Learning with Positive and Unlabeled Examples Using Weighted Logistic Regression

Wee Sun Lee LEEWS@COMP.NUS.EDU.SG
Department of Computer Science and Singapore-MIT Alliance, National University of Singapore, Singapore 117543

Bing Liu LIUB@CS.UIC.EDU
Department of Computer Science, University of Illinois, Chicago, 851 South Morgan St., Chicago IL 60607-7053

Abstract

The problem of learning with positive and unlabeled examples arises frequently in retrieval applications. We transform the problem into a problem of learning with noise by labeling all unlabeled examples as negative and use a linear function to learn from the noisy examples. To learn a linear function with noise, we perform logistic regression after weighting the examples to handle noise rates of greater than a half. With appropriate regularization, the cost function of the logistic regression problem is convex, allowing the problem to be solved efficiently. We also propose a performance measure that can be estimated from positive and unlabeled examples for evaluating retrieval performance. The measure, which is proportional to the product of precision and recall, can be used with a validation set to select regularization parameters for logistic regression. Experiments on a text classification corpus show that the methods proposed are effective.

1. Introduction

In retrieval applications it is very common to have situations where positive and unlabeled examples are available but negative examples cannot be obtained without paying an additional cost. For example, in trying to learn a classifier for a user's preference for web pages, the user's bookmarks can be considered as positive examples while unlabeled examples can be sampled from the web. In direct marketing, it is desirable to have a classifier that can identify future customers from the customer profiles. The company's current list of customers can be considered as positive examples, while new databases of unlabeled examples can be purchased at a low cost compared to the cost of obtaining negative examples.

In this paper, we use the following simple model for learning with positive and unlabeled examples: positive examples are randomly labeled positive with probability $1 - \alpha$ and are left unlabeled with probability α (see (Denis, 1998)). Under this assumption, if we labeled all the unlabeled examples as negative, we will never make an error on a negative example but will randomly label positive examples as negative with probability α.

One problem with this formulation is that the value of α is unknown. Blum and Mitchell (Blum & Mitchell, 1998) observed that given $\alpha < 1$, function f, noisy observed label Y', actual label Y and input X, $\Pr[f(X) = 1|Y' = -1] + \Pr[f(X) = -1|Y' = 1]$ is linearly related to $\Pr[f(X) = 1|Y = -1] + \Pr[f(X) = -1|Y = 1]$. The expression $\Pr[f(X) = 1|Y' = -1] + \Pr[f(X) = -1|Y' = 1]$ is the expected sum of observed false positive and false negative error frequencies. Since the target function has zero actual error, minimizing the sum of observed false positive and false negative frequencies will give a good approximation of the target function when the target function is in the function class used and the sample size is large enough. This avoids the need to know the value of α.

Minimizing the expected sum of false positive and false negative error frequencies can be shown to be equivalent to minimizing expected weighted error where false positives are multiplied by $\Pr[Y' = -1]$ and false negatives are multiplied by $\Pr[Y' = 1]$. Unfortunately, minimizing weighted errors is NP-hard for linear functions (Hoffgen et al., 1995). In this paper, instead of minimizing weighted errors, we learn the real-valued conditional probability of observing a positive label given the input by performing logistic regression. We perform regularization by forming the cost function to optimize from the sum of squared of the weights of the linear function and the sum of weighted logit losses. The resulting cost function is convex and we optimize using simple gradient descent.

In practice, the function class that we use may not be the correct function class to use for learning. Even if the correct target function class is known, we may want to use a simpler approximating function (for example by regularization) when the training sample is small in order to

Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003), Washington DC, 2003.

obtain better generalization. In this paper, we show that $pr/\Pr[Y=1]$ where p is the precision $\Pr[Y=1|f(X)=1]$ and r is the recall $\Pr[f(X)=1|Y=1]$, can be estimated from positive and unlabeled examples. This function is maximized by the target function and generally we want both the precision and recall to be high in a retrieval situation. Being able to estimate the function from positive and unlabeled examples allows us to use performance on a validation set to select the appropriate regularization parameter to use for learning. We performed experiments using a validation set to select the regularization parameter for logistic regression on a text classification task. The results show that the method used is effective.

The paper has two main contributions. The first is the use of a real valued output instead of thresholded binary outputs for linear function on weighted examples for learning with positive and unlabeled examples. This allows the use of maximum likelihood which gives a convex cost function for optimization. The second contribution is the introduction of a performance measure that can be estimated from positive and unlabeled examples. This performance measure can be used for selecting the regularization parameter from a validation set when only positive and unlabeled examples are available.

In Section 2, we discuss related work. Section 3 describes in detail our algorithm for learning linear functions. We derive our estimate for $pr/\Pr[Y=1]$ using positive and unlabeled data in Section 4 and give experimental results in Section 5.

2. Related Works

A theoretical study of Probably Approximately Correct (PAC) learning from positive and unlabeled examples was done in (Denis, 1998). Using the model where a positive example is left unlabeled with constant probability, it was shown that function classes learnable under the statistical queries model (Kearns, 1998) is also learnable from positive and unlabeled examples. Learning from positive example was also studied theoretically in (Muggleton, 2001) within a Bayesian framework where the distribution of functions and examples are assumed known. Sample complexity for the case where the positive and unlabeled examples can be sampled is given in (Liu et al., 2002), where it was shown that maximizing the number of examples classified as negative while constraining the function to correctly classify positive examples will give good performance with large enough sample size.

In (Liu et al., 2002), the expectation maximization (EM) algorithm was used with the naive bayes model to approximately maximize the number of examples classified as negative while approximately constraining the positive examples to be classified correctly. This is done by initializing the generative model for the negative examples with a subset of unlabeled examples that are highly likely to be negative, followed by performing EM iterations where the labels of the positive examples are kept positive but the labels of unlabeled examples are allowed to change. A similar algorithm using support vector machines (SVM) was proposed in (Yu et al., 2002). The algorithm first finds some examples that can be confidently labeled as negative and then trains the support vector machine with the positive examples and the examples that have been confidently labeled as negative. As the support vector machine tries to find the maximal margin hyperplane, more unlabeled examples may be classified as negative than the initially labeled negative examples. This forms a new negative set which is again used to train the support vector machine along with the positive set. The process is iterated in order to label more and more unlabeled examples as negative while retaining the positive examples correctly labeled. Both the EM and iterated SVM algorithms are not guaranteed to find functions that label large number of unlabeled examples as negative even when such functions exists in the function class.

The naive bayes algorithm was modified to learn from positive and unlabeled examples in (Denis et al., 2002). This is done by subtracting an estimate of the positive examples from the negative model. However, the algorithm requires prior knowledge of the probability of positive examples. An alternative to using such knowledge would be to use performance estimates that can be estimated from positive and unlabeled examples, such as the function $pr/\Pr[Y=1]$ that is proposed in this paper, on a validation set. When the probability that the positive example is left unlabeled is constant, it is also possible to modify the perceptron algorithm so that it is able to learn from positive and unlabeled examples using ideas from (Bylander, 1994; Blum et al., 1996). However, so far our attempts on using such algorithms have not been very successful, most probably due to the lack of good regularization criterion as we have been experimenting with datasets with very large input dimensions.

It is also possible to discard the unlabeled data and learn only from the positive data. This was done in the one-class SVM (Scholkopf et al., 1999), which tries to learn the support of the positive distribution. This method appears to be highly sensitive to the input representation and dimensionality (Manevitz & Yousef, 2001) and did not perform well on the feature set that we used in this paper.

Besides learning from positive and unlabeled examples, there has been considerable interest in learning from a small number of labeled positive and negative examples with a large number of unlabeled examples. Works on this topic include (Nigam et al., 1998) which uses naive bayes and the EM algorithm, (Joachims, 1999) which uses transductive SVM and (Blum & Mitchell, 1998) which exploits conditional independence of multiple views of the data to do co-training.

3. Learning Linear Functions

Linear functions of the form $g(x) = \sum_{j=1}^{k} w_j x_j + b$, where $x_j, j = 1, \ldots, n$ are the components of the input vector x and b is the bias, are practically effective and commonly used in machine learning.

To model the process generating the positive and unlabeled examples, we assume that positive examples are randomly left unlabeled with probability α while negative examples are always left unlabeled. By labeling all unlabeled examples as negative, positive examples are wrongly labeled with probability α while negative examples are never wrongly labeled. Let X be the random variable representing the input vector, Y be the actual label and Y' be the observed noisy label. For any function f we define its expected observed sum of false positive and false negative error frequencies $C(f) = \Pr[f(X) = -1|Y' = 1] + \Pr[f(X) = 1|Y' = -1]$ and expected actual sum of false positive and false negative error frequencies $C'(f) = \Pr[f(X) = -1|Y = -1] + \Pr[f(X) = 1|Y = -1]$. Blum and Mitchell (Blum & Mitchell, 1998) showed that if the positive examples have constant noise rate α and the negative examples have constant noise rate β, then

$$C(f) = 1 - \frac{(1-\alpha-\beta)\gamma(1-\gamma)(1-C'(f))}{\Pr[Y'=1]\Pr[Y'=-1]}$$

where $\gamma = \Pr[Y = 1]$. If we label the unlabeled examples as negative, the positive examples will have noise rate α while the negative examples will have noise rate $\beta = 0$. Minimizing $C(f)$ will also minimize $C'(f)$. Observing that

$$\begin{aligned}
&\Pr[f(X) = -1|Y' = 1] + \Pr[f(X) = 1|Y' = -1] \\
= \quad & \frac{\Pr[f(X) = -1, Y' = 1]}{\Pr[Y' = 1]} + \frac{\Pr[f(X) = 1, Y' = -1]}{\Pr[Y' = -1]} \\
= \quad & \frac{\Pr[Y' = -1]\Pr[f(X) = -1, Y' = 1]}{\Pr[Y' = 1]\Pr[Y' = -1]} \\
& + \frac{\Pr[Y' = 1]\Pr[f(X) = 1, Y' = -1]}{\Pr[Y' = 1]\Pr[Y' = -1]}
\end{aligned}$$

minimizing $\Pr[f(X) = -1|Y' = 1] + \Pr[f(X) = 1|Y' = -1]$ is equivalent to minimizing the expected weighted errors, where false negatives are weighted by $\Pr[Y' = -1]$ while false positives are weighted by $\Pr[Y' = 1]$.

However, minimizing weighted errors is NP-hard (Hoffgen et al., 1995). Instead of minimizing the false positive and false negative error frequencies, we assume that the function class is a real valued function class that is powerful enough to represent the conditional probability that the label is positive given the input. We show that if we multiply examples that are labeled positive by $\Pr[Y' = -1]$ and examples that are labeled negative by $\Pr[Y' = 1]$, the conditional probability of a positive label given that the example is a positive example is greater than 0.5 and the conditional probability of a positive label given that the example is a negative example is less than 0.5. This allows us to threshold the real valued conditional probability at 0.5 to obtain the correct classification.

Let $\psi = \Pr[Y' = 1]$ and $\gamma = \Pr[Y = 1]$. The expected fraction of examples that are labeled positive is

$$\psi = \gamma(1-\alpha).$$

Similarly, the expected fraction of examples that are labeled negative is

$$(1-\psi) = \gamma\alpha + (1-\gamma).$$

We first consider the behaviour of a positive instance x. The probability that it is labeled positive is $1 - \alpha$. Each positively labeled example is multiplied by the weight $1-\psi$ giving an expected positive weight of

$$(1-\alpha)(1-\psi) = (1-\alpha)\left[\gamma\alpha + (1-\gamma)\right]$$

on x. Similarly the expected negative weight on x is

$$\alpha\psi = \alpha\left[\gamma(1-\alpha)\right].$$

Normalizing to equate the weights with probabilities, we see that the conditional probability of the positive label has been transformed into

$$\frac{(1-\alpha)\left[\gamma\alpha + (1-\gamma)\right]}{(1-\alpha)\left[\gamma\alpha + (1-\gamma)\right] + \alpha\left[\gamma(1-\alpha)\right]} = \frac{\gamma\alpha + (1-\gamma)}{2\gamma\alpha + (1-\gamma)}.$$

This is greater than 0.5 as long as $\alpha < 1$ and $\gamma < 1$. Hence, after weighting, we can see that if we set the threshold at 0.5, we will obtain the correct classification. Since the probability of a negative example being labeled as positive is zero, thresholding at 0.5 will also give the correct classification for negative examples.

To learn the conditional probability, we compose the linear function $g(x)$ with the sigmoid function to obtain $h(x) = 1/(1 + e^{-g(x)})$. We then perform maximum likelihood estimation on weighted positive and negative examples, where the weighting can be interpreted as multiple copies of the same examples. For each unweighted example, we obtain the logit loss $l(y, g(x)) = \ln(1 + e^{-yg(x)})$. Summing over weighted examples, we obtain $\frac{1}{n^{(+)}} \sum_{y^i=1} l(y^i, g(x^i)) + \frac{1}{n^{(-)}} \sum_{y^i=-1} l(y^i, g(x^i))$ where $n^{(+)}$ is the number of positive examples, $n^{(-)}$ is the number of negative examples and $(x^1, y^1), \ldots, (x^n, y^n)$ is the set of examples. This is equivalent to minimizing the cost $\sum_{y^i=1} \frac{n^{(-)}}{n^{(+)}} l(y^i, g(x^i)) + \sum_{y^i=-1} l(y^i, g(x^i))$. Finally, to prevent overfitting, we add the sum of squared values of the weights as a regularization term to obtain the final cost function $\sum_{y^i=1} \frac{n^{(-)}}{n^{(+)}} l(y^i, g(x^i)) + \sum_{y^i=-1} l(y^i, g(x^i)) + c(\sum_{j=1}^{k} w_j^2 + b^2)$, where c is the regularization parameter that can be adjusted to prevent overfitting. This cost function is convex.

If the function class is powerful enough to represent the conditional probability, then maximum likelihood estimation will give us accurate approximation when the sample size is large enough. In the case where the function class is not powerful enough, it is useful to view the logit loss as an upper bound to the $0-1$ loss (Mason et al., 1999). In this case, we are trying to minimize an upper bound to the sum of false positive and false negative frequencies, which still makes good sense even when the function class is not powerful enough to give a good approximation to the real valued conditional probability but can give a good approximation to the classifier.

To optimize the cost, we do simple gradient descent. The gradient of the loss function $l(y, g(x))$ with respect to w_j is simply $\frac{dl(y,g(x))}{dw_j} = -x_j y e^{-yg(x)}/(1+e^{-yg(x)})$. Let $w_{j,t}$ be the jth component of the weight vector at epoch t and let $g_t(x) = \sum_{j=1}^{k} w_{j,t}x_j + b$. Let

$$\begin{aligned}\Delta_{j,t} &= \sum_{y_i=1} \frac{n^{(-)}}{n^{(+)}} x_j y^i e^{-y^i g_t(x^i)}/(1+e^{-y^i g_t(x^i)}) \\ &\quad + \sum_{y^i=-1} x_j y^i e^{-y^i g_t(x^i)}/(1+e^{-y^i g_t(x^i)})\end{aligned}$$

be the jth component of the negative gradient of the weighted sum of losses at epoch t. The jth component negative gradient of the sum of squared weights is simply $-w_j$. We add a momentum term $\gamma\Delta_{j,t-1}$, $\gamma < 1$, to the gradient of the sum of losses (see e.g. (Mitchell, 1997)) to accelerate convergence of the gradient descent. Our update at each epoch t then becomes

$$w_{j,t} = (1-c)w_{j,t-1} + \eta(\Delta_{j,t} + \gamma\Delta_{j,t-1}),$$

where η is the learning rate. Bias is implemented by extending the feature vector by an additional component and setting the additional component to a constant for all examples. Updates for the bias is treated in the same way as updates for the weights.

4. Estimating Performance using Positive and Unlabeled Examples

Prevention of overfitting is crucial when learning with noise. This can often be done by using a validation set to select the regularization parameter c. Although the target function minimizes $\Pr[f(X)=1|Y'=-1] + \Pr[f(X)=-1|Y'=1]$, the sum of positive and negative error frequencies is not necessarily good for selecting the regularization parameter when the function class is not able to represent the target function accurately.

The need to learn from positive and unlabeled examples often arise in retrieval situations, where we have a collection of positive examples and would like to retrieve more positive examples from a source of unlabeled examples. In these scenarios, the ratio of positive to negative examples is often quite small. A commonly used performance measure in retrieval situations is the F score, where $F = 2pr/(p+r)$ with precision $p = \Pr[Y=1|f(X)=1]$ and recall $r = \Pr[f(X)=1|Y=1]$. The F score is the harmonic mean of the precision and recall. To get a high F score, both precision and recall must be high. Unfortunately we do not know how to estimate the F score from positive and unlabeled examples. Instead, we propose a performance criteria for comparing models (or regularization parameters) $pr/\Pr[Y=1]$ that can be estimated directly from the validation set without making additional assumptions. To see this, note that

$$\begin{aligned}&\Pr[f(X)=1|Y=1]\Pr[Y=1] \\ &= \Pr[Y=1|f(X)=1]\Pr[f(X)=1] \\ &\Leftrightarrow \frac{\Pr[f(X)=1|Y=1]}{\Pr[f(X)=1]} = \frac{\Pr[Y=1|f(X)=1]}{\Pr[Y=1]} \\ &\Leftrightarrow \frac{r}{\Pr[f(X)=1]} = \frac{p}{\Pr[Y=1]}\end{aligned}$$

Multiplying both sides by r, we find that $pr/\Pr[Y=1] = r^2/\Pr[f(X)=1]$. Note that the recall $r = Pr[f(X)=1|Y=1]$ can be estimated from the performance of the hypothesis on the positive labeled examples of the validation set, while $\Pr[f(X)=1]$ can be estimated from the validation set, giving us an estimate of the desired model selection criteria. The performance measure is proportional to the square of the geometric mean of precision and recall. It has roughly the same behaviour as the F score in the sense that it is large when both p and r are large and is small if either p or r is small.

5. Experiments

We performed experiments using the 20 Newsgroup dataset (Lang, 1995). The dataset consists of documents from 20 newsgroups with roughly 1000 documents in each group. The preprocessing is as follows:

- Removal of the headers of each document.
- Removal of stop words.
- Removal of words that ocurred no more than 5 times in the entire corpus.
- Each document is represented by a vector where the components of the vector are the term frequencies of the bag of words in the document.
- The vectors are normalized to have length 1 except when naive bayes based methods (which use the raw word counts) are used.
- An additional component with value 1 is added to implement the bias of the linear function.

Positive Set	Best $\alpha = 0$	Crit. I $\alpha = 0$	Crit. II $\alpha = 0$	Best $\alpha = 0.3$	Crit. I $\alpha = 0.3$	Crit. II $\alpha = 0.3$	Best $\alpha = 0.7$	Crit. I $\alpha = 0.7$	Crit. II $\alpha = 0.7$
atheism	0.670	0.670	0.512	0.675	0.655	0.532	0.616	0.552	0.477
autos	0.814	0.814	0.790	0.804	0.804	0.716	0.731	0.710	0.710
space	0.875	0.875	0.800	0.887	0.887	0.797	0.747	0.738	0.747
graphics	0.633	0.633	0.478	0.643	0.643	0.601	0.518	0.498	0.483
motorcycles	0.888	0.888	0.798	0.873	0.870	0.713	0.799	0.799	0.703
christian	0.772	0.772	0.694	0.741	0.741	0.672	0.645	0.636	0.643
ms-windows	0.723	0.723	0.723	0.717	0.717	0.643	0.644	0.633	0.601
baseball	0.833	0.833	0.833	0.830	0.830	0.804	0.727	0.727	0.638
guns	0.750	0.750	0.720	0.730	0.730	0.552	0.628	0.618	0.596
pc	0.630	0.630	0.508	0.619	0.619	0.475	0.574	0.541	0.503
hockey	0.897	0.897	0.897	0.897	0.897	0.768	0.826	0.826	0.763
mideast	0.888	0.888	0.816	0.877	0.873	0.873	0.815	0.796	0.796
mac	0.749	0.749	0.710	0.767	0.767	0.627	0.677	0.677	0.614
crypt	0.887	0.887	0.874	0.877	0.868	0.824	0.843	0.843	0.816
politics	0.634	0.634	0.470	0.620	0.620	0.482	0.564	0.564	0.454
xwindows	0.773	0.773	0.739	0.760	0.760	0.573	0.681	0.636	0.625
electronics	0.670	0.670	0.647	0.666	0.666	0.540	0.527	0.527	0.435
religion	0.545	0.520	0.520	0.540	0.540	0.366	0.495	0.495	0.386
forsale	0.760	0.760	0.760	0.750	0.732	0.579	0.657	0.591	0.591
med	0.872	0.872	0.872	0.867	0.867	0.786	0.795	0.782	0.789
Average	0.763	0.762	0.708	0.757	0.754	0.646	0.675	0.659	0.619

Table 1. The F scores for using weighted logistic regression to learn with positive and unlabeled examples. Criteria I uses $pr/\Pr[Y=1]$ for the selecting the regularization parameter. Criteria II uses the sum of false positive and false negative frequencies for selecting the regularization parameter. Best indicates performance of the best regularization parameter on the test set.

We performed a random splitting of the data into 3 sets: the training set containing 50% of the documents, the validation set containing 20% of the documents and the test set containing 30% of the documents. For each experiment, each of the newsgroup was alternately made the positive group with the remaining 19 newsgroups made the negative group. The linear function was trained on the training data for $T = 500$ epochs with learning rate $\eta = 1/(\text{number examples})$, momentum parameter $\gamma = 0.99$ and four decay parameters $c \in \{0.005, 0.01, 0.05, 0.1\}$. We tested both the sum of the false positive and false negative frequencies and the estimate of $pr/\Pr[Y=1]$ on the validation set to select the best c parameter. The linear function is retrained on the combined training and validation set using the selected c parameter and then tested on the test data.

Three sets of experiments were performed. The first compares the performance of linear function trained with the logit loss on weighted examples (henceforth refered to as weighted logistic regression) using two methods for selecting the regularization parameter: the sum of false positive and false negative frequencies and the estimate of $pr/\Pr[Y=1]$. Three values of α are used. In this case, different values of α will give a different number of positive examples. The second set of experiments gives the result of S-EM (Liu et al., 2002) and one-class SVM (Scholkopf et al., 1999; Manevitz & Yousef, 2001) on the same feature set for comparison purposes. The last set of experiments uses a slightly different set-up where the number of positive examples are held constant but different number of positive examples are added into the negative set to form the unlabeled set. This simulates the situation where we already have a set of positive examples and try to draw unlabeled examples from a source with unknown probability of obtaining a positive example.

We measure performance on the test set in terms of the F score which is commonly used and familiar to information retrieval practitioners.

5.1. Experiment 1

We created three different datasets. In the first set, no additional errors ($\alpha = 0$) were added. In the second, 30% ($\alpha = 0.3$) of the positive documents in the training and validation sets respectively were made into unlabeled documents (labeled negative) while in the third data set, 70% ($\alpha = 0.7$) of the training and validation documents were made into unlabeled documents.

The results are shown in Table 1, where we compare the new criteria for selecting regularization parameter $pr/\Pr[Y=1]$ (Criteria I) with the best regularization parameter on the test set (Best) and using the sum of positive

Positive Set	$\alpha = 0$ SVM	$\alpha = 0$ NB	$\alpha = 0.3$ S-EM	$\alpha = 0.3$ One-Cls	$\alpha = 0.7$ S-EM	$\alpha = 0.7$ One-Cls
atheism	0.515	0.616	0.546	0.141	0.577	0.138
autos	0.697	0.768	0.644	0.136	0.542	0.139
space	0.771	0.891	0.844	0.117	0.631	0.115
graphics	0.514	0.595	0.513	0.123	0.480	0.122
motorcycles	0.833	0.814	0.784	0.167	0.698	0.161
christian	0.690	0.766	0.679	0.177	0.644	0.181
ms-windows	0.614	0.586	0.572	0.190	0.471	0.216
baseball	0.774	0.844	0.705	0.140	0.705	0.134
guns	0.593	0.777	0.673	0.167	0.580	0.166
pc	0.484	0.540	0.505	0.161	0.470	0.172
hockey	0.880	0.899	0.830	0.156	0.803	0.167
mideast	0.837	0.910	0.855	0.171	0.840	0.185
mac	0.697	0.560	0.515	0.166	0.450	0.173
crypt	0.834	0.926	0.849	0.197	0.721	0.193
politics	0.500	0.678	0.567	0.134	0.488	0.131
xwindows	0.701	0.677	0.628	0.137	0.618	0.141
electronics	0.476	0.641	0.527	0.108	0.349	0.107
religion	0.185	0.501	0.440	0.152	0.421	0.151
forsale	0.660	0.734	0.709	0.149	0.573	0.145
med	0.738	0.893	0.838	0.121	0.745	0.121
Average	0.650	0.731	0.661	0.150	0.590	0.153

Table 2. The F scores when there is no errors for svmLight with default parameters and naive bayes with additive smoothing of 0.1. Also performance of S-EM and for one class support vector machine with different number of positive examples.

and negative error frequencies (Criteria II) for different values of α. The results show that selecting the regularization parameter using $pr/\Pr[Y=1]$ gives only a slight degradation in performance compared to the best regularization parameter on the test set. In comparison, the sum of false positive and false negative frequencies does not perform very well for selecting the regularization parameter when performance is measured in terms of the F score. As α increases, degradation in performance is slight when most of the positive examples are labeled as positive. Even when 70% of the positive examples are unlabeled, the classifier still achieves reasonable performance.

5.2. Experiment 2

For comparison (see Table 2) , we tried other methods on the same data and feature sets. As we do not use validation sets for these methods, we combine the training and validation sets of the previous experiments to form new training sets for this set of experiments.

We ran the S-EM algorithm from (Liu et al., 2002) which tries to find a good initialization for the generative model of the negative examples and then uses EM with naive bayes in order to label the unlabeled examples. The implementation used is the same as that described in (Liu et al., 2002) except that an additive smoothing parameter of 0.1 (Agrawal et al., 2000) is used instead of 1 (Laplacian smoothing) in the naive bayes model as it performs better.

The other method used is the one-class support vector machines (Scholkopf et al., 1999) which does not use the unlabeled examples but instead tries to learn the support of the distribution of positive examples. One-class SVM was used in (Manevitz & Yousef, 2001) to solve a text classification problem when only positive examples are available. The one-class support vector machines software libSVM (Chang & Lin, 2001) is used in the experiment. We use the combined positive sets of our training and validation sets as the positive set. The default parameters of libSVM were used as there does not appear to be a simple method for tuning parameters using only positive examples. Only the performance for the linear kernel is shown. (Results for Gaussian kernels are poorer.)

To compare performance of the weighted logistic regression on noiseless data, we ran the SVM algorithm (svmLight (Joachims, 1998) with the default parameters) and naive bayes with an additive smoothing parameter of 0.1 on the noiseless dataset using the same feature set. The average F score for the SVM is 0.650 while the average F score for naive bayes is 0.731. The average F score for weighted logistic regression in this case is 0.763. As the parameters and features used by the SVM and naive bayes are not tuned with a validation set, this does not give a fair comparison but merely gives an indication that weighted

Positive Set	$\alpha = 0$ Best	$\alpha = 0$ Crit.I	$\alpha = 0$ NB	$\alpha = 0.3$ Best	$\alpha = 0.3$ Crit. I	$\alpha = 0.3$ S-EM
atheism	0.646	0.642	0.658	0.637	0.628	0.588
autos	0.773	0.773	0.757	0.763	0.761	0.596
space	0.796	0.796	0.878	0.786	0.783	0.697
graphics	0.575	0.575	0.607	0.567	0.567	0.504
motorcycles	0.849	0.849	0.812	0.826	0.826	0.747
christian	0.678	0.678	0.772	0.660	0.656	0.667
ms-windows	0.669	0.669	0.609	0.660	0.658	0.494
baseball	0.789	0.789	0.826	0.768	0.768	0.733
guns	0.685	0.685	0.782	0.677	0.667	0.640
pc	0.619	0.619	0.575	0.608	0.608	0.505
hockey	0.871	0.871	0.909	0.862	0.862	0.844
mideast	0.865	0.865	0.895	0.857	0.852	0.853
mac	0.721	0.721	0.633	0.713	0.713	0.503
crypt	0.858	0.858	0.906	0.854	0.854	0.756
politics	0.596	0.591	0.683	0.585	0.585	0.532
xwindows	0.757	0.757	0.716	0.729	0.726	0.633
electronics	0.604	0.604	0.619	0.574	0.574	0.401
religion	0.513	0.509	0.525	0.507	0.503	0.434
forsale	0.710	0.710	0.702	0.694	0.685	0.581
med	0.830	0.830	0.878	0.819	0.816	0.764
Average	0.720	0.720	0.737	0.707	0.705	0.624

Table 3. The F scores for $\alpha = 0$ and $\alpha = 0.3$ when the positive set remains constant but the number of positive in the unlabeled is varied.

logistic regression does not perform poorly relative to other methods in the noiseless case.

The S-EM algorithm performs reasonably but not as well as weighted logistic regression. This is probably because of a mismatch between the generative model which uses a single model for the negative examples with the data which is composed of many news groups for the negative examples. The one-class SVM performs poorly on this feature set. Experiments in (Manevitz & Yousef, 2001) indicate that one-class SVM is highly sensitive to the features used. Better performance was obtained with one-class SVM in (Manevitz & Yousef, 2001) for the Reuters data set. However, the number of features is between 10 to 20 in those experiments. We did not try feature selection for improving the performance of one-class support vector machines. In general, we think that utilizing the unlabeled examples as noisy information is better than throwing them away.

5.3. Experiment 3

In the experiments above, the number of positive examples decreases as the noise level increases. In practice, the number of positive examples is often constant as the learner already has the positive examples and may be using a different source of unlabeled examples which may contain a different fraction of positive examples in the unlabeled set. We perform another set of experiments where the number of positive examples are held constant but the number of positive in the unlabeled set is varied. We started with the number of positive examples used in the $\alpha = 0.7$ case in the first set of experiments. The number of positive examples in the unlabeled set is varied to create the situations where $\alpha = 0.3$ and $\alpha = 0$. The case $\alpha = 0.7$ in this experiment is identical to the $\alpha = 0.7$ case in the first set of experiments. The same test set is used as in the previous experiments, so the algorithm is further disadvantaged by having a different test distribution compared to the training distribution (because some of the positive examples have been removed from the unlabeled set).

The result in Table 3 shows that the regularization parameter selection criteria performs well even in this situation. Comparison with the previous experiment also shows that performance improves with a larger positive set. Interestingly, naive bayes performs well in the noiseless case when the number of positive examples is small (Table 3) but its performance does not improve when the number of positive examples is increased (Table 2).

5.4. Discussion

Overall, the method seems to be effective when linear function is a good classifier and the F score is an appropriate performance measure. By adding the regularization parameter which is selected using a validation set, it is also fairly tolerant to very high dimensional data such as text. It is able to perform better than methods based on the naive bayes assumption such as S-EM on datasets where the naive bayes assumption is not well satisfied. It uses more information than methods such as one-class SVM and hence may be preferable when the unlabeled data is available.

More experiments need to be done on other text data sets such as the Reuters data set and on non-text data such as direct marketing data to confirm the effectiveness of the method.

6. Conclusion

We studied the problem of using linear functions to learn from positive and unlabeled examples. We propose using logistic regression on weighted examples together with a performance measure that can be estimated from positive and unlabeled examples to select the regularization parameter from a validation set. Experiments show that the method performs well.

Acknowledgements

The authors would like to thank Xiaoli Li who performed the experiments using the S-EM algorithm. Bing Liu was supported in part by the National Science Foundation under the NSF grant IIS-0307239

References

Agrawal, R., Bayardo Jr., R., & Srikant, R. (2000). Athena: Mining-based interactive management of text databases. *International Conference on Extending Database Technology*.

Blum, A., Frieze, A. M., Kannan, R., & Vempala, S. (1996). A polynomial-time algorithm for learning noisy linear threshold functions. *IEEE Symposium on Foundations of Computer Science* (pp. 330–338).

Blum, A., & Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. *COLT: Proceedings of the Workshop on Computational Learning Theory*.

Bylander, T. (1994). Learning linear threshold functions in the presence of classification noise. *COLT: Proceedings of the Workshop on Computational Learning Theory*.

Chang, C.-C., & Lin, C.-J. (2001). *LIBSVM: a library for support vector machines*. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

Denis, F. (1998). PAC learning from positive statistical queries. *Proc. 9th International Conference on Algorithmic Learning Theory - ALT '98* (pp. 112–126).

Denis, F., Gilleron, R., & Tommasi, M. (2002). Text classification from positive and unlabeled examples. *The 9th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, IPMU*.

Hoffgen, K.-U., Simon, H.-U., & Horn, K. S. V. (1995). Robust trainability of single neurons. *Journal of Computer and System Sciences*, *50*, 114–125.

Joachims, T. (1998). Making large-scale svm learning practical. *LS8-Report, 24, Universitat Dortmund, LS VIII-Report*.

Joachims, T. (1999). Transductive inference for text classification using support vector machines. *Proceedings of ICML-99, 16th International Conference on Machine Learning* (pp. 200–209).

Kearns, M. (1998). Efficient noise-tolerant learning from statistical queries. *Journal of the ACM*, *45*, 983–1006.

Lang, K. (1995). Newsweeder: Learning to filter netnews. *International Conference on Machine Learning* (pp. 331–339).

Liu, B., Lee, W. S., Yu, P. S., & Li, X. (2002). Partially supervised classification of text documents. *International Conf. on Machine Learning* (pp. 387–394).

Manevitz, L. M., & Yousef, M. (2001). One class SVMs for document classification. *Journal of Machine Learning Research*, *2*, 139–154.

Mason, L., Baxter, J., Bartlett, P., & Frean, M. (1999). Boosting algorithms as gradient descent in function space. *Technical report, RSISE, Australian National University, 1999.*.

Mitchell, T. (1997). *Machine learning*. New York: McGraw-Hill.

Muggleton, S. (2001). Learning from positive data. *Machine learning, to appear*.

Nigam, K., McCallum, A., Thrun, S., & Mitchell, T. (1998). Learning to classify text from labeled and unlabeled documents. *AAAI-98* (pp. 792–799). Madison, US: AAAI Press, Menlo Park, US.

Scholkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A. J., & Williamson, R. C. (1999). Estimating the support of high dimensional distribution. *Technical Report. Microsoft Research MSR-TR-99-87*.

Yu, H., Han, J., & Chang, K. C.-C. (2002). PEBL: Positive example based learning for web page classification using SVM. *Proc. 2002 Int. Conf. on Knowledge Discovery in Databases (KDD'02)*.