Understanding the Behavior of Co-training

Kamal Nigam KNIGAM@CS.CMU.EDU
Rayid Ghani RAYID@CS.CMU.EDU
School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213

ABSTRACT
Recently there has been significant interest in supervised learning algorithms that combine labeled and unlabeled data for text learning tasks. The co-training setting (Blum & Mitchell, 1998) applies to datasets that have a natural separation of their features into two disjoint sets. We demonstrate that when learning from labeled and unlabeled data, algorithms explicitly leveraging a natural independent split of the features outperform algorithms that do not. When a natural split does not exist, co-training algorithms that manufacture a feature split may out-perform algorithms not using a split. These results help explain why co-training algorithms are both discriminative in nature and robust to the assumptions of their embedded classifiers.

1. INTRODUCTION
There has been much recent interest in supervised learning algorithms that combine information from labeled and unlabeled data. Datasets whose features naturally partition into two sets, and algorithms that use this division, fall into the *co-training* setting (Blum & Mitchell, 1998). For such datasets, cotraining algorithms explicitly leverage this split when learning from labeled and unlabeled data. We explore questions of why co-training algorithms are successful: do they actually leverage independent divisions of features, or do these algorithms use unlabeled data only as well as those that ignore the feature division? How sensitive are co-training algorithms to the correctness of their assumptions? Can co-training algorithms be applied to datasets without natural feature divisions?

In this paper we use two algorithms and their variants to explore these questions: the co-training algorithm (Blum & Mitchell, 1998) that uses the feature split, and an algorithm using Expectation-Maximization (Nigam, McCallum, Thrun, & Mitchell, 2000) that ignores it. We instantiate both algorithms with the underlying representation of a naive Bayes text classifier (Lewis, 1998) to facilitate more direct comparisons.

2. INDEPENDENT FEATURE SPLITS
We create a semi-artificial dataset that has the independence properties desired by co-training. We select four newsgroups from the 20 Newsgroups dataset (see e.g. (Nigam et al., 2000)) and create a two-class problem with class-conditional independence by joining together randomly selected documents from each of the first two newsgroups to make positive examples, and joining together randomly selected documents from each of the second two newsgroups to make negative ones. This joining is done such that the words in the first and third newsgroups come from the same vocabulary, while words from the second and fourth newsgroups come from a separate vocabulary. This dataset has two nice properties: (1) true class-conditional independence between the words in the two feature sets and (2) redundancy between the feature sets (each feature set is sufficient for accurate classification). This represents a best-case scenario for the application of co-training. We run ten random test/train splits for co-training, EM and naive Bayes using 6 labeled, 1000 unlabeled, and 976 test documents.

Table 1: Error rates using unlabeled data on a dataset with an independent feature split.

Method	Labeling	Feature Split	Error
co-training	incremental	uses	3.7%
co-EM	iterative	uses	3.3%
EM	iterative	ignores	8.9%
self-training	incremental	ignores	5.8%

The results show that co-training performs best on this dataset. Starting from a baseline naive Bayes error of 34.0%, using just the labeled data, co-training reduces error down to 3.7%, matching the accuracy of naive Bayes if all the labels of the unlabeled documents were known. EM also reduces the error, but only to 8.9%. These results suggest that the co-training algorithm performs better than EM when there is indeed an independent division of the feature space.

Algorithmically, co-training differs from EM in more ways than just its use of the feature split. The other significant difference between them is that co-training labels just a few documents per round; that is, co-training *incrementally* uses the unlabeled data. In contrast, EM probabilistically labels all the data at each round; EM *iteratively* uses the unlabeled data. This difference could account for the benefit of the co-training algorithm instead of its use of the feature split.

In order to tease apart the effects of the feature splitting from the effects of the labeling process, we specify two hybrids of EM and co-training to fill the space of algorithms along these dimensions. *co-EM*, is an iterative algorithm that uses the feature split. It first initializes the A-feature-set naive Bayes classifier from the labeled data only. Then, A probabilistically labels all the unlabeled data. The B-feature-set classifier trains using the labeled data and the unlabeled data with A's labels, relabels the unlabeled data, and then iterates this process. The second hybrid is *self-training*, an incremental algorithm that does not explicitly use the feature split. Initially, self-training builds a single classifier with the labeled data pooling all the features. Then it labels the unlabeled data and converts the most confidently predicted document of each class into a labeled training example, and iterates until all the unlabeled documents are labeled.

Table 1 shows classification error rates for the four algorithms above. Both algorithms that explicitly use the feature split have lower error rates than either algorithm that does not. Co-training and co-EM have error rates of 3.7% and 3.3%, where self-training and EM have error rates of 5.8% and 8.9%

Table 2: Error rates using labeled and unlabeled data on the News 2x2 dataset with random splits.

Method	Labeling	Feature Split	Error
co-training	incremental	uses	5.5%
co-EM	iterative	uses	5.1%
EM	iterative	ignores	8.9%
self-training	incremental	ignores	5.8%

respectively. These results suggest that algorithms explicitly using an existing independent and redundant division of the features can be expected to perform better than algorithms that do not.

3. CREATING FEATURE SPLITS

Although co-training is a powerful paradigm, it is not widely applicable. The large majority of datasets have a single set of features with no obvious or natural way to divide them. Yet, if there were sufficient redundancy among the features, and we could identify a fairly reasonable division of them, then co-training algorithms may show similar advantages to those seen in the previous section. To test this idea, we begin by randomly dividing a feature set into two. Although this will not be an independent split, we can apply co-training to it.

The first dataset we use is the News 2x2 dataset, but without the knowledge of the natural split. Since we know there is a natural split, we can evaluate how much we lose by choosing a random split instead. The second dataset we use is the News5 dataset, used previously by McCallum and Nigam (1998).

Table 2 shows classification error rates for the News 2x2 dataset. The results for EM and self-training are identical to those in Section 2. Notice that again, the two co-training algorithms perform better than the non-co-training ones, but by a smaller margin than if the correct feature division were known. These results show promise for applying co-training algorithms to flat data. However, recall that this dataset is constructed to be redundant in that it contains twice as much data as is needed for classification. The results indicate that, with enough redundancy in the text, there may be enough natural independence of words to allow co-training to flourish.

For News5 dataset, Naive Bayes with access to all the labels gets 16.7% accuracy, while with just the initialy labeled data it gets 50.1% error. The results for this dataset do not set such a clear trend. The worst performer is EM (31.2% error)—the algorithm with a strong probabilistic foundation. The best performer is self-training(27% error), an algorithm that does not use feature splitting. However, the co-training algorithm(28% error) does reduce error over EM by 10%. This seems like encouraging evidence to support our hypothesis that splitting regular datasets into disjoint feature sets and running co-training-like algorithms on it can result in a decrease in classification error compared to regular algorithms like EM.

Note that we split our feature sets in a random fashion; the next step is to develop a splitting algorithm that results in feature sets that are maximally independent. Ideally, we'd like the sets of features to have low *conditional mutual information*. However, it is not tractable to compute this for text data. Instead, we approximate this as the sum of the pairwise mutual independencies for all pairs of words in different sets. If we create a graph, where words are nodes, and the weights on the edges are the pairwise conditional mutual information, then a 2-way balanced min-cut of the graph may partition the features into sets that will be helpful to co-training. This cut will create two sets of features that have low pairwise conditional mututal information. However, finding the min-cut is NP-hard so we need to use efficient approximation algorithms. Fortunately, much is known about efficient approximate min-cut graph partitioning algorithms (Fjallstrom, 1998). Experiments using this approach, and ideas in similar directions, are an area of ongoing research.

4. DISCUSSION AND FUTURE WORK

Given the results from the previous sections, what can we conclude about the behavior of co-training? Certainly, results on the News 2x2 dataset show that co-training performs better than EM when the feature set independence assumption is valid. We hypothesize that the co-training algorithm succeeds in part because it is more robust to the assumptions made by its underlying classifiers.

EM uses the naive Bayes classifier to assign class probabilities to each unlabeled document. However, these probabilities are poorly estimated because the word independence assumption is violated by text data. Co-training, on the other hand, makes limited use of the assumptions of the underlying classifier. It uses the classifier for ranking the documents, and does not make use of the actual posterior scores. This suggests that EM should be more sensitive to the correctness of these assumptions.

The robustness of co-training to the underlying classifier assumptions can also be understood in another way. EM is a likelihood-based approach, and nothing about the technique is geared specifically towards classification. Thus, as EM fits the generative model to the unlabeled data, if the natural clustering of the unlabeled data does not correspond to class-based clusters, EM will suffer. Co-training, on the other hand, is a more discriminative approach, in that it tries to add documents to its labeled set that will help with classification. Most incremental co-training algorithms approximate this by adding documents about which it is most confident. In future work we plan to make this criteria more explicitly discriminative.

References

Blum, A., & Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. In *COLT '98*.

Fjallstrom, P.-O. (1998). Algorithms for graph partitioning: A survey. *Linkoping Electronic Atricles in Computer and Information Science, 3*.

Lewis, D. D. (1998). Naive (Bayes) at forty: The independence assumption in information retrieval. In *ECML-98*.

McCallum, A., & Nigam, K. (1998). Employing EM in pool-based active learning for text classification. In *Proceedings of ICML '98*.

Nigam, K., McCallum, A., Thrun, S., & Mitchell, T. (2000). Text classification from labeled and unlabeled documents using EM. *Machine Learning, 39*.