Web Page Classification using Iterative Cross-Training Algorithm

Nuanwan Soonthornphisaj[1] and Boonserm Kijsirikul[2]
Machine Intelligence & Knowledge Discovery Laboratory
Department of Computer Engineering
Chulalongkorn University,
Phathumwan, Bangkok, 10330, Thailand.
E-Mail: nuanwan[1], boonserm[2]@mind.cp.eng.chula.ac.th

Abstract: The paper presents a generalization of Iterative Cross-Training algorithm (ICT) which was previously applied to Thai Web pages identification [1]. The main concept of ICT is to iteratively train two sub-classifiers by using unlabeled examples in crossing manner. In this paper, we extend the algorithm in order to classify Web pages into course or non-course ones, which is a more challenging problem. We compare ICT against Supervised Naï ve Bayes classifier and Co-Training classifier. The experimental results show that ICT still preserves its robustness on this new problem's characteristic.
Key words: Web pages classification, Iterative Cross-Training

1. Introduction

Nowadays, there is a massively increase of web pages in Internet. An ideal search engine should be the one that has the most updated information of all web pages to provide the best search result for the user. Therefore it should have an effective web robot which can crawl the web and automatically classify Web pages into categories, since Web page classification task is a tedious job and time consuming process if it is done by human. We want it to be automatic with a reliable classification.

The problem of text classification has been explored by many researchers with variety of learning algorithms [2,3,4,5,6]. When we give a sufficient set of labeled training examples, supervised learning is the most effective method for the classification. However, the construction of hand-labeled data must be done by a human and thus this is a painfully time-consuming process. Though it is costly to construct hand-labeled data, in some domains it is easy to obtain unlabeled ones, such as data in the World Wide Web. Therefore, we propose a new learning algorithm called incremental iterative cross-training (incremental-ICT) in order to utilize the available unlabeled data.

Our incremental-ICT is based on the ICT algorithm that has been successfully applied for identifying Thai Web pages [1]. ICT employs two sub-classifiers to iteratively train each other by using unlabeled examples in crossing manner. ICT is based on the assumption that one of the sub-classifiers has some knowledge about the domain. However, this assumption is violated on some domains where we cannot give domain knowledge to the classifier. In such a problem, ICT does not perform well. In this paper, we propose a new algorithm, called incremental-ICT, which requires no such assumption.

To evaluate the effectiveness of our algorithm, we apply it to a more difficult problem than Thai Web pages identification. The problem we are interested in is the classification of Web pages into course or non-course pages. In this problem, each Web page contains two sets of features: (1) words appearing on the page, and (2) words appearing on the hyperlinks that link to that page. Therefore, each page can be viewed in two different ways, i.e., page-based features and hyperlink-based features. With these two feature sets, we construct two naï ve Bayes classifiers; the first one learns it model from hyperlink-features and the other learns from page-features.

We run experiments to evaluate the effectiveness of our method. In the experiments, we compare our method with *Co-Training* algorithm [2] and a supervised learning algorithm which uses a naï ve Bayes classifier. The results show that incremental-ICT gives better performance than the other classifiers.

The paper is organized as follows. Section 2 presents our learning algorithm, and gives the details of a naï ve Bayes classifier. Section 3 describes other learning methods used in our comparison. Section 4 describes the experimental results. Finally, Section 5 concludes our work.

2. Incremental Iterative Cross-Training

The architecture of our learning algorithm consists of two naï ve Bayes classifiers each of which learns from different features of a Web page, i.e., words on hyperlinks linking to the page and words on the

page. Starting with a small number of labeled data, each classifier estimates its parameters and uses the learned parameters to classify unlabeled data for the other as shown in Figure 1. The classification for unlabeled data is done in incremental way, i.e., the algorithm incrementally labels a small number of data. The training data is duplicated into two sets: *TrainingData1* for training hyperlink-based classifier and *TrainingData2* for training page-based one, respectively. The concept of our algorithm is that if we can obtain reliable statistical information from the first classifier, it should be useful in classifying training data for the second classifier. After receiving training from each other, the parameters of the classifiers should be more reliable every iteration.

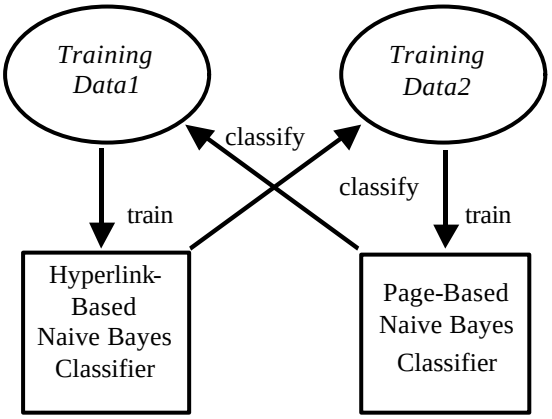

Figure 1: The architecture of iterative cross-training.

Table 1: Incremental-ICT algorithm

Given:

- Two training sets *TrainingData1* of hyperlink-based data and *TrainingData2* of page-based data (*TrainingData1* and *TrainingData2* both contains *U* labeled examples).

– Use labeled data in *TrainingData1* to estimate the parameter set θ_h of hyperlink-based classifier.
– Use labeled data in *TrainingData2* to estimate the parameter set θ_p of page-based classifier.
– Loop until all data are labeled
  - Use the page-based classifier with current θ_p to classify *TrainingData1* into positive and negative examples.
    - Check consistency of the classification with the hyperlink-based classifier. Label the class for the most confident *p* positive examples and *n* most confident negative examples.
  - Train hyperlink-based classifier by the labeled examples in *TrainingData1* to estimate the parameter set θ_h of the classifier.
  - Use the hyperlink-based classifier with current θ_h to classify *TrainingData2* into positive and negative examples.
    - Check consistency of the classification with the page-based classifier. Label the class for the most confident *p* positive examples and *n* most confident negative examples.
  - Train page-based classifier by the labeled examples in *TrainingData1* to estimate the parameter set θ_p of the classifier.

The training algorithm of incremental-ICT is shown in Table 1. As shown in the table, the training process starts with the parameter estimation of both classifiers, i.e. hyperlink-based and page-based, using initial labeled data. For each round of iteration, the page-based classifier with the current parameter, θ_p, will classify training data into positive and negative examples. Then it will ask for the confirmation from the hyperlink-based classifier which consider another view of each example to make decision about which class the example should be. If both classifiers agree with the same classifying result, the most confident *p* positive and *n* negative examples will be labeled.

The hyperlink-based classifier is then trained by the labeled examples in *TrainingData1* to estimate the parameter set θ_h. With this current θ_h, the hyperlink-based classifier will classify *TrainingData2* into positive and negative examples. Then the consistency checking process is performed again to ask for the agreement from the page-based classifier. The most confident *p* positive and *n* negative examples will be labeled. The page-based classifier starts again with parameter estimation by using labeled examples in *TrainingData1*. These processes will be repeatedly done until all data are labeled.

The classification mechanisms for these two classifiers are the same which use the naï ve Bayes algorithm. The algorithm is a well-known approach and is considered to be one of the most effective way for text classification [7]. This algorithm employs *bag-of-words* to represent the document. The method is described below.

Given a set of class labels L = {l_1, l_2,…, l_m} and a document d of n words (w_1, w_2,..,w_n), the most likely class label l^* estimated by naï ve Bayes is the one that maximizes $Pr(l_j|w_1,\ldots w_n)$:

$$l^* = \underset{l_j}{argmax}\ Pr(l_j|w_1,\ldots,w_n) \quad (1)$$

$$= \underset{l_j}{argmax}\ \frac{Pr(l_j)Pr(w_1,\ldots,w_n|l_j)}{Pr(w_1,\ldots,w_n)} \quad (2)$$

$$= \underset{l_j}{argmax}\ Pr(l_j)Pr(w_1,\ldots,w_n|l_j) \quad (3)$$

For our data set, L is the set of positive and negative class labels which are course homepage and non-course homepage, respectively. $Pr(w_1,\ldots, w_n)$ in equation 2 can be ignored, as we are interested in finding the most likely class label. As there are usually an extremely large number of possible values for $d = (w_1,w_2,\ldots,w_n)$, calculating the term $Pr(w_1,\ldots,w_n|l_j)$ requires a huge number of examples to obtain reliable estimation. Therefore, to reduce the number of required examples and improve reliability of the estimation, assumptions of naï ve Bayes are made. These assumptions are (1) the conditional independent assumption, i.e. the presence of each word is conditionally independent of all other words in the document given the class label, and (2) an assumption that the position of a word is unimportant, e.g. encountering the word "subject" at the beginning of a document is the same as encountering it at the end [7]. Equation 3 can be rewritten as:

$$l^* = \underset{l_j}{argmax}\ Pr(l_j)\prod_{i=1}^{n} Pr(w_i \mid l_j,w_1,\ldots,w_{i-1}) \quad (4)$$

$$= \underset{l_j}{argmax}\ Pr(l_j)\prod_{i=1}^{n} Pr(w_i \mid l_j) \quad (5)$$

The probabilities $Pr(l_j)$ and $Pr(w_i|l_j)$ are used as the parameter sets θ_h and θ_p of the classifiers, and are estimated from the training data. The prior probability $Pr(l_j)$ is estimated as the ratio between the number of examples belonging to the class l_j, and the number of all examples. The conditional probability $Pr(w_i|l_j)$, of seeing word w_i given class label l_j, is estimated by the following equation:

$$Pr(w_i|l_j) = \frac{1 + N(w_i,l_j)}{T + N(l_j)} \quad (6)$$

Where $N(w_i,l_j)$ is the number of times word w_i appears in the training examples from class label l_j, $N(l_j)$ is the total number of unique word in the training set. Equation 6 employs Laplace smoothing (add one to all of word counts), to avoid assigning probability values of zero to words that do not occur in the training examples for a particular class.

To evaluate our method, we will compare it with the other two techniques, which are Co-Training and supervised naï ve Bayes classifiers. The classifiers are described in the following sections.

3. Co-Training Classifier

The Co-Training algorithm explicitly uses the split of the features when learning from labeled and unlabeled data. Its approach is to build the naï ve Bayes classifier for each of the distinct feature sets. Each classifier is initialized using a few labeled documents. Then every round of Co-Training, each classifier chooses the most confident p positive and n negative labeled examples to add to the labeled set of documents. The documents selected are those that have the highest posterior class probability, $Pr(l_j|d)$. Then, each classifier rebuilds from the augmented labeled set and the process repeats [2].

Table *2*: The Co-Training algorithm [2]

Given:

A set *LE* of labeled training examples
A set *UE* of unlabeled examples

Create a pool *UE'* of examples by choosing *u* examples at random from *UE*.
Loop while there exist documents without class labels:

- Use *LE* to estimate θ_h of hyperlink-based classifier using the hyperlink portion of each document.
- Use *LE* to estimate θ_p of page-based classifier using the page portion of each document.
- Allow the hyperlink-based classifier with current θ_h to label *p* positive and *n* negative examples from *UE'*.
- Allow the page-based classifier with current θ_p to label *p* positive and *n* negative examples from *UE'*.
- Add these self-labeled examples to *LE*.
- Randomly choose *2p+2n* examples from *UE* to replenish *UE'*.

4. Supervised Naï ve Bayes Classifier

The basic method of *supervised learning* for building a classifier is that it requires a set of examples with predefined classes. The classifier is then try to find out some common properties of the different classes in order to make correct

classification for other data. Thus, this kind of classifiers need a large number of labeled examples to correctly model the character of the class during learning process. Labeling must be done by a person to train the classifier accurately. In our experiment below, we employ the naï ve Bayes classifier as a supervised learning algorithm. The algorithm of the naï ve Bayes is the same as one described in Section 2, except that it is trained by hand-labeled data.

5. Experimental Results

In order to test the robustness of incremental-ICT algorithm with a more difficult problem, we set up experiments on the problem of course/non-course Web page classification, and compare the performance of incremental-ICT to the other classifiers, i.e., Co-Training algorithm and the supervised naï ve Bayes classifier.

5.1 Data Set & Experimental Setting

The data for our experiment is obtained via ftp from Carnegie Mellon University [8]. It consists of 1,051 Web pages collected from Computer Science department Web sites at four universities: Cornell, University of Washington, University of Wisconsin, and University of Texas. These Web pages have been hand-labeled into two categories. We consider the category "course home page" as the positive class and the other as the negative class. In this dataset , 22% of the Web pages were course home pages and the rest were non-course home pages.

A Course home page gives information about the subject such as the course outline, the class schedule, reference books. A non-course home page is a personal homepage or organization web page.

Each sample is filtered to remove words which give no significance in predicting to the class of the document. Words to be eliminated are auxiliary verbs, prepositions, pronouns, possessive pronouns, phone numbers, digit sequences, dates and special characters. We have 230 course Web pages and 821 non-course Web pages. Each Web page has two views, i.e. page-based and hyperlink-based. The training set contains 172 course Web pages and 616 non-course Web pages.

Three positive examples and nine negative examples were randomly selected from the training dataset to be initial labeled data. Therefore, each data set contains 12 initial labeled examples, 776 training examples and 263 testing examples. We then used 3-fold cross-validation for averaging the results. Three positive and nine negative samples are used as initial labeled data for incremental-ICT and Co-Training algorithms. The parameters *p* and *n* in Table 1 and Table 2 is set to 1 and 3, respectively.

5.2 The Results

Standard precision (P), recall (R), accuracy (A) and F1-measure (F1) are used to evaluate the performance of the classifiers. These are defined as follows.

$$P = \frac{\text{no. of correctly predicted positive examples}}{\text{no. of predicted positive examples}}$$

$$R = \frac{\text{no. of correctly predicted positive examples}}{\text{no. of all positive examples}}$$

$$A = \frac{\text{no. of correctly predicted examples}}{\text{no. of all examples}}$$

$$F1 = \frac{2PR}{P+R}$$

Table 3: Performance of classifiers using 3-fold cross-validation: P = Precision , R = Recall, A = Accuracy, F1 = F1-measure.

Classifier	P(%)	R(%)	A (%)	F1
I-ICT (page)	94.04	80.46	94.55	86.72
S-Bayes (page)	77.48	94.35	92.76	85.09
S-Bayes (hyperlink)	87.81	62.17	89.81	72.80
I-ICT (hyperlink)	67.54	72.41	85.17	69.89
Co-Training (hyperlink)	62.41	59.19	81.62	60.75
Co-Training (page)	91.91	34.49	85.17	50.15

The experimental results are shown in Table 3. In Table 3, I-ICT (page) and I-ICT (hyperlink) stand for the page-based and hyperlink-based naï ve Bayes classifiers of incremental-ICT, respectively. Co-Training (page) and Co-Training (hyperlink) are page-based and hyperlink-based naï ve Bayes classifiers of Co-Training algorithm, respectively. S-Bayes (page) and S-Bayes (hyperlink) are supervised naï ve-Bayes classifiers, which classify Web pages based on words in Web pages and words in hyperlinks, respectively.

As shown in the table, I-ICT(page) gives the best performance followed by S-Bayes (page), S-Bayes (hyperlink), I-ICT (hyperlink), Co-Training (hyperlink) and Co-Training (page). The reason that I-ICT (page) gives better performance compared to S-Bayes is because I-ICT (page) cooperates with I-ICT (hyperlink) while S-Bays uses single classifier. The performance of I-ICT (hyperlink) is not good

as that of I-ICT (page). This is because hyperlinks contain fewer words and thus are less capable of building accurate classifier. The training technique of I-ICT is also an effective way as its performance is better than that of Co-Training which uses a different training technique.

6. Discussion and conclusion

ICT algorithm has been proved to be robust under new assumption that each example can be viewed in two different views. With the consistency checking process which is used to compensate the lack of domain's knowledge of the classifiers, our algorithm reached 94.04% precision and 80.46% recall and outperformed supervised naï ve Bayes algorithm, and Co-Training.

Acknowledgments

This work was supported by National Electronics and Computer Technology Center, Research, development and engineering project NT-B-06-4F-13-311.

References

[1] Kijsirikul, B., Sasipongpairoege, P., Soonthornphisaj, N. and Meknavin, S. Supervised and Unsupervised Learning Algorithms for Thai Web Pages Identification, Proceeding of the Pacific Rim International Conference on Artificial Intelligence (PRICAI-2000), 690- 700, August 2000.

[2] Blum, A. and Mitchell, T. Combining Labeled and Unlabeled Data with Co-Training, Proceeding of the Eleventh Annual Conference on Computational Learning Theory, 1998.

[3] Cohen, W. and Singer, Y. Context-sensitive learning methods for text categorization, ACM Transactions on Information Systems, 17(2): 141-173, 1999.

[4] Joachims, T. Text categorization with support vector machines: Learning with many relevant feature, Proceedings Tenth European Conference on Machine Learning, Springer Verlag, 1998.

[5] Nigam, K., McCallum, A., Thrun, S. and Mitchell, T. Text classification from labeled and unlabeled documents using EM. Machine Learning, 39(2/3): 103-134, 2000.

[6] Apte, C., and Damerau, F. Automated learning of decision rules for text categorization, ACM TOES, 12(2): 233-251, 1994.

[7] Mitchell, T. Machine Leaming, McGraw-Hill. New York, 180-184, 1997.

[8] The Word Wide Knowledge Base (web-kb) project, http://www.cs.cmu.edu/afs/cs.cmu.edu/project /theo-51/www/co-training/data/course-cotrain-data.tar.gz, Carnegie Mellon University, U.S.A.