Partially Supervised Text Classification: Combining Labeled and Unlabeled Documents Using an EM-like Scheme

Carsten Lanquillon

DaimlerChrysler Research and Technology
D-89013 Ulm, Germany
Phone: +49 731 505 2809, Fax: +49 731 505 4210
carsten.lanquillon@daimlerchrysler.com

Abstract. Supervised learning algorithms usually require large amounts of training data to learn reasonably accurate classifiers. Yet, in many text classification tasks, labeled training documents are expensive to obtain, while unlabeled documents are readily available in large quantities. This paper describes a general framework for extending any text learning algorithm to utilize unlabeled documents in addition to labeled document using an Expectation-Maximization-like scheme. Our instantiation of this partially supervised classification framework with a similarity-based single prototype classifier achieves encouraging results on two real-world text datasets. Classification accuracy is reduced by up to 38% when using unlabeled documents in addition to labeled documents.

1 Introduction

With the enormous growth of on-line information available through the World Wide Web, electronic news feeds, digital libraries, corporate intranets, and other sources, the problem of automatically classifying text documents into predefined categories is of great practical importance in many information organization and management tasks.

This classification problem can be solved by applying supervised learning algorithms which learn reasonably accurate classifiers when provided with enough labeled training examples [4,14]. For complex learning tasks, however, providing sufficiently large sets of labeled training examples becomes prohibitive because hand-labeling examples is expensive. Therefore, an important issue is to reduce the need for labeled training documents. As shown in [9], a promising approach in text domains is to use *unlabeled* documents in addition to *labeled* documents during the learning process. While labeled documents are expensive to obtain, unlabeled documents are often readily available in large quantities.

Why does using unlabeled data help? As pointed out by [9] and [6], it is well known in information retrieval that words in natural language occur in strong co-occurrence patterns [13]. While some words are likely to co-occur in

R. López de Mántaras, E. Plaza (Eds.): ECML 2000, LNAI 1810, pp. 229–237, 2000.

one document, others are not. When using unlabeled documents we can exploit information about word co-occurrences that is not accessible from the labeled documents alone. This information can increase classification accuracy.

Nigam *et al.* [9] use a multinomial Naïve Bayes classifier in combination with the Expectation Maximization (EM) algorithm [3] to make use of unlabeled documents in a probabilistic framework. They show that augmenting the available labeled documents with unlabeled documents can significantly increase classification accuracy. In this paper we drop the probabilistic framework and extend the EM-like scheme to be used with any text classifier.

The remainder of the paper is organized as follows. Section 2 gives a brief introduction to text classification and two traditional learning algorithms which are used later on. In Section 3, our algorithm for combining labeled and unlabeled documents in an EM-like fashion is described. Some experimental results are presented in Section 4. Section 5 lists some related work, and Section 6 concludes this paper.

2 Text Classification

The task of text classification is to automatically classify documents into a predefined number of classes. Each document can be in multiple, exactly one, or no class. In the experiments presented in Section 4, the task is to assign each document to exactly one class. Using supervised learning algorithms in this particular setting, a classifier can try to represent each class simultaneously. Alternatively, each class can be treated as a separate binary classification problem where each binary problem answers the question of whether or not a document should be assigned to the corresponding class [6].

2.1 Document Representation

In information retrieval, documents are often represented as feature vectors, and a subset of all distinct words or word stems occurring in the given documents are used as features. Words that frequently occur in many documents (*stop words* like "and", "or" etc.) or words that occur only in very few documents may be removed. Further, measures such as the *average mutual information* with the class labels can be used for feature selection [15]. Each feature is given a weight which depends on the learning algorithm at hand. This leads to an attribute-value representation of text. Possible weights are, e.g., binary indicators for the presence or absence of features, plain feature counts—*term frequency (tf)*—or more sophisticated weighting schemes, such as multiplying each term frequency with the *inverted document frequency (idf)* [12]. Finally, each feature vector may be normalized to unit length to abstract from different document lengths.

2.2 Learning Algorithms

A variety of text learning algorithms have been studied and compared in the literature, e.g. see [4] and [14].

Naïve Bayes Classifier For comparison we apply the multinomial Naïve Bayes classifier which uses the term frequency as feature weights as described in [9]. The idea of the Naïve Bayes classifier is to use the joint probabilities of words (features) and classes to estimate the probabilities of the classes given a document. A document is then assigned to the most probable class.

Single Prototype Classifier Further, we use a similarity-based method based on *tfidf* weights which we denote as *single prototype classifier (SPC)*. It is a variant of Rocchio's method for relevance feedback [10] applied to text classification and is also described as the *Find Similar* algorithm in [4]. The classifier models each class with exactly one prototype computed as the average (centroid) of all available training documents. We use a scheme for setting feature weights which is denoted as *ltc* in SMART [11] notation. A document is assigned to the class of the prototype to which it has the largest cosine similarity.

3 Partially Supervised Learning

This section describes a family of partially supervised learning algorithms for combining labeled and unlabeled documents, extending the work of [9].

3.1 General Framework

A general approach for utilizing information given by unlabeled data is to apply some form of clustering. Treating the class labels of the unlabeled documents as missing values, an EM-like scheme can be applied as described below. Table 1 gives an outline of this framework.

Given a set of training documents D, for some subset of the documents $d_i \in D^l$ we know the class label y_i, and for the rest of the documents $d_i \in D^u$, the class labels are unknown. Thus we have a disjoint partitioning of our training documents into a labeled set and an unlabeled set of documents $D = D^l \cup D^u$. The task is to build a classifier based on the training documents, D, for predicting the class label of unseen unlabeled documents.

First, an initial classifier, H, is build based only on the labeled documents, D^l. Then the algorithm iterates the following three steps until the class memberships given to the unlabeled documents, D^u, by the current classifier, H, do not change from one iteration to the next. Corresponding to the **E-step**, the current classifier, H, is used to obtain classification scores for each unlabeled document. The classifier may respond with any type of classification scores, they need not be probabilistic. In order to abstract from the classifier's response, in the next step we transform these scores into class memberships, yielding a class membership matrix, $U^u \in [0,1]^{c \times |D^u|}$, where c is the number of classes. The sum of class memberships of a document over all classes is assumed to be one. Possible transformations are, for instance, normalizing the scores or using hard memberships, e.g. setting the largest score to one and all other scores to zero. The transformation function should depend on the classifier at hand such that it knows how to make use of the class membership matrix, U^u. Using hard memberships always

Table 1. EM-like algorithmic framework for partially supervised learning

- **Inputs:** Sets D^l and D^u of labeled and unlabeled documents.
- Build initial classifier, H, based only on the labeled documents, D^l.
- Loop while classifying the unlabeled documents, D^u, with the current classifier, H, changes as measured by the class memberships of the unlabeled documents, U^u:
  - **(E-step)** Use the current classifier, H, to evaluate classification scores for each unlabeled document.
  - Transform classification scores into class memberships of the unlabeled documents, U^u.
  - **(M-step)** Re-build the classifier, H, based on labeled documents, D^l, and unlabeled documents, D^u, with labels obtained from U^u.
- **Output:** Classifier, H, for predicting class labels of unseen unlabeled documents.

allows us to use any traditional classifier. Now, provided with the class membership matrix, U^u, a new classifier, H, can be build from both, the labeled and unlabeled documents. This corresponds to the **M-step**. The final classifier, H, can then be used to predict the class labels of unseen test examples.

3.2 Instantiations

In order to apply this algorithmic framework, the underlying classification algorithm and the function for transforming classification scores have to be specified.

Naïve Bayes Classifier When using a Naïve Bayes classifier and leaving the resulting probabilistic classification scores unchanged, we end up with the algorithm given in [9]. This instantiation has a strong probabilistic framework and is guaranteed to converge to a local minimum as stated by [9].

Single Prototype Classifier Next, we will use the single prototype classifier in combination with a transformation of classification scores into hard class memberships. Hence, this instantiation of our partially supervised algorithmic framework turns out to be a variation of the well known hard *k-means* clustering algorithm [7]. The difference is that the memberships of the labeled documents remain fixed during the clustering iterations. The traditional k-means algorithm is guaranteed to converge to a local minimum after a finite number of iterations. What about our partially supervised variant?

The proof of convergence for the traditional k-means algorithm is based on the fact that there is only a finite number of hard partitionings of training documents into classes and that the sum of squared distances between prototypes and training documents, J, does not increase while iteratively updating the class memberships and the prototypes. Therefore, the algorithm must converge in a finite number of steps.

The calculation of cluster prototypes based on training documents and their hard class labels is the same in our partially supervised algorithm. Hence, this

step does not increase J. As mentioned above, the update rule for the class membership matrix in our algorithm differs from the traditional k-means algorithm. The class labels of the labeled documents remain fixed while the unlabeled documents are assigned to the closest prototype. The latter is equivalent to the traditional k-means algorithm and thus does not lead to an increase in J either. Further, note that fixed class memberships cannot cause J to change. Thus, our partially supervised algorithm will also converge to a local minimum after a finite number of steps.

4 Experimental Results

This section gives empirical evidence that combining labeled and unlabeled documents with certain text classifiers using the algorithmic framework in Table 1 can improve traditional text classifiers. Experimental results are reported on two different text corpora which are available at *http://www.cs.cmu.edu/∼textlearning*. We use a modified version of the *Rainbow* system [8] to run our experiments. Following the setups in [9], we run the experiments with the partially supervised single prototype classifier as described Section 3. The results are compared to the partially supervised Naïve Bayes approach as given in [9].

4.1 Datasets and Protocol

The 20 Newsgroups dataset consists of 20017 articles divided almost evenly among 20 different UseNet discussion groups. The task is to classify an article into the one of the twenty newsgroups to which it was posted. When tokenizing the documents, UseNet headers are skipped, and tokens are formed from contiguous alphabetic characters. We do not apply stemming, but remove common stop words. While all features are used in the experiments with the Naïve Bayes classifier, for the single prototype classifier, we limit the vocabulary to the 10000 most informative words, as measured by average mutual information with the class labels. We create a test set of 4000 documents and an unlabeled set of 10000 documents. Labeled training sets are formed by partitioning the remaining 6000 documents into non-overlapping sets. All sets are created with equal number of documents per class. Where applicable, up to ten trials with disjunct labeled training sets are run for each experiment. Results are reported as averages over these trials.

The WebKB dataset contains 8145 web pages gathered from four university computer science departments. Only the 4199 documents of the classes *course, faculty, project,* and *student* are used. The task is to classify a web page into the appropriate one of the four classes. We do not apply stemming and stopword removal. The vocabulary is limited to the top 300 words according to average mutual information with the class labels in all experiments. To test in *leave-one-university-out* fashion, we create four test sets, each containing all the pages form one of the four complete computer science departments. For each test set, an unlabeled set of 2500 pages is created by randomly selecting from

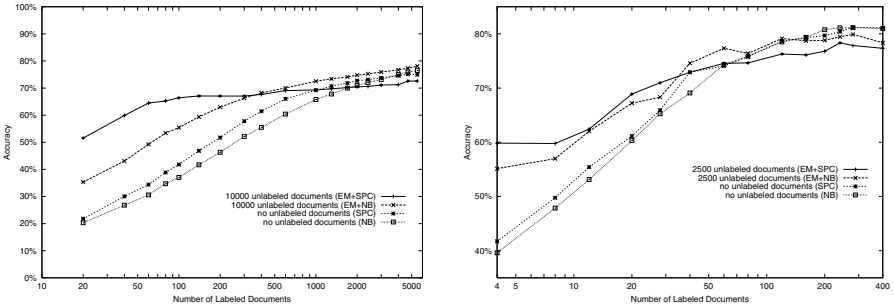

Fig. 1. Classification accuracy of the partially supervised learning framework (EM) using the Naïve Bayes classifier (NB) and the single prototype classifier (SPC) compared to the traditional classifiers on the 20 Newsgroups dataset (left) and on the WebKB dataset (right). Note the magnified vertical scale on the right

the remaining pages. Different non-overlapping labeled training sets are created from the remaining web pages. Results are reported as averages over the four different test sets.

4.2 Results

Figure 1 shows the effect of using the partially supervised learning framework with the Naïve Bayes classifier (NB) and the single prototype classifier (SPC) on the 20 Newsgroups dataset and the WebKB dataset. The horizontal axis indicates the amount of labeled training data on a log scale. Note that, for instance, 20 training documents for the 20 Newsgroups and four documents for the WebKB dataset correspond to one training document per class. The vertical axis indicates the average classification accuracy on the test sets. We vary the number of labeled training documents for both datasets and compare the results to the traditional classifiers which do not use any unlabeled documents.

In all experiments, the partially supervised algorithms perform substantially better when the amount of labeled training documents is small. For instance, with only 20 training examples for the 20 Newsgroups dataset, the partially supervised SPC reaches about 52% accuracy while the traditional SPC achieves 22%. Thus, the classification error is reduced by about 38%. For the NB, accuracy increases from 20% to about 35% when using unlabeled documents with 20 labeled training examples. For the WebKB dataset, the performance increase is much smaller, especially for the SPC. However, note that there are four times less unlabeled documents for the experiments on this dataset. As can be expected, the more labeled documents are available, the smaller the performance increase. Note that especially for the SPC, accuracy even degrades when using unlabeled documents with a lot of labeled documents. We hypothesize that when the number of labeled documents is small, the learning algorithm is desperately in need for help and makes even good use of uncertain information as provided

by unlabeled documents. However, when the accuracy is already high without any unlabeled documents, i.e. when there are enough labeled documents, adding uncertain information by means of unlabeled documents does not help but rather hurts classification accuracy.

5 Related Work

The family of Expectation-Maximization (EM) algorithms and its application to classification is broadly studied in the statistics literature. R.J.A. Little [3] mentions the idea of using an EM-like approach to improve a classifier by treating the class labels of unlabeled documents as missing values. Emde describes a conceptual clustering algorithm that tries to take advantage of the information inherent to the unlabeled data in a setting where the number of labeled data is small [5]. Blum and Mitchell [2] use co-training to make use of labeled and unlabeled data in the case that each example has at least two redundantly sufficient representations. Bensaid and Bezdek try to use information inherent to the labeled data to help clustering the unlabeled data [1]. In current work by Bensaid and the author, this approach is applied to text classification. As mentioned in Sections 1 and 3, this paper describes a generalization of the work done by Nigam *et al.* [9]. They use a multinomial Naïve Bayes classifier in combination with the EM-algorithm to make use of unlabeled documents. Joachims explores transductive support vector machines for text classification [6]. This approach uses the unlabeled test documents in addition to the labeled training documents to better adjust the parameters of the support vector machine. Although designed for classifying the documents of just this test set, the resulting support vector machine could as well be applied to classify new, unseen documents as done in this paper. However, as yet there is no empirical evidence of how well this works.

6 Conclusions and Future Work

This paper presents a general framework for partially supervised learning from labeled and unlabeled documents using an EM-like scheme in combination with an arbitrary text learning algorithm. This is an important issue when hand-labeling documents is expensive but unlabeled documents are readily available in large quantities.

Empirical results with two real-world text classification tasks and a similarity-based single prototype classifier show that this EM-scheme can successfully be applied to non-probabilistic classifiers. The applied instantiation of our framework is a variant of the traditional hard k-means clustering algorithm where the class memberships of some training documents, namely the labeled documents, are fixed. The single prototype classifier seems to be well suited for classification tasks where the number of labeled documents is very scarce. For larger numbers of labeled documents, the Naïve Bayes classifier is superior.

Adding unlabeled documents to a larger number of labeled training documents may even hurt classification accuracy when using the single prototype classifier. Future work will focus on preventing the unlabeled documents from degrading performance. An interesting approach is to introduce a weight to adjust the contribution of unlabeled documents as discussed in [9].

So far we applied only very simple learning algorithms because the successful application of more sophisticated methods seems doubtful when only very few labeled training documents are present. Nevertheless, other learning algorithms are being tested in current research. Our conjecture is that this framework works well for learning algorithms that aggregate document information for each class into a single representative like the two methods applied in this paper. By contrast, approaches like the nearest neighbor rule are likely to fail since they do not generalize and thus cannot exploit information inherent to unlabeled documents.

Acknowledgements

Thanks to Tom Mitchell, the CMU text learning group, and Amine Bensaid for help and discussions. Special thanks to Kamal Nigam for providing his results and scripts to setup the experiments.

References

1. A. Bensaid and J. Bezdek. Semi-supervised point-prototype clustering. *Int. Journal of Pattern Recognition and Artifical Intelligence*, 12(5):625–643, 1998. 235
2. A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Annual Conference on Computational Learning Theory (COLT-98)*, 1998. 235
3. A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. of the Royal Stat. Soc., Series B*, 39:1–38, 1977. 230, 235
4. S. Dumais, J. Platt, D. Heckerman, and M. Sahami. Inductive learning algorithms and representation for text categorization. In *Proceedings of the Seventh International Conference on Information and Knowledge Management*, 1998. 229, 230, 231
5. W. Emde. Inductive learning of characteristic concept descriptions from small sets of classified examples. In *Proc. 7th European Conf. on Machine Learning*, 1994. 235
6. T. Joachims. Transductive inference for text classification using support vector machines. In *Proc. 16th Int. Conf. on Machine Learning*, 1999. 229, 230, 235
7. J. MacQueen. Some methods for classification and analysis of multivariate observations. In L.M. Le Cam and J. Neyman, editors, *Proc. 5th Berkeley Symp. Math. Stat. Prob.*, volume I, pages 281–297. Univ. of California Press, 1967. 232
8. A.K. McCallum. Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering. `http://www.cs.cmu.edu/~mccallum/bow`, 1996. 233
9. K. Nigam, A. McCallum, S. Thrun, and T. Mitchell. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 1999. To appear. 229, 230, 231, 232, 233, 235, 236

10. J.J. Jr. Rocchio. Relevance feedback in information retrieval. In G. Salton, editor, *The SMART Retrieval System: Experiments in Automatic Document Processing*, pages 313–323. Prentice Hall, 1971. 231
11. G. Salton. *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*. Addison-Wesley, 1989. 231
12. G. Salton and C. Buckley. Term weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523, 1988. 230
13. C. van Rijsbergen. A theoretical basis for the use of co-occurrence data in information retrieval. *Journal of Documentation*, 33(2):106–119, 1977. 229
14. Y. Yang and X. Liu. A re-examination of text categorization methods. In *Proceeding of ACM SIGIR Conference*, 1999. 229, 230
15. Y. Yang and J.O. Pedersen. A comparative study on feature selection in text categorization. In *Proc. 14th Int. Conf. on Machine Learning*, 1997. 230