Uncertainty Reduction in Collaborative Bootstrapping: Measure and Algorithm

Yunbo Cao
Microsoft Research Asia
5F Sigma Center,
No.49 Zhichun Road, Haidian
Beijing, China, 100080
i-yucao@microsoft.com

Hang Li
Microsoft Research Asia
5F Sigma Center,
No.49 Zhichun Road, Haidian
Beijing, China, 100080
hangli@microsoft.com

Li Lian
Computer Science Department
Fudan University
No. 220 Handan Road
Shanghai, China, 200433
leelix@yahoo.com

Abstract

This paper proposes the use of *uncertainty reduction* in machine learning methods such as co-training and bilingual bootstrapping, which are referred to, in a general term, as 'collaborative bootstrapping'. The paper indicates that uncertainty reduction is an important factor for enhancing the performance of collaborative bootstrapping. It proposes a new measure for representing the degree of uncertainty correlation of the two classifiers in collaborative bootstrapping and uses the measure in analysis of collaborative bootstrapping. Furthermore, it proposes a new algorithm of collaborative bootstrapping on the basis of uncertainty reduction. Experimental results have verified the correctness of the analysis and have demonstrated the significance of the new algorithm.

1 Introduction

We consider here the problem of collaborative bootstrapping. It includes co-training (Blum and Mitchell, 1998; Collins and Singer, 1998; Nigam and Ghani, 2000) and bilingual bootstrapping (Li and Li, 2002).

Collaborative bootstrapping begins with a small number of labelled data and a large number of unlabelled data. It trains two (types of) classifiers from the labelled data, uses the two classifiers to label some unlabelled data, trains again two new classifiers from all the labelled data, and repeats the above process. During the process, the two classifiers help each other by exchanging the labelled data. In co-training, the two classifiers have *different feature structures*, and in bilingual bootstrapping, the two classifiers have *different class structures*.

Dasgupta et al (2001) and Abney (2002) conducted theoretical analyses on the performance (generalization error) of co-training. Their analyses, however, cannot be *directly* used in studies of co-training in (Nigam & Ghani, 2000) and bilingual bootstrapping.

In this paper, we propose the use of uncertainty reduction in the study of collaborative bootstrapping (both co-training and bilingual bootstrapping). We point out that uncertainty reduction is an important factor for enhancing the performances of the classifiers in collaborative bootstrapping. Here, the uncertainty of a classifier is defined as the portion of instances on which it cannot make classification decisions. Exchanging labelled data in bootstrapping can help reduce the uncertainties of classifiers.

Uncertainty reduction was previously used in active learning. We think that it is this paper which for the first time uses it for bootstrapping.

We propose a new measure for representing the uncertainty correlation between the two classifiers in collaborative bootstrapping and refer to it as 'uncertainty correlation coefficient' (UCC). We use UCC for analysis of collaborative bootstrapping. We also propose a new algorithm to improve the performance of existing collaborative bootstrapping algorithms. In the algorithm, one classifier always asks the other classifier to label the most *uncertain* instances for it.

Experimental results indicate that our theoretical analysis is correct. Experimental results also indicate that our new algorithm outperforms existing algorithms.

2 Related Work

2.1 Co-Training and Bilingual Bootstrapping

Co-training, proposed by Blum and Mitchell (1998), conducts two bootstrapping processes in parallel, and makes them collaborate with each other. More specifically, it repeatedly trains two classifiers from the labelled data, labels some unlabelled data with the two classifiers, and exchanges the newly labelled data between the two classifiers. Blum and Mitchell assume that the two classifiers are based on two subsets of the entire feature set and the two subsets are conditionally independent with one another given a class. This assumption is called 'view independence'. In their algorithm of co-training, one classifier always asks the other classifier to label the most certain instances for the collaborator. The word sense disambiguation method proposed in Yarowsky (1995) can also be viewed as a kind of co-training.

Since the assumption of view independence cannot always be met in practice, Collins and Singer (1998) proposed a co-training algorithm based on 'agreement' between the classifiers.

As for theoretical analysis, Dasgupta et al. (2001) gave a bound on the generalization error of co-training within the framework of PAC learning. The generalization error is a function of 'disagreement' between the two classifiers. Dasgupta et al's result is based on the view independence assumption, which is strict in practice.

Abney (2002) refined Dasgupta et al's result by relaxing the view independence assumption with a new constraint. He also proposed a new co-training algorithm on the basis of the constraint.

Nigam and Ghani (2000) empirically demonstrated that bootstrapping with a random feature split (i.e. co-training), even violating the view independence assumption, can still work better than bootstrapping without a feature split (i.e., bootstrapping with a single classifier).

For other work on co-training, see (Muslea et al 200; Pierce and Cardie 2001).

Li and Li (2002) proposed an algorithm for word sense disambiguation in translation between two languages, which they called 'bilingual bootstrapping'. Instead of making an assumption on the features, bilingual bootstrapping makes an assumption on the classes. Specifically, it assumes that the classes of the classifiers in bootstrapping do not overlap. Thus, bilingual bootstrapping is different from co-training.

Because the notion of agreement is not involved in bootstrapping in (Nigam & Ghani 2000) and bilingual bootstrapping, Dasgupta et al and Abney's analyses cannot be directly used on them.

2.2 Active Learning

Active leaning is a learning paradigm. Instead of passively using all the given labelled instances for training as in supervised learning, active learning repeatedly asks a supervisor to label what it considers as the most critical instances and performs training with the labelled instances. Thus, active learning can eventually create a reliable classifier with fewer labelled instances than supervised learning. One of the strategies to select critical instances is called 'uncertain reduction' (e.g., Lewis and Gale, 1994). Under the strategy, the most uncertain instances to the current classifier are selected and asked to be labelled by a supervisor.

The notion of uncertainty reduction was not used for bootstrapping, to the best of our knowledge.

3 Collaborative Bootstrapping and Uncertainty Reduction

We consider the collaborative bootstrapping problem.

Let $\mathcal{X}$ denote a set of instances (feature vectors) and let $\mathcal{Y}$ denote a set of labels (classes). Given a number of labelled instances, we are to construct a function $h : \mathcal{X} \to \mathcal{Y}$. We also refer to it as a classifier.

In collaborative bootstrapping, we consider the use of two partial functions h_1 and h_2, which either output a class label or a special symbol $\perp$ denoting 'no decision'.

Co-training and bilingual bootstrapping are two examples of collaborative bootstrapping.

In co-training, the two collaborating classifiers are assumed to be based on two different views, namely two different subsets of the entire feature set. Formally, the two views are respectively interpreted as two functions $X_1(x)$ and $X_2(x)$, $x \in \mathcal{X}$. Thus, the two collaborating classifiers h_1 and h_2 in co-training can be respectively represented as $h_1(X_1(x))$ and $h_2(X_2(x))$.

In bilingual bootstrapping, a number of classifiers are created in the two languages. The classes of the classifiers correspond to word senses and do not overlap, as shown in Figure 1. For example, the classifier $h_1(x \mid E_1)$ in language 1 takes sense 2 and sense 3 as classes. The classifier $h_2(x \mid C_1)$ in language 2 takes sense 1 and sense 2 as classes, and the classifier $h_2(x \mid C_2)$ takes sense 3 and sense 4 as classes. Here we use E_1, C_1, C_2 to denote different words in the two languages. Collaborative bootstrapping is performed between the classifiers $h_1(*)$ in language 1 and the classifiers $h_2(*)$ in language 2. (See Li and Li 2002 for details).

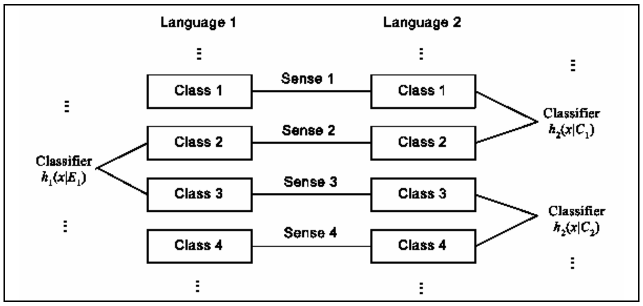

Figure 1: Bilingual Bootstrapping

For the classifier $h_1(x \mid E_1)$ in language 1, we assume that there is a *pseudo* classifier $h_2(x \mid C_1, C_2)$ in language 2, which functions as a collaborator of $h_1(x \mid E_1)$. The pseudo classifier $h_2(x \mid C_1, C_2)$ is based on $h_2(x \mid C_1)$ and $h_2(x \mid C_2)$, and takes sense 2 and sense 3 as classes. Formally, the two collaborating classifiers (one real classifier and one pseudo classifier) in bilingual bootstrapping are respectively represented as $h_1(x \mid E)$ and $h_2(x \mid C)$, $x \in \mathcal{X}$.

Next, we introduce the notion of uncertainty reduction in collaborative bootstrapping.

Definition 1 The uncertainty $U(h)$ of a classifier h is defined as:

$$U(h) = P(\{x \mid h(x) = \perp, x \in \mathcal{X}\}) \qquad (1)$$

In practice, we define $U(h)$ as

$$U(h) = P(\{x \mid C(h(x) = y) < \theta,\ \forall y \in \mathcal{Y}, x \in \mathcal{X}\}) \quad (2)$$

where θ denotes a predetermined threshold and $C(*)$ denotes the confidence score of the classifier h.

Definition 2 The conditional uncertainty $U(h \mid y)$ of a classifier h given a class y is defined as:

$$U(h \mid y) = P(\{x \mid h(x) = \perp, x \in \mathcal{X}\} \mid Y = y) \qquad (3)$$

We note that the uncertainty (or conditional uncertainty) of a classifier (a partial function) is an indicator of the *accuracy* of the classifier. Let us consider an ideal case in which the classifier achieves 100% accuracy when it can make a classification decision and achieves 50% accuracy when it cannot (assume that there are only two classes). Thus, the total accuracy on the entire data space is $1 - 0.5 \times U(h)$.

Definition 3 Given the two classifiers h_1 and h_2 in collaborative bootstrapping, the uncertainty reduction of h_1 with respect to h_2 (denoted as $UR(h_1 \setminus h_2)$), is defined as

$$UR(h_1 \setminus h_2) = P(\{x \mid h_1(x) = \perp, h_2(x) \neq \perp, x \in \mathcal{X}\}) \qquad (4)$$

Similarly, we have

$$UR(h_2 \setminus h_1) = P(\{x \mid h_1(x) \neq \perp, h_2(x) = \perp, x \in \mathcal{X}\})$$

Uncertainty reduction is an important factor for determining the performance of collaborative bootstrapping. In collaborative bootstrapping, the more the uncertainty of one classifier can be reduced by the other classifier, the higher the performance can be achieved by the classifier (the more effective the collaboration is).

4 Uncertainty Correlation Coefficient Measure

4.1 Measure

We introduce the measure of uncertainty correlation coefficient (UCC) to collaborative bootstrapping.

Definition 4 Given the two classifiers h_1 and h_2, the conditional uncertainty correlation coefficient (CUCC) between h_1 and h_2 given a class y (denoted as $r_{h_1 h_2 y}$), is defined as

$$r_{h_1 h_2 y} = \frac{P(h_1(x) = \perp, h_2(x) = \perp \mid Y = y)}{P(h_1(x) = \perp \mid Y = y) P(h_2(x) = \perp \mid Y = y)} \qquad (5)$$

Definition 5 The uncertainty correlation coefficient (UCC) between h_1 and h_2 (denoted as $R_{h_1 h_2}$), is defined as

$$R_{h_1 h_2} = \sum_y P(y) r_{h_1 h_2 y} \qquad (6)$$

UCC represents the degree to which the uncertainties of the two classifiers are related. If UCC is high, then there are a large portion of instances which are uncertain *for both of the classifiers*. Note that UCC is a symmetric measure from both classifiers' perspectives, while UR is an asymmetric measure from one classifier's perspective (either $UR(h_1 \setminus h_2)$ or $UR(h_2 \setminus h_1)$).

4.2 Theoretical Analysis

Theorem 1 reveals the relationship between the CUCC (UCC) measure and uncertainty reduction.

Assume that the classifier h_1 can collaborate with either of the two classifiers h_2 and h'_2. The two classifiers h_2 and h'_2 have equal conditional uncertainties. The CUCC values between h_1 and h'_2 are smaller than the CUCC values between h_1 and h_2. Then, according to Theorem 1, h_1 should collaborate with h'_2, because h'_2 can help reduce its uncertainty more, thus, improve its accuracy more.

Theorem 1 Given the two classifier pairs (h_1, h_2) and (h_1, h'_2), if $r_{h_1 h_2 y} \geq r_{h_1 h'_2 y}, y \in \mathcal{Y}$ and $U(h_2 \mid y) = U(h'_2 \mid y),\ y \in \mathcal{Y}$, then we have

$$UR(h_1 \setminus h_2) \leq UR(h_1 \setminus h'_2)$$

Proof:

We can decompose the uncertainty $U(h_1)$ of h_1 as follows:

$$\begin{aligned}
U(h_1) &= \sum_y P(\{x \mid h_1(x) = \perp, x \in \mathcal{X}\} \mid Y = y) P(Y = y) \\
&= \sum_y (P(\{x \mid h_1(x) = \perp, h_2(x) = \perp, x \in \mathcal{X}\} \mid Y = y) \\
&\quad + P(\{x \mid h_1(x) = \perp, h_2(x) \neq \perp, x \in \mathcal{X}\} \mid Y = y)) P(Y = y) \\
&= \sum_y (r_{h_1 h_2 y} P(\{x \mid h_1(x) = \perp, x \in \mathcal{X}\} \mid Y = y) \\
&\quad \cdot P(\{x \mid h_2(x) = \perp, x \in \mathcal{X}\} \mid Y = y) \\
&\quad + P(\{x \mid h_1(x) = \perp, h_2(x) \neq \perp, x \in \mathcal{X}\} \mid Y = y)) P(Y = y) \\
&= \sum_y (r_{h_1 h_2 y} U(h_1 \mid y) U(h_2 \mid y) \\
&\quad + P(\{x \mid h_1(x) = \perp, h_2(x) \neq \perp, x \in \mathcal{X}\} \mid Y = y)) P(Y = y) \\
&= \sum_y (r_{h_1 h_2 y} U(h_1 \mid y) U(h_2 \mid y) P(Y = y) \\
&\quad + P(\{x \mid h_1(x) = \perp, h_2(x) \neq \perp, x \in \mathcal{X}\}))
\end{aligned}$$

Thus,

$$\begin{aligned}
UR(h_1 \setminus h_2) &= P(\{x \mid h_1(x) = \perp, h_2(x) \neq \perp, x \in \mathcal{X}\}) \\
&= U(h_1) - \sum_y r_{h_1 h_2 y} U(h_1 \mid y) U(h_2 \mid y) P(Y = y)
\end{aligned}$$

Similarly we have

$$UR(h_1 \setminus h'_2) = U(h_1) - \sum_y r_{h_1 h'_2 y} U(h_1 \mid y) U(h'_2 \mid y) P(Y = y)$$

Under the conditions, $r_{h_1 h_2 y} \geq r_{h_1 h'_2 y}, y \in \mathcal{Y}$ and $U(h_2 \mid y) = U(h' \mid y_2),\ y \in \mathcal{Y}$, we have

$$UR(h_1 \setminus h_2) \leq UR(h_1 \setminus h'_2) \qquad \square$$

Theorem 1 states that the lower the CUCC values are, the higher the performances can be achieved in collaborative bootstrapping.

Definition 6 The two classifiers in co-training are said to satisfy the view independence assumption (Blum and Mitchell, 1998), if the following equations hold for any class y.

$$P(X_1 = x_1 \mid Y = y, X_2 = x_2) = P(X_1 = x_1 \mid Y = y)$$

$$P(X_2 = x_2 \mid Y = y, X_1 = x_1) = P(X_2 = x_2 \mid Y = y)$$

Theorem 2 If the view independence assumption holds, then $r_{h_1 h_2 y} = 1.0$ holds for any class y.

Proof:

According to (Abney, 2002), view independence implies classifier independence:

$$P(h_1 = u \mid Y = y, h_2 = v) = P(h_1 = u \mid Y = y)$$

$$P(h_2 = v \mid Y = y, h_1 = u) = P(h_2 = v \mid Y = y)$$

We can rewrite them as

$$P(h_1 = u,, h_2 = v \mid Y = y) = P(h_1 = u \mid Y = y) P(h_2 = v \mid Y = y)$$

Thus, we have

$$\begin{aligned}
&P(\{x \mid h_1(x) = \perp, h_2(x) = \perp, x \in \mathcal{X}\} \mid Y = y) \\
&= P(\{x \mid h_1(x) = \perp, x \in \mathcal{X}\} \mid Y = y) P(\{x \mid h_2(x) = \perp x \in \mathcal{X}\} \mid Y = y)
\end{aligned}$$

It means

$$r_{h_1 h_2 y} = 1.0, \quad \forall y \in \mathcal{Y} \qquad \square$$

Theorem 2 indicates that in co-training with view independence, the CUCC values ($r_{h_1 h_2 y}, \forall y \in \mathcal{Y}$) are small, since by definition $0 < r_{h_1 h_2 y} < \infty$. According to Theorem 1, it is easy to reduce the uncertainties of the classifiers. That is to say, co-training with view independence can perform well.

How to conduct theoretical evaluation on the CUCC measure in bilingual bootstrapping is still an open problem.

4.3 Experimental Results

We conducted experiments to empirically evaluate the UCC values of collaborative bootstrapping. We also investigated the relationship between UCC and accuracy. The results indicate that the theoretical analysis in Section 4.2 is correct.

In the experiments, we define accuracy as the percentage of instances whose assigned labels agree with their 'true' labels. Moreover, when we refer to UCC, we mean that it is the UCC value on the test data. We set the value of θ in Equation (2) to 0.8.

Co-Training for Artificial Data Classification

We used the data in (Nigam and Ghani 2000) to conduct co-training. We utilized the articles from four newsgroups (see Table 1). Each group had 1000 texts.

Table 1: Artificial Data for Co-Training

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

By joining together randomly selected texts from each of the two newsgroups in the first row as positive instances and joining together randomly selected texts from each of the two newsgroups in the second row as negative instances, we created a two-class classification data with *view independence*. The joining was performed under the condition that the words in the two newsgroups in the first column came from one vocabulary, while the words in the newsgroups in the second column came from the other vocabulary.

We also created a set of classification data without view independence. To do so, we randomly split all the features of the pseudo texts into two subsets such that each of the subsets contained half of the features.

We next applied the co-training algorithm to the two data sets.

We conducted the same pre-processing in the two experiments. We discarded the header of each text, removed stop words from each text, and made each text have the same length, as did in (Nigam and Ghani, 2000). We discarded 18 texts from the entire 2000 texts, because their main contents were binary codes, encoding errors, etc.

We randomly separated the data and performed co-training with random feature split and co-training with natural feature split in five times. The results obtained (cf., Table 2), thus, were averaged over five trials. In each trial, we used 3 texts for each class as labelled training instances, 976 texts as testing instances, and the remaining 1000 texts as unlabelled training instances.

From Table 2, we see that the UCC value of the natural split (in which view independence holds) is lower than that of the random split (in which view independence does not hold). That is to say, in natural split, there are fewer instances which are *uncertain* for both of the classifiers. The accuracy of the natural split is higher than that of the random split. Theorem 1 states that the lower the CUCC values are, the higher the performances can be achieved. The results in Table 2 agree with the claim of Theorem 1. (Note that it is easier to use CUCC for theoretical analysis, but it is easier to use UCC for empirical analysis).

Table 2: Results with Artificial Data

Feature	Accuracy	UCC
Natural Split	0.928	1.006
Random Split	0.712	2.399

We also see that the UCC value of the natural split (view independence) is about 1.0. The result agrees with Theorem 2.

Co-Training for Web Page Classification

We used the same data in (Blum and Mitchell, 1998) to perform co-training for web page classification.

The web page data consisted of 1051 web pages collected from the computer science departments of four universities. The goal of classification was to determine whether a web page was concerned with an academic course. 22% of the pages were actually related to academic courses. The features for each page were possible to be separated into two independent parts. One part consisted of words occurring in the current page and the other part consisted of words occurring in the anchor texts pointed to the current page.

We randomly split the data into three subsets: labelled training set, unlabeled training set, and test set. The labelled training set had 3 course pages and 9 non-course pages. The test set had 25% of the pages. The unlabelled training set had the remaining data.

Table 3: Results with Web Page Data and Bilingual Bootstrapping Data

Data		Accuracy	UCC
Web Page		0.943	1.147
Word Sense Disambiguation	bass	0.925	2.648
	drug	0.868	0.986
	duty	0.751	0.840
	palm	0.924	1.174
	plant	0.959	1.226
	space	0.878	1.007
	tank	0.844	1.177

We used the data to perform co-training and web page classification. The setting for the

Table 4: Data for Bilingual Bootstrapping

Word	Unlabelled instances		Seed words	Test instances
	English	Chinese		
bass	142	8811	fish / music	200
drug	3053	5398	treatment / smuggler	197
duty	1428	4338	discharge / export	197
palm	366	465	tree / hand	197
plant	7542	24977	industry / life	197
Space	3897	14178	volume / outer	197
tank	417	1400	combat / fuel	199
Total	16845	59567	-	1384

experiment was almost the same as that of Nigam and Ghani's. One exception was that we did not conduct feature selection, because we were not able to follow their method from their paper.

We repeated the experiment five times and evaluated the results in terms of UCC and accuracy. Table 3 shows the average accuracy and UCC value over the five trials.

Bilingual Bootstrapping

We also used the same data in (Li and Li, 2002) to conduct bilingual bootstrapping and word sense disambiguation.

The sense disambiguation data were related to seven ambiguous English words, each having two Chinese translations. The goal was to determine the correct Chinese translations of the ambiguous English words, given English sentences containing the ambiguous words.

For each word, there were two seed words used as labelled instances for training, a large number of unlabeled instances (sentences) in both English and Chinese for training, and about 200 labelled instances (sentences) for testing. Details on data are shown in Table 4.

We used the data to perform bilingual bootstrapping and word sense disambiguation. The setting for the experiment was exactly the same as that of Li and Li's. Table 3 shows the accuracy and UCC value for each word.

From Table 3 we see that both co-training and bilingual bootstrapping have low UCC values (around 1.0). With lower UCC (CUCC) values, higher performances can be achieved, according to Theorem 1. The accuracies of them are indeed high.

Note that since the features and classes for each word in bilingual bootstrapping and those for web page classification in co-training are different, it is not meaningful to directly compare the UCC values of them.

5 Uncertainty Reduction Algorithm

5.1 Algorithm

```
Input: A set of labeled instances and a set of unlabelled instances.
Loop while there exist unlabelled instances{
    Create classifier $h_1$ using the labeled instances;
    Create classifier $h_2$ using the labeled instances;
    For each class ($Y = y$){
        Pick up $b_y$ unlabelled instances whose labels ($Y = y$) are most certain for $h_1$ and are most uncertain for $h_2$, label them with $h_1$ and add them into the set of labeled instances;

        Pick up $b_y$ unlabelled instances whose labels ($Y = y$) are most certain for $h_2$ and are most uncertain for $h_1$, label them with $h_2$ and add them into the set of labeled instances;
    }
}
Output: Two classifiers $h_1$ and $h_2$
```

Figure 2: Uncertainty Reduction Algorithm

We propose a new algorithm for collaborative bootstrapping (both co-training and bilingual bootstrapping).

In the algorithm, the collaboration between the classifiers is driven by *uncertainty reduction*. Specifically, one classifier always selects the most uncertain unlabelled instances for it and asks the other classifier to label. Thus, the two classifiers can help each other more effectively.

There exists, therefore, a similarity between our algorithm and active learning. In active learning the learner always asks the supervisor to label the most uncertain examples for it, while in our algorithm one classifier always asks the other classifier to label the most uncertain examples for it.

Figure 2 shows the algorithm. Actually, our new algorithm is different from the previous algorithm only in one point. Figure 2 highlights the point in italic fonts. In the previous algorithm, when a classifier labels unlabeled instances, it labels those instances whose labels are most *certain* for the classifier. In contrast, in our new algorithm, when a classifier labels unlabeled instances, it labels those instances whose labels are most certain for the classifier, but at the same time most *uncertain* for the other classifier.

As one implementation, for each class *y*, h_1 first selects its most certain a_y instances, h_2 next selects from them its most uncertain b_y instances ($a_y \geq b_y$), and finally h_1 labels the b_y instances with label *y* (Collaboration from the opposite direction is performed similarly.). We use this implementation in our experiments described below.

5.2 Experimental Results

We conducted experiments to test the effectiveness of our new algorithm. Experimental results indicate that the new algorithm performs better than the previous algorithm. We refer to them as 'new' and 'old' respectively.

Co-Training for Artificial Data Classification

Table 5: Accuracies with Artificial Data

Feature	Accuracy		UCC
	Old	New	
Natural Split	**0.928**	0.924	1.006
Random Split	0.712	**0.775**	2.399

We used the artificial data in Section 4.3 and conducted co-training with both the old and new algorithms. Table 5 shows the results.

We see that in co-training the new algorithm performs as well as the old algorithm when UCC is low (view independence holds), and the new algorithm performs *significantly* better than the old algorithm when UCC is high (view independence does not hold).

Co-Training for Web Page Classification

We used the web page classification data in Section 4.3 and conducted co-training using both the old and new algorithms. Table 6 shows the results. We see that the new algorithm performs as well as the old algorithm for this data set. Note that here UCC is low.

Table 6: Accuracies with Web Page Data

Data	Accuracy		UCC
	Old	New	
Web Page	**0.943**	**0.943**	**1.147**

Bilingual Bootstrapping

We used the word sense disambiguation data in Section 4.3 and conducted bilingual bootstrapping using both the old and new algorithms. Table 7 shows the results. We see that the performance of the new algorithm is slightly better than that of the old algorithm. Note that here the UCC values are also low.

Table 7: Accuracies with Bilingual Bootstrapping Data

Word	Accuracy		UCC
	Old	New	
bass	0.925	**0.955**	2.648
drug	**0.868**	0.863	0.986
duty	0.751	**0. 797**	0.840
palm	**0.924**	0.914	1.174
plant	**0.959**	0.944	1.226
space	0.878	**0.888**	1.007
tank	0.844	**0.854**	1.177
Average	0.878	**0.888**	-

We conclude that for both co-training and bilingual bootstrapping, *the new algorithm performs significantly* better *than the old algorithm when UCC is high, and performs as well as the old algorithm when UCC is low.* Recall that when UCC is high, there are more instances which are uncertain for both classifiers and when UCC is low, there are fewer instances which are uncertain for both classifiers.

Note that in practice it is difficult to find a situation in which UCC is completely low (e.g., the view independence assumption completely holds), and thus the new algorithm will be more useful than the old algorithm in practice. To verify this, we conducted an additional experiment.

Again, since the features and classes for each word in bilingual bootstrapping and those for web page classification in co-training are different, it is not meaningful to directly compare the UCC values of them.

Co-Training for News Article Classification

In the additional experiment, we used the data from two newsgroups (comp.graphics and comp.os.ms-windows.misc) in the dataset of (Joachims, 1997) to construct co-training and text classification.

There were 1000 texts for each group. We viewed the former group as positive class and the latter group as negative class. We applied the new and old algorithms. We conducted 20 trials in the experimentation. In each trial we randomly split the data into labelled training, unlabeled training and test data sets. We used 3 texts per class as labelled instances for training, 994 texts for testing, and the remaining 1000 texts as unlabelled instances for training. We performed the same preprocessing as that in (Nigam and Ghani 2000).

Table 8 shows the results with the 20 trials. The accuracies are averaged over each 5 trials. From the table, we see that co-training with the new algorithm *significantly* outperforms that using the old algorithm and also 'single bootstrapping'. Here, 'single bootstrapping' refers to the conventional bootstrapping method in which a single classifier repeatedly boosts its performances with all the features.

Table 8: Accuracies with News Data

Average Accuracy	Single Bootstrapping	Collaborative Bootstrapping	
		Old	New
Trial 1-5	0.725	0.737	**0.768**
Trial 6-10	0.708	0.702	**0.793**
Trial 11-15	0.679	0.647	**0.769**
Trial 16-20	0.699	0.689	**0.767**
All	0.703	0.694	**0.774**

The above experimental results indicate that our new algorithm for collaborative bootstrapping performs significantly better than the old algorithm when the collaboration is difficult. It performs as well as the old algorithm when the collaboration is easy. *Therefore, it is better to always employ the new algorithm.*

Another conclusion from the results is that we can apply our new algorithm into any *single bootstrapping* problem. More specifically, we can randomly split the feature set and use our algorithm to perform co-training with the split subsets.

6 Conclusion

This paper has theoretically and empirically demonstrated that *uncertainty reduction is the essence of collaborative bootstrapping*, which includes both *co-training* and *bilingual bootstrapping*.

The paper has conducted a new theoretical analysis of collaborative bootstrapping, and has proposed a new algorithm for collaborative bootstrapping, both on the basis of uncertainty reduction. Experimental results have verified the correctness of the analysis and have indicated that the new algorithm performs better than the existing algorithms.

References

S. Abney, 2002. Bootstrapping. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*.

A. Blum and T. Mitchell, 1998. Combining Labeled Data and Unlabelled Data with Co-training. In P*roceedings of the 11th Annual Conference on Computational learning Theory.*

M. Collins and Y. Singer, 1999. Unsupervised Models for Named Entity Classification. In *Proceedings of the 1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora.*

S. Dasgupta, M. Littman and D. McAllester, 2001. PAC Generalization Bounds for Co-Training. In *Proceedings of Neural Information Processing System, 2001.*

T. Joachims, 1997. A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization. In *Proceedings of the 14th International Conference on Machine Learning.*

D. Lewis and W. Gale, 1994. A Sequential Algorithm for Training Text Classifiers. In *Proceedings of the 17th International ACM-SIGIR Conference on Research and Development in Information Retrieval.*

C. Li and H. Li, 2002. Word Translation Disambiguation Using Bilingual Bootstrapping. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics.*

I. Muslea, S.Minton, and C. A. Knoblock 2000. Selective Sampling With Redundant Views. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence.*

K. Nigam and R. Ghani, 2000. Analyzing the Effectiveness and Applicability of Co-Training. In *Proceedings of the 9th International Conference on Information and Knowledge Management*.

D. Pierce and C. Cardie 2001. Limitations of Co-Training for Natural Language Learning from Large Datasets. In *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing (EMNLP-2001).*

D. Yarowsky, 1995. Unsupervised Word Sense Disambiguation Rivaling Supervised Methods. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics.*