When does Co-Training Work in Real Data?

Charles X. Ling[1], Jun Du[1], and Zhi-Hua Zhou[2]

[1] Department of Computer Science
The University of Western Ontario, London, Ontario, N6A 5B7, Canada
[2] National Key Laboratory for Novel Software Technology
Nanjing University, Nanjing, 210093, China
cling@csd.uwo.ca, jdu42@csd.uwo.ca, zhouzh@lamda.nju.edu.cn

Abstract. Co-training, a paradigm of semi-supervised learning, may alleviate effectively the data scarcity problem (i.e., the lack of labeled examples) in supervised learning. The standard two-view co-training requires the dataset be described by two views of attributes, and previous theoretical studies proved that if the two views satisfy the sufficiency and independence assumptions, co-training is guaranteed to work well. However, little work has been done on how these assumptions can be empirically verified given datasets. In this paper, we first propose novel approaches to verify empirically the two assumptions of co-training based on datasets. We then propose simple heuristic to split a single view of attributes into two views, and discover regularity on the sufficiency and independence thresholds for the standard two-view co-training to work well. Our empirical results not only coincide well with the previous theoretical findings, but also provide a practical guideline to decide when co-training should work well based on datasets.

1 Introduction

Co-training, a paradigm of semi-supervised learning, has drawn considerable attentions and interests recently (see, for example, [1, 2] for review). The standard two-view co-training [3] assumes that there exist two disjoint sets of features or views that describe the data.[3] The standard co-training utilizes an initial (small) labeled training dataset and a (large) set of unlabeled data from the same distribution, and it works roughly as follows [3]. Two separate classifiers are first trained on the initial labeled training dataset using the two views respectively. Then, alternately, each classifier classifies the unlabeled data, chooses the few unlabeled examples whose labels it predicts most confidently, and adds those examples and the predicted labels to the training dataset of the other classifier. The classifiers are retrained, and the process repeats, until some stopping criterion is met. That is, the two classifiers "teach" each other with the additional

[3] Another form of co-training, called the single-view co-training in our paper, generates diverse learners on the single view of attributes [4, 5]. In this paper, we mainly study the standard two-view co-training [3], which will be referred to as two-view co-training, or simply co-training in the paper.

examples whose labels are given by the other classifier to improve the classification accuracy, compared to a classifier learned only from the initial labeled training data.

Two assumptions are proposed for co-training to work well [3]. The first one assumes that the views are sufficient; that is, each view (thus also the combined view) is sufficient to predict the class perfectly. We call it the *sufficiency assumption*. The second assumption requires that the two views be conditionally independent; that is, the two views are independent given the class. We call it the *independence assumption*. Theoretical results have shown that if the sufficiency and independence assumptions are satisfied, co-training is guaranteed to work well. (The assumptions can be relaxed for co-training to still work well [6, 7]. Nevertheless, the sufficiency and independence assumptions are a "sufficient condition" for co-training to work well). In addition, the two-view co-training has been applied quite successfully to many real-world tasks, such as statistical parsing [8], noun phrase identification [9], and image retrieval [10].

However, the two assumptions that guarantee co-training to work well may not be true in most real-world applications. Given a real-world dataset with two views of attributes, how can we judge if the two-view co-training would work well? How can we verify if the sufficiency and independence assumptions are satisfied to guarantee co-training to work well? If the real-world dataset has only one view, can the two-view co-training still work? This paper is our first attempt to answer these questions.

2 Verifying Co-training Assumptions Empirically

Given a whole dataset (with labels) and two views of attributes ($X = x_1, \ldots, x_m$ and $Y = y_1, \ldots, y_n$), how can we verify if the two assumptions on sufficiency and independence for the standard co-training are satisfied? If the assumptions are satisfied, co-training is guaranteed to work well, and thus can be applied. Note that sometimes the domain knowledge can ensure the satisfaction of the two assumptions, but in most real-world applications, such assumptions cannot be guaranteed. Thus it is important that these assumptions be empirically verified based on the dataset given. Here we will use the whole labeled dataset (or a very large training set) that *represents the learning task* to verify the two assumptions. This is because the theoretical assumptions on sufficiency and independence are based on the whole domain (for example, it is assumed that there exist target functions that map from the single view, the X view, and the Y view perfectly [3]).

The sufficiency assumption is relatively easy to verify. Sufficiency means that $X \times Y$ can accurately predict the class, so can X and Y individually. We can build a classifier to estimate the accuracy on the whole dataset D using $X \times Y$ with the 10-fold cross-validation. We denote this accuracy as p. The sufficiency says that p should be close to 1. Note that the theoretical results assume that there exist (target) functions that map from $X \times Y$, X, and Y to the class label perfectly. As we are verifying the assumption empirically, we use learning algorithms on

the whole dataset to establish if such functions exist or not. Similarly, we build a classifier using attributes in X to estimate the accuracy (call it p_x) of X predicting the class, and accuracy (call it p_y) of Y predicting the class.

Thus, the sufficiency assumption of co-training can be defined as: there exists a small positive number δ_1 (such as 0.1) such that

$p > 1 - \delta_1$,
$p_x > 1 - \delta_1$, and
$p_y > 1 - \delta_1$.

We call δ_1 the *sufficiency threshold*. In Section 3.2, we will discover ranges of δ_1 that make co-training work well.

Conditional independence is a bit harder to verify. It means that given the class, the two views are independent. One way to verify this is, for each class label, if each x_i is independent of Y, and each y_i is independent of X. To verify if x_i is independent of Y empirically, we build a classifier (or many classifiers) to predict x_i using Y on the whole dataset. If x_i is independent of Y, then Y cannot predict x_i well — not better than the default accuracy of x_i. Again we establish empirically if Y can predict x_i better than its default accuracy on the whole dataset. Assume that the 10-fold cross-validated accuracy of Y predicting x_i on D is p_{x_i}, then it should not be much larger than the default accuracy of x_i — the accuracy (denoted as p'_{x_i}) of the majority value of the class. The same is true for using X to predict y_j. Thus, the independence assumption can be defined as: there exists a small positive number δ_2 (such as 0.1) such that for each class value

$p_{x_i} < p'_{x_i} + \delta_2$ for all $1 \leq i \leq m$, and
$p_{y_i} < p'_{y_i} + \delta_2$ for all $1 \leq i \leq n$.

We call δ_2 the *independence threshold*. We will establish the ranges of δ_2 to make co-training work in Section 3.2.

3 Splitting Single Views to Two Views

In the previous section we describe an empirical approach to verify, when given the whole dataset and two views, if the two views satisfy the sufficiency and independence assumptions for co-training to work well. However, the standard two-view co-training has limited success in most real-world datasets with single views, such as most UCI datasets [11]. (One could also apply directly the single-view co-training on the datasets with single views, but other complications may be entailed.)

In this section we propose a simple heuristic to split single views into two views such that if the two views satisfy the sufficiency and independence assumptions, the two-view co-training is guaranteed to work well. The heuristic works as follows. We first calculate the entropy of each attribute in the single view based on the whole dataset D, similar to the entropy calculation for all attributes when deciding which attribute should be chosen as the root of the decision tree [12]. Intuitively, the larger the entropy, the more predictive of the class that the attribute would be. In order to distribute high-entropy attributes

evenly in the two views, we simply assign attributes with the first, third, and so on (the odd number of), highest entropy to the first view. We then assign attributes with the second, fourth, and so on (the even number of), highest entropy to the second view. Our proposed method is closely related to [13], which also splits single-views into two views. However, it simply splits the attributes randomly into two views. Later in this section, we make a comparision between our entropy splitting approach and the random splitting approach. After the two views are formed, the two assumptions (sufficiency and independence) for co-training are verified using the approaches described in the previous section.

We choose 32 UCI datasets coming with the WEKA package [14] to see if we can split the single view into two views for co-training to work. The continuous attributes are discredited into 10 equal-width bins in order to utilize naive Bayes [15] for checking the sufficiency and independence assumptions. As most previous co-training researches are based on binary classification problem, datasets with multiple classes are converted to binary by using the majority class value as one class, and the rest of the other values as the other class. These datasets are named with "_new" appended on the end of their original names.

In order to study the range of the sufficiency and independence thresholds for co-training to work well, we set $\delta_1 = 0.5$ for now (a very relaxed value, as any weak binary classifier should predict better than 50%). We apply both entropy splitting and random splitting on these 32 datasets for comparison. Entropy splitting yields smaller δ_1 on most datasets (31 out of 32) and smaller δ_2 on about half datasets (15 out of 32) compared to random splitting, thus we utilize it to verify the working of co-training in the rest of the paper. The cross-validated accuracies on the single view, the X view, and the Y view using naive Bayes on D are listed in Table 1. We use "Acc(X,Y)", "Acc(X)", and "Acc(Y)" to denote them respectively in the table.

Our experiments of applying co-training on these UCI datasets are conducted in the following two high-level steps. In the first step, we run the standard co-training on these datasets to see if co-training would work. For each dataset we also obtain the tightest (smallest) sufficiency and independence thresholds that would make it pass the verification. In the second step, we apply a meta-learning algorithm [16] on the results of the first step to discover proper ranges of the thresholds that can predict when co-training works well. We describe these two steps in details below.

3.1 Applying Co-training on UCI Datasets

To apply co-training on these 32 datasets, the whole datasets D are first split randomly into three disjoint subsets: the training set (R), unlabeled set (U), and test set (T). The test set T is always 25% of D. To make sure that co-training can possibly show improvement when the unlabeled data are added, we choose a small training set for each dataset such that the "optimal gain" in accuracy when using the unlabeled data optimally is large enough (greater than 10%). The "optimal gain", denoted as "OptGain" in Table 1, is thus the difference between the accuracy on the initial training set R plus all unlabeled data with

correct labels and the accuracy on R alone (without any benefit of unlabeled examples). The "optimal gain" reflects the upper bound that co-training can achieve in accuracy. The unlabeled set is the whole dataset taking away the test set and the training set. The proper training set size (with the optimal gain greater than 10%) is also listed in Table 1. The standard co-training [3] is then applied. The process is repeated 20 times with different split of R, U, and T.

The average accuracy before applying co-training (test accuracy of applying naive Bayes on the initial training set; denoted as "IniAcc"), and the average accuracy after applying co-training (denoted as "CtAcc") are recorded in the table. A significance test, a paired t-test with 95% confidence, is applied to see if the test accuracy after co-training is significantly better than the test accuracy before co-training (i.e., if co-training really works or not). If it is, then co-training wins, denoted by W in the "CtWorks?" column; if it is significantly worse, then co-training loses (L); else co-training ties (T) with no co-training. These results are presented together in Table 1 for easy viewing.

From Table 1, we can see that overall, co-training wins in 6 datasets, loses in 3, and ties in the rest 23 datasets. Of course this does not imply that co-training does not work well for most single-view real-world datasets, as the sufficiency and independence thresholds (δ_1 and δ_2) are set very relaxed ($\delta_1 = 0.5$), thus the two views of these datasets may not be sufficient or independent. For each dataset, we can obtain the tightest (smallest) threshold values for the sufficiency and independence assumptions to pass. These threshold values (δ_1 and δ_2) are also listed in Table 1. These values provide us with an opportunity to discover the hidden regularity of these thresholds that make co-training win.

3.2 Meta-learning Co-training Thresholds

Results in Table 1 do seem to indicate that co-training would win when δ_1 and δ_2 are relatively small. In order to obtain a more precise range of δ_1 and δ_2, we use the idea of meta-learning to find hidden regularity of δ_1 and δ_2 that makes co-training work (win). We simply take, from Table 1, the numerical values in columns δ_1 and δ_2 as attributes, and W, L or T from "CtWorks?" as the class label. We obtain 32 training examples on which we can apply meta-learning.

We first assign W (win) as one class, and group L (lose) and T (tie) as the "others" class, to discover when co-training would win (W). As we expect simple rules for the thresholds, we apply WEKA's j48, the standard decision-tree algorithm [12] on the 32 training examples with pruning. The decision tree found is surprisingly simple:

```
d1 <= 0.23
|   d2 <= 0.15: W (7.0/2.0)
|   d2 > 0.15: others (8.0/1.0)
d1 > 0.23: others (17.0)
```

The tree discovered by j48 clearly indicates that co-training would win if the sufficiency threshold (δ_1) is less than or equals to 0.23, and the independence

Dataset	Acc(X,Y)	Acc(X)	Acc(Y)	δ_1	δ_2	Training	OptGain	IniAcc	CtAcc	CtWorks?
breast-cancer	75.5%	73.3%	74.0%	0.27	0.32	1/50	28.5%	0.44	0.48	T
breast-w	97.3%	96.6%	95.4%	0.05	0.13	1/100	17.2%	0.80	0.90	W
colic	78.8%	75.0%	81.8%	0.25	0.13	1/50	21.0%	0.60	0.63	T
credit-a	84.8%	84.6%	74.1%	0.26	0.31	1/50	23.0%	0.62	0.60	T
credit-g	76.3%	72.0%	73.4%	0.28	0.18	1/100	16.2%	0.58	0.58	T
diabetes	75.4%	69.7%	74.9%	0.30	0.12	1/50	15.1%	0.62	0.64	T
heart-statlog	83.7%	79.6%	76.7%	0.23	0.06	1/50	26.6%	0.59	0.73	W
hepatitis	83.9%	83.2%	82.6%	0.17	0.19	1/50	17.6%	0.69	0.70	T
ionosphere	90.9%	89.7%	90.1%	0.10	0.35	1/50	25.2%	0.66	0.69	T
kr-vs-kp	87.9%	70.4%	80.1%	0.30	0.27	1/200	24.0%	0.64	0.50	L
mushroom	95.8%	92.9%	98.5%	0.07	0.54	1/500	15.9%	0.80	0.82	T
sonar	77.4%	76.4%	76.0%	0.24	0.23	1/50	24.1%	0.53	0.52	T
tic-tac-toe	69.6%	70.5%	65.3%	0.35	0.17	1/100	15.5%	0.58	0.56	T
vote	90.1%	86.9%	91.0%	0.13	0.26	1/50	23.0%	0.68	0.60	T
anneal_new	92.4%	82.9%	92.5%	0.17	0.49	1/50	13.9%	0.77	0.71	L
arrhythmia_new	75.2%	74.6%	72.8%	0.27	0.34	1/50	21.9%	0.52	0.56	T
autos_new	79.0%	81.0%	72.2%	0.28	0.62	1/50	24.9%	0.57	0.55	T
cmc_new	65.4%	66.1%	61.0%	0.39	0.21	1/100	14.5%	0.52	0.50	T
cylinder-bands_new	73.9%	67.4%	73.8%	0.33	0.51	1/50	16.8%	0.57	0.51	L
dermatology_new	99.7%	99.5%	99.7%	0.01	0.20	1/50	27.1%	0.73	0.86	W
ecoli_new	97.7%	85.4%	83.9%	0.16	0.11	1/50	40.5%	0.57	0.70	W
flags_new	75.3%	73.7%	74.2%	0.26	0.32	1/50	20.0%	0.54	0.52	T
glass_new	72.4%	69.2%	68.2%	0.32	0.24	1/50	21.5%	0.53	0.55	T
haberman_new	75.8%	75.2%	74.2%	0.26	0.02	1/50	29.0%	0.46	0.49	T
heart-c_new	84.2%	80.2%	76.9%	0.23	0.10	1/50	20.0%	0.65	0.69	T
heart-h_new	84.0%	78.6%	80.3%	0.21	0.12	1/50	20.2%	0.66	0.75	W
liver-disorders_new	64.9%	60.0%	64.9%	0.40	0.03	1/50	18.0%	0.47	0.49	T
primary-tumor_new	85.0%	81.7%	79.1%	0.21	0.12	1/50	17.4%	0.67	0.66	T
solar-flare_1_new	74.0%	73.4%	74.9%	0.27	0.27	1/50	27.5%	0.47	0.50	T
solar-flare_2_new	100%	100%	81.4%	0.19	0.18	1/50	24.7%	0.75	0.69	T
spambase_new	84.5%	79.9%	78.6%	0.21	0.15	1/300	28.8%	0.57	0.75	W
splice_new	92.3%	84.1%	86.3%	0.16	0.19	1/200	25.9%	0.66	0.67	T

Table 1. Applying the standard co-training on UCI datasets after view split. The cross-validated accuracies on the single view, the X view, and the Y view using naive Bayes on the whole datasets are listed as "Acc(X,Y)", "Acc(X)", and "Acc(Y)" respectively. "Training" indicates the size of training set, the test set is always 25% of the whole dataset, and the rest is the unlabeled set. "OptGain" is the difference between the accuracy on the initial training set plus all unlabeled data with correct labels and the accuracy on initial training set alone (without any benefit of unlabeled examples). "IniAcc" and "CtAcc" indicate the average test accuracy before and after applying co-training respectively. "CtWorks?" shows if co-training wins, ties or loses based on a paired t-test with 95% confidence.

threshold (δ_2) is less than or equals to 0.15. There are 3 exceptions to the simple tree, but the overall accuracy of this tree is quite high at 91%, much higher than the default accuracy of 81% (26/32).

It would also be interesting to find out when co-training would lose (L) so it should be avoided. We use L (lose) as one class, and group win (W) and tie (T) as the "others" class, and run j48 again.[4] The result is also surprisingly simple:

```
d2 <= 0.26: others (21.0)
d2 > 0.26
|   d1 <= 0.28: others (9.0/1.0)
|   d1 > 0.28: L (2.0)
```

This indicates that co-training would lose (so it should not be used) if the sufficiency threshold (δ_1) is greater than 0.28, and the independence threshold (δ_2) is greater than 0.26. Clearly our empirical results coincide well with theoretical findings that if the two views are sufficient and independent, co-training must work well (win). However, theoretical guarantee on sufficiency and independence is often impossible to obtain. What is more important is that the actual range of the sufficiency and independence thresholds, though discovered empirically here, provides a simple guideline for deciding and applying the standard co-training in real-world datasets.

4 Conclusions

To summarize, in this paper we propose empirical verification of the sufficiency and independence assumptions of the standard two-view co-training algorithm. We design heuristic to split datasets with a single view into two views, and if the two views pass the sufficiency and independence verification discovered by meta-learning, co-training is highly likely to work well. Our conclusions coincide well with the previous theoretical results, but our work provides a practical guide as to when co-training can work in datasets with two views. Our current work is based on the whole dataset. In our future work, we will study co-training verification on small training data.

Acknowledgments

Z.-H. Zhou was supported by NSFC (60635030, 60721002), JiangsuSF (BK2008018) and 863 Program (2007AA01Z169).

References

1. Chapelle, O., Schölkopf, B., Zien, A., eds.: Semi-Supervised Learning. MIT Press, Cambridge, MA (2006)

[4] We tried to run j48 on the training examples with three classes: W, L and T, but it always returns a one-node tree, with or without pruning.

2. Zhu, X.: Semi-supervised learning literature survey. Technical Report 1530, Department of Computer Sciences, University of Wisconsin at Madison, Madison, WI (2006)
3. Blum, A., Mitchell, T.: Combining labeled and unlabeled data with co-training. In: Proceedings of the 11th Annual Conference on Computational Learning Theory, Madison, WI (1998) 92–100
4. Goldman, S., Zhou, Y.: Enhancing supervised learning with unlabeled data. In: Proceedings of the 17th International Conference on Machine Learning, San Francisco, CA (2000) 327–334
5. Zhou, Z.H., Li, M.: Tri-training: exploiting unlabeled data using three classifiers. IEEE Transactions on Knowledge and Data Engineering **17**(11) (2005) 1529–1541
6. Abney, S.: Bootstrapping. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, Philadelphia, PA (2002) 360–367
7. Balcan, M.F., Blum, A., Yang, K.: Co-training and expansion: Towards bridging theory and practice. In Saul, L.K., Weiss, Y., Bottou, L., eds.: Advances in Neural Information Processing Systems 17. MIT Press, Cambridge, MA (2005) 89–96
8. Sarkar, A.: Applying co-training methods to statistical parsing. In: Proceedings of the 2nd Annual Meeting of the North American Chapter of the Association for Computational Linguistics, Pittsburgh, PA (2001) 95–102
9. Pierce, D., Cardie, C.: Limitations of co-training for natural language learning from large data sets. In: Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing, Pittsburgh, PA (2001) 1–9
10. Zhou, Z.H., Chen, K.J., Jiang, Y.: Exploiting unlabeled data in content-based image retrieval. In: Proceedings of the 15th European Conference on Machine Learning, Pisa, Italy (2004) 525–536
11. Blake, C., Keogh, E., Merz, C.J.: UCI repository of machine learning databases. [http://www.ics.uci.edu/∼mlearn/MLRepository.html], Department of Information and Computer Science, University of California, Irvine, CA (1998)
12. Quinlan, R.J.: C4.5: programs for machine learning. Morgan Kaufmann Publishers Inc. (1993)
13. Nigam, K., Ghani, R.: Analyzing the effectiveness and applicability of co-training. In: Proceedings of the 9th ACM International Conference on Information and Knowledge Management, Washington, DC (2000) 86–93
14. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques. Second edn. Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann (June 2005)
15. Langley, P., Iba, W., Thompson, K.: An analysis of bayesian classifiers. In: National Conference on Artificial Intelligence. (1992) 223–228
16. Vilalta, R., Drissi, Y.: A perspective view and survey of meta-learning. Artificial Intelligence Review **18**(2) (June 2002) 77–95