Learning from Positive and Unlabeled Examples with Different Data Distributions

Xiao-Li Li[1], Bing Liu[2]

[1] Institute for Infocomm Research, Heng Mui Keng Terrace, 119613, Singapore
xlli@i2r.a-star.edu.sg
[2] Department of Computer Science, University of Illinois at Chicago,
851 S. Morgan Street, Chicago, IL 60607-7053
liub@cs.uic.edu

Abstract. We study the problem of learning from positive and unlabeled examples. Although several techniques exist for dealing with this problem, they all assume that positive examples in the positive set *P* and the positive examples in the unlabeled set *U* are generated from the same distribution. This assumption may be violated in practice. For example, one wants to collect all printer pages from the Web. One can use the printer pages from one site as the set *P* of positive pages and use product pages from another site as *U*. One wants to classify the pages in *U* into printer pages and non-printer pages. Although printer pages from the two sites have many similarities, they can also be quite different because different sites often present similar products in different styles and have different focuses. In such cases, existing methods perform poorly. This paper proposes a novel technique A-EM to deal with the problem. Experiment results with product page classification demonstrate the effectiveness of the proposed technique.

1 Introduction

Learning from positive and unlabeled examples (or PU learning) can be regarded as a two-class (positive and negative) classification problem, where there are only labeled positive training data, but no labeled negative training data. Since traditional classification techniques require both labeled positive and negative examples to build a classifier, they are thus not suitable for this problem. Although it is possible to manually label some negative examples, it is labor-intensive and time consuming. In the past three years, several techniques [12][13][23][11][9] were proposed to solve the problem. These techniques mainly use a two-step strategy. The first step tries to identify a set of reliable negative documents from the unlabeled set. The second step builds a classifier by iteratively applying a classification algorithm, i.e. EM [5] or SVM [20].

All the existing techniques assume that positive examples in the positive set *P* and positive examples (which are not known) in the unlabeled set *U* are generated from the same distribution. In the context of the Web or text documents, this means that the word features of positive documents in both *P* and *U* are similar and with similar frequencies. This assumption may be violated in practice. For example, one wants to

collect all printer pages from Web. One can use the printer pages from one site as the set P of positive pages and use product pages from each of the other Web sites as U. One wants to classify all the pages in U into printer pages and non-printer pages. Although printer pages from the two sites have many similarities, they may also be quite different. The reason is that different Web sites present similar products in different styles and have different focuses. In such cases, directly applying the existing methods gives poor results. The reason is that the first step of these methods is unable to find reliable negative pages. Consequently, the second step builds poor classifiers.

This paper proposes a novel technique to deal with the problem. The proposed method (called A-EM for Augmented EM) is in the framework of EM [5]. The proposed technique has two novelties:

- We add a large set of irrelevant documents O (which contains almost no positive document) to U. This reduces the level of noise in U (here we regard positive documents in U as noise), which enables us to compute the parameters of the classifier more accurately.
- The EM algorithm generates a sequence of classifiers. However, the performances of this sequence of classifiers may not be necessarily improving. This is a well-known phenomenon due to the mismatch of mixture components and document classes [16][12][11]. We propose a classifier selection (or catch) criterion to select a good classifier from the set of classifiers produced by EM. Although there exist classifier selection methods given in [12] and [9], they perform poorly also due to the different data distributions identified above.

Note that although a classifier can be built using positive documents P (positive class) and irrelevant documents O (negative class), our experiments show that classifiers built with P and O are very poor since irrelevant documents in O can be totally different from the negative documents in U. For example, irrelevant documents in O are about sports, finance, and politics, while the negative documents in U are about computers, TV and digital cameras. In PU learning, the unlabeled set U is usually also the test set. Since irrelevant documents in O are not representative of the negative documents in U, O thus cannot be used as the negative set to build an accurate classifier to classify U.

We have performed a large number of experiments using Web product pages. Classifying such data is critical for many e-commerce applications, which also provides an ideal test case for our technique. Our results show that the new method outperforms existing methods dramatically.

2 Related Work

In [6], a theoretical study of PAC learning from positive and unlabeled examples is reported. [15] studies the problem in a Bayesian framework where the distribution of functions and examples are assumed known. [12] reports sample complexity results and shows from a theoretical point of view how the problem may be solved.

A few practical algorithms were also proposed in [9][11][12][13][23]. They conform to the theory given in [12], and follow a two-step strategy: (1) automatically identifying a set of reliable negative examples from the unlabeled set; and (2) build-

ing a classifier using EM or SVM iteratively. The differences among these methods are in the details of the two steps.

In [12], Liu et al. proposed a method (called S-EM) to solve the problem. It is based on naïve Bayesian classification (NB) and the EM algorithm. The main idea of the method is to first use a spy technique to identify some reliable negative documents from the unlabeled set. It then runs EM to build the final classifier. In [23], a SVM based technique (called PEBL) is proposed to classify Web pages given positive and unlabeled pages. It reports a different method for identifying reliable negative examples and then uses SVM iteratively for classifier building. [24] estimates SVM boundary of positive class for small positive data. [11] reports a technique called Roc-SVM. In this technique, reliable negative documents are extracted by using the information retrieval technique Rocchio [17]. Again, SVM is used in the second step. A classifier selection criterion is also proposed to catch a good classifier from iterations of SVM. In [13], a more principled approach based on a biased formulation of SVM is proposed. The method does not have the first step. Lee and Liu propose a logistic regression based method with a classifier selection method [9].

In [19], one-class SVM is proposed. This technique uses only positive data to build a SVM classifier. However, [11] shows that the results are poor. Unlabeled data does help classification. One-class SVM is also studied in [8] and [4].

Other related works include semi-supervised learning (from a small labeled set and a large unlabeled set), co-training and cross-training [1][2][3][7][16][18][25][21]. They are different from our work as we use no labeled negative data.

In this paper, we only focus on texts in Web pages. It is known that structures and hyperlink information in Web pages also help classification.

3 The Proposed Technique

We now introduce algorithm A-EM to deal with the problem that positive examples in P and hidden positive examples in U may be generated from different distributions.

3.1 NB Classification and EM Algorithm

As mentioned in the introduction section, this work employs the EM framework as in [14][16][12]. Our EM algorithm here is based on naïve Bayesian classification (NB). Before presenting the proposed method, we give an overview of both NB and EM.

The Naive Bayesian method is an effective technique for text classification [10][16]. Given a set of training documents D, each document is considered an ordered list of words. We use $w_{d_i,k}$ to denote the word in position k of document d_i, where each word is from the vocabulary $V = < w_1, w_2, \ldots, w_{|V|} >$. The vocabulary is the set of all words we consider for classification. We also have a set of predefined classes, $C = \{c_1, c_2, \ldots, c_{|C|}\}$ (in this paper we only consider two class classification, so, $C=\{c_1, c_2\}$). In order to perform classification, we need to compute the posterior probability, $Pr(c_j|d_i)$, where c_j is a class and d_i is a document. Based on the Bayesian probability and the multinomial model, we have

$$\Pr(c_j) = \frac{\sum_{i=1}^{|D|} \Pr(c_j \mid d_i)}{|D|} \quad (1)$$

and with Laplacian smoothing,

$$\Pr(w_t \mid c_j) = \frac{1 + \sum_{i=1}^{|D|} N(w_t, d_i) \Pr(c_j \mid d_i)}{|V| + \sum_{s=1}^{|V|} \sum_{i=1}^{|D|} N(w_s, d_i) \Pr(c_j \mid d_i)} \quad (2)$$

where $N(w_t,d_i)$ is the count of the number of times that the word w_t occurs in document d_i and $Pr(c_j|d_i) \in \{0,1\}$ depending on the class label of the document.

Finally, assuming that the probabilities of the words are independent given the class, we obtain the NB classifier:

$$\Pr(c_j \mid d_i) = \frac{\Pr(c_j) \prod_{k=1}^{|d_i|} \Pr(w_{d_i,k} \mid c_j)}{\sum_{r=1}^{|C|} \Pr(c_r) \prod_{k=1}^{|d_i|} \Pr(w_{d_i,k} \mid c_r)} \quad (3)$$

In the naive Bayesian classifier, the class with the highest $Pr(c_j|d_i)$ is assigned as the class of the document.

The Expectation-Maximization (EM) algorithm [5] is a popular class of iterative algorithms for maximum likelihood estimation in problems with incomplete data. It is often used to fill the missing values in the data using existing values by computing the expected value. The EM algorithm consists of two steps, the *Expectation* step, and the *Maximization* step. The *Expectation step* basically fills in the missing data. The parameters are estimated in the *Maximization step* after the missing data are filled. This leads to the next iteration. For the naive Bayesian classifier, the steps used by EM are identical to that used to build the classifier (equations (3) for the *Expectation step*, and equations (1) and (2) for the *Maximization step*). In EM, the probability of the class given a document takes the value in [0, 1] instead of {0, 1}.

3.2 The General Algorithm: A-EM

We now present the general algorithm A-EM in this work. The proposed technique consists of three steps: (1) initialization by introducing irrelevant documents, (2) running EM, and (3) selecting the final classifier.

Algorithm A-EM(P, U, O)

1. Let $N = U \cup O$;
2. **For** each $d_i \in N$, let $Pr(+|d_i) = 0$, $Pr(-|d_i) = 1$;
3. **For** each $d_i \in P$, let $Pr(+|d_i) = 1$, $Pr(-|d_i) = 0$;
4. Build the initial naïve Bayesian classifier *NB-C* using equations (1) and (2);
5. **Loop while** classifier parameters change
6. **For** each document $d_i \in N$
7. Compute $Pr(+|d_i)$ and $Pr(-|d_i)$ using *NB-C*;
8. Update $Pr(c_j)$ and $Pr(w_t|c_j)$ by replacing equations (1) and (2) with the new probabilities in step 7 (a new *NB-C* is being built in the process)
9. Select a good classifier from the series of classifiers produced by EM.

Figure 1: The A-EM algorithm with the NB classifier

Initially, each positive document d_i in P is assigned the class label "+" (positive). As we have no labeled negative documents, each document d_j in unlabeled set U is assigned the class label "-" (negative). Our problem is turned into a one-side error problem, i.e., there is a large error in the negative set (which is U here). Using this initial labeling a NB classifier can be built, which is applied to classify documents in U to obtain the posterior probabilities ($Pr(+|d_j)$ and $Pr(-|d_j)$) for each document in U. We then iteratively employ the revised posterior probabilities to build a new NB classifier. The process goes on until the parameters converge. This is the EM algorithm with U as negative N. Figure 1 shows the A-EM algorithm. O (line 1) can be ignored for now. We assume N is U (Section 3.4 will explain why we need to use O).

We now discuss in what situation this algorithm will work well. In our problem, the difficulty in building an accurate classifier lies in the fact that we do not have labeled negative data (but a noisy unlabeled set U). As a result, initially we use all documents in U as negative documents. After learning, the NB classifier *NB-C* (line 4) will use the values of $Pr(c_j)$ and $Pr(w_t|c_j)$ to classify the documents in the unlabeled set U. Equivalently, we can say that we use the classifier to compute the posterior probabilities for each document d_i in U, i.e., $Pr(+|d_i)$ and $Pr(-|d_i)$. If the *NB-C* classifier is good, it will assign most positive documents in U small probabilities of $Pr(-|d_i)$ while high probabilities of $Pr(+|d_i)$. When the EM algorithm is applied to estimate $Pr(c_j)$ and $Pr(w_t|c_j)$ again in the next iteration, it will be more accurate than the first *NB-C* classifier because the EM algorithm does not regard U as negative. Instead, for each document in U, it uses the revised posterior probabilities. So the key issue of this problem is whether the original *NB-C* classifier is able to assign positive documents in U high probabilities. If this is possible, EM will be able to improve the first *NB-C* classifier. However, in practice this may not be the case.

3.3 Problems and Solutions

As discussed previously, in practice positive examples in P and hidden positive examples in U may be generated from different distributions. Thus, they may not be sufficiently similar. As a result, positive examples in U may not be assigned right probabilities by the algorithm in Figure 1. Then EM will produce poor classifiers. We experimented with the above algorithm using the Web page data. The results were quite poor. We also tried the classifier selection methods in [12] and [9], but they do not work either. We believe that the reasons are:

1) Positive documents in P are not sufficiently representative of positive documents in U, although these documents are still similar to a large extend.
2) Due to the problem above, it is difficult to estimate the behavior of positive documents in U using positive documents in P. This makes the existing classifier selection methods ineffective because they all estimate the information about positive documents in U using P.

To deal with these two problems, we propose the following strategies: For the first problem, we amplify or boost the similarity of positive documents in U and P (their similarities are small initially) while reducing the similarity of positive and negative documents in U. We do this by introducing a large number of irrelevant documents

(they are definitely negative documents). This can be easily done because irrelevant documents are easy to find. For example, we are interested in product page classification. We can add a large number of news articles from newspapers.

To deal with the second problem, we need to design a classifier selection method that does not depend on positive documents in P, but only on U. Thus, the distribution difference will not cause problem.

3.4 Introducing Irrelevant Documents: Initialization of A-EM

Recall that the key piece of information needed for classification is $Pr(w_t|c_j)$, where w_t is a word and c_j is a class. If there are a large number of positive examples in U or there are many keywords that are indicative of positive documents also occurring in U very often, then the NB classifier will not be able to separate positive and negative class well.

To deal with this problem, we introduce additional irrelevant documents O (negative) into the original unlabeled set U (line 1 in Fig 1), which reduces the error in U. Basically, adding O will change the probability $Pr(w_t|-)$. Obviously, the proportion of positive documents in $O+U$ is reduced and consequently $Pr(w_t|-)$ is reduced for a positive keyword w_t. Note that $Pr(w_t|+)$ does not change because we do not add anything in the positive set P. In effect, we amplify the positive features in P. In classifying documents in U, those positive documents are likely to get much higher values of $Pr(+|d_i)$, and lower values of $Pr(-|d_i)$. This means that we have boosted the similarity of positive documents in P and U, which allows us to build more accurate classifiers.

3.5 Selecting a Good Classifier

EM works well when the local maximum that the classifier is approaching to separate positive and negative documents. In practice, the behavior of EM can be quite unpredictable due to mismatch of mixture components and document classes [16][9][11] (due to model misfit). Thus, each iteration of EM gives a classifier that may potentially be a better classifier than the classifier produced at convergence. If we run EM n iterations, we have n classifiers from which we need to select a good classifier (to choose the best is difficult).

There are two existing techniques for selecting a classifier in EM. S-EM [12] estimates the change in the probability of error in order to decide which iteration of EM is the final classifier. More specifically, in the selection formula, it estimates the probability that positives in U are classified as negative through checking how many positives in P are classified as negative, i.e., using $Pr_P(f(x)=-|Y=+)$ as an approximation of $Pr_U(f(x)=-|Y=+)$. Lee and Liu proposes another performance criterion in [9], $Pr(f(x)=+|Y=+)^2 / Pr(f(x)=+)$, which also use positive documents from P in a validation set in computation. Note that $f()$ is the classifier, x is a document vector and Y is the class attribute.

As we discussed in Section 3.3, since the positive documents in P are not sufficiently similar to the positive documents in U, the two techniques do not work because they both depend on P. We now propose a new technique that depends primar-

ily on the unlabeled set U. Since it does not use P in evaluation, it is thus independent of positive set P.

Since our task here is to identify positive documents from the unlabeled set U, therefore it is appropriate to use information retrieval measures for our purpose. Here we use the F value to evaluate the performance of the classifier. F value is commonly used in text classification: $F = 2pr/(p+r)$, where p is the precision and r is the recall. We try to select a classifier from EM iterations with the maximal F value.

We now use the confusion matrix to introduce our method (Table 1). A confusion matrix contains information about actual and predicted results given by a classifier.

Table 1. Confusion Matrix of a classifier

	Classified Positive	Classified Negative
Actual Positive	TP	FN
Actual Negative	FP	TN

Here TP is the number of correct predictions of positive documents (true positive); FN is the number of incorrect predictions of positive documents (false negative); FP is the number of incorrect predictions of negative documents (false positive); and TN is the number of correct predictions of negative documents (true negative). Based on the matrix, the precision and recall of positive class can be written as:

$$p = \frac{TP}{TP + FP} \qquad r = \frac{TP}{TP + FN} \tag{4}$$

F value can be written as

$$\begin{aligned} F &= 2 * \frac{TP}{TP + FP} * \frac{TP}{TP + FN} / (\frac{TP}{TP + FP} + \frac{TP}{TP + FN}) \\ &= \frac{2TP}{(TP + FP) + (TP + FN)} \end{aligned} \tag{5}$$

Note that $TP+FP$ is the number of documents that are classified as positive (we denote the document set as CP) and $TP+FN$ is the actual number of positive documents in U (we denote it as PD, and it is a constant).

So F value can then be expressed as:

$$F = \frac{2TP}{|CP| + PD} \tag{6}$$

We use an estimate of change in F value to decide which iteration of EM to select as the final classifier. The change in F value from iteration i-1 to i (F value in ith iteration divided the F value in (i-1)th iteration) is

$$\Delta_i = \frac{F_i}{F_{i-1}} = \frac{TP_i}{TP_{i-1}} * \frac{|CP_{i-1}| + PD}{|CP_i| + PD} \tag{7}$$

In the EM algorithm, we select iteration i as our final classifier if Δ_i is the last iteration with value greater than 1. $|CP_{i\text{-}1}|$ and $|CP_i|$ are the number of documents classified as positive in iteration i and i+1 respectively. We estimate PD by using the number of documents classified as positive when EM converges. Next, we estimate $TP_i/TP_{i\text{-}1}$ since it is impossible to directly estimate either TP_i or $TP_{i\text{-}1}$.

Our idea here is that first we get a set K of representative keywords for the positive

class. This is done as follows: first we compute $Pr(w_t|+)$ for each word in the positive set P. We then rank the probabilities from large to small and fetch the top $|K|$ keywords. We observe that although the distributions of positive documents in U and P are different, we can still find good keywords from the positive class. For example, in category *printer* of our data, we obtain representative keywords "printer", "inkjet", "Hewlett", "Packard", "ppm" etc.

For a document, the more positive keywords it contains, the more likely it belongs to the positive class. So, we use

$$\frac{\sum_{t}^{|K|} N(w_t, d_i), d_i \in CP_i}{\sum_{t}^{|K|} N(w_t, d_i), d_i \in CP_{i-1}} \quad (8)$$

to estimate TP_i/TP_{i-1}, where $\sum_{t}^{|K|} N(w_t, d_i), d_i \in CP_i)$ is the total number of keywords in the document set CP_i. Intuitively, for a set CP_i (documents classified as positive) in an EM iteration, the larger the total number of positive keywords are in CP_i, the more true positive documents it contains. For instance, if CP_i contains more printer keywords, then it is likely that CP_i contains more true printer pages. It is thus reasonable to use equation (8) to estimate TP_i/TP_{i-1}.

4 Empirical Evaluation

This section evaluates the proposed technique. We compare it with existing methods, i.e., Roc [17], Roc-SVM [11], and PEBL [23]. Roc-SVM is available on the Web as a part of the LPU system (http://www.cs.uic.edu/~liub/LPU/LPU-download.html). We implemented PEBL and Roc as they are not publicly available. We do not include the results of S-EM [12] as it does not perform well due to its spy technique which heavily depends on the similarity of positive pages in U and in P.

4.1 Datasets

Our empirical evaluation is done using Web pages from 5 commercial Web sites, Amazon, CNet, PCMag, J&R and ZDnet. These sites contain many description pages of different kinds of products. We use Web pages that focus on the following categories of products: Notebook, Digital Camera, Mobile Phone, Printer and TV. Table 2 lists the number of documents downloaded from each Web site, and their corresponding classes.

The construction of positive set P and unlabeled set U is done as follows: we use web pages of a particular type of product from a single site ($Site_i$) as positive pages P, e.g., camera pages from Amazon. The unlabeled set U is the set of all product pages from another site ($Site_j$) ($i \neq j$), e.g., CNet. We also use U as the test set in our experiments because our objective is to extract or to recover those hidden positive pages in U, e.g., camera pages in CNet. In preprocessing, we removed stopwords but did not perform stemming.

Note that traditional text classification corpora, e.g., 20-Newsgroups and Reuters, are not used as the primary datasets in our experiments because these datasets do not have the different distribution problem discussed in this paper.

Table 2. Number of Web pages and their classes

Web sites	Amazon	CNet	J&R	PCMag	ZDnet
Notebook	434	480	51	144	143
Camera	402	219	80	137	151
Mobile	45	109	9	43	97
Printer	767	500	104	107	80
TV	719	449	199	0	0

The irrelevant document set O is from the corpora: 20-Newsgroups and Reuters-21578. In each experiment, we randomly select a% of documents from the Reuters or 20-Newsgroups collection and add them to U. In our experiments, we use 6 a values to create different settings, i.e., 0%, 20%, 40%, 60%, 80%, and 100%.

4.2 Results

We performed a comprehensive set of experiments using all 86 P and U combinations. For each combination dataset, we randomly add irrelevant documents from Reuters and 20-Newsgroups respectively with different a values. In other words, we select every entry (one type of product from each Web site) in Table 2 as the positive set P and use each of the other 4 Web sites as the unlabeled set U. As we discussed in Section 3.5, we use F value to evaluate the performance of our classifiers. F value only measures retrieval results of the positive class. This is suitable for us as we want to see if we can identify positive pages from the unlabeled set.

Table 3 shows the results when P is the set of camera pages from Amazon and U is the set of all product pages from CNet. We added Reuters data to U. Column 1 gives the percentage of Reuters documents added to U (a=0% means that no Reuters document is added). Column 2 to Column 10 show the results of NB, 1EM, …, 8EM (EM usually converges within 8 iterations). From Table 3, we observe that for this dataset when a = 0%, the results of NB and all EM iterations are zero. For other values of a, EM improves NB's results tremendously if we are able to select a good EM classifier. Note that we can see that the converged EM may not give the best classifier.

Table 3. A-EM results (F values in %) for P (camera pages from Amazon) and U (pages from Cnet) with different a settings

Add%	NB	1EM	2EM	3EM	4EM	5EM	6EM	7EM	8EM
a=0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
a=20	2.70	4.50	12.0	30.9	83.8	**100**	100	100	100
a=40	19.0	93.7	**100**	100	100	100	100	100	100
a=60	44.7	**100**	98.9	98.0	96.5	95.6	95.0	93.6	89.2
a=80	70.4	**98.9**	96.1	92.8	82.8	75.1	72.9	69.5	60.5
a=100	80.2	**96.3**	87.8	74.9	70.1	59.2	41.8	40.9	37.0

In our experiments, we use $|K| = 10$ (10 keywords) in equation (8). We found that different values of $|K|$ give similar results as long as $|K|$ is not too small.

For this dataset, our technique in Section 3.5 is able to catch the best classifier for each a value. That is, for a=20%, a=40%, a=60%, a=80% and a=100%, the selected classifiers are from 5EM (F = 100%), 2EM (F = 100%), 1EM (F = 100%), 1EM (F = 98.9%), 1EM (F = 96.3%) respectively. These results and those in Table 3 show that the amount of irrelevant data added to the unlabeled set also has an effect on the final classifier. Thus we also need to decide which classifier to use from the series of classifiers produced using different a values. We select the final classifier with the following formula:

$$\max_{a} \left(\frac{\sum_{t}^{|K|} N(w_t, d_i), d_i \in CP_{a,i}}{|\, CP_{a,i} \,|} \right), \tag{9}$$

where $CP_{a,i}$ is the set of documents in U classified as positive using the selected classifier i for a particular a. This formula shows that we choose the classifier of the a value that has the highest average keyword count per page in $CP_{a,i}$.

Table 3 gives a good sample of the type of results that we get using EM with or without adding irrelevant documents to the unlabeled set. We observe that adding irrelevant data helps tremendously (see NB and 1EM).

Since we have conducted a large number of experiments (all 86 combinations) and compared A-EM with existing methods, to avoid gory details we summarize all the results here. Table 4 shows the summarized results when Reuters data and 20-Newsgroups data (as the irrelevant data set O) are added to U. Columns 2, 3 and 4 give the average F values of Roc, Roc-SVM and PEBL respectively. Column 5 shows the corresponding results of A-EM. For techniques Roc-SVM and PEBL, we use the same classifier selection methods as for A-EM.

Table 4. Results (F values in %) of adding Reuters and 20 Newsgroups data

Dataset added	Roc	Roc-SVM	PEBL	A-EM
Reuters	**64.5**	**73.4**	**72.3**	**87.2**
20 Newsgroups	**66.7**	**72.6**	**72.1**	**89.1**

Clearly, we can see that A-EM outperforms the other techniques dramatically. For adding Reuters data, the average F value for A-EM is 87.2%, much higher than other methods. The results of adding 20-Newsgroups are similar and A-EM's average F value is 89.1%, which is also much higher than other methods. This shows that the type of irrelevant data is not important. Another important observation is that A-EM's results are consistent, while the results of the other methods vary a great deal.

Next, we show the effectiveness of adding irrelevant documents. Table 3 already gives a good indication. However, it only shows a single data set. Table 5 summaries the average all 86 experiments of adding 20-Newsgroups documents with different a values using A-EM. From the table we can see that after adding irrelevant documents to the unlabeled sets, the results of both NB and EM (1EM-4EM) improve as compared to adding nothing (a=0%). The situation is similar for adding Reuters data.

From these results, we conclude that adding irrelevant data to unlabeled sets can improve the results dramatically with almost no negative effect. Comparing the results of A-EM in Tables 4 and results of EM in Table 5, it is also clear that fixing any particular iteration of EM (Table 5) as the final classifier is not a good solution. Classifier selection is an essential step in A-EM.

Table 5. Average results (*F* values) of adding 20 Newsgroup documents with different *a* values

Add%	NB	1EM	2EM	3EM	4EM	5EM	6EM	7EM	8EM
a=0	20.0	28.4	36.3	40.9	43.4	45.9	47.0	47.6	48.5
a=20	54.9	76.1	77.8	75.5	73.9	71.2	68.9	67.5	66.1
a=40	65.2	78.2	71.9	64.4	59.7	56.8	54.3	53.3	52.4
a=60	69.5	78.4	67.4	59.1	53.8	51.6	50.0	49.0	48.6
a=80	71.9	76.2	62.6	53.9	49.5	46.9	45.0	44.2	44.2
a=100	73.4	75.7	60.4	51.8	47.2	44.6	43.7	43.4	43.4

Table 6 further illustrates the effectiveness of our classifier selection method, which lists another set of summarized results. It compares the optimal results (manually selected best results after checking the test results from all EM iterations) and our selection results. We observe that the selection results are very close to the optimal results for both adding 20-Newsgroups data and Reuters data to the unlabeled sets, which means our classifier selection method is very effective.

Table 6. Best results and results of A-EM selected classifiers

Dataset Added	Best results	Selection results
20 Newsgroups	92.0	89.1
Reuters	89.8	87.2

Finally, we also tried to use *O* as the negative set (instead of *U*) to build a classifier (SVM or NB) with *P*, the results were very poor. On average, the NB classifier learned based on *P* and *O* only gets 12.1% *F* value because most negative pages in *U* are very different from those in *O*.

5 Conclusion

This paper studied the PU learning problem with different distributions of positive examples in *P* and positive examples in *U*. We showed that existing techniques performed poorly in this setting. This paper proposed an effective technique called A-EM to deal with the problem. The algorithm first boosts the similarity of the positive documents in *U* and *P* by introducing a large number of irrelevant documents into *U*. It then runs EM to construct a series of classifiers. A novel method for selecting a good classifier from the set of classifiers was also presented. Experimental results with product page classification show that the proposed technique is more effective than existing techniques.

Acknowledgments: The work of Bing Liu is supported by the National Science Foundation (NSF) under the NSF grant IIS-0307239. Xiaoli Li would like to thank Dr See-Kiong Ng for his help.

References

1. Basu S., Banerjee A., and Mooney R. 2002. Semi-supervised clustering by seeding. ICML-2002.
2. Blum A. and Mitchell T. 1998. Combining labeled and unlabeled data with co-training. COLT-98.
3. Bockhorst J. and Craven M. 2002. Exploiting relations among concepts to acquire weakly labeled training data. ICML-2002.
4. Crammer K., Chechik G. 2004. A needle in a haystack: local one-class optimization, ICML-2004.
5. Dempster A., Laird N. and Rubin D. 1977. Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society, 1977.
6. Denis F. 1998. PAC learning from positive statistical queries. ALT-98, pages 112-126. 1998.
7. Goldman S., Zhou Y. 2000. Enhancing supervised learning with unlabeled data. ICML-2000.
8. Koppel M. and Schler J. 2004. Authorship Verification as a one-class classification problem, ICML-2004.
9. Lee W., Liu B. 2003. "Learning with positive and unlabeled examples using weighted logistic regression. ICML-2003.
10. Lewis D. and Gale W. 1994. A sequential algorithm for training text classifiers. SIGIR-1994.
11. Li X., Liu B. 2003. Learning to classify text using positive and unlabeled data. IJCAI-2003.
12. Liu B., Lee W., Yu *P*., and Li X. 2002. Partially supervised classification of text documents. ICML-2002.
13. Liu B., Dai Y., Li X., Lee W., and Yu *P*. 2003. Building text classifiers using positive and unlabeled examples. ICDM-2003.
14. McCallum A. 1999. Multi-label text classification with a mixture model trained by EM. In AAAI 99 Workshop on Text Learning.
15. Muggleton S. 2001. Learning from the positive data. Machine Learning, 2001.
16. Nigam K., McCallum A., Thrun S., Mitchell T. 2000. Text classification from labeled and unlabeled documents using EM. Ma-chine Learning, 2000.
17. Rocchio J. 1971. Relevant feedback in information retrieval. In Salton G. eds. The smart retrieval system: experiments in auto-matic document processing. Englewood Cliffs, 1971.
18. Sarawagi S., Chakrabarti S. and Godbole S. 2003. Cross-training: learning probabilistic mappings between topics. KDD-2003.
19. Scholkopf B., Platt J., Shawe J., Smola A. and Williamson R. 1999. Estimating the support of a high-dimensional distribution. Technical Report MSR-TR-99-87, Microsoft Research, 1999.
20. Vapnik V. 1995. The nature of statistical learning theory.
21. Wu P., Dietterich T., G. 2004. Improving SVM accuracy by training on auxiliary data sources, ICML-2004.
22. Yang Y. and Liu X. 1999. A re-examination of text categorization methods. SIGIR-1999.
23. Yu H., Han J., and Chang K. 2002. PEBL: Positive example based learning for Web page classification using SVM. KDD-2002.
24. Yu H. 2003. General MC: Estimating boundary of positive class from small positive data. ICDM-2003.
25. Zelikovitz S. and Hirsh H. 2000. Improving short text classification using unlabeled background knowledge to assess document similarity. ICML-2000.