Co-training with a Single Natural Feature Set Applied to Email Classification

Jason Chan
School of Information Technologies
The University of Sydney, Australia
jchan3@it.usyd.edu.au

Irena Koprinska
School of Information Technologies
The University of Sydney, Australia
irena@it.usyd.edu.au

Josiah Poon
School of Information Technologies
The University of Sydney, Australia
josiah@it.usyd.edu.au

Abstract

When dealing with information overload from the Internet, such as the classification of Web pages and the filtering of email spam, a new technique called co-training has been shown to be a promising approach to help build more accurate classifiers. Co-training allows classifiers to learn with fewer labelled documents by taking advantage of the more abundant unclassified documents. However, conventional co-training requires the dataset to be described by two disjoint and natural feature sets that are sufficiently redundant. In many practical situations, it is not intuitively obvious how to obtain two natural feature sets. This paper shows that when only a single natural feature set is used, the performance of co-training is beneficial in the application of email classification.

1. Introduction

One of the greatest problems facing users of the Internet is dealing with information overload. Usually, the great majority of information available consists of unwanted or unhelpful instances. Even after implementing narrowing searches or applying email filters, there still exists a significant quantity of undesired documents.

Research in Text Categorization [6] has shown that it is possible to build effective classifiers to filter unwanted documents given a sufficiently large set of training examples. However, obtaining labelled Web pages or emails is very costly, because it usually requires a great deal of human effort to classify unlabelled documents.

A new technique to overcome this problem, called *co-training* [1], was shown to be capable of converting unlabelled Web documents into labelled Web documents by initially starting off with only a small pool of classified examples. One of the main requirements that were stated for co-training to be successful was that the dataset must be described by two disjoint sets of natural features that were redundantly sufficient. That is, using only either one of the natural sets of attributes, a classifier can be built with reasonably high accuracy. For example, in their experiment that dealt with the problem of classifying Web pages, the two sets of features used to describe a page where the words in the body of the page and the words in hyperlinks of other documents referring to that particular page.

In the great majority of practical situations, there do not exist two natural sets of features that can describe the dataset. This paper investigates the applicability of co-training to such datasets. We compare co-training of a single natural feature set and co-training with two natural feature sets. By analysing the results, we address the question of when co-training with a random split of features is likely to be useful. The experiments are based on the application of email classification, which attempts to determine whether a given message is a genuine email or spam.

2. The co-training algorithm

In a given application with redundantly sufficient features, a classifier with reasonable performance can be built with each of the two sets of features separately. Co-training employs these two classifiers in a loop to label all the unlabelled examples. Each classifier takes turns to select the most confidently predicted examples and add these into the training set. Both classifiers then re-learn on the enlarged training set so that they take into account the newly added (and previously unlabelled) data. The loop is then repeated for a number of iterations to maximize performance on a separate validation set.

The idea behind the co-training algorithm is that one classifier, with its set of features, can confidently predict the class of an unlabelled example because it is similar to the training instances. However, it may only be similar to the training instances for this classifier's set of features. Because of the confidence with which this classifier predicts this example's class, it will be labelled accordingly and placed into the training set. Hence, the other classifier will be able to learn from this instance and adjust better in future.

3. Previous work on co-training

Blum and Mitchell [1] first introduced the technique of co-training. In their application of identifying academic course home pages from a set of Web documents, co-training was shown to be able to reduce the error rate of a classifier. Theoretical insights were also given, among them being the requirement of redundantly sufficient feature sets.

Kiritchenko and Matwin [2] applied co-training to the task of predicting which folder a user would place an email into. They found that the performance of co-training is sensitive to the learning algorithm used. In particular, co-training with Naïve Bayes (NB) worsens performance, while Support Vector Machines (SVM) improves it. The authors explained this with the inability of NB to deal with large sparse datasets. This explanation was confirmed by significantly better results after feature selection.

Nigam and Ghani [4] investigated the sensitivity of co-training to the assumption of redundant sufficiency. One of their experiments involved performing co-training on a dataset whereby a natural split of feature sets is not used. The two feature sets were chosen by randomly assigning all the features of the dataset into two different groups. This was tried for two datasets: one with a clear redundancy of features, and one with an unknown level of redundancy and non-evident natural split in features. The results indicated that the presence of redundancy in the feature sets gave the co-training algorithm a bigger advantage over expectation maximization. Together with theoretical insights, this result led the researchers to conclude that co-training has a considerable dependence on the assumption of redundant sufficiency. However, even when this assumption is violated, as in many practical settings, the performance of co-training can still be quite useful in improving a classifier's performance.

4. Experimental setup

4.1 Objective

In the large majority of cases, datasets consist of only a single set of features with no obvious or natural way to divide them into two separate sets. Hence, the question of whether co-training can be useful with only a single natural feature set is of great practical importance.

This paper investigates the performance of co-training with only one natural feature set in comparison to the use of two natural feature sets. The main question that we address is: *how useful is co-training with a single natural feature set?*

4.2 Dataset, preprocessing and classifiers used

We applied tests on email classification using the LingSpam[1] corpus. This dataset consists of 2883 emails of which 479 are spam and 2404 are genuine emails. Each email is broken up into two sections: the text found in the subject header of the email and the words found in the main body of the message. After applying a stop list[2], a word count of each word type was kept with a distinction made between the words that appeared in the subject header and those that appeared in the body.

The standard bag-of-words representation was used and feature selection was performed with Information Gain [8]. Upon inspection of the word lists, it was decided that the top 100 words was a suitable cut-off, resulting in a dimensionality reduction of about 98%. This value is similar to thresholds used in other experiments, such as [2]. Each of the email documents was then represented using the term frequencies of the selected 100 features. This term weighting was motivated by its successful use in the domain of Web page classification [5].

Three types of classifiers were tested: Decision Tree (DT), NB and SVM. In previous work on co-training [2], NB has often been used as a benchmark. The SVM was used in text categorization and email classification [2] with great success. Implementations of these classifiers were obtained from WEKA [7].

4.2 The feature sets

Below is a summary of the feature sets used in the experiments.

- *Body*: all words that appear in the body of an email
- *Subject*: all words that appear in subject of an email
- *Half1*: a random selection of half of the feature set consisting of the combination of *Subject* and *Body*
- *Half2*: the other half of the features not found in *Half1*

The two feature sets *Half1* and *Half2* are created to test the hypothesis that it is possible to randomly split a natural feature set into two smaller feature sets to obtain useful results in co-training. The *Body* and *Subject* feature sets will hereon be referred to as the *natural feature sets,* while the other feature sets that contain a random selection of words will be referred to as the *random selection feature sets*.

5. Experiment 1: investigating the redundancy of the feature sets

In this experiment, supervised learning without co-training was performed with the aim of determining how redundantly sufficient each of the feature sets were. Table 1 summarizes the results. 10-fold cross validation was performed, with 2595 instances used in the training set and 288 instances in the test set. The f-measure[3] was used as a performance index.

Table 1. Supervised classification using various feature sets

	Body	Subject	Random halves	All features
DT	92.7	45.5	89.2	92.7
NB	86.9	45.5	85.0	86.9
SVM	88.9	63.0	87.3	88.9

5.1 The good performance of co-training using random selection feature sets

On inspection of Table 1, it is not surprising that of all the feature sets, supervised classification with the *Body* feature set performs best, since the classifiers have access

[1] http://www.mlnet.org/cgi-bin/mlnetois.pl/?File=dataset-details.html&Id=963839410Ling-Spam

[2] http://alt-usage-english.org/excerpts/fxcommon.html

[3] In this paper, f-measure will refer to macro-averaged f1-measure, which is given by the formula:

$$f1 = 2pr / (p+r)$$

where p is the macro-averaged precision and r is the macro-averaged recall. The macro-averaged precision is the average of the precision of each of the two classes; similarly for recall.

to the majority of words in the email message. However, it is very interesting to notice that using the feature sets consisting of randomly selected words does not result in performance that is much worse than using all the words available. For example, using half of the words randomly selected from all the available words only decreases the classification performance by no more than 4%.

The positive experimental results observed for the random feature sets show that there exists sufficient redundancy within the feature set of this particular dataset. This suggests that co-training with a random split of a single natural feature set can indeed result in the improved performance of a classifier, provided that this dataset is sufficiently redundant.

5.2 The poor performance of the *Subject* feature set

Interestingly, the worse performing set of attributes is easily the *Subject* feature set. The classification performance achieved with this feature set is much worse than using only half of the all the available features.

The most likely reason for the poor performance of the S*ubject* feature set is the significantly lower number of word tokens present in the subject header in comparison to the entire email message[1]. As a result, the few words often found in the subject header of an email are not very good indicators of whether that email is genuine or spam.

Another reason for poorer performance with this feature set is the increased variance in noise to be expected with using fewer word tokens. That is, performance becomes more sensitive to the presence of words that would not generally help to distinguish between spam and genuine email, since there are fewer useful (or helpful) words contributing to the overall performance.

Words found in the *Subject* feature set tend to be more meaningful than other words found in the main body of the document, since they summarise the main topic of the email and their words are usually more selectively chosen than the words found in the main body. However, the results suggest that this advantage has been outweighed by the increased sensitivity to noise in the form of unhelpful words as well as the decreased number of word tokens.

6. Experiment 2: co-training with a random split of all features

The goal of this experiment was to compare the overall performance of co-training with a random split of features against co-training with natural split of features. Table 2 summarises the results. Shown are the maximum classification performance (max) that is achieved and the performance improvement (increase) achieved over the base classifier trained with only the initial labelled set. Figure 1 illustrates the difference in performance over the number of co-training iterations completed for naï ve Bayes.

Table 2. Co-training with natural split versus random split

	Natural Feature Split		Random Feature Split	
	max	increase	max	increase
DT	45.5	0.0	45.5	0.0
NB	84.9	8.2	86.8	10.1
SVM	78.0	0.0	79.5	1.5

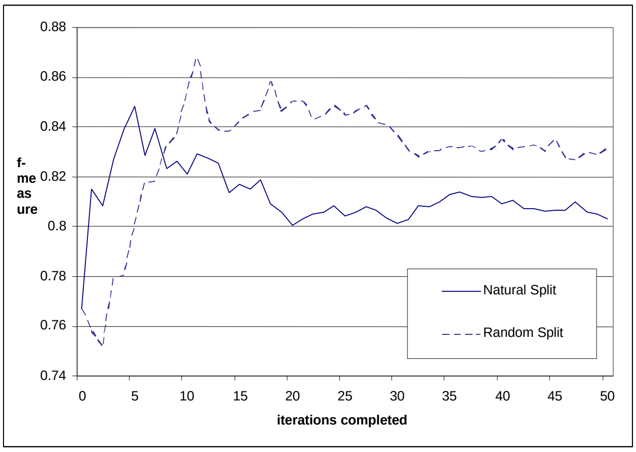

Figure 1. Performance plotted against iterations completed

We started off with a labelled set of 1 spam and 1 genuine email. 10% of the dataset (288 instances) were retained for testing, while the rest were used as initially unlabelled data. 1 newly-labelled spam and 5 new-labelled genuine emails were transferred from the unlabelled set to the labelled set on each iteration of co-training. We repeated each trial 5 times, obtaining a different sample of labelled, unlabelled and test set each time. The above results are averaged over these 5 trials.

6.1 General observations

Very encouraging results can be seen in Table 2. They show that the performance of co-training with a random split of features is comparable to co-training with the natural selection of features. Indeed, the classification performance achieved by the NB and SVM classifiers is in fact better with the random split than the natural split.

Due to space restrictions, a general summary of the many results and observations we obtained are given here:

- The general trends discussed were found to be true for various parameter settings of co-training, such as different initial labelled set sizes, unlabelled cache sizes, number of spam and genuine instances added per iteration, and ratio of spam and genuine instances in the entire dataset.
- As shown in Figure 1, co-training with a random split of features improves the performance of a classifier in comparison to only using the initial labelled set, but classification performance is still lower than using all the instances labelled correctly (see Table 1).
- Despite the DT classifier's superior performance in experiment 1, it proved to be very poor in the co-training setting because it is difficult to construct a reasonable classifier with the small initial labelled set.
- The SVM classifier also obtained positive results, but generally NB was found to have better performance.

[1] In [1], Blum and Mitchell also reported that their hyperlink-based classifier has inferior performance due to similar reasoning.

6.2 Comparison between natural feature sets and random selection feature sets

As seen in Table 2, with all the classifiers in the experiment, co-training with a random split of the features produces results that are comparable with using the natural feature sets. In many cases, the f-measure is higher for the random feature split in comparison to the natural feature split. Why is co-training with a random split of the features so comparable, and in some cases, even better than using the natural feature sets?

There are two reasons for this. Firstly, one of the classifiers used in co-training with the natural feature sets is even weaker than either of the two classifiers using randomly generated feature sets. As shown in Experiment 1, the *Subject* feature set is much weaker than any of the random feature sets that were produced. As a result, the classifier using *Subject* is incorrectly labelling many instances in comparison with classifiers built using the random selection feature sets, hence transferring many incorrectly labelled instances into the labelled set.

Secondly, as discovered in Experiment 1, it was found that using a random selection of half of the features from all the features results in classifiers that only perform slightly worse than a classifier using all the attributes available. As a result, when performing co-training, both classifiers using their respective half of the features are able to improve the training set by labelling unlabelled instances with a sufficiently high classification performance.

These two reasons combined suggest that co-training with a random split of a redundantly sufficient feature set can be just as competitive as and even better than co-training with two natural feature sets. This is especially the case when there exists a considerable difference between the classification performances of the two classifiers using the natural feature sets separately. As hinted in Experiment 1, a natural feature set consisting of fewer words, such as using the hyperlinks of a Web page, or using the subject header of an email, may produce significantly poorer results. In this event, co-training without the use of this lower quality feature set is likely to be more beneficial.

7. Conclusion

Helping users efficiently cope with excessive quantities of unwanted Web documents is an important objective in making the Internet more useful. Classifiers can be built to filter out unwanted information but typically require many labelled examples. Co-training has been shown to be a beneficial tool in improving the performance of a classifier that is given only a small training set. However, conventional co-training requires the Web documents to be able to be described by two natural sets of features, which is not always possible.

The primary objective of experiment 1 was to determine whether the LingSpam corpus was redundantly sufficient. It was found that classifiers could be built using a random selection of only half of all available features and still obtain very good classification performance. This implies that the dataset is redundantly sufficient.

In experiment 2, the performance of co-training with a random split of the entire feature set was compared to co-training with two naturally occurring feature sets. The first natural feature set contained the words used in the main body, while the second consisted of the words occurring in the subject header. It was found that co-training using a random split of all the features was just as competitive as, and often outperformed co-training with the natural feature sets. Also, classification performance generally improved over the initial classifier trained on the small initial labelled set with random-split co-training.

An important element that is needed in a feature set for co-training with a random split to work well is a dataset with high redundancy. When this condition is met, a random split of the feature set will produce two subsets, each of which can still be used on its own by a classifier to achieve a sufficiently high classification performance.

Also, co-training with a random split of a single natural feature set should be more preferable than co-training with two natural feature sets if one of the two natural feature sets is considerably weaker than the other. This is particularly the case with Web pages and emails, where feature sets other than the words in the main body will be typically weak because of their small quantity. In such cases, co-training with a random split of all the features should produce comparable and possibly better results.

One possible tool to determine whether there exists sufficient redundancy in a dataset for random-split co-training is Principal Component Analysis[1] [3]. This technique measures the importance of the attributes in a dataset, which can be used to determine whether there exists sufficient redundancy. We leave this as a possible area for future research.

8. References

[1] A. Blum, T. Mitchell. Combining Labeled and Unlabeled Data with Co-Training. In Proceedings of the Workshop on Computational Learning Theory, 1998.

[2] S. Kiritchenko, S. Matwin. Email Classification with Co-Training. In Proceedings of CASCON, 2001.

[3] F. Korn, H.V. Jagadish, C. Faloutsos. Efficiently Supporting Ad Hoc Queries in Large Datasets of Time Sequences. In Proceedings of SIGMOD, 1997.

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

[5] M. Pazzani, J. Muramatsu, D. Billsus. Syskill & Webert: Identifying interesting Websites. In Proceedings of the 13th National Conference on Artificial Intelligence, 1996.

[6] F. Sebastiani, Machine Learning in Automated Text categorization, *ACM Comp. Surveys*, 2002.

[7] I. H. Witten, E. Frank. Data Mining: Practical machine learning tools with Java implementations. Morgan Kaufmann, 1999.

[8] Y. Yang, J. O. Pedersen. A Comparative Study on Feature Selection in Text Categorization. In Proceedings of the 14th International Conference on Machine Learning, 1997.

[1] Also known as Singular Value Decomposition.