Combinatorial Learning Using Uniview Co-Self-Training and Semi-Supervised K-Means Clustering

Bryant Pham
Alan Garcia
Department of Computer Science
University of Houston
Houston, TX 77004

Abstract

The scarcity of labeled training data is one of the principal challenges in machine learning. This paper presents a novel approach to broaden the scope of exploration of the unlabeled training data space through a combinatorial implementation of co-self-training and semi-supervised K-means clustering. Unlike existing co-training strategies, our combinatorial learning approach focuses on fully exhausting the unlabeled training data by incorporating a semi-supervised K-means clustering technique to explore the unlabeled examples co-self-training cannot confidently predict. By exploring more of the unlabeled training data, classifiers will better represent the target concept and increase the classification performance on unseen data. We assess our Combinatorial Learning approach with experiments on three datasets from the UCI repository.

1. Introduction

Semi-supervised learning models are motivated by practical scenarios in which the amount of unlabeled data is significantly larger than labeled data. Human effort has been the primary means of labeling of data and is consequently an expensive approach. Furthermore, the usage of human perspective in data labeling may introduce noise in the dataset generated from natural human bias and error. In an effort to reduce the cost of data labeling, self-labeling methodologies have become an area of interest to the machine learning community as Triguero, Garcia and Herrera [4] detail in their survey and empirical study of self-labeled techniques for semi-supervised learning. However, the scarcity of labeled training data continues to be one of the principal challenges in optimizing the performance of the data-driven machine learning algorithms.

Amid this ever present disparity of labeled and unlabeled data, several attempts have been made to utilize the abundance of unlabeled data to enhance machine learning algorithms. The implementation of self-training algorithms by Zhou, Kantarcioglu and Thuraisingham [6] stands as one of the earliest efforts to improve the classification performance of learning algorithms by incorporating subsets unlabeled data to compensate for the lack of labeled data. In contrast, Blum and Mitchell [1] opted to boost the performance of a learning algorithm when only a small set of labeled examples are available through multi-view co-training. In this approach, Blum and Mitchell enlarge the training datasets by separately training two learning algorithms on each of the multiple views; the predictions on unlabeled examples of one algorithm becomes a pseudo-labeled example for the other algorithm.

Similarly, our Combinatorial Learning approach is motivated foremost by the scarcity of labeled training data, but also by the general shortage of training data. In an effort to individually address both concerns, our learning approach is a combination of two methods: uniview co-self-training and supervised k-means clustering. Our uniview co-self-training technique addresses the issue concerning a short supply of labeled data through a concurrent self-training variation of the uniview co-training work performed by Goldman and Zhou [2]. Furthermore, the second concern regarding a general shortage of training data is addressed by our implementation of semi-supervised K-means clustering. In reviewing the methodology of several self-training and co-training models, it was evident that a significant portion of the unlabeled training data remained unlabeled following the semi-supervised learning implementation due to the lacking threshold requirements of varying statistical metrics. In review of pattern recognition systems, Jain and Chandrasekaran [3] acknowledge the cost and limitations of acquiring large number of training data samples, however they urge the machine learning community that every effort should be made to obtain as much training data as possible to improve the confidence in performance of the classifier. In an effort to maximize our exploration and usage of the available unlabeled training data, our Combinatorial learning approach implements a semi-supervised K-means clustering technique following the completion of uniview co-self-training.

The content of the paper is organized as follows. In section 2, we discuss the related work we used as the basis of the variations implemented as part of our Combinatorial Learning approach. In section 3, we state the methodology behind our variations on co-training, k-means clustering and the union of the two aforementioned sequential methods that form our Combinatorial Learning approach. The performance of our Combinatorial Learning implementation is assessed through testing of three separate datasets from the UCI repository and a support vector machine supervised-learning classifier. In section 4, we present our experiments and results in detail. Finally, in section 6, we conclude by summarizing our work, presenting the learnings behind our results and stating our future work on the subject of Combinatorial Learning.

2. Related Work

Many approaches have been studied in how to best leverage unlabeled data in a semi-supervised learning setting. In this section, we'll mention prior works from two categories of approaches our idea extends from: multi-classifier co-training and semi-supervised clustering.

2.1 Multi-Classifier Co-Training

Co-training is a machine learning algorithm introduced by Blum and Mitchell [1] to supplement a small set of labeled data with a large unlabeled dataset. Blum and Mitchell's semi-supervised approach consisted of training two independent classifiers with each one utilizing a small labeled dataset based on a different view of the same instances. The two trained classifiers separately classified the same set of unlabeled data and added their k-most confident predictions to the other classifier's labeled set for use in improved classification performance upon retraining.

Goldman and Zhou [2] furthered Blum and Mitchell's multi-view co-training approach to avoid the requirement for data to naturally embody an even split of features. Goldman and Zhou's co-training strategy incorporates a single view of the data and instead allows two different supervised learning algorithms to reveal unique patterns in the data to transfer knowledge amongst the two in the form of high confidence predictions. Our variation of the co-training approach resembles some behavioral characteristics of self-training as described in Zhu's survey of semi-supervised learning [7]. The co-self-training technique we present is essentially a uniview concurrent self-training model that features two different learners, a neural network and a support vector machine. Rather than each learner updating the labeled training dataset of the other learner with pseudo-labeled instances as in pure co-training, our co-self-training approach will update the shared labeled dataset with pseudo-labeled instances meeting a specific confidence threshold T in favor of the same classification. Through this concurrent self-training approach, high confidence pseudo-labeled instances must also be assigned the same class label by both learners. In effect, both learners are able to benefit from the same pseudo-labeled instance in the following training iteration on the labeled dataset.

2.2 Semi-Supervised Clustering

Clustering is an unsupervised learning method where data is grouped in clusters to learn the underlying structure or distribution. Data points close in distance can assume to be similar. This idea extends into semi-supervised learning in which unlabeled data takes the label of the nearest labeled data point.

Wang, Wang, and Shen [5] proposed an improved K-means clustering algorithm by using labeled data to generate good initial clusters to bias clustering towards a global optimum. Their method utilizes optimal initial clusters inferred by labeled data as well as a novel strategy to generate optimal cluster centers for classes without a representative labeled data point. Their results show that all variations of their label-biased K-means clustering out performed the classical K-means approach in all experiments. This study shows that semi-supervised clustering can be used to associate labeled data with unlabeled data, which we will leverage to perform data labeling in our approach.

Our approach differs by aiming to combine the methodologies of co-training and semi-supervised clustering to provide pseudo-labels to unlabeled data. We'll take labels of the highest confidence predictions from co-training, then use labels from clustering to supplement low confidence co-training predictions. In effect, our labeling strategy will capture more of the unlabeled data space with confidence which will allow models to learn on more data.

3. Methodology

In this section, we'll detail our approaches to uniview co-self-training, semi-supervised clustering, and a combination of the two for a hybrid model.

3.1 Uniview Co-Self-Training

As part of the Uniview Co-Self-Training approach, we maintain a set *U* of unlabeled training data, a set *L* consisting of the labeled training data, a set L_{NN} containing the class predictions of the unlabeled training data (initially empty), and a set L_{SVM} containing the class predictions of the unlabeled training data (initially empty).

The shared labeled training set will be used to train both the neural network and a support vector machine. Initially, the complete unlabeled training set will be fed into the neural network model and the support vector machine model to generate each learner's respective pseudo-labeled sets, L_{NN} and L_{SVM}.

Each instance of the pseudo-labeled sets, L_{NN} and L_{SVM}, will be compared with each other to determine if two requirements are met: first, the label prediction of the neural network and support vector machine models agree and second, both learners' prediction confidence must be above a confidence threshold T_C. If an instance satisfies both conditions then it will be added to the labeled training set *L* and removed from the unlabeled training set *U*. Otherwise, the instance remains in the unlabeled set *U*. Once all the instance predictions have been allocated to their corresponding set on a single iteration, then the newly expanded labeled training set will be fed into both supervised learning algorithms, new versions of the neural network and support vector machine models will be generated, the reduced unlabeled training set will be fed into the learners to generate new predictions. This process will be repeated until no label predictions meet the two aforementioned conditions required for inclusion into the labeled training set *L*.

The final output of the co-self-training method is a set of confident pseudo-labeled data and a set of residual low confidence unlabeled data.

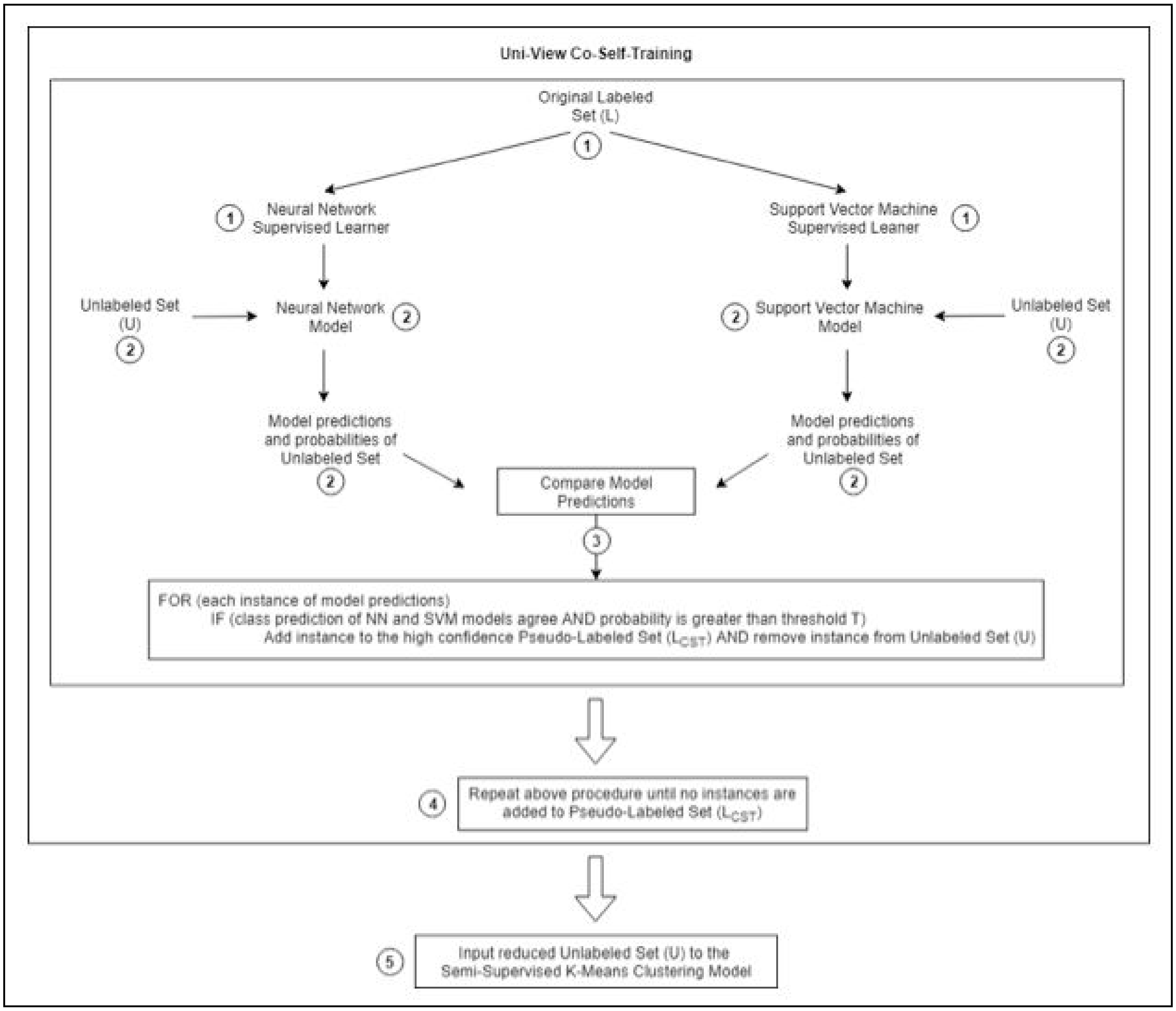

Uni-View Co-Self-Training
Original Labeled Set (L)
1
Neural Network Supervised Learner
Support Vector Machine Supervised Leaner
Unlabeled Set (U)
2
Neural Network Model
Support Vector Machine Model
Unlabeled Set (U)
Model predictions and probabilities of Unlabeled Set
Compare Model Predictions
Model predictions and probabilities of Unlabeled Set
3
FOR (each instance of model predictions)
IF (class prediction of NN and SVM models agree AND probability is greater than threshold T)
Add instance to the high confidence Pseudo-Labeled Set (L$_{CST}$) AND remove instance from Unlabeled Set (U)
4
Repeat above procedure until no instances are added to Pseudo-Labeled Set (L$_{CST}$)
5
Input reduced Unlabeled Set (U) to the Semi-Supervised K-Means Clustering Model

Figure 1. Uniview Co-Self-Training Diagramed Pseudocode.

3.2 Semi-Supervised K-means Clustering

Let *U* be the set of unlabeled data, *L* be the set of labeled data, and *t* be the distance threshold.

Our approach to semi-supervised K-means clustering starts by initializing cluster centroids from points in *L*. Each centroid will represent a respective label in the dataset. For each centroid, we'll take all points from *U* within the Euclidean distance threshold *t*, label them according to their closest centroid, and recompute each cluster center. Repeat giving labels to unlabeled points until no points lie within distance *t* for each centroid. An example can be found in Figure 2.

Because clustering results vary with the parameter *t,* it is important to define what deems the best cluster and optimal *t* for our problem. First, let's define a so called cluster score C_{score}:

$$C\_score = \frac{\text{number of points labeled}}{\text{total distance traveled by centroid}}$$

Where the numerator is the total number of *U* points labeled for the cluster and the denominator is the total distance the centroid traveled till convergence. Note that C_{score} varies with *t* but is not explicit in the above equation because our clustering approach is iterative. Maximizing C_{score} encourages labeling more *U* points while moving the centroid as little as possible; this results in the tightest cluster that captures the most points which we'll refer to as the optimal cluster. The optimal *t* maximizes the sum of C_{score} values for all clusters:

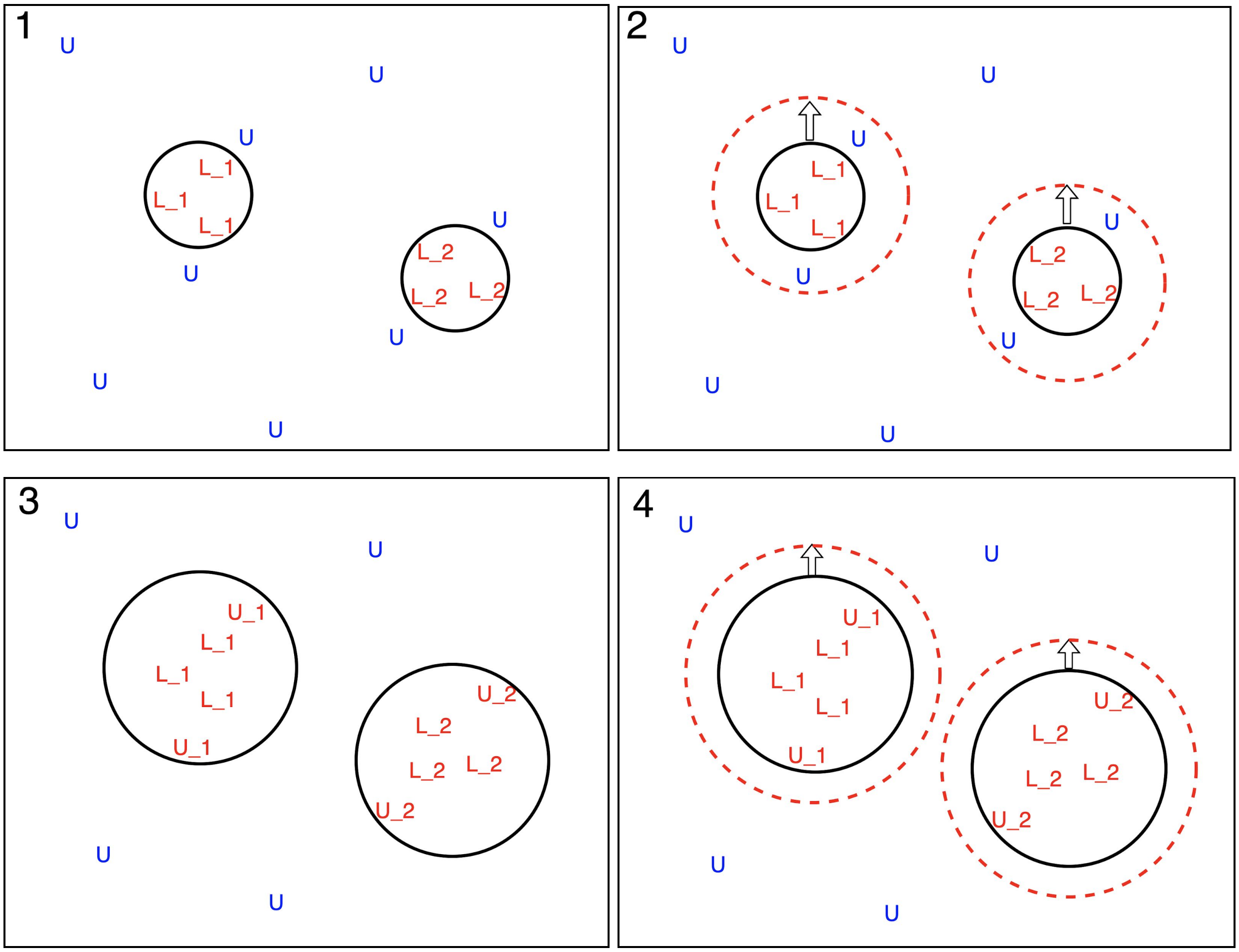
$$\text{t}^* = argmax_t \sum_{i=1}^{n} C\_score_i$$

Maximizing this sum results in optimality for all clusters. This definition of optimal clusters accounts for possible issues in semi-supervised clustering:

1. *U* points too far from the nearest cluster will be considered outliers and will not be labeled. This restricts the algorithm to giving labels only with the highest confidence i.e. the closest *U* points.
2. It is possible the classes in *L* are not representative of all possible classes. Assuming the clusters are well separated, optimizing for C_{score} avoids capturing points from clusters of unknown classes and giving incorrect labels to an entire class of points.

Currently, there is no closed form solution for the optimal *t*, so it is important to compare different *t* values during experimentation to maximize C_{score}.

Figure 2. *Semi-supervised K-means. 1. Clusters are initialized with labeled points. 2. Look for unlabeled points within distance t. 3. Give labels to U points and recompute centroid. 4. Converge if no U points within distance t.*



3.3 Combinatorial Learning

Our Combinatorial Learning approach is the sequential unison of our aforementioned uniview co-self-training and semi-supervised K-means clustering implementations.

The Combinatorial Learning sequence starts with the co-self-training model by training two independent classifiers on the labeled dataset *L*. Next, the co-self-training procedure will iteratively predict pseudo-labels on dataset *U* until no predictions meet the confidence threshold T_C. In effect, the result of the co-self-training procedure will be both a residual unlabeled training dataset and a high confidence pseudo-labeled set L_{CST}.

The semi-supervised K-means clustering model is first trained on dataset *L*, then used to predict pseudo-labels on the residual unlabeled dataset. By training the clustering model only on dataset *L*, we avoid any biases from the co-self-training model in the pseudo-labeled set L_{CST}. The output is a pseudo-labeled set L_{KMC}.

The segmented outputs of our Combinatorial Learning sequence consolidates into a single expanded training dataset featuring the original labeled set L, the confident pseudo-labeled set L_{CST} generated from uniview co-self-training and the confident pseudo-labeled set L_{KMC} generated from semi-supervised K-means clustering. Lastly, the final expanded training dataset is used to train a supervised learning algorithm to produce a final classifier.

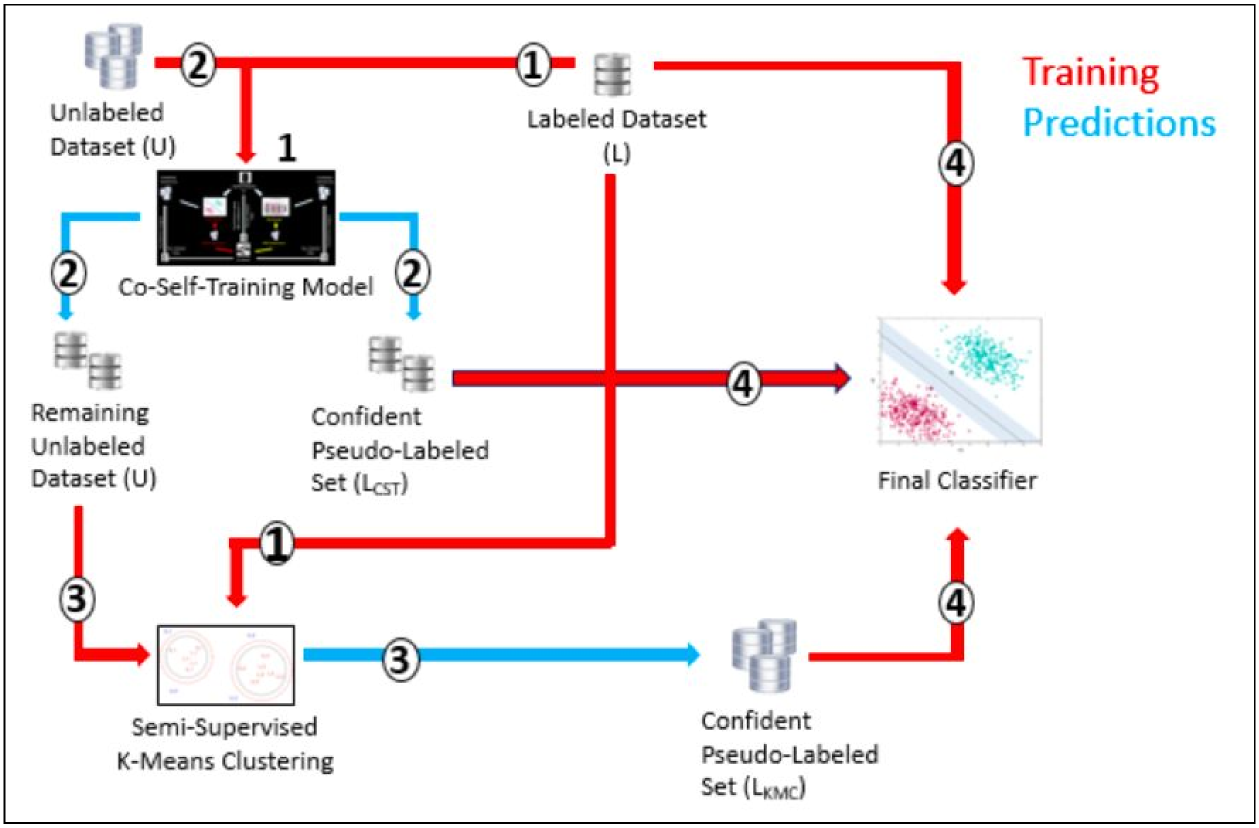

Unlabeled Dataset (U)
2
1
Labeled Dataset (L)
Training
Predictions
1
4
2
Co-Self-Training Model
2
Remaining Unlabeled Dataset (U)
Confident Pseudo-Labeled Set (L_CST)
4
Final Classifier
1
3
3
4
Semi-Supervised K-Means Clustering
Confident Pseudo-Labeled Set (L_KMC)

Figure 3.

Combinatorial Learning Model using Uniview Co-Self-Training and Semi-Supervised K-Means Clustering.

4. Experiments

In this section, we present our experimental results and analyze our findings.

4.1 Methodology

Our experiment starts with building a standalone neural network (NN) and SVM trained and tuned only on labeled data. Parameters from these individual models are then used in the NN and SVM for the co-self-training model. The clustering model is trained as described in section 3.2. The combinatorial model composes the trained co-self-training and clustering models as described in section 3.3. Pseudo-labeled data produced from the clustering, co-training, and combinatorial models go on to train respective final SVMs; with which we use to measure impact of pseudo-labeled data on classification performance.

We tested on three UCI datasets: Adult Income, Breast Cancer, and Congressional Voting Records. For the Breast Cancer and Congressional Voting datasets, sixty percent of the data was sampled to be unlabeled and forty percent as labeled. For the Adult Income dataset, we reduced the dataset by fifty percent then did a sixty/forty split for the unlabeled and labeled sets. The reduced dataset helps alleviate computational load during training on our machine. Samples taken for labeled datasets were class balanced. Final SVM accuracy results are reported from 10-fold cross validation. It is important to note the validation set used to measure accuracy is sampled only from labeled data for all experiments. We cannot assume pseudo-labels to be correct, so it is imperative to validate only on data known to be true.

Table 1a. Breast Cancer results. Confidence threshold: 0.65, Clustering distance threshold: 7

Model	Labeled	Unlabeled	Cotrain Labels Given	Cotrain Label Accuracy	Remaining Unlabeled	Cluster Labels Given	Cluster Label Accuracy	Final SVM Accuracy
Combined	114	172	160	68.34%	12	12	18.93%	68.19%

Table 1b. Breast Cancer results. Confidence threshold: 0.65, Clustering distance threshold: 7

Model	Labeled	Unlabeled	Labels Given	Label Accuracy	Final SVM Accuracy
Co-train	114	172	156	64.65%	67.31%
Clustering	114	172	172	69.77%	67.50%
SVM	114	-	-	-	64.00%
NN	114	-	-	-	56.95%

Table 2a. Adult Income results. Confidence threshold: 0.7, Clustering distance threshold: 4500

Model	Labeled	Unlabeled	Cotrain Labels Given	Cotrain Label Accuracy	Remaining Unlabeled	Cluster Labels Given	Cluster Label Accuracy	Final SVM Accuracy
Combined	6512	9768	4498	88.15%	5270	241	73.86%	58.32%

Table 2b. Adult Income results. Confidence threshold: 0.7, Clustering distance threshold: 4500

Model	Labeled	Unlabeled	Labels Given	Label Accuracy	Final SVM Accuracy
Co-train	6512	9768	4459	88.25%	58.68%
Clustering	6512	9768	512	80.66%	60.41%
SVM	6512	-	-	-	60.46%
NN	6512	-	-	-	50.02%

Table 3a. Voting Records results. Confidence threshold: 0.95, Clustering distance threshold: 4

Model	Labeled	Unlabeled	Cotrain Labels Given	Cotrain Label Accuracy	Remaining Unlabeled	Cluster Labels Given	Cluster Label Accuracy	Final SVM Accuracy
Combined	174	261	219	83.71%	42	42	41.97%	95.24%

Table 3b. Voting Records results. Confidence threshold: 0.95, Clustering distance threshold: 4

Model	Labeled	Unlabeled	Labels Given	Label Accuracy	Final SVM Accuracy
Co-train	174	261	220	83.56%	95.30%
Clustering	174	261	261	74.71%	90.21%
SVM	174	-	-	-	94.72%
NN	174	-	-	-	90.14%

Table 4. Average label confidence per pseudo-label model

Dataset	Model	Avg SVM Label Confidence	Avg NN Label Confidence
Breast Cancer	Combined	64.01%	78.45%
	Co-train	63.69%	78.03%
Adult Income	Combined	61.69%	100.00%
	Co-train	61.52%	100.00%
Voting Records	Combined	90.15%	91.14%
	Co-train	90.06%	92.18%

4.1 Experiment Results

Table 1a and 1b shows results of the Breast Cancer dataset. Pseudo-label models generally outperformed baseline models. Our combined model labeled most unlabeled data in the co-training phase, which led to small performance differences compared to the other pseudo-label models. Although label accuracy for pseudo-label models hover around sixty percent, the number of correct pseudo-labels double the amount of correctly labeled data we have for training: resulting in overall positive transfer. This is only possible because the dataset is small. Because the dataset is so small, correct pseudo-labeled data have large influence in guiding models towards the target concept so long as they outnumber incorrect pseudo-labels.

Table 2a and 2b shows results of the Adult Income dataset. Our combined model increases the labeled data space by about seventy three percent with an overall pseudo-label accuracy of eighty seven percent but still performs worse than the best baseline model (SVM). One reason for this could be because correct pseudo-labeled data are too similar to the given labeled data causing them to be uninformative points; leading to only negative transfer from incorrect pseudo-labeled data. This shows that our approach may decrease classification accuracy in large datasets unless pseudo-label accuracy is extremely high.

Table 3a and 3b shows results of the Voting Records dataset. The dataset is small, so results are similar to findings in the Breast Cancer experiment: our approach more than doubles the labeled set with highly informative and correct pseudo-labeled data, guiding the model closer to the target concept. The combined model only marginally outperforms the best baseline model (SVM) because baseline accuracy is already high, which reduces information gain from correct pseudo-labeled data.

Results from these experiments show that the co-train labeling phase is the bottleneck of our approach. Table 4 displays the average pseudo-label confidence for each predictor in the co-training model. The SVM is the least confident in predicting labels across all experiments. Its average label confidence bottlenecks the confidence threshold parameter used; this forces us to take pseudo-labels predictions without reasonable certainty. Using a higher bias predictor could allow for a higher confidence threshold and therefore more accurate pseudo-label predictions.

5. Contrarian Idea

This next thought is so important that we decided it deserves its own section.

Current co-training literature places high emphasis on only taking pseudo-labels predicted with the highest confidence possible. The issue with this: you're effectively taking pseudo-labels that are most like the labeled points you already have: leading to no information gain and negative transfer from the incorrect pseudo-labels. A contrarian idea would be to test co-training with a confidence threshold that is subpar: possibly forty to sixty percent. Taking pseudo-labels we're not entirely certain about represents our best guess to the most informative queries. If the pseudo-label is correct, we make a considerable leap towards the target concept. If it turns out to be wrong, then it'll reflect in low final model accuracy and we'll increment the confidence threshold. This idea could benefit large dataset-fitted models where label predictions outside the fitted model can help generalization.

6. Conclusions and Future Work

In this paper, we presented a model to combine co-training and semi-supervised K-means clustering to predict pseudo-labels for unlabeled data. Experimental results show that pseudo-labels from our combined model can improve classification accuracy on small datasets, assuming pseudo-label accuracy is adequately high. Datasets with a large amount of labeled data may not benefit from our approach due to the high probability of uninformative pseudo-labels, even if they are correct.

There are many further research opportunities branching off our approach. Our co-self-training model can benefit from replacing the SVM with a higher bias predictor as mentioned in section 4.1.

Another interesting direction would be to flip our approach to have the clustering model as the main pseudo-labeler with the co-self-training model to supplement uncertain predictions. This reversal of roles prioritizes predicting pseudo-labels based on similarity in Euclidean distance as opposed to a probabilistic measure in similarity, which can account for highly informative points a probabilistic approach may miss.

Finally, the contrarian idea explained in section 5 is worth exploring to further understand the relationship between labeled data, unlabeled data, and the various prediction techniques in co-training.

References

1. Blum, A. & Mitchell, T. (2000). Combining Labeled and Unlabeled Data with Co-Training. Proceedings of the Annual ACM Conference on Computational Learning Theory (pp. 92-100).
2. Goldman, S.A., & Zhou, Y. (2000). Enhancing Supervised Learning with Unlabeled Data. *ICML*.
3. Jain, A. K., & Chandrasekaran, B. (1982). Dimensionality and sample size considerations in pattern recognition practice. In *Handbook of Statistics* (Vol. 2, pp. 835-855). Elsevier.
4. Triguero, I., Garcia, S., Herrera, F. (2015) Self-labeled techniques for semi-supervised learning: taxonomy, software and empirical study. Knowledge and Information Systems, vol 42 (pp. 245-284).
5. Wang X., Wang C., Shen J. (2011) Semi–supervised K-Means Clustering by Optimizing Initial Cluster Centers. In: Gong Z., Luo X., Chen J., Lei J., Wang F.L. (eds) Web Information Systems and Mining. WISM 2011. Lecture Notes in Computer Science, vol 6988. Springer, Berlin, Heidelberg
6. Zhou, Y., Kantarcioglu, M. & Thuraisingham, B. (2012) Self-Training with Selection-by-Rejection. *2012 IEEE 12th International Conference on Data Mining*, Brussels (pp. 795-803).
7. Zhu, X. (2006) Semi-supervised learning literature survey. Tech. Rep. TR 1530, University of Wisconsin, Madison.

Appendix

The code is available at https://github.com/bryant-pham/advmlproject