A Collaborative Ability Measurement for Co-training

Dan Shen[1,2], Jie Zhang[1,2], Jian Su[1], Guodong Zhou[1], and Chew-Lim Tan[2]

[1] Institute for Infocomm Research, 21 Heng Mui Keng Terrace, Singapore, 119613
{shendan,zhangjie,sujian,zhougd}@i2r.a-star.edu.sg
[2] Department of Computer Science, National University of Singapore, Singapore, 117543
{shendan,zhangjie,tancl}@comp.nus.edu.sg

Abstract. This paper explores collaborative ability of co-training algorithm. We propose a new measurement (CA) for representing the collaborative ability of co-training classifiers based on the overlapping proportion between certain and uncertain instances. The CA measurement indicates whether two classifiers can co-train effectively. We make theoretical analysis for CA values in co-training with independent feature split, with random feature split and without feature split. The experiments justify our analysis. We also explore two variations of the general co-training algorithm and analyze them using the CA measurement.

1 Introduction

Co-training and several viable alternatives of co-training, such as Co-Boosting (Collins and Singer [1]), Co-Testing (Muslea et al. [2]) and Self-Training (Ng and Cardie [3]), have been successfully applied to a number of natural language processing (NLP) tasks. Most of the works follow the general co-training algorithm proposed by Blum and Mitchell [4] as shown in Fig. 1.

Given:
- a small labelled data set L
- a large unlabelled data set U
- two views V_1 and V_2 of a classification task

Loop for k iterations:
- Use L to train the classifier h_1 that considers V_1
- Use L to train the classifier h_2 that considers V_2
- Allow h_1, h_2 to label the data in U
- Select the labelled data from U and add them to L

Fig. 1. General co-training algorithm.

The performance of co-training relies on the *collaborative ability* of the two classifiers. With strong collaborative ability, each classifier can provide reliable and informative labelled data to the other iteratively and complement the other's predictions when labelling test data. On the contrary, they can not help each other with weak collaborative ability. In this paper, we propose a new measurement (CA) to represent the collaborative

K.-Y. Su et al. (Eds.): IJCNLP 2004, LNAI 3248, pp. 436–445, 2005.

ability based on the overlapping proportion between the classifiers' certain and uncertain instances in test data set. We also make the theoretical analysis for the collaborative ability of classifiers with independent view split, with random view split and without view split. Experiments on text classification task justify our analysis. Based on the CA measurement, we are able to understand more about the learning behavior of co-training and give the suggestions that whether the two classifiers can be used for effective co-training. It will be useful for designing the co-training model in practice. In addition, we discuss the difference and relation between the CA measurement and the Uncertainty Correlation Coefficient (UCC) Measure proposed by Cao et al. [5]. Furthermore, we explore two variations of the general co-training algorithm when independent view split does not exist. Using the CA measurement, we analyze the collaborative abilities of the classifiers in these variations.

2 Collaborative Ability of Co-training

Collaborative ability of co-training can be considered from two aspects as follows:

- Can one classifier provide reliable and informative labelled data to the other iteratively?
- Can one classifier complement the other's prediction when labeling test data?

We estimate the collaborative ability based on the overlapping proportion between two classifiers' certain and uncertain instances in the test data set.

2.1 Collaborative Ability (CA) Measurement

We define the co-training model as follow:
X: a set of data Y: a set of class
Given a training data set, two co-training classifiers are to learn two functions h_1 and h_2, which map X to Y: $h_1 : X \rightarrow Y$ $h_2 : X \rightarrow Y$

Definition 1: The *Certainty Set* of classifier h on X is defined as:

$$CSet(h) = \{x | Conf(h(x)) \geq \alpha_1, x \in X\}$$

Definition 2: The *Uncertainty Set* of classifier h on X is defined as:

$$UCSet(h) = \{x | Conf(h(x)) < \alpha_2, x \in X\}$$

where α_1 and α_2 denotes a predefined threshold for certainty and uncertainty respectively ($\alpha_1 \geq \alpha_2$). $Conf(h(x))$ denotes the confidence score of $h(x)$.

Definition 3: The *collaborative ability of one classifier h_1 to the other classifier h_2* is defined as:

$$CA(h_2|h_1) = P(\{x | x \in CSet(h_1) \cap UCSet(h_2)\})$$

$CA(h_2|h_1)$ indicates the portion of the reliable and informative labelled data which h_1 can provide to h_2. If $CA(h_2|h_1)$ is high, there are a large portion of data which is very confident for h_1 and is not confident for h_2. In this case, h_1 can provide such data to h_2, and help h_2 to enhance the performance.

Similarly, we define the collaborative ability of the classifier h_2 to the classifier h_1 as follows:

$$CA(h_1|h_2) = P(\{x|x \in CSet(h_2) \cap UCSet(h_1)\})$$

Generally speaking, if $CA(h_2|h_1)$ is much higher than $CA(h_1|h_2)$, it is probably because h_2 is not sufficient enough for the learning task or the features h_2 used are too weak. In this case, the co-training process may benefit from h_1 rather than from both classifiers. Furthermore, when labelling test data, the predictions made by h1 will play the dominant role and the whole performance of the combined classifier may be close to the performance of h_1.

Definition 4: The *collaborative ability between the two classifiers* h_1 *and* h_2 is defined as:

$$\begin{aligned} CA_{h_1h_2} &= CA(h_2|h_1) + CA(h_1|h_2) \\ &= P(\{x|x \in (CSet(h_1) \cap UCSet(h_2))\}) \\ &\quad + P(\{x|x \in (CSet(h_2) \cap UCSet(h_1))\}) \end{aligned}$$

$CA_{h_1h_2}$ presents the degree of collaboration between h_1 and h_2. It can be used to determine whether the feature split is suitable for the learning task and the two classifiers are able to help each other.

If $CA_{h_1h_2}$ is low, it can be explained that the two classifiers' confidence degree for most of the data are consistent. There are two cases:

First, there is a large number of data of which both classifiers are very confident. These data are reliable but not informative. In the co-training, the occurrence of this case may indicate that the two classifiers have been bootstrapped successfully and the room for further improvement may not be large.

Second, there is a large number of data for which both classifiers cannot confidently predict. These data may be informative but not reliable. In the co-training, it indicates that the two classifiers are not effective enough, since one classifier can not provide reliable labelled data to the other.

Our CA measurement is different from the Uncertainty Correlation Coefficient (UCC) Measure proposed by Cao et al. [5]. UCC is defined as the portion of instances of which both classifiers are uncertain. The authors also state that the lower the UCC values are, the higher the performances can be achieved in co-training. The idea of our CA measurement is consistent with their statement. Furthermore, we consider not only the Uncertainty Correlation Coefficient (UCC) factor but also Certainty Correlation Coefficient (CCC) factor. As analyzed above, the lower the UCC and CCC values are, the stronger the collaborative ability of the classifiers is. In practice, we may compute the CA value as following:

$$\begin{aligned} N_{CUC1} &= |CSet(h_1) \cap UCSet(h_2)| \\ N_{CUC2} &= |UCSet(h_1) \cap CSet(h_2)| \\ N_{CC} &= |CSet(h_1) \cap CSet(h_2)| \\ N_{UCUC} &= |UCSet(h_1) \cap UCSet(h_2)| \\ CA_{h_1h_2} &= \frac{N_{CUC1} + N_{CUC2}}{N_{CUC1} + N_{CUC2} + N_{CC} + N_{UCUC}} \end{aligned}$$

2.2 Theoretical Analysis of CA Measurement

In order to make our theoretical analysis simpler, we set the threshold α_1 equal to α_2. Therefore, for all labelled data x,

$$P(\{x|x \in CSet(h)\}) + P(\{x|x \in UCSet(h)\}) = 1$$

Suppose: two events A and B, where A is the event that an instance belongs to $CSet$ of h_1 and B is the event that an instance belongs to $CSet$ of h_2.

$$P(A) = P(\{x|x \in CSet(h_1)\})$$
$$P(B) = P(\{x|x \in CSet(h_2)\})$$

Then:

$$P(\neg A) = P(\{x|x \in UCSet(h_1)\}) = 1 - P(A)$$
$$P(\neg B) = P(\{x|x \in UCSet(h_2)\}) = 1 - P(B)$$

Compute the CA value:

$$\begin{aligned} CA_{h_1h_2} &= P(A, \neg B) + P(\neg A, B) \\ &= P(\neg B|A)P(A) + P(\neg A|B)P(B) \\ &= P(A) + P(B) - 2P(A, B) \end{aligned}$$

Let's consider the CA values in the co-training with independent view split, with random view split and without view split.

- **Independent View Split**
  In this case, the two classifiers have independent conditional probability for labelling the data with some confidence. That is, $P(A, B) = P(A)P(B)$. Therefore, the CA value will be:

  $$CA^{INDEP}_{h_1h_2} = P(A) + P(B) - 2P(A)P(B)$$

- **Random View Split**
  If an independent view split is not available, the event A and B may be dependent on each other. That is, if one classifier is certain of an instance, the other classifier is more likely to be certain than uncertain of it and vice versa. That is,

  $$P(A|B) \geq P(A) \text{ and } P(B|A) \geq P(B)$$

  Therefore,

  $$P(A) + P(B) - 2P(A)P(B) \geq P(A) + P(B) - 2P(A, B)$$

  $$CA^{INDEP}_{h_1h_2} \geq CA_{h_1h_2}$$

  With the theoretical analysis above, we show that the CA value in the co-training with independent view split is higher than that in the co-training without independent feature split. It indicates that the classifiers with separate and redundant

views will have the better collaborative ability. Our analysis has been supported by many previous research works (Blum and Mitchell [4]; Nigam and Ghani [6]), as they also stated that co-training with independent view split outperforms other co-training and variations. Furthermore, we find that the two classifiers may still be bootstrapped from each other given the acceptable collaborative ability even if there does not exist an independent view split. In most practical applications, we can use the CA measurement to estimate the collaborative ability and decide whether the two classifiers with certain feature split can be used for effective co-training.

– **Single View (Self-training)**
We also evaluate the collaborative ability of the co-training without view split (self-training), which may be considered as a special case of general co-training algorithm. In this case,

$$P(A) = P(B) = P(A|B)$$

Therefore, the CA value of self-training is zero, i.e. one classifier can only be bootstrapped by itself.

2.3 Experiment Results

We choose text classification task to verify the effectiveness of CA measurement. Our experiment is similar to Nigam and Ghani [6] and Cao et al. [5]. We use four newsgroups from the 20 Newsgroups dataset to produce a new data set, as shown in Table 1. A preprocessing procedure is conducted on the texts in each group in order to remove newsgroup header and stop words, scale the length of text to the same, and eliminate some invalid texts, as in (Nigam and Ghani [6]).

Table 1. Data set with natural split feature set.

Class	Feature Set A	Feature Set B
Pos	comp.os.ms-windows.misc	talk.politics.misc
Neg	comp.sys.ibm.pc.hardware	talk.politics.guns

We produce three data sets as follows:

– **Data Set with Independent View Split**
The two newsgroups in the first row of Table 1 constitute the positive class by joining together randomly selected documents from each of them. Similarly, the two newsgroups in the second row constitute the negative class. Thus, a two-class data set is produced. Since the vocabulary of the first column (Feature Set A) is different from the vocabulary of the second column (Feature Set B), the data set can be regarded as view independent. We denote this data set as *IndepViewSet*.
– **Data Set with Random View Split**
We also produce a data set with random split view as in (Cao et al. [5]). In which, the whole feature set in the above data set is randomly split into two subsets, which may not be independent. We denote this data set as *RandViewSet*.

– **Data Set with Single View**
  We use the whole feature set as a single view for this data set. It is produced for the purpose of self-training and two co-training variations which will be introduced in Section 4. We denote this data set as *SingleViewSet*.

In order to show the CA measurement in an empirical way, we conduct experiments based on the three data sets. We separate each data set into 3 positive and 3 negative instances as an initial training set, 500 positive and 500 negative instances as an unlabelled set and the rest about 900 instances as a test set. The separation made on the three data sets is identical. We use Naive Bayes classifiers as the learning algorithms in this co-training task. The certainty threshold α_1 is set to 0.8 and the uncertainty threshold α_2 is set to 0.6.

We train two Naive Bayes classifiers on the *IndepViewSet* for co-training, two naive Bayes classifiers on the *RandViewSet* for co-training and one naive Bayes classifier on the *SingleViewSet* for self-training. In each iteration, we select one most confident positive instance and one most confident negative instance of each classifier from the unlabelled set into the training set until the unlabelled set is empty.

Fig. 2 shows the three charts of experiment results. From the three charts, we can find that CA measurement of *IndepViewSet* co-training (around 0.5 0.6) is higher than those of *RandViewSet* co-training (around 0.3) and *SingleViewSet* self-training (zero). All of the findings above are consistent with our theoretical analysis.

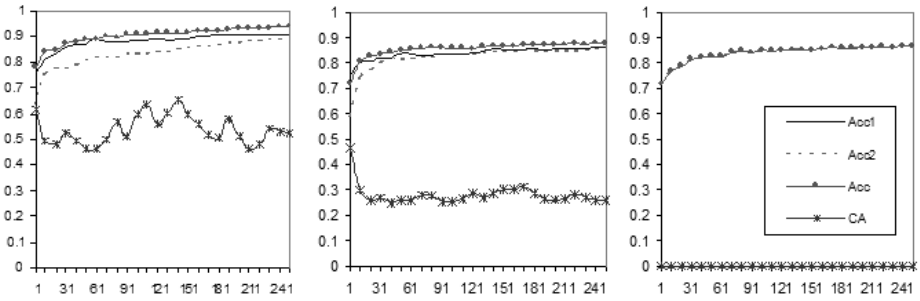

Fig. 2. Performance charts with CA measurement of three experiments. (Acc: accuracy of the combined classifier. Acc1/Acc2: accuracies of two co-training classifiers. 1st chart: co-training on *IndepViewSet*. 2nd chart: co-training on *RandViewSet*. 3rd chart: self-training on *SingleViewSet*.

Furthermore, from the first chart, we can find that the performances of two classifiers are constantly increasing during the whole process of co-training, and the performance of the combined classifier is much higher than that of the either co-training classifier. It shows that the classifiers have strong collaborative ability to help each other, which is presented by our CA measurement. We also find that the performance is still trending up at the end of 250 iterations, since CA is still at a high level. This suggests that more data may further improve the performance.

From the second chart, the classifiers can help each other improve the performance at the beginning iterations as CA is relatively high. While in later iterations, performance can only be improved slightly, as CA drops to a low level. The performance of the combined classifier is only slightly higher than that of the either co-training classifier. All these findings suggest that the collaborative ability on the *RandViewSet* is weaker than that on the *IndepViewSet*.

The third chart shows a special case for co-training which means no collaborative ability. Table 2 shows the summary of the final performances.

Table 2. Performance of three experiments in the 250 iteration. Acc: accuracy of the combined classifier. Acc1/Acc2: accuracies of two co-training classifiers.

	Acc	Acc1	Acc2	CA range
IndepViewSet	**93.9**	91.1	88.4	around 0.5 0.6
RandViewSet	**89.8**	87.6	86.0	around 0.3
SingleViewSet	**86.5**	86.5	86.5	0

3 Two Variations of Co-training

In this section, we study two alternatives of co-training when the independent view split is not available.

3.1 Co-training Using Two Learning Algorithms

We denote this variation as VCT-1 in Fig. 3. Actually, there are two different points between VCT-1 and the general co-training (bold in Fig. 3). VCT-1 uses two different learning algorithms to collaborate each other and both of them use a single feature set rather than two separate feature sets.

3.2 Co-training from Different Seed Sets

We denote this variation as VCT-2. Fig. 4 shows the VCT-2 algorithm and highlights the difference between VCT-2 and the general co-training algorithm in bold fonts. VCT-2 uses two different seed sets to start co-training. Furthermore, in the co-training iterations, the labelled data are transformed from the two unlabelled data sets U_1, U_2 to the two training data sets L_1, L_2 ($U_1 \rightarrow L_2$ and $U_2 \rightarrow L_1$) respectively. In VCT-2, two classifiers are based on the same learning algorithms and the single feature set.

In the next part, we will use CA measurement to estimate the effectiveness of VCT-1 and VCT-2.

3.3 Experiment Results

We also conduct experiments on the two variations of co-training above. For VCT-1, we use Naive Bayes and Support Vector Machine (SVM) as two learning algorithms C_1 and

Given:
  a small labelled data set L
  a large unlabelled data set U
  a single view V for a classification task
Loop until U is empty:
  Use L to train **classifier** C_1 **based on** V **using learning algorithm 1**
  Use L to train **classifier** C_2 **based on** V **using learning algorithm 2**
  Allow C_1, C_2 to label the data in U
  Select most confident p positive and n negative labelled data of C_1 from U and add them to L
  Select most confident p positive and n negative labelled data of C_2 from U and add them to L

Fig. 3. Co-training using two learning algorithms.

Given:
  two small labelled data sets L_1 **and** L_2
  a large unlabelled data set U
  a single view V for a classification task
Initialize:
  Divide U into **two unlabelled data set** U_1 and U_2
Loop until either U_1 or U_2 is empty:
  Use L_1 to train the classifier h_1 based on V
  Use L_2 to train the classifier h_2 based on V
  Allow h_1 to label the data in U_1
  Allow h_2 to label the data in U_2
  Select most confident p positive and n negative data labelled by h_1 **from** U_1 **and add them to** L_2
  Select most confident p positive and n negative data labelled by h_2 **from** U_2 **and add them to** L_1

Fig. 4. Co-training from different seed sets.

C_2. The Naive Bayes classifier is the same as used in the experiments of Section 3.3. SVM-light (Joachims 1999 [7]) with linear kernel function is used as C_2 in VCT-1. The certainty threshold α_1 is set to 1.0 and the uncertainty threshold α_2 is set to 0.8 in SVM. For VCT-2, the two classifiers we use are also the same Naive Bayes classifiers used in Section 3.3. Initial training sets and unlabelled data sets of the two classifiers are different.

The data set used in these experiments is the *SingleViewSet* as introduced in the Section 3.3. We also separate the *SingleViewSet* into 3 positive and 3 negative instances as an initial training set, 500 positive and 500 negative instances as an unlabelled set and the rest about 900 instances as a test set. In each iteration, we select one most confident positive and one most confident negative instance of each classifier from the unlabelled set into the training set until the unlabelled set is empty.

Fig. 5 shows the performance charts of VCT-1 and VCT-2. From the first chart (VCT-1), we can find that the CA measurement (around 0.3) is close to that of the *RandViewSet* co-training in the Section 3.3. We can also find that the two learning algo-

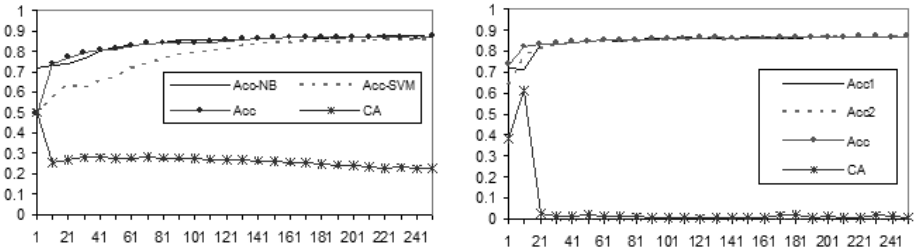

Fig. 5. Performance chart with CA measurement of VCT-1 and VCT-2.

rithms, Naive Bayes classifier and SVM classifier, can help each other slightly improve the performances. The final combined result (87.4) is also slightly higher than either of the co-training result and also higher than the result of self-training (86.5) shown in Section 3.3 which means it benefits from the collaboration of two learning algorithms. It is clear that VCT-1's collaborative ability is weaker than that of the *IndepViewSet* co-training. However, practically speaking, if we cannot split features into two views (view insufficiency) for certain task, we could consider VCT-1 by measuring its CA value. If CA is at an acceptable level, say above 0.3, the VCT-1 may be practical.

From the second chart (VCT-2), we can find that the CA value drops dramatically to nearly zero after iteration 21. In the first 20 iterations, the CA value is high which indicates that the two classifiers are collaborating well. Actually, from the accuracy curves, we can find that performances of the two co-training classifiers are improving fast and the combined performance is much higher than the performance of the either co-training classifier in the first 20 iterations. However, after the 21st iteration, the CA value is close to zero, which indicates that the classifiers do not have collaborative ability any more. In fact, the accuracy curves also show that the performances of the two co-training classifiers are too close to each other. In this case, it suggests that the two classifiers highly agree with each other and do not have ability to collaborate, which becomes a similar case to the self-training. Therefore, in this application, VCT-2 is not suitable for co-training. In addition, it also suggests that CA measurement can be used to determine when the co-training process can be stopped as well as the agreement value.

4 Conclusions

In the paper, we theoretically and empirically analyze the relation between CA measurement and collaborative ability in co-training algorithm. We show that the CA measurement can well represent the collaborative ability of two co-training classifiers. We use the CA value to compare three co-training settings and show that higher CA value leads to better co-training ability. Furthermore, we propose two variations of co-training and use CA measurement to evaluate them. The results on text classification task show that co-training with different learning algorithms is a viable alternative and co-training from different seed sets is not effective. The CA measurement enables us to understand

more about the learning behavior of co-training and suggests whether the classifiers can be used for effective co-training. It will be useful for designing the co-training model in practice.

References

1. Collins, M., Singer, Y.: Unsupervised models for named entity classification. In: Proceedings of the 1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Proceeding and Very Large Corpora. (1999)
2. I. Muslea, S.M., Knoblock, C.A.: Selective sampling with redundant views. In: Proceedings of the Seventeenth National Conference on Artificial Intelligence. (2000)
3. Ng, V., Cardie, C.: Weakly supervised natural language learning without redundant views. In: Proceedings of the Main Conference on HLT-NAACL 2003. (2002)
4. Blum, A., Mitchell, T.: Combining labeled data and unlabelled data with co-training. In: Proceedings of the 11th Annual Conference on Computational learning Theory. (1998)
5. Y. B. Cao, H.L., Lian, L.: Uncertainty reduction in collaborative bootstrapping: Measure and algorithm. In: Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics. (2003)
6. Nigam, K., Ghani, R.: Analyzing the effectiveness and applicability of co-training. In: Proceedings of the 9th International Conference on Information and Knowledge Management. (2000)
7. Joachims, T.: Making large-scale svm learning practical. In: Advances in Kernel Methods - Support Vector Learning, B. Scholkopf and C. Burges and A. Smola (ed.), MIT-Press (1999)