Iterative Labeling for Semi-Supervised Learning

Steve Hanneke   HANNEKE@UIUC.EDU
Department of Computer Science
University of Illinois
Urbana, IL 61801, USA

Dan Roth   DANR@CS.UIUC.EDU
Department of Computer Science
University of Illinois
Urbana, IL 61801, USA

Abstract

We propose a unified perspective of a large family of semi-supervised learning algorithms, which select and label unlabeled data in an iterative process. We discuss existing approaches that label examples based on the confidence of the current hypothesis, and propose an alternative approach that labels examples based on empirical risk. This new approach is shown to be statistically reasonable, allows for worst-case performance guarantees and, as we show, significantly outperforms confidence-based approaches in experiments.

1. Introduction

In recent years, there has been heightened interest in learning algorithms that exploit both labeled and unlabeled data. This setting is referred to as semi-supervised learning. In many situations, obtaining unlabeled examples for learning is fast and easy, while choosing accurate labels for them may be difficult, expensive, or time-consuming. For example, when training a vision system to recognize birds, one can obtain millions of unlabeled training examples by setting up a video camera in a park and recording all images for a week. However, labeling those same examples could require more time than one is willing to commit. A more pointed example occurs in the domain of medical diagnosis, where in some cases the only way to obtain labeled examples of a certain kind is by the death of a patient. In the Natural Language Processing domain, one can obtain almost limitless examples of text from digital libraries and the world wide web, but almost none of it is annotated for supervised learning. One would like to be able to supply a relatively small amount of labeled data, supplemented by a vast amount of unlabeled data, and learn a reliable hypothesis based on these. Because of this, semi-supervised learning algorithms are generally designed to operate in scenarios where the amount of unlabeled data is vast relative to labeled.

Formally, let $\mathcal{D}$ be an (unknown) distribution over a sample space $\mathcal{X} \times \mathcal{Y}$. A *labeled* example is any $(x, y) \in \mathcal{X} \times \mathcal{Y}$, and an *unlabeled* example is any $x \in \mathcal{X}$. Define a *learning algorithm* $\mathcal{A}$ as a function mapping a set of examples S to a trained classifier $h : \mathcal{X} \to \mathcal{Y}$, such that for $x \in \mathcal{X}$, $h(x)$ denotes the label that h assigns to x. A *supervised* learning algorithm has the additional constraint that all examples in S be labeled, whereas a *semi-supervised* learning algorithm can take mixed labeled and unlabeled examples. For a learning algorithm $\mathcal{A}$, and a set of examples

S, let $\mathcal{A}(S)$ denote the classifier produced by training $\mathcal{A}$ with S. For a classifier $f:\mathcal{X} \rightarrow \mathcal{Y}$, define $error(f) = Pr\{f(X) \neq Y\}$, where (X,Y) is distributed according to $\mathcal{D}$. Given a set $\mathcal{U}$ of iid unlabeled examples and a set $\mathcal{L}$ of iid labeled examples, the goal of semi-supervised learning is to minimize the quantity $error(\mathcal{A}(\mathcal{L} \cup \mathcal{U}))$.

The task of semi-supervised learning includes problems and approaches markedly different from those found in other subfields of machine learning. Questions such as how to label examples, how much to trust these new labels, and how to prevent overfitting with a small labeled set arise naturally in this setting. Perhaps the most interesting but disturbing problem observed in semi-supervised learning algorithms is a tendency for the accuracy to *degrade* when the amount of unlabeled data is *increased*. In fact, many algorithms have been shown to perform worse than strictly supervised learning with the labeled data *alone* in certain situations. Although this phenomenon is not yet fully understood, it is important to keep in mind when deciding how to approach the semi-supervised learning task.

2. Approaches to the Task

To address the question of how one should approach semi-supervised learning, it is important to examine previous approaches for strengths and weaknesses. Semi-supervised learning includes algorithms ranging from augmenting traditional supervised learning, to those using labeled data to improve clustering algorithms (Cohn et al.), to those using unlabeled data for model selection (Cohen, 2003; Schuurmans, 1997). We shall attempt to discuss only what seems to be the two main types of semi-supervised learning algorithms.

The first type evolved largely as extensions or modifications of the EM algorithm (Dempster et al., 1977). These algorithms proceed iteratively, each time labeling or relabeling some subset of the unlabeled data. We shall refer to these as *Iterative Labeling* algorithms. The second type seeks extreme points in some global criteria, based on the relation of the hypothesis to the data. We shall refer to these as *Global Optimization* algorithms. These two appraoches represent different ends of a tradeoff between accuracy and efficiency. In this work, we shall investigate the former type in detail.

2.1 Iterative Labeling

One approach to semi-supervised learning that has achieved some success involves wrapping a meta-algorithm around a supervised learning algorithm. The combined algorithm proceeds iteratively, each time assigning new labels to a subset of the data. Formally, define the Iterative Labeling family of meta-algorithms[1] such that a particular member is specified by fixing three function parameters: *Choose-Relabel-Set*, *Assign-Labels*, and *Stopping-Condition*. These may be viewed as subroutines used in the operation of the algorithm. Each is described in below.

Choose-Relabel-Set(S, $\mathcal{L}$, $\mathcal{A}$) is a function taking as input a set of examples S, a set of labeled examples $\mathcal{L}$, and a supervised learning algorithm $\mathcal{A}$, and outputting a subset of S. Its role is to select a subset of the data to be assigned labels on each iteration.
Assign-Labels($\mathcal{R}$, $\mathcal{L}$, $\mathcal{A}$) is a function mapping a set of examples $\mathcal{R}$, a set of labeled examples $\mathcal{L}$, and a supervised learning algorithm $\mathcal{A}$, to a set of examples. The role of this parameter is to label

---

1. We say it is a family of *meta*-algorithms because until a supervised learner is specified, the exact sequence of steps is not strictly defined.

the examples selected by *Choose-Relabel-Set*. It returns the set $\mathcal{R}$, but with new labels on the examples.

Stopping-Condition(S, S') is a function mapping two sets of examples into the set {*True*, *False*}. This parameter indicates when the algorithm should halt.

In addition, we introduce a function *Replace*(S, $\mathcal{Q}$), which maps the two sets S and $\mathcal{Q}$ to a set containing all elements of S, except that whenever an example in S differs only by its label from an example in $\mathcal{Q}$, the label from $\mathcal{Q}$ is used instead[2]. We also introduce a function *Get-Labeled*(S), which returns all labeled examples in S. Finally, in order to use the meta-algorithm, one must also specify a supervised learner. Note that the choice of these parameters may impose constraints on the type of learner allowed. For example, if the parameters make use of soft labels, the learner must be able to accommodate this. For an Iterative Labeling meta-algorithm IL, and a supervised learning algorithm $\mathcal{A}$, let us denote by IL$_\mathcal{A}$ the semi-supervised learning algorithm produced by using $\mathcal{A}$ with IL. Note that although many current Iterative Labeling algorithms explicitly use the trained classifier to assign new labels, it is not required in this framework. The execution proceeds as in Figure 1.

Given: supervised learning algorithm $\mathcal{A}$, dataset S
$i \leftarrow 0$
do
  $i \leftarrow i + 1$
  S' $\leftarrow$ S
  $\mathcal{L}_i \leftarrow$ *Get-Labeled*(S)
  $\mathcal{R} \leftarrow$ *Choose-Relabel-Set*(S, $\mathcal{L}_i$, $\mathcal{A}$)
  $\mathcal{Q} \leftarrow$ *Assign-Labels*($\mathcal{R}$, $\mathcal{L}_i$, $\mathcal{A}$)
  S $\leftarrow$ *Replace*(S, $\mathcal{Q}$)
while *Stopping-Condition*(S, S') evaluates to False
return *classifier f* = $\mathcal{A}(\mathcal{L}_i)$

Figure 1: Iterative Labeling meta-algorithm

Many of the current approaches to semi-supervised learning in use today are from this family. Prominent examples include EM (Dempster et al., 1977; Nigam et al., 2000), Yarowsky's algorithm (Yarowsky, 1995), and co-training. Figure 2 gives the parameter definitions for a representative set of algorithms. In particular, Truncated EM, which is sometimes referred to as self-training (Mihalcea, 2004), simply allows the classifier to label all examples for which it has confidence above a fixed threshold. Truncated EM has been used extensively in natural language; in particular, the most basic form of Yarowsky's algorithm presented in (Yarowsky, 1995) is exactly the semi-supervised learning algorithm formed by using Truncated EM with a decision list learner. In contrast to Truncated EM, the Auction algorithm labels a fixed number of examples per iteration, and examples are seen as bidding for the labels, which are awarded to high-confidence examples. Auction is based on a common pattern observed in several prominent algorithms, and in particular can be viewed

---

2. We assume that all examples are initially unique, or contain some kind of unique identifier so that even without labels, examples are distinguishable.

3

as a one-sided version of the co-training algorithm presented in (Blum and Mitchell, 1998). The Co-training algorithm given here is taken almost exactly from (Blum and Mitchell, 1998), with the simplifying exception that here *Choose-Relabel-Set* is allowed to select from the entire unlabeled set instead of being restricted to a smaller subset. One extension of the co-training approach is the Generalized Co-training algorithm (Goldman and Zhou, 2000). It replaces the independent views of traditional co-training with the use of two different supervised learning algorithms, both operating on the same view. Other Iterative Labeling algorithms include ASSEMBLE (Bennett et al., 2002), which uses a boosting approach, Label Propagation (Zhu and Ghahramani, 2002), and many variants of co-training, such as Co-EM and Co-Boost (Collins and Singer, 1999).

EM
Choose-Relabel-Set(S, $\mathcal{L}$, $\mathcal{A}$) = $\mathcal{U}$
Assign-Labels($\mathcal{R}$, $\mathcal{L}$, $\mathcal{A}$) =
$\{(x, \vec{p}(y)) | (x, z) \in \mathcal{R}, \ \vec{p}(y) = \mathcal{A}(\mathcal{L})$*'s estimate of the conditional probability* $p(y|x)$
for each value $y \in \mathcal{Y}\}$
Stopping-Condition(S, S′) = *True* iff *distance*(S,S′) $\leq \gamma$ *for some* $\gamma > 0$

Truncated EM
Choose-Relabel-Set(S, $\mathcal{L}$, $\mathcal{A}$) = $\{x \in S | conf(\mathcal{A}(\mathcal{L})|x) \geq \theta\}$ *where* θ *is a fixed constant threshold, and conf is the classifier's confidence in its prediction*
Assign-Labels($\mathcal{R}$, $\mathcal{L}$, $\mathcal{A}$) = $\{(x, y) | x \in \mathcal{R}, \ y = h(x)\}$*, where h =* $\mathcal{A}(\mathcal{L})$
Stopping-Condition(S, S′) = *True* iff S = S′

Auction
Choose-Relabel-Set($S, \mathcal{L}, \mathcal{A}$) = $\{x \in S | conf(\mathcal{A}(\mathcal{L})|x$ *is among the top* $N\hat{p}(h(x))$ *confidences of all unlabeled examples* $z \in S$ *such that* $h(z) = h(x)\}$ *where* $\hat{p}(c)$ *is the frequency of label c in the initial labeled set, h =* $\mathcal{A}(\mathcal{L})$*, and N is the number of examples labeled per iteration*
Assign-Labels($\mathcal{R}$, $\mathcal{L}$, $\mathcal{A}$) = $\{(x, y) | (x, z) \in \mathcal{R}, \ y = h(x)\}$*, where h =* $\mathcal{A}(\mathcal{L})$
Stopping-Condition(S, S′) = *True* iff S = S′

Co-Training
Here, S can be decomposed into $S^{(1)}$ and $S^{(2)}$ for views 1 and 2 respectively.
Choose-Relabel-Set($S, \mathcal{L}, \mathcal{A}$) = $\{x \in S | \exists i \in \{1,2\}$ *such that* $conf(h^{(i)})|x)$ *is among the top* $\frac{1}{2}N\hat{p}(h^{(i)}(x))$ *confidences of all unlabeled examples* $z \in S^{(i)}$ *such that* $h^{(i)}(z) = h^{(i)}(x)\}$ *where* $\hat{p}(c)$ *is the frequency of label c in the initial labeled set,* $h^{(i)} = \mathcal{A}(\mathcal{L}^{(i)})$*, and N is the number of examples to be labeled per iteration*
Assign-Labels($\mathcal{R}$, $\mathcal{L}$, $\mathcal{A}$) = $\{(x, y) | x \in \mathcal{R}, \ y = h^{(j)}(x)$ *where* $j = argmin_{i \in \{1,2\}} \|\{z \in \mathcal{R} | conf(h^{(i)}|z) > conf(h^{(i)}|x)\}\|\}$
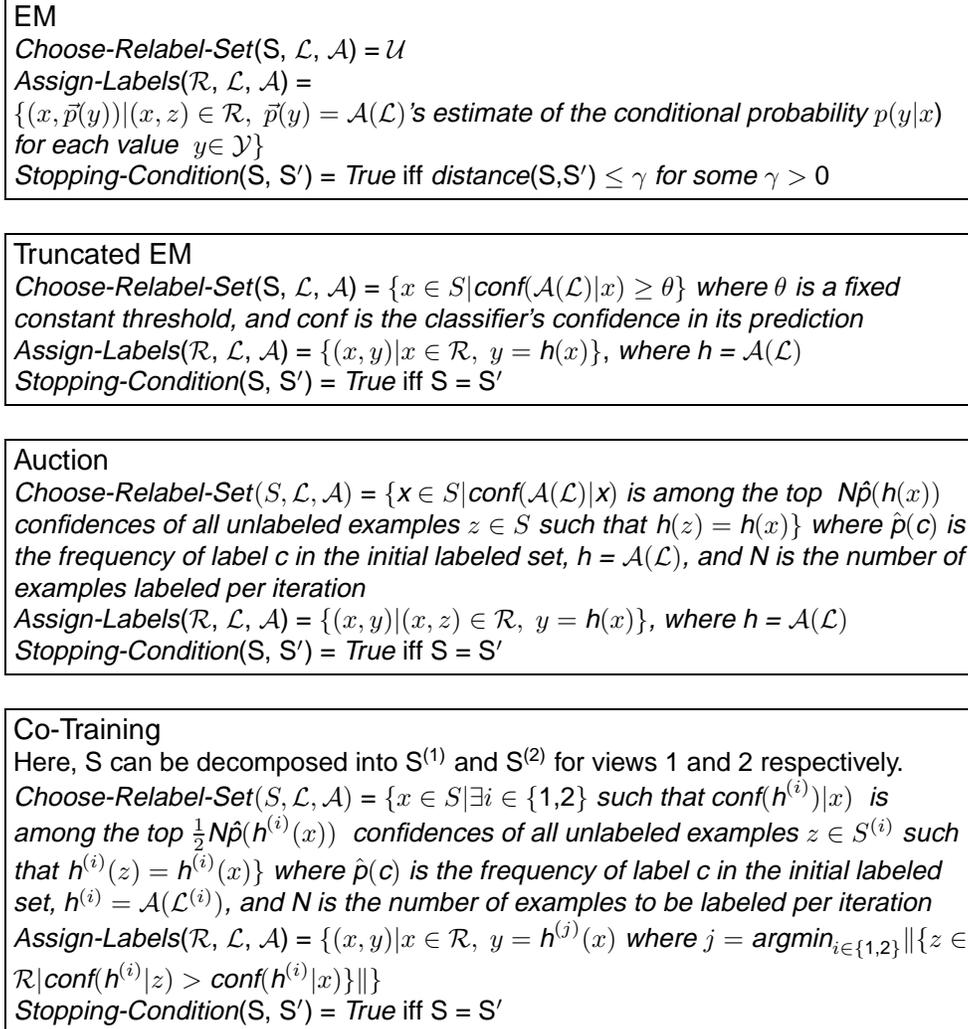*Stopping-Condition*(S, S′) = *True* iff S = S′

Figure 2: Examples of Iterative Labeling meta-algorithms.

Since the only way an Iterative Labeling algorithm can update its hypothesis is by labeling examples and retraining the supervised learner, the goal of Iterative Labeling algorithms is to label the examples in such a way as to minimize the error of the classifier produced by training on all

examples labeled when the algorithm halts. That is, the algorithm should seek a labeling of the unlabeled data that provides an ideal training set for the supervised learning algorithm, so that it produces the best classifier possible from training on the given data. However, as the following theorem states, even if we have an oracle that can determine the true error of of a classifier, the problem of labeling data so as to minimize the error is hard in the general case.

Theorem 2.1 *Let IL^* be an Iterative Labeling meta-algorithm, such that for any supervised learning algorithm $\mathcal{A}$, set of labeled data $\mathcal{L}$, and set of unlabeled data $\mathcal{U}$, error($IL^*_{\mathcal{A}}(\mathcal{L} \cup \mathcal{U})) \leq error(IL_{\mathcal{A}}(\mathcal{L} \cup \mathcal{U}))$, for all Iterative Labeling meta-algorithms IL. Then executing IL^* is NP-Hard.*

Proof The proof is by a polynomial reduction from Subset-Sum, a problem known to be NP-Complete (Garey and Johnson, 1979). The problem of Subset-Sum is stated as follows. Given a set $S \subset \mathcal{N}$, and a positive integer k, determine whether $\exists \mathcal{R} \subseteq S$ such that $\sum_{x \in \mathcal{R}} x = k$.

Define $\mathcal{X} \subset \mathcal{N}$ such that $|\mathcal{X}| < \infty$. Let $\sigma = \sum_{x \in \mathcal{X}} x$, $\mu = \max(\mathcal{X})$, and let k be a positive integer. Now define $\mathcal{Y} = \{0, 1, ..., \max(\sigma\mu, k\mu)\}$, and let $f : \mathcal{X} \rightarrow \mathcal{Y}$ be a target function such that $f(x) = xk \; \forall x \in \mathcal{X}$. Let $\mathcal{D}$ be a distribution over $\mathcal{X} \times \mathcal{Y}$, such that $\mathcal{D}$ assigns positive probability to $(x, y) \in \mathcal{X} \times \mathcal{Y}$ if and only if $y = f(x)$. Define a supervised learning algorithm $\mathcal{A}$, over the sample space $\mathcal{X} \times \mathcal{Y}$, such that for any set of labeled examples $\mathcal{Q}$, and any $x \in \mathcal{X}$, if $h = \mathcal{A}(\mathcal{Q})$, then h$(x) = \sum_{(q,y) \in \mathcal{Q}} qx$.[3] Now imagine the following scenario. Let $S \subseteq \mathcal{X}$ be a set of unlabeled examples with X component selected according to the marginal distribution of $\mathcal{D}$ over $\mathcal{X}$. Then $\exists \mathcal{R} \subseteq S$ such that $\sum_{x \in \mathcal{R}} x = k$ if and only if $error(IL^*_{\mathcal{A}}(S)) = 0$. That is, there is a solution to Subset-Sum if and only if IL^* can perfectly learn the target from the given data. ∎

Remarks. For a finite dataset with a finite number of possible labels, IL^* can generally be applied in exponential time using exhaustive methods. Although this shows the goal of Iterative Labeling algorithms is NP-Hard in the general case, the artificial flavor of the proof leaves open the possibility of interesting specific cases for which the problem may be solved efficiently. In particular, it would be interesting to examine for a specific supervised learning algorithm, under what assumptions on the data the problem becomes tractable.

In the absence of an oracle capable of providing information about the true error of a classifier, these algorithms generally attempt to select and label examples so as to optimize some quantity believed to be roughly related to the error rate of the resulting classifier. For example, Abney shows that a variant of Yarowsky's algorithm can be viewed as optimizing a bound on the likelihood of the data (Abney, 2004). While theorem 2.1 is phrased for error minimization, one might speculate that a similar statement could be posed for many other nontrivial quantities that one might attempt to optimize via Iterative Labeling.

Because of the computational infeasibility of such an optimal algorithm, Iterative Labeling approaches tend to be greedy. Specifically, they tend to select and label examples based on individual properties of those examples instead of selecting and labeling entire sets of examples. A direct consequence of this localized process of selecting and labeling examples is that these algorithms are typically extremely efficient, even for large high-dimensional datasets.

The family of Iterative Labeling meta-algorithms can be further subdivided into two classes, namely those that use the trained classifier's predictions to directly label examples that will be used

---

3. To account for the technicality of restricting predictions of h to the set $\mathcal{Y}$, if $\sum_{(q,y) \in \mathcal{Q}} qx > \sigma\mu$, then h$(x) = 0$.

to update that classifier, and those that label examples via some other process. Let us call the former class *confidence-based* since generally algorithms of this type select examples for which the trained classifier has high confidence in its predictions. Truncated EM and Auction are both confidence-based algorithms.

3. The Problems of Confidence-Based Iterative Labeling

Let us now examine more carefully the distinction between confidence-based and other types of Iterative Labeling algorithms. As mentioned, confidence-based algorithms update the classifier using examples labeled with the classifier's own predictions, and generally selected because the classifier has high confidence in its prediction. It turns out that algorithms of this type generally tend toward excessively conservative updates to the hypothesis, since training on high-confidence examples that the current hypothesis already agrees with will tend to have little effect. The extreme case of this occurs for mistake-driven learning algorithms Littlestone (1988), which can derive no benefit at all from confidence-based semi-supervised learning. Since a classifier produced by a mistake-driven algorithm will always agree with its own predictions, it will never make a mistake on examples it has labeled, and thus the algorithm will never update its hypothesis. One can imagine a continuum of supervised learning algorithms, distinguished by the degree to which they can learn from examples labeled in agreement with their predictions. Thus the closer the learner is to the mistake-driven extreme of this continuum, the less effective confidence-based semi-supervised algorithms will be.

However, despite the ultra-conservative updates of these confidence-based algorithms, they provide no worst case performance guarantees. This problem has been widely observed in iterative semi-supervised learning algorithms. Indeed, it has been shown that in certain situations, which are not yet fully understood, many semi-supervised learning algorithms can significantly *degrade* the performance relative to strictly supervised learning. For example, Pierce and Cardie (2001) addresses the fact that after a number of iterations, co-training's past mistakes can create a snowball effect, during which the accuracy declines. Cohen (2003) discusses situations in which one would be better off discarding the unlabeled data than to use EM for semi-supervised learning. For classification, we can think of these degradations as the algorithm leading itself astray; that is, if we rely on the predictions of the classifier to label examples, then making a mistake on one iteration and training on the result reinforces the mistake, producing a classifier that is more likely to make similar mistakes on the next iteration.

We can thus observe two desirable properties of an Iterative Labeling semi-supervised learning algorithm. The first is that it should allow for significant updates to the hypothesis, and the second is that it should provide worst-case performance guarantees not significantly worse than strictly supervised learning. This work proposes one possible method for selecting and labeling examples based on minimizing the empirical error of the resulting classifier. This approach differs from most existing algorithms in that it does not rely on the predictions of the current hypothesis to select and label examples. Thus, it is able to effect significant updates to the hypothesis using newly labeled examples. Additionally, we provide both worst-case performance guarantees and statistical analysis that justifies the decisions made by the algorithm. We then provide experimental results on a large scale data set, showing that the proposed method significantly outperforms confidence-based approaches when labeled data are scarce.

The preceding discussion suggests that for an Iterative Labeling semi-supervised algorithm to be successful in general, we should seek to select and label examples using a criterion other than the

confidence and predictions of the classifier that will be training on them. One approach which seems to hinge on this reasoning is co-training Blum and Mitchell (1998); Collins and Singer (1999). In the co-training framework, two classifiers are maintained, ideally based on different views of the data. Like the above confidence-based approach, the algorithm selects examples to be labeled for which at least one of the classifiers has high confidence, and takes the prediction of that classifier as the new label for that example. The benefit is that the examples labeled by one classifier are also presented to the other half to update the hypothesis on the complementary view. Thus, the examples, as represented in each view, receive at least some of their labels from a source other than the classifier that will be updated with them.

Co-training has been widely investigated for the case of data containing multiple conditionally-independent views of the data Blum and Mitchell (1998); Dasgupta et al. (2002). However, it has also been observed to improve performance when using two different learning algorithms operating over the same view Goldman and Zhou (2000); Clark et al.. While we know of no existing formal analysis of this latter setting, it can be justified if we assume that the *confidence of a classifier* is indicative of the *accuracy of its prediction*. Assume we have a supervised learning algorithm $\mathcal{A}$, with current hypothesis h, and we wish to train an accurate classifier. In addition, there is a function f separate from this classifier that provides (possibly incorrect) labels for any example, along with a confidence rating for its prediction. A confidence-based algorithm would train $\mathcal{A}$ on examples labeled with high confidence by h. However, if whenever f and h disagree in their predictions for these examples, the prediction with higher confidence is prefered, we would expect the average accuracy of prediction to increase. If we can (naïvely) say that the expected accuracy of a classifier is an increasing function of the fraction of correctly labeled examples in its training set, then in this case accepting f's predictions would improve the accuracy of the resulting classifier. Indeed, we would expect that such a situation of f "overruling" h's confident prediction would cause a significant update to $\mathcal{A}$'s hypothesis. Thus, in addition to providing worst-case performance guarantees, another desirable property of an Iterative Labeling semi-supervised algorithm is to use a source of labels other than the trained classifier.

4. Compatibility-Based Iterative Labeling

Inspired by the above considerations, we propose an Iterative Labeling algorithm that selects and labels examples based on the *empirical performance* of a classifier tested on the initial labeled set. We present a property which represents the ability of an Iterative Labeling algorithm to improve its hypothesis using a given set of unlabeled examples. If there is no labeling of these examples that can be used by the supervised learner to improve the hypothesis, then we say the set is *incompatible* with the learner. We may thus define a new notion of compatibility between a set of unlabeled examples and a learning algorithm as follows.

Definition 4.1 *For a supervised learning algorithm $\mathcal{A}$, a set of labeled training data $\mathcal{L}$, and a set of unlabeled examples $\mathcal{R}$, define the compatibility[4] of $\mathcal{R}$ with $\mathcal{A}$ given $\mathcal{L}$ as*

$$compat(\mathcal{R}, \mathcal{A}|\mathcal{L}) = error(\mathcal{A}(\mathcal{L})) \text{ - } \min_{\lambda \in \Lambda} error(\mathcal{A}(\mathcal{L} \cup \lambda(\mathcal{R})))$$

---

4. This should not be confused with the notion of compatibility introduced in Blum and Mitchell (1998), which is a relation between a *distribution* and a *target function*.

Intuitively, $compat(\mathcal{R}, \mathcal{A}|\mathcal{L}_\rangle)$ is the improvement in error from one iteration to the next, given that *Choose-Relabel-Set* returns $\mathcal{R}$ and *Assign-Labels* labels examples so as to minimize the error of the resulting classifier.

4.1 A Compatility-Based Meta-Algorithm

We would generally like an algorithm for which, on each iteration, *Choose-Relabel-Set* returns a set with positive compatibility, and *Assign-Labels* labels the examples to minimize the error of the resulting classifier. We call such algorithms *compatibility-based*. There are two issues to be addressed along this line of reasoning. The first is computational feasibility and the second involves the lack of direct knowledge of the *true* compatibility of a set.

Ideally, one could propose an algorithm which on the first iteration, selects the set of examples with maximum compatibility, labels them so as to minimize the error of the resulting classifier, and halts. It is trivial to show that if one has access to the true error of the hypothesis, then this meta-algorithm is equivalent to the aforementioned IL* meta-algorithm, and thus by Theorem 2.1, would be NP-Hard to run. As such, we suggest being more greedy, restricting *Choose-Relabel-Set* to select from sets with fewer than some fixed number of elements.

To address the second fact, that we do not generally have access to the true compatibility of a set, we suggest using the initial labeled data as a test set to approximate the error of a classifier. This estimate of the error can then be used to compute the compatibility. To simplify the analysis in this section, we assume that all of the initially labeled examples are used as this test set, and that all examples used to train the classifier originate in the unlabeled set and are labeled by the iterative labeling process.

The idea of using the empirical error as a guiding measure in decisions has been used extensively in various contexts. For example, in the context of part-of-speech tagging, Brill (1995) selects transformation rules by comparing the observed accuracies that the new hypotheses would have after appending these rules, as determined by evaluation on an annotated corpus. In the context of semi-supervised learning, Goldman and Zhou (2000) propose a co-training algorithm based on two learning algorithms rather than redundant views of the data; they use estimated changes in the accuracy of the hypothesis as part of their process of determining whether or not it is worthwhile to label a given example, determined via cross-validation on the initial labeled set.

4.2 Justification

We would now like to determine whether compatibility-based algorithms have the desirable properties outlined in the previous section. It should be clear that the labels are not assigned based on the *predictions* of the classifier. Rather, they are assigned based on the *resulting error* of the classifier *after training* on those examples. Intuitively, one could view this as a step toward the independent source of labels provided in cotraining, and a step away from the uninformative labels of confidence-based algorithms like Truncated EM and Auction. Thus we would expect the hypothesis to experience significant updates from training on examples labeled in this way.

Now we turn to the question of worst-case performance guarantees. As noted earlier, algorithms that label examples with the predictions of the current hypothesis can lead themselves astray, accumulating labeling errors as they progress. However, because compatibility-based algorithms are ultimately based on empirical risk minimization, we can draw upon well-known bounds on its resulting performance Vapnik (1998). In particular, we can take the VC dimension of the family of

classifiers to be upper bounded by $\min(\|\mathcal{U}\| \lg(\|\mathcal{Y}\| + 1), \mathcal{VC}(\mathcal{A}))$, where $\mathcal{VC}(\mathcal{A})$ is the VC dimension of the hypothesis space of algorithm $\mathcal{A}$. Thus, if the unlabeled set is large enough, we expect that compatibility-based algorithms would never perform significantly worse than a strictly supervised learning algorithm that minimizes the empirical risk on the initial labeled set. This guarantee is not possible for other approaches that do not verify their accuracy on a labeled set.

Examining the decisions made at each step of the algorithm, we can discuss the statistical significance of labeling examples in this way. This process can be modeled as a classical hypothesis testing scenario . We compare the possible labelings of the data using the classifiers they would produce after training. The labeling corresponding to a classifier with lowest empirical error on a labeled set $\|\mathcal{L}\|$ is selected. Assume binary classification for simplicity, and that we are deciding whether to label a single example as positive or negative. Let f_+ and f_- be the corresponding classifiers produced by training on all examples labeled in previous iterations, plus this new example with positive or negative label respectively. We consider that for any labeled set $\mathcal{L}$, there are 3 disjoint types of examples in $\mathcal{L}$, namely those $(x, y) \in \mathcal{L}$ for which $f_+(x) = f_-(x)$, $f_+(x) = y \neq f_-(x)$, or $f_+(x) \neq y = f_-(x)$. Let a, b, and c denote the number of examples with each of these respective qualities, and assume $b \geq c + \varepsilon$, for integer $\varepsilon > 0$, so that our method labels the example +. Then the decision criterion employed in selecting this label forms a statistical test for a trinomial distribution, and has an approximate significance level of at most

$$\sum_{i=0}^{\frac{\|\mathcal{L}\|-\varepsilon}{2}} \sum_{j=i+\varepsilon}^{\|\mathcal{L}\|-i} \frac{\|\mathcal{L}\|!}{i!j!(\|\mathcal{L}\| - i - j)!} (\frac{c+b}{2\|\mathcal{L}\|})^{i+j}(1 - \frac{c+b}{\|\mathcal{L}\|})^{\|\mathcal{L}\|-i-j}.$$

This formula may be evaluated numerically for any given set $\mathcal{L}$. For example, take $\|\mathcal{L}\| = 200$, and say the two possible classifiers disagree on some number of examples, of which f_+ gets 3 more right than f_-. Then the significance of this test is approximately 0.13, or in other words, we are approximately 87% sure that this is the right choice. Thus, we can say that the proposed process for labeling examples makes reasonable choices in a strictly formal sense.

5. Experimental Results

We performed experiments on confidence-based and compatibility-based algorithms. All algorithms here use the SNoW learning architecture Carlson et al. (1999) with Winnow update rules Littlestone (1988) for a core supervised learning algorithm[5]. For the confidence-based algorithms, we use softmax Bishop (1996) over the raw activation values as confidences. Specifically, suppose the number of classes is n, and the raw activation values of class i is act_i. The confidence for class i is derived by $\frac{e^{act_i}}{\sum_{1 \leq j \leq n} e^{act_j}}$, a form known to be a good transformation of activations into conditional probabilities. The confidence-based algorithms we evaluate include Truncated EM and Auction, as described above. For these experiments, Truncated EM uses a threshold of 0.85, selected based on good performance on a related dataset, and Auction labels approximately 0.1% of the initial amount of unlabeled data on each iteration.

As an example of a compatibility-based semi-supervised learning algorithm, we use the following greedy approach. *Choose-Relabel-Set* examines the unlabeled examples $x \in \mathcal{U}$ in random order and returns the first example x such that $\{x\}$ has positive compatibility. *Assign-Labels* labels this

---

5. Since the multiclass classification is performed via a winner-take-all approach, the overall algorithm is not actually mistake-driven, and so it has the potential to affect confidence-based algorithms.
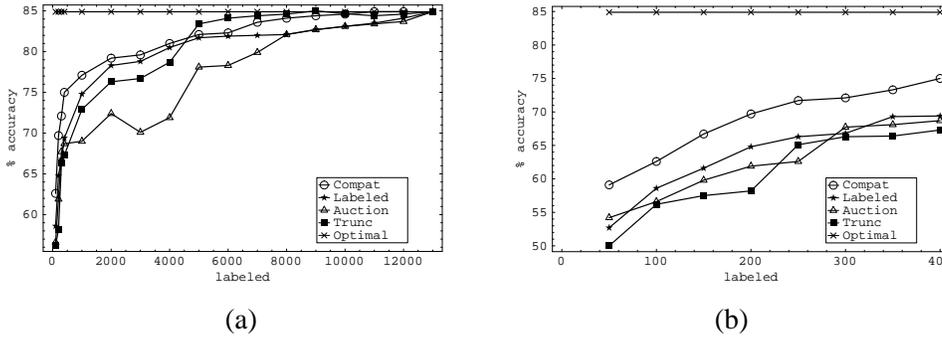
(a)   (b)

Figure 3: Iterative Labeling meta-algorithms using a Winnow-based supervised learner on a named entity classification task. Here, "Labeled" indicates the Winnow hypothesis trained strictly on the initial labeled set, "Compat" is the proposed compatibility-based algorithm, "Auction" is the Auction algorithm, "Trunc" is Truncated EM, and "Optimal" is the performance when all data is labeled. The horizontal axis represents the initial number of labeled examples, and vertical is the percent accuracy. The results are averaged over at least 20 runs per point. (b) displays a magnification of the region in which labeled data are scarce, revealing that the compatibility-based approach significantly outperforms confidence-based approaches for small initial labeled sets, which is typically the most important situation for semi-supervised learning.

example so as to minimize an estimate of the error of the classifier that results from adding this newly labeled example to the labeled set and retraining. That is, the algorithm tries all possible labels and calculates the updated hypothesis, which is then evaluated on a set of labeled examples to estimate its error. We estimate the error via a form of 5-fold cross-validation in which the initial labeled data is split into 5 disjoint sets; the classifier is then trained on four of the sets along with all newly labeled data, then evaluated on the fifth set; the process is then repeated for each of the five such combination of these labeled sets. The algorithm halts when there are no more examples $x \in \mathcal{U}$ such that $compat(\{x\}, \mathcal{A}|\mathcal{L}) > 0$, and concludes by training the classifier on all examples that have labels at that point. Additionally, for small initial labeled sets, we allow the algorithm to add a limited number of newly labeled examples to the labeled set to be used for evaluation. While this could potentially increase the bias of the algorithm, we have observed that the technique tends to improve performance in general when labeled data are scarce, due to decreasing the variance of the estimates.

Evaluation was performed on a named entity classification task as in Collins and Singer (1999), though our data comes from a TREC dataset. The task of named entity classification is to identify the type of a given entity as encountered in a textual context. We are given features that describe a word or phrase and the context in which it occurs. We are then asked to classify it as one of several types. For the dataset used in these experiments, the traditional categories of *person, location, organization,* and *other* are used. The task of named entity classification for these categories is a well-studied problem, with excellent results obtained by supervised learning with large quantities of labeled data. However, the task is particularly relevant to semi-supervised learning, since one

10

may become interested in a new type of entity not present in the training data. In such a situation, one should not be required to gather large amounts of data and annotate them by hand. Rather, we would like to provide a relatively small number of labeled examples and a large number of unlabeled examples (perhaps automatically gathered from the web). The dataset consists of 14,177 examples, roughly balanced between the four classes. Results are displayed in figure 2. Each point is generated by selecting a random subset of the required size as labeled data, a subset of size 1000 as a test set unavailable to the algorithm and used to evaluate the performance, and the remainder is used for unlabeled data; this process is repeated and averaged over at least 20 runs per point so that the observed differences are significant.

The results are encouraging for the compatibility-based approach. The differences are especially seen for small initial labeled sets, which is the situation in which semi-supervised learning is most useful in practice. The fact that Truncated EM eventually outperforms the compatibility-based algorithm for large labeled sets can be understood in light of the confidence-based algorithm making fewer mistakes. Indeed, there appears to be threshold around 4000 labeled examples, past which the confidence-based algorithms dramatically improve. It might be interesting to propose a hybrid algorithm that uses a compatibility-based approach to boost the performance in settings with few labeled examples, but switches to a more confidence-based approach for large labeled sets. We interpret the poor performance of Auction as caused by its requirement to label *all* of the unlabeled data.

6. Discussion and Open Problems

We defined the Iterative Labeling family of semi-supervised meta-algorithms and described a sub-family of confidence-based algorithms. We discussed the fact that these algorithms generally tend toward excessively conservative updates to the hypothesis. This paper also discussed situations in which confidence-based algorithms actually degrade performance. We suggested an alternate method of selecting and labeling the examples based on minimizing the empirical error of the resulting classifier, and explained why this is a reasonable approach. Finally, the strength of this technique was demonstrated empirically on the task of named entity classification.

In this paper, we have compared one possible method of selecting and labeling examples based on a criterion other than the confidence of the current hypothesis; one clear direction for investigation, which we are currently pursuing, is a comparison with other non-confidence-based algorithms, such as co-training. We might expect co-training to perform comparable to Truncated EM, as is indicated by Collins and Singer (1999), but it seems an interesting avenue of investigation to determine under what circumstances either approach is dominated by the other.

Much work remains in looking for accurate, efficient compatibility-based algorithms, possibly with generalization guarantees better than the worst-case bounds given here. One disadvantage of a compatibility-based approach is the added computational expense of estimating the error many times for each unlabeled example. While this can be somewhat alleviated by parallelizing the cross-validation and using an online supervised learning algorithm, identifying new ways to expedite the learning process without sacrificing the error-driven approach would be an interesting direction. Additionally, a thorough exploration of various types of supervised learnering algorithms and how much benefit each can derive from an Iterative Labeling meta-algorithm, for example a confidence-based algorithm, is needed. Comparisons between Iterative Labeling algorithms and non-Iterative Labeling algorithms, as well as an investigation into the relation between Iterative La-

beling and Active Learning are also desirable. Finally, it is essential to investigate the applicability of compatibility-based algorithms in real-world settings.

References

Steven Abney. Understanding the yarowsky algorithm. *Computational Linguistics*, 2004.

Kristin Bennett, Ayhan Demiriz, and Richard Maclin. Exploiting unlabeled data in ensemble methods. In *SIGKDD*, 2002.

Christopher M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1996.

Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the Workshop on Computational Learning Theory, Morgan Kaufmann Publishers*, 1998.

Eric Brill. Error-driven learning and natural language processing: A case study in part-of-speech tagging. In *Computational Linguistics*, 1995.

A. Carlson, C. Cumby, J. Rosen, and D. Roth. The SNoW learning architecture. Technical Report UIUCDCS-R-99-2101, UIUC Computer Science Department, May 1999.

Stephen Clark, James Curran, and Miles Osborne. Bootstrapping pos-taggers using unlabelled data. In *CoNLL-03*.

Ira Cohen. *Semisupervised Learning of Classifiers with Applications to Human-Computer Interaction*. PhD thesis, Department of Electrical Engineering, University of Illinois at Urbana-Champaign, 2003.

David Cohn, Rich Caruana, and Andrew McCallum. Semi-supervised clustering with user feedback.

M. Collins and Y. Singer. Unsupervised models for named entity classification. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, 1999.

S. Dasgupta, M. L. Littman, and D. McAllester. Pac generalization bounds for co-training. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, pages 375–382, Cambridge, MA, 2002. MIT Press.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38, 1977.

M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman and Co., New York, 2nd edition, 1979.

Sally Goldman and Yan Zhou. Enhancing supervised learning with unlabeled data. In *Proc. 17th ICML*, pages 327–334. Morgan Kaufmann, San Francisco, CA, 2000.

N. Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning*, 2:285–318, 1988.

Rada Mihalcea. Co-training and self-training for word sense disambiguation. In *Proceedings of CoNLL-2004*, pages 33–40. Boston, MA, USA, 2004.

Kamal Nigam, Andrew K. McCallum, Sebastian Thrun, and Tom M. Mitchell. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39(2/3):103–134, 2000.

David Pierce and Claire Cardie. Limitations of co-training for natural language learning from large datasets. In *Conference on Empirical Methods in Natural Language Processing*, 2001.

Dale Schuurmans. A new metric-based approach to model selection. In *Fourteenth National Conference on Artificial Intelligence*, 1997.

Vladimir Vapnik. *Statistical Learning Theory*. John Wiley & Sons, Inc., 1998.

David Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *Meeting of the Association for Computational Linguistics*, pages 189–196, 1995.

X. Zhu and Z. Ghahramani. Learning from labeled and unlabeled data with label propagation. Technical report, Carnegie Mellon University, 2002.