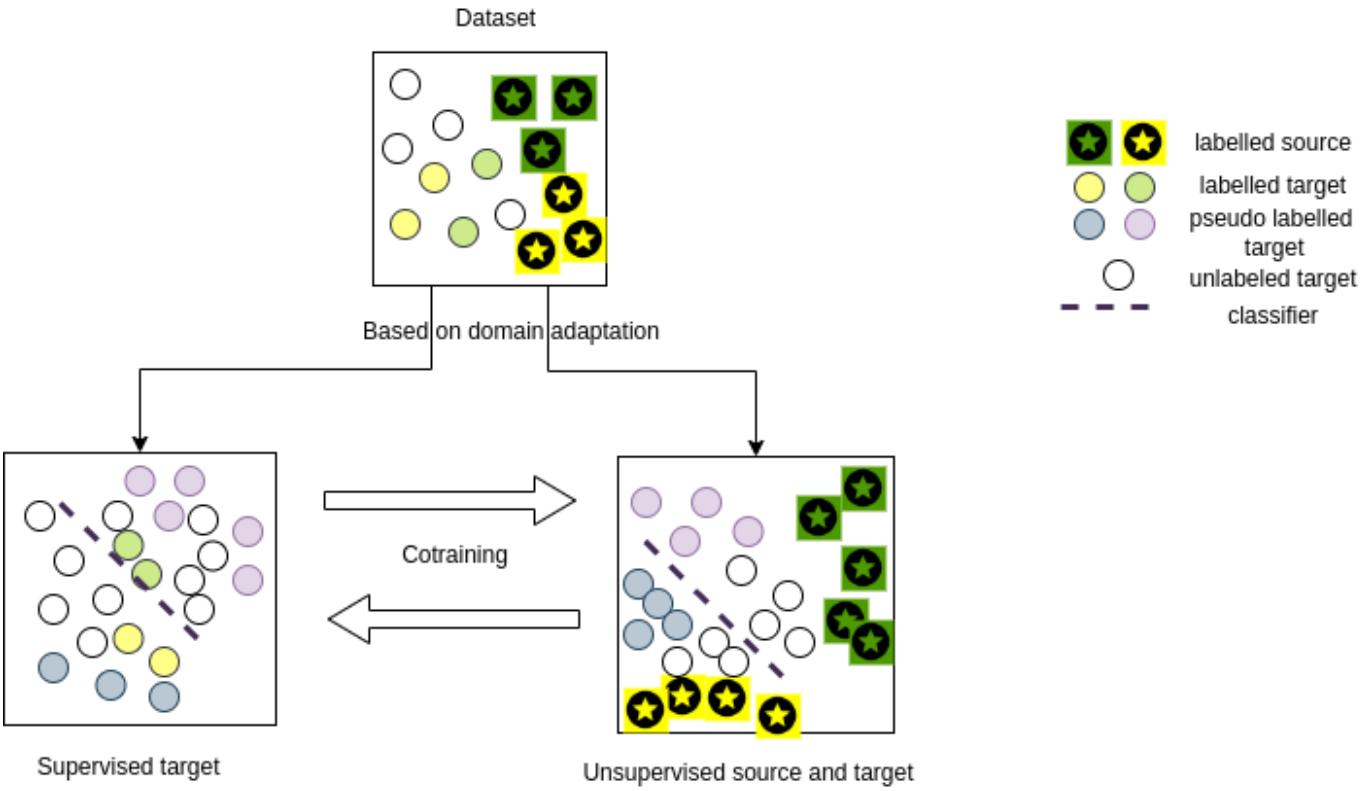

Fig. 5: Using domain adaptation for co-training

process as seen in [30].

Co-training algorithms that use random subspaces are more suitable for processing high-dimensional data, since they require no prior knowledge or complicated mathematical design to find the perfect division. This algorithm needs to calculate how many subspaces are optimal, which is computationally intensive. Information theory and statistics are used to evaluate the independence or sufficiency of views in split algorithms based on correlation or conditional mutual information. In general, it is difficult to determine which should come first, independence or sufficiency. Automatic splitting algorithms are useful because they automatically split views based on the cost function and the optimization algorithm without considering their relationships. The optimization algorithm, however, may limit this kind of algorithm to the local optimal case. This algorithm design eliminates the computational costs associated with exploring better segmentation by using prior knowledge, but requires in-depth knowledge of relevant fields for the algorithm designer.

IV. Calculating Label Confidence

According to [31], a key step in self-training algorithms such as co-training is determining label confidence once the views are generated in accordance with independence and sufficiency assumptions. This prevents learners from labelling unlabelled samples incorrectly, which in turn would result in the algorithm performing poorly by increasing the labelled set incorrectly. Label confidence can be estimated explicitly or implicitly depending on the estimation method. Implicit estimation algorithms calculate the confidence of a pseudo label based on the difference between the learner's results. A confidence score can be calculated explicitly by either adjusting the loss function, comparing classifier predictions using similarity measures, or using a confidence interval.

A. *Implicit calculation*

As a result of implicit estimation, there is no need to calculate the label confidence value, and this method has a lower calculation cost. It avoids the time-consuming ten-fold cross validation process by using a voting mechanism, which operates on the principle that minority obeys majority. As the training process progresses, implicit estimation is highly dependent on base classifiers. The accuracy of implicit estimation is lower than that of explicit estimation since unlabelled data may be mislabelled, thereby degrading the classifiers' performance during iteration. Ensemble learning, as explained in [32], in addition, could be applied to avoid the influence of noise as much as possible and determine what hypothesis to use for the unlabelled data to be selected.

[33] proposed a tri-training version of the co-training algorithm that implicitly calculates label confidence. A supervised model can be trained using unlabelled data using tri-training. The approach addresses the issue of labelled data being scarce, but unlabelled data being plentiful. The labelled

data is used to train three models. By making predictions on the unlabelled data, they add samples to the labelled set where the predictions agree. This results in a larger dataset that is labelled. This is then repeated and re-trained. [34] integrated the voting mechanism into the standard co-training algorithm. [35] addressed mislabelling and noise effects with ensemble methods through random forests.

B. Explicit calculation

A further comparison of average confidence levels between majority learners and minority learners is made by the voting mechanism algorithm, as explained earlier. Unlabelled data can be labeled as long as the confidence average of the majority learners is greater than the average of the minority learners, otherwise it cannot be. To ensure co-training is successful, there must be a specific score or range that can be used to compare the label predictions. This section shall discuss some approaches to confidently pick up the right label.

- **Adjusting the loss function** - Probability distributions can be viewed as labels on training instances. A binary classification can be expressed as a vector $(p0, p1)$, where $p0$ represents the probability that a label will be 0 and p1 the probability of a label being 1. If the correct label of a particular instance x is 0, then that corresponds to the probability distribution (1,0); if the correct label is 1, then that corresponds to the probability distribution (0,1). This distribution is then compared to the prediction from the classifier using the cross-entropy loss.
  Cross-entropy loss has the advantage of being able to evaluate any two distributions. Therefore, the probability distribution (0.8,0.2) corresponds to a confidence level of 0.8 that the correct label for instance x is 0. This cross-entropy can be used to determine how much loss the training instance X contributed to the classifier's predictions based on distribution (0.8,0.2). By summing all instances in the training set, an adjusted loss function can be calculated. Confidence scores can be directly incorporated into a classifier by minimizing this adjusted loss function.
  Self-supervised video representation learning is introduced in [36]. This involves fine-tuning a linear layer and the whole feature encoder end-to-end with cross-entropy loss. [37] co-trains deep neural networks to recognize visual objects by adding layers of self-supervised networks as intermediate inputs, with their outputs diversified through cross-entropy regularization.
- **Use of similarity measures** - Based on clustering and manifold hypotheses, label confidence can be estimated as well. According to clustering hypotheses, similar data belong to the same cluster and have similar labels. It is possible to calculate the confidence score of each unlabelled sample by calculating the classification consistency of the samples closest to the unlabelled sample. The consistency and similarity of classifications can be measured in various ways. Using a cosine similarity score, the similarities between the unlabelled data can be identified to predict the labels. It is beneficial to use cosine similarity to avoid negative correlation because of a lack of data and to deal with scalability and sparsity issues. In the approach described in [38], the classification performance is improved through cosine similarity scores and matrix factorization.
  By calculating the predicted confidence of samples based on these combined likelihoods, [39] illustrates how to use a confidence score that summarizes the adjusted likelihoods across views for multi-label classification. During the iterative co-training process, the selected samples are filtered label-wise, and then filtered labels are communicated to learners.
- **Use a confidence interval** - A confidence interval can be calculated and presented along with the classifier prediction rather than just a single error score. There are two components to a confidence interval:
  - Range - This is the lower and upper limit on the prediction probability that can be expected from the classifier.
  - Probability - It represents the likelihood of the classifier's prediction falling within the range.

  Wilson score intervals [40], are generally used to calculate confidence intervals for classification errors. When a probabilistic sample follows a binomial distribution, the Wilson score is used to estimate the population probability. An expected confidence interval with a range of probabilities is achieved through this.
  In order to measure confidence intervals, cross-validation is used. Authors in [41] have shown that confidence intervals can be used to select the most confident unlabelled examples and to effectively enhance the classification performance.

C. Conclusions

Co-training approaches approaches enable the application of machine learning in cases where the labelled dataset available is of limited size. This paper has provided an overview and categorisation of the potential problems with co-training algorithms and summarized developments and innovations in recent years.

Through co-training, machines can learn from multiple perspectives, whether they are single-view learners or multi-view learners. The primary problems faced by co-training algorithms are how to divide data views effectively and how to estimate label confidence accurately. Although a range of approaches have been suggested and experimental work has been conducted, additional work is still required on co-training algorithms to demonstrate their value and application in industrial settings.

In terms of future work and directions to develop the co-training approach, we suggest the following avenues are explored. Co-training algorithms still rely on currently specific and limited datasets to demonstrate new innovations. Despite this, there does not appear to be a shared dataset for the whole co-training family, and most experiments in literature have

not used a unified dataset where approaches can be compared side by side. Furthermore, larger scale experiments beyond experimental datasets would be helpful. To demonstrate which innovations are effective, some practical or theoretical verification is required because the practical situation is much more complex than the experimental dataset. In both single view and multi view co-training methods, multiple models have to be trained simultaneously, which directly results in high computational costs. Therefore, reducing the computational cost of algorithms needs to be explored. The co-training algorithm should be explored further in terms of the proportion of unlabelled and labelled data it uses. Machine learning regression tasks are difficult to evaluate because the output is continuous. Due to this, co-training algorithms can be used to solve regression problems less frequently, therefore more research is needed to make them applicable in this range of problems.

V. Acknowledgment

The authors would like to acknowledge the support of the Business and Local Government Data Research Centre (grant number ES/L011859/1) funded by the Economic and Social Research Council (ESRC) for undertaking this work

References

[1] S. Ray, "A quick review of machine learning algorithms," in *2019 International conference on machine learning, big data, cloud and parallel computing (COMITCon)*. IEEE, 2019, pp. 35–39.

[2] X. J. Zhu, "Semi-supervised learning literature survey," 2005.

[3] J. E. Van Engelen and H. H. Hoos, "A survey on semi-supervised learning," *Machine Learning*, vol. 109, no. 2, pp. 373–440, 2020.

[4] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *Proceedings of the eleventh annual conference on Computational learning theory*, 1998, pp. 92–100.

[5] K. Nigam and R. Ghani, "Analyzing the effectiveness and applicability of co-training," in *Proceedings of the ninth international conference on Information and knowledge management*, 2000, pp. 86–93.

[6] I. Muslea, S. Minton, and C. A. Knoblock, "Active+ semi-supervised learning= robust multi-view learning," in *ICML*, vol. 2. Citeseer, 2002, pp. 435–442.

[7] S. Kiritchenko and S. Matwin, "Email classification with co-training," in *Proceedings of the 2001 conference of the Centre for Advanced Studies on Collaborative research*. Citeseer, 2001, p. 8.

[8] R. Mihalcea, "Co-training and self-training for word sense disambiguation," in *Proceedings of the Eighth Conference on Computational Natural Language Learning (CoNLL-2004) at HLT-NAACL 2004*, 2004, pp. 33–40.

[9] W. Wang and Z.-H. Zhou, "A new analysis of co-training," in *ICML*, 2010.

[10] S. Qiao, W. Shen, Z. Zhang, B. Wang, and A. Yuille, "Deep co-training for semi-supervised image recognition," in *Proceedings of the european conference on computer vision (eccv)*, 2018, pp. 135–152.

[11] J. Wang, S.-w. Luo, and X.-h. Zeng, "A random subspace method for co-training," in *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*. IEEE, 2008, pp. 195–200.

[12] M. Chen, K. Q. Weinberger, and Y. Chen, "Automatic feature decomposition for single view co-training," in *ICML*, 2011.

[13] M. A. Hall, "Correlation-based feature selection for machine learning," Ph.D. dissertation, The University of Waikato, 1999.

[14] F. Feger and I. Koprinska, "Co-training using rbf nets and different feature splits," in *The 2006 IEEE International Joint Conference on Neural Network Proceedings*. IEEE, 2006, pp. 1878–1885.

[15] J. M. Sotoca and F. Pla, "Supervised feature selection by clustering using conditional mutual information-based distances," *Pattern Recognition*, vol. 43, no. 6, pp. 2068–2081, 2010.

[16] A. Salaheldin and N. E. Gayar, "New feature splitting criteria for co-training using genetic algorithm optimization," in *International Workshop on Multiple Classifier Systems*. Springer, 2010, pp. 22–32.

[17] J. Slivka, A. Kovačević, and Z. Konjović, "Co-training based algorithm for datasets without the natural feature split," in *IEEE 8th International Symposium on Intelligent Systems and Informatics*. IEEE, 2010, pp. 279–284.

[18] S. Sun and F. Jin, "Robust co-training," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 25, no. 07, pp. 1113–1126, 2011.

[19] T. Mitsumori, S. Fation, M. Murata *et al.*, "Gene/protein name recognition based on support vector machine using dictionary as features," *BMC bioinformatics*, vol. 6, no. 1, pp. 1–10, 2005.

[20] K. Weiss, T. M. Khoshgoftaar, and D. Wang, "A survey of transfer learning," *Journal of Big data*, vol. 3, no. 1, pp. 1–40, 2016.

[21] M. Chen, K. Q. Weinberger, and J. Blitzer, "Co-training for domain adaptation," *Advances in neural information processing systems*, vol. 24, 2011.

[22] C. Wan, R. Pan, and J. Li, "Bi-weighting domain adaptation for cross-language text classification," in *Twenty-Second International Joint Conference on Artificial Intelligence*, 2011.

[23] B. Chen, W. Lam, I. Tsang, and T.-L. Wong, "Extracting discriminative concepts for domain adaptation in text mining," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2009, pp. 179–188.

[24] L. Duan, D. Xu, I. W.-H. Tsang, and J. Luo, "Visual event recognition in videos by learning from web data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 9, pp. 1667–1680, 2011.

[25] L. Duan, D. Xu, and S.-F. Chang, "Exploiting web images for event recognition in consumer videos: A multiple source domain adaptation approach," in *2012 IEEE Conference on computer vision and pattern recognition*. IEEE, 2012, pp. 1338–1345.

[26] P. Chaovalit and L. Zhou, "Movie review mining: A comparison between supervised and unsupervised classification approaches," in *Proceedings of the 38th annual Hawaii international conference on system sciences*. IEEE, 2005, pp. 112c–112c.

[27] T. Munkhdalai, M. Li, T. Kim, O.-E. Namsrai, S.-p. Jeong, J. Shin, and K. H. Ryu, "Bio named entity recognition based on co-training algorithm," in *2012 26th International Conference on Advanced Information Networking and Applications Workshops*, 2012, pp. 857–862.

[28] C. M. Nguyen, X. Li, R. D. Blanton, and X. Li, "Partial bayesian co-training for virtual metrology," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 5, pp. 2937–2945, 2020.

[29] F. Sun, H. Wu, Z. Luo, W. Gu, Y. Yan, and Q. Du, "Informative feature selection for domain adaptation," *IEEE Access*, vol. 7, pp. 142 551–142 563, 2019.

[30] R. Kihlman and M. Fasli, "Classifying human rights violations using deep multi-label co-training," in *2021 IEEE International Conference on Big Data (Big Data)*, 2021, pp. 4887–4895.

[31] Z.-H. Zhou, M. Li *et al.*, "Semi-supervised regression with co-training." in *IJCAI*, vol. 5, 2005, pp. 908–913.

[32] H. Kim, H. Kim, H. Moon, and H. Ahn, "A weight-adjusted voting algorithm for ensembles of classifiers," *Journal of the Korean Statistical Society*, vol. 40, no. 4, pp. 437–449, 2011.

[33] Z.-H. Zhou and M. Li, "Tri-training: Exploiting unlabeled data using three classifiers," *IEEE Transactions on knowledge and Data Engineering*, vol. 17, no. 11, pp. 1529–1541, 2005.

[34] S. Karlos, G. Kostopoulos, and S. Kotsiantis, "A soft-voting ensemble based co-training scheme using static selection for binary classification problems," *Algorithms*, vol. 13, no. 1, p. 26, 2020.

[35] M. Li and Z.-H. Zhou, "Improve computer-aided diagnosis with machine learning techniques using undiagnosed samples," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 37, no. 6, pp. 1088–1098, 2007.

[36] T. Han, W. Xie, and A. Zisserman, "Self-supervised co-training for video representation learning," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 5679–5690. [Online]. Available: https://proceedings.neurips.cc/paper/2020/file/3def184ad8f4755ff269862ea77393dd-Paper.pdf

[37] G. Díaz, B. Peralta, L. Caro, and O. Nicolis, "Co-training for visual object recognition based on self-supervised models using a cross-entropy regularization," *Entropy*, vol. 23, no. 4, 2021. [Online]. Available: https://www.mdpi.com/1099-4300/23/4/423

[38] R. Gokhale and M. Fasli, "Matrix factorization for co-training algorithm to classify human rights abuses," in *2018 IEEE International Conference on Big Data (Big Data)*. IEEE, 2018, pp. 2170–2179.

[39] W. Zhan and M.-L. Zhang, "Inductive semi-supervised multi-label learning with co-training," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017, pp. 1305–1314.

[40] R. Bender, "Calculating confidence intervals for the number needed to treat," *Controlled clinical trials*, vol. 22, no. 2, pp. 102–110, 2001.

[41] Y. Zhou and S. Goldman, "Democratic co-learning," in *16th IEEE International Conference on Tools with Artificial Intelligence*. IEEE, 2004, pp. 594–602.