Efficient Semi-supervised and Active Learning of Disjunctions

Maria-Florina Balcan NINAMF@CC.GATECH.EDU
Christopher Berlind CBERLIND@GATECH.EDU
Steven Ehrlich SEHRLICH@CC.GATECH.EDU
Yingyu Liang YLIANG39@GATECH.EDU
School of Computer Science, College of Computing, Georgia Institute of Technology, Atlanta, GA, USA

Abstract

We provide efficient algorithms for learning disjunctions in the semi-supervised setting under a natural regularity assumption introduced by (Balcan & Blum, 2005). We prove bounds on the sample complexity of our algorithms under a mild restriction on the data distribution. We also give an active learning algorithm with improved sample complexity and extend all our algorithms to the random classification noise setting.

1. Introduction

In many modern applications, like web-based information gathering, unlabeled data is abundant but labeling it is expensive. Consequently, there has been substantial effort in understanding and using semi-supervised learning (using large amounts of unlabeled data to augment limited labeled data) and active learning (where the algorithm itself asks for labels of carefully chosen examples with the goal of minimizing the human labeling effort) (Zhu, 2011; Dasgupta, 2011).

Conceptually, what makes unlabeled data useful in the semi-supervised learning context (Balcan & Blum, 2010; Zhu, 2011), is that for many learning problems, the natural regularities of the problem involve not only the form of the function being learned but also how this function relates to the distribution of data; for example, that it partitions data by a wide margin as in Transductive SVM (Joachims, 1999a) or that data contains redundant sufficient information as in Co-training (Blum & Mitchell, 1998). Unlabeled data is useful in this context because it allows one to reduce the search space from the whole set of hypotheses, down to the set of hypotheses satisfying the regularity condition with respect to the underlying distribution. Such insights have been exploited for deriving a variety of sample complexity results (Dasgupta et al., 2001; Kaariainen, 2005; Rigollet, 2007; Balcan & Blum, 2010). However, while in principle semi-supervised learning can provide benefits over fully supervised learning (Balcan & Blum, 2010; Zhu, 2011), the corresponding algorithmic problems become much more challenging. As a consequence there has been a scarcity of efficient semi-supervised learning algorithms.

In this paper we provide efficient algorithms with nearly optimal sample complexity for semi-supervised and active learning of disjunctions under a natural regularity assumption introduced in (Balcan & Blum, 2005). In particular we consider the so called two-sided disjunctions setting, where we assume that the target function is a monotone disjunction satisfying a margin like regularity assumption[1]. In the simplest case resolved in (Balcan & Blum, 2005), the notion of "margin" is as follows: every variable is either a positive indicator for the target function (i.e., the true label of any example containing that variable is positive) or a negative indicator (i.e., the true label of any example containing that variable is negative), and no example contains both positive and negative indicators. In this work, we consider the much more challenging setting left open in (Blum & Balcan, 2007) where *non-indicators* or *irrelevant variables*, i.e., variables that appear in both positive and negative examples, are also present.

In the semi-supervised learning setting, we present an algorithm that finds a consistent hypothesis that furthermore is compatible (in the sense that it satisfies our regularity assumption). This algorithm is proper (it outputs a disjunction), has near-optimal labeled data sample complexity provided that each irrelevant variable appears with non-negligible probability, and it is efficient when the number of irrelevant variables

Proceedings of the 30^{th} International Conference on Machine Learning, Atlanta, Georgia, USA, 2013. JMLR: W&CP volume 28.

[1] See Section 6 for further discussion.

is $O(\log n)$. We next present a non-proper algorithm that PAC learns two-sided disjunctions with nearly the same sample complexity and whose running time is polynomial for any k.

In the active learning setting, we present an efficient active learning algorithm for two-sided disjunctions. This algorithm outputs a consistent, compatible hypothesis, with sample complexity linear in the number of irrelevant variables and independent of the probability of irrelevant variables appearing, a quantity that appears in the bounds in the semi-supervised setting. Combined with the NP-hardness result we show for two-sided disjunctions (see supplementary material), the algorithm also shows that the active query ability can help to overcome computational difficulty.

We also discuss how our algorithms can be adapted to deal with random classification noise.

Discussion: To see why the presence of irrelevant variables significantly complicates the algorithmic problem, note that in the absence of non-indicators (the case studied in (Balcan & Blum, 2005)), we could construct an approximation of the so called commonality graph, defined on n vertices (one per variable), by putting an edge between two vertices i and j if there is any example x in our unlabeled sample with x_i, x_j set to 1. If the target function indeed satisfies the regularity assumption, then no component will get multiple labels, so all we need to learn is a labeled example in each component. Furthermore, if the number of components in the underlying graph is small, then both in the semi-supervised and active learning setting we can learn with many fewer labeled examples then in the supervised learning setting.

Introducing non-indicators into the target concept complicates matters, because components can now have multiple labels. We could think of the non-indicators as forming a vertex cut in the commonality graph separating variables corresponding to positive indicators from those corresponding to negative ones. To learn well, one could try to find such a cut with the necessary properties to ensure compatibility with the unlabeled data (i.e. no examples are composed only of non-indicators). Unfortunately, this is a difficult combinatorial problem in general. Interestingly, we will be able to find such cut for $k = O(\log n)$ and for general k we will be still be able to learn with nearly optimal rates, if each non-indicator appears with non-negligible probability; we do this by identifying a superset of non-indicators and carefully making inferences using it. Furthermore, since mistakes reveal vertices in both sides of the cut, our adaptive query ability in the active learning model will allow us to actively search for vertices in the cut.

Related work: While several semi-supervised learning methods have been introduced (Chapelle et al., 2006; Zhu et al., 2003; Joachims, 1999b), much of the theoretical work has focused either on sample complexity (e.g., (Dasgupta et al., 2001; Kaariainen, 2005; Rigollet, 2007)) or on providing polynomial time algorithms with error bounds for surrogate losses only (e.g., (Rosenberg & Bartlett, 2007)). The few existing results with guarantees on the classification error loss hold under very stringent conditions about the underlying data distribution (e.g., independence given the label (Blum & Mitchell, 1998)). By contrast, we provide (PAC-style) polynomial time algorithms for learning disjunctions with general guarantees on the classification error loss.

We note that while a lot of the research on active learning (Dasgupta, 2005; Balcan et al., 2006; Hanneke, 2007a;b; Dasgupta et al., 2007; Beygelzimer et al., 2010; Koltchinskii, 2010) has *not* made an explicit regularity assumption as in semi-supervised learning, this is an interesting direction to study. As our results reveal, active learning could help overcome computational hardness limitations over (semi-supervised) passive learning in these settings.

2. Preliminaries and Notation

Let $X = \{0,1\}^n$ be the instance space, $Y = \{-1, 1\}$ be the label set, and D denote any fixed probability distribution over X. Following (Balcan & Blum, 2005), a two-sided disjunction h is defined as a pair of monotone disjunctions[2] (h_+, h_-) such that $h_+(x) = -h_-(x)$ for all $x \sim D$, and h_+ is used for classification. Let the concept class C be the set of all pairs[3] of monotone disjunctions and for any hypothesis $h = (h_+, h_-) \in C$, define $h(x) = h_+(x)$.

For a two-sided disjunction (h_+, h_-), variables included in h_+ are the *positive indicators*, and variables in h_- are *negative indicators*. Variables appearing neither in h_+ nor in h_- are called *non-indicators*, as the value of any such variable has no effect on the label of any example. To simplify the discussion, we will often identify binary strings in $X = \{0,1\}^n$ with subsets of the variables $V = \{x_1, \ldots, x_n\}$. In other words, we say an example x contains x_i if the i-th coordinate of x is set to 1. This allows us to speak of variables "appearing in" or "being present in" examples rather than variables being set to 1. We will use similar language when

[2] Recall that a monotone disjunction is an OR function of positive literals only, e.g. $h(x) = x_1 \vee x_3 \vee x_4$.

[3] Although we are actually interested in learning a single monotone disjunction, we need to associate each disjunction with a second disjunction in order to test compatibility.

referring to hypotheses, so that a two-sided disjunction $h = (h_+, h_-)$ consists of a set h_+ of positive indicators and a set h_- of negative indicators (which completely determine a third set of non-indicators).

In the semi-supervised learning setting, we will assume that both labeled examples L and unlabeled examples U are drawn i.i.d. from D and that examples in L are labeled by the target concept h^*, where h^* is a two-sided disjunction with at most k non-indicators. We will let $|L| = m_l$ and $|U| = m_u$; both m_l and m_u will be polynomial throughout the paper. In the active setting, the algorithm first receives a polynomially sized unlabeled sample U and it can adaptively ask for the label $\ell(x) = h^*(x)$ of any example $x \in U$. In the random classification noise model (studied in Section 5) we assume that the label of each labeled example is flipped with probability α.

The generalization error of a hypothesis h is given by $\mathsf{err}(h) = \Pr_{x\sim D}[h(x) \neq h^*(x)]$, the probability of h misclassifying a random example drawn from D. For a set L of labeled examples, the empirical error is given by $\mathsf{err}_L(h) = |L|^{-1} \sum_{x\in L} I[h(x) \neq h^*(x)]$. If $\mathsf{err}_L(h) = 0$ for some h we say that h is *consistent* with the data.

To formally encapsulate the regularity or compatibility assumption for two-sided disjunctions described in the introduction, we consider the *compatibility function* χ: $\chi(h, x) = I[h_+(x) = -h_-(x)]$ for any hypothesis h and example $x \in X$. In addition, we define (overloading notation) the compatibility between h and the distribution D as $\chi(h, D) = \mathrm{E}_{x\sim D}[\chi(h, x)] = \Pr_{x\sim D}[h_+(x) = -h_-(x)]$. For a set U of unlabeled examples, define the empirical compatibility between h and U as $\chi(h, U) = |U|^{-1} \sum_{x\in U} I[h_+(x) = -h_-(x)]$. If $\chi(h, U) = 1$ we say that h is *compatible* with the data. Thus a hypothesis is consistent and compatible with a set of examples if every example contains exactly one type of indicator and every labeled example contains an indicator of the same type as its label. We will assume throughout that the target function is compatible.

We define, for any $\epsilon > 0$, the reduced hypothesis class $C_{D,\chi}(\epsilon) = \{h \in C : 1 - \chi(h, D) \leq \epsilon\}$, the set of hypotheses with "unlabeled error" at most ϵ. Similarly, $C_{U,\chi}(\epsilon) = \{h \in C : 1 - \chi(h, U) \leq \epsilon\}$ for an unlabeled sample U. The key benefit of using unlabeled data and our regularity assumption is that the number of labeled examples will only depend on $\log |C_{D,\chi}(\epsilon)|$ which for helpful distributions will be much smaller than $\log |C|$.

2.1. The Commonality Graph

The basic structure used by all of our algorithms is a construct we call the *commonality graph*. As mentioned in the introduction, the commonality graph is the graph on variables that contains an edge between two vertices if the corresponding variables appear together in a common example. That is, given the set U of unlabeled examples, define the commonality graph $G_{\mathrm{com}}(U) = (V, E)$ where $V = \{x_1, \ldots, x_n\}$ and E contains an edge (x_i, x_j) if and only if there is some $x \in U$ such that x_i and x_j are both set to 1 in x. Furthermore, given the set L of labeled examples, let V_L^+ be the set of variables appearing in positive examples and V_L^- be the set of variables appearing in negative examples.

The edge structure of the commonality graph and the labeled examples will allow us to draw inferences about which vertices in the graph correspond to positive indicators, negative indicators, and non-indicators in the target concept. Any variable that appears in a labeled example cannot be an indicator of the type opposite of the label. In addition, an edge between two variables implies they cannot be indicators of different types. This means that any path in the commonality graph between positive and negative indicators must contain a non-indicator. Similarly, paths that pass only through indicator variables can be used to propagate labels to the unlabeled examples.

3. Semi-supervised Learning

Our general strategy is to identify non-indicators and remove them from the commonality graph, reducing this problem to the simpler case. Notice that each non-indicator that appears in the unlabeled data is significant; failing to identify it can lead to incorrect inferences about a large probability mass of examples. A variable is obviously a non-indicator if it appears in both positive and negative examples. A naïve approach would be to draw enough labeled examples so that every significant non-indicator appears in examples with both labels. The problem with this approach is that some non-indicator can appear much more frequently in positive examples than in negative examples. In this case the number of examples needed by the naïve approach is inversely proportional to the probability of that non-indicator appearing in negative examples. This sample complexity can be worse than in the fully supervised case.

In our approach, it is enough to ensure each non-indicator appears in a labeled example, but not necessarily in both positive and negative examples. The number of examples needed in this case will now depend on the minimum probability of a non-indicator appearing. This allows the sample complexity to be significantly smaller than that of the naïve approach; for example, when a non-indicator appears in positive

examples with constant probability while in negative examples with probability ϵ/n.

Our approach can still identify non-indicators, now by examining paths in the commonality graph. In paths whose interior vertices appear only in unlabeled examples (i.e. are indicators) and whose endpoints appear in oppositely labeled examples, one of the endpoints must be a non-indicator. When $k = O(\log n)$ we can enumerate over all consistent compatible hypotheses efficiently by restricting our attention to a small set of paths.

If the number of non-indicators is larger, we can still find a good hypothesis efficiently by finding the non-indicators one at a time. Each time our working hypothesis makes a mistake this reveals a path whose endpoint is a non-indicator.

The number of labeled examples we require will depend on the *minimum non-indicator probability* defined by

$$\epsilon_0(D, h^*) = \min_{x_i \notin h^*_+ \cup h^*_-} \Pr_{x \sim D}[x_i = 1].$$

For notational convenience denote it simply by ϵ_0 without ambiguity. To guarantee with high probability that each non-indicator appears in some labeled example, it suffices to use $\tilde{O}(\frac{1}{\epsilon_0} \log k)$ labeled examples.

3.1. Finding a Consistent, Compatible Hypothesis Efficiently when $k = O(\log n)$

We now give an algorithm, along with some intuition, for finding a two-sided disjunction that is consistent and compatible with a given training set. We note that this problem is NP-hard in general (see Section 2 in supplementary material). Given example sets L and U, the algorithm begins by constructing the commonality graph $G = G_{\text{com}}(U)$ and setting G to $G \setminus (V_+ \cap V_-)$. This removes any variables that appear in both positive and negative examples as these must be non-indicators.

To identify the rest of the non-indicators, we consider a new graph. Using $u \leftrightarrow_G v$ to denote the existence of a path in the graph G between vertices u and v, we define the *indicator graph* $G_{\text{ind}}(G, V_+, V_-)$ to be the bipartite graph with vertex set $V_+ \cup V_-$ and edge set $\{(u, v) \in V_+ \times V_- : u \leftrightarrow_{G \setminus (V_+ \cup V_-)} v\}$. The key idea is that an edge in this graph implies that at least one of its endpoints is a non-indicator, since the two variables appear in oppositely labeled examples but are connected by a path of indicators.

Note that the target set of non-indicators must form a vertex cover in the indicator graph. By iterating over all minimal vertex covers, we must find a subset of the target non-indicators whose removal disconnects positive examples from negative examples, and this corresponds to a consistent compatible hypothesis. The algorithm is summarized in Algorithm 1.

Algorithm 1 Finding a consistent compatible two-sided disjunction

Input: unlabeled set U, labeled set L
Set $G = G_{\text{com}}(U)$, $V_+ = V_L^+$, $V_- = V_L^-$
Set $G = G \setminus (V_+ \cap V_-)$
Set $V_+ = V_+ \cap G$, $V_- = V_- \cap G$
Set $G_I = G_{\text{ind}}(G, V_+, V_-)$
for each minimal vertex cover S of G_I **do**
  Set $G' = G \setminus S$, $V'_+ = V_+ \setminus S$, $V'_- = V_- \setminus S$
  Set $h_+ = \{v \in G' : \exists u \in V'_+,\ u \leftrightarrow_{G'} v\}$
  if $(h_+, G' \setminus h_+)$ is consistent and compatible **then break**
Output: hypothesis $h = (h_+, G' \setminus h_+)$

The key step in Algorithm 1 is enumerating the minimal vertex covers of the indicator graph G_I. One way to do this is as follows. First find a maximum matching M in G_I, and let m be the number of disjoint edges in M. Enumerate all 3^m subsets of vertices that cover M (for every edge in M, one or both of the endpoints can be included in the cover). For each such cover S, extend S to a minimal vertex cover of G_I by adding to S every variable not covered by M that has no neighbors already in S. This extension can always be done uniquely, so there is a one-to-one correspondence between covers of M and minimal vertex covers of G_I.

This enumeration method gives us both a concrete way to implement Algorithm 1 and a way to bound its running time. We prove in Theorem 1 that given enough data, Algorithm 1 correctly outputs a consistent compatible hypothesis with high probability. We then bound its time and sample complexity.

Theorem 1. *For any distribution D over $\{0,1\}^n$ and target concept $h^* \in C$ such that $\chi(h^*, D) = 1$, h^* has at most k non-indicators, and the minimum non-indicator probability is ϵ_0, if $m_u \geq \frac{1}{\epsilon}\left[\log \frac{2|C|}{\delta}\right]$ and*

$$m_l \geq \max\left\{\frac{1}{\epsilon_0}\log\frac{k}{\delta}, \frac{1}{\epsilon}\left[\log\frac{2|C_{D,\chi}(\epsilon)|}{\delta}\right]\right\}$$

then with probability at least $1-2\delta$, Algorithm 1 outputs a hypothesis $h \in C$ such that $\text{err}_L(h) = 0$, $\chi(h, U) = 1$, and $\text{err}(h) \leq \epsilon$. Furthermore, when $k = O(\log n)$ the algorithm runs in time at most poly(n).

Proof. We separate the proof into three sections, first proving consistency and compatibility of the output hypothesis, then giving the sample sizes required to

guarantee good generalization, and finally showing the overall running time of the algorithm.

Consistency and Compatibility: The exit condition for the loop in Algorithm 1 guarantees that the algorithm will output a consistent compatible hypothesis, so long as a suitable minimal vertex cover of G_I is found. Thus, it suffices to show that such a vertex cover exists with high probability when L is large enough.

By the definition of ϵ_0 along with the independence of the samples and a union bound, if $m_l > \frac{1}{\epsilon_0} \log \frac{k}{\delta}$, then with probability at least $1 - \delta$, all non-indicator variables appear in some labeled example. We will assume in the remainder of the proof that all variables not in $V_L^+ \cup V_L^-$ are indicators.

Since an edge in G between indicators forces both endpoints to be of the same type, a path through indicators does the same. Edges in G_I correspond to such paths, but the endpoints of such an edge cannot be indicators of the same type because they appear in differently labeled examples. It follows that at least one endpoint of every edge in G_I must be a non-indicator.

Now let V_0 be the set of non-indicators in the target hypothesis. The above observations imply that V_0 contains a vertex cover of G_I. It follows that there must exist a subset $\tilde{S} \subseteq V_0$ that is a minimal vertex cover of G_I. Let $\tilde{h} = (\tilde{h}_+, \tilde{h}_-)$ be the hypothesis h formed from the minimal vertex cover $S = \tilde{S}$. We only need to show that $\tilde{h}$ is both consistent and compatible.

Every indicator of h^* is also an indicator of $\tilde{h}$ since only true non-indicators were removed from G and all remaining variables became indicators in $\tilde{h}$. Since every example contains an indicator of h^*, every example must contain an indicator of $\tilde{h}$ of the correct type. Furthermore, if an example contained both positive and negative indicators, this would imply an edge still present in G_I. But removing a vertex cover removes all edges, so this is impossible. Hence $\tilde{h}$ is a consistent, fully compatible hypothesis.

Generalization Error: If $m_u \geq \frac{1}{\epsilon}[\log \frac{2|C|}{\delta}]$ and $m_l \geq \max\{\frac{1}{\epsilon_0} \log \frac{k}{\delta}, \frac{1}{\epsilon}[\log \frac{2|C_{D,\chi}(\epsilon)|}{\delta}]\}$, the above analysis states that Algorithm 1 will fail to produce a consistent compatible hypothesis with probability at most δ. Furthermore, an algorithm with true error rate greater than ϵ will be fully consistent with a labeled set of size m_l with probability at most $\delta / C_{D,\chi}(\epsilon)$. Union bounding over all compatible hypotheses we see that a consistent compatible hypothesis will fail to have an error rate less than ϵ with probability at most δ. By a union bound over the two failure events, the overall probability of failure is $\leq 2\delta$.

Running Time: Since checking consistency and compatibility can be done in time polynomial in the number of examples, the limiting factor in the running time is the search over minimal vertex covers of G_I. In a bipartite graph, the size of the minimum vertex cover is equal to the size of the maximum matching. The set of k non-indicators in the target hypothesis includes a vertex cover of G_I, so the size m of the maximum matching is at most k. There is one minimal vertex cover for each of the 3^m covers of the maximum matching, so the number of minimal vertex covers to search is at most 3^k. □

3.2. A General Semi-supervised Algorithm

Algorithm 1 is guaranteed (provided the labeled set is large enough) to find a hypothesis that is both consistent and compatible with the given data but is efficient only when k is logarithmic in n. When k is instead polylogarithmic in n, our algorithm is no longer efficient but still achieves a large improvement in sample complexity over supervised learning. We now present an efficient algorithm for finding a low-error (but not necessarily consistent and compatible) hypothesis which matches the sample complexity of Algorithm 1.

The algorithm, summarized in Algorithm 2, begins by constructing the commonality graph from the unlabeled examples and identifying potential indicators from a subset of the labeled examples. We use sample(m, S) to denote a random sample of m elements from set S. An initial hypothesis is built and tested on the sequence of remaining labeled examples. If the hypothesis makes a mistake, it is updated and testing continues. Each update corresponds to either identifying a non-indicator or labeling all indicators in some connected component in the commonality graph, so the number of updates is bounded. Furthermore, if the hypothesis makes no mistakes on a large enough sequence of consecutive examples, then with high probability it has a small error rate overall. This gives us a stopping condition and allows us to bound the number of examples seen between updates.

The hypotheses in Algorithm 2 use a variation on nearest neighbor rules for classification. Given a commonality graph G and a set L of labeled examples, the associated nearest neighbor hypothesis $h_{G,L}$ classifies an example x the same as the nearest labeled example in L. The distance between two examples x and x' is the measured by the minimum path distance in G between the variables in x and the variables in x'. If no examples in L are connected to x, then $h_{G,L}$ classifies x negative by default. For convenience, we use $\text{nn}_{G,S}(x)$ to denote the vertex in the set S nearest to a variable

Algorithm 2 Learning a Low-error Hypothesis

Input: data sets U and L, parameters ϵ, δ, k
Set $L' = \text{sample}(\frac{1}{\epsilon_0}\log\frac{k}{\delta}, L)$ and $L = L \setminus L'$
Set $G = G_{\text{com}}(U) \setminus (V^+_{L'} \cap V^-_{L'})$
Set $P = G \cap (V^+_{L'} \cup V^-_{L'})$
Set $h = h_{G,L'}$ and $c = 0$
while $L \neq \emptyset$ and $c \leq \frac{1}{\epsilon}\log\frac{k+T}{\delta}$ **do**
  Set $x = \text{sample}(1, L)$
  Set $L = L \setminus \{x\}$, and $L' = L' \cup \{x\}$
  if $h(x) \neq \ell(x)$ **then**
    Set $G = G \setminus \text{nn}_{G,P}(x)$
    Set $h = h_{G,L'}$ and $c = 0$
  else
    Set $c = c + 1$
Output: the hypothesis h

in the example x via a path in G. If no such vertex exists, $\text{nn}_{G,S}(x)$ returns the empty set. Using hypotheses of this form ensures that the neighbor variable used to classify an example x is connected to x by a path through indicators, which allows us to propagate its label to the new example. If the example is misclassified, we must have been fooled by a non-indicator.

The number of examples used by Algorithm 2 depends on T, the number of connected components in the commonality graph after removing all non-indicators. Lemma 1 bounds T by the number of compatible hypotheses and is proved in the supplementary material. Theorem 2 bounds the number of examples sufficient for Algorithm 2 to output a low-error hypothesis.

Lemma 1. *Let G be the graph that results from removing all non-indicators from $G_{\text{com}}(U)$, and suppose G is divided into T connected components. If $m_u \geq \frac{2n^2}{\epsilon}\log\frac{n}{\delta}$, then $T \leq \log_2 |C_{D,\chi}(\epsilon)|$ with probability at least $1-\delta$.*

Theorem 2. *For any distribution D over $\{0,1\}^n$ and target concept $h^* \in C$ such that $\chi(h^*, D) = 1$, h^* has at most k non-indicators, and the minimum non-indicator probability is ϵ_0, if $m_u \geq \frac{2n^2}{\epsilon}\log\frac{n}{\delta}$ and*

$$m_l \geq \frac{1}{\epsilon_0}\log\frac{k}{\delta} + \frac{k + \log|C_{D,\chi}(\epsilon)|}{\epsilon}\left[\log\frac{k+\log|C_{D,\chi}(\epsilon)|}{\delta}\right]$$

then with probability at least $1-3\delta$, Algorithm 2 outputs a hypothesis h in polynomial time such that $\mathsf{err}(h) \leq \epsilon$.

Proof. **Generalization Error:** First note that according to the loop exit condition, Algorithm 2 outputs the first hypothesis it encounters that correctly classifies a sequence of at least $\frac{1}{\epsilon}\log\frac{k+T}{\delta}$ i.i.d. examples from D. If $\mathsf{err}(h) > \epsilon$ for some hypothesis h, then the probability that h correctly classifies such a sequence of examples is at most $(1-\epsilon)^{\frac{1}{\epsilon}\log\frac{k+T}{\delta}} \leq \frac{\delta}{k+T}$. Assuming Algorithm 2 updates its hypothesis at most $k+T$ times, a union bound over the $k+T$ hypotheses considered guarantees that with probability at least $1-\delta$, the hypothesis output by Algorithm 2 has error rate at most ϵ. In the remainder of the proof, we will bound the total number of samples required and show that it makes at most $k+T$ updates to its hypothesis.

Mistake Bound: By the definition of ϵ_0, the initial set of m_l labeled examples ensures that with probability at least $1-\delta$ all non-indicators are included in the potential indicator set P, so all variables outside P (call this set Q) are indicators. We will assume such an event holds throughout the remainder of the proof. In particular, this means that any paths through Q must consist entirely of indicators of the same type.

Suppose at some point during the execution of Algorithm 2, the intermediate hypothesis h misclassifies an example x. There are two types of such mistakes to consider. If the variables in x are not connected to any variables in P, then by the above observation, all variables connected to x are indicators of the same type, and in particular, they are indicators of the type corresponding to the label of x. This means that this type of mistake can occur only when h knows of no labeled examples connected to x. Once h is updated to be $h_{G,L'}$ where $x \in L'$, h can make no further mistakes of this type on any examples connected to x. Thus, Algorithm 2 can make at most T mistakes of this type before all components have labeled examples.

The hypothesis $h_{G,L'}$ labels x with the label of the example of L' containing $\text{nn}_{G,P}(x)$. If x is labeled incorrectly, then this must be an example with label opposite that of x. But since the path between $\text{nn}_{G,P}(x)$ and x consists only of vertices not in P, i.e. indicators, we conclude that $\text{nn}_{G,P}(x)$ must be a non-indicator. Algorithm 2 can make at most k mistakes of this type before all non-indicators are removed from G.

Sample Complexity and Running Time: We have shown that after Algorithm 2 makes $k+T$ updates, all non-indicators have been removed from G and all connected components in G contain a variable that has appeared in a labeled example. Since at most $\frac{1}{\epsilon}\log\frac{k+T}{\delta}$ examples can be seen between updates, the total number of labeled examples needed is at most

$$\frac{1}{\epsilon_0}\log\frac{k}{\delta} + \frac{k+T}{\epsilon}\log\frac{k+T}{\delta}.$$

Straightforward algebra and an application of Lemma 1 yields the bound given in the theorem statement, and a union bound over the three failure events of probability δ yields the stated probability of success. The time

complexity is clearly polynomial in n per example and therefore polynomial overall. □

4. Active Learning

We now consider the problem of learning two-sided disjunctions in the active learning model, where the learner has access to a set U of unlabeled examples and an oracle that returns the label of any example in U we submit. The additional power provided by this model allows us to use the same strategy as in the semi-supervised algorithm in Section 3.2 while achieving sample complexity bounds independent of ϵ_0.

As in Section 3.2, the goal will be to identify and remove non-indicators from the commonality graph and obtain labeled examples for each of the connected components in the resulting graph. In the semi-supervised model we could identify a mistake when there was a path connecting a positive labeled example and a negative labeled example. To identify non-indicators we guaranteed that they would lie on the endpoints of these labeled paths. In the active learning setting, we are able to check the labels of examples along this path, and thus are able to remove our dependence on the minimum non-indicator probability.

The algorithm we propose can be seen as a slight modification of Algorithm 2. The idea is to maintain a set of at least one labeled example per connected component and to test the corresponding nearest neighbor hypothesis on randomly chosen examples. If the hypothesis misclassifies some example, it identifies a path from the example to its nearest neighbor. Since these examples have opposite labels, a non-indicator must be present at a point on the path where positive indicators switch to negative indicators, and such a non-indicator can found in logarithmically many queries by actively choosing examples to query along this path in a binary search pattern. The search begins by querying the label of an example containing the variable at the midpoint of the path. Depending on the queried label, one of the endpoints of the path is updated to the midpoint, and the search continues recursively on the smaller path whose endpoints still have opposite labels. Let $\text{BinarySearch}_{G,L}(x)$ return the non-indicator along the path in G from a variable in x to $\text{nn}_{G,L}(x)$. The algorithm halts after removing all k non-indicators or after correctly labeling a long enough sequence of random examples. The details are described in Algorithm 3, and the analysis is presented in Theorem 3.

Theorem 3. *For any distribution D over $\{0,1\}^n$ and target concept $h^* \in C$ such that $\chi(h^*, D) = 1$ and h^* has at most k non-indicators, let T be the number of connected components in the graph G that results from removing all non-indicators from $G_{\text{com}}(U)$. If $m_u \geq \frac{2n^2}{\epsilon} \log \frac{n}{\delta}$ then after at most*

$$m_q = O\left(\log |C_{D,\chi}(\epsilon)| + k\left[\log n + \frac{1}{\epsilon}\log\frac{k}{\delta}\right]\right)$$

label queries, with probability $\geq 1 - 2\delta$, Algorithm 3 outputs a hypothesis h in polynomial time such that $\mathsf{err}(h) \leq \epsilon$.

Algorithm 3 Active Learning Two-Sided Disjunctions

Input: unlabeled data U, parameters ϵ, δ, k
Set $G = G_{\text{com}}(U)$ and $L = \emptyset$
for each connected component R of G **do**
  Choose $x \in U$ such that $x \subseteq R$
  Set $L = L \cup \{(x, \ell(x))\}$
Set $h = h_{G,L}$ and $c = 0$
while $c \leq \frac{1}{\epsilon}\log\frac{k}{\delta}$ **do**
  Set $x = \text{sample}(1, U)$ and $L = L \cup \{(x, \ell(x))\}$
  if $h(x) \neq \ell(x)$ **then**
    Set $v = \text{BinarySearch}_{G,L}(x)$
    Set $G = G \setminus \{v\}$
    for each new connected component R of G **do**
      Choose $x \in U$ such that $x \subseteq R$
      Set $L = L \cup \{(x, \ell(x))\}$
    Set $h = h_{G,L}$ and $c = 0$
  else
    Set $c = c + 1$
Output: the hypothesis h

Proof. **Generalization Error:** According to the exit condition of the loop in Step 3, Algorithm 3 outputs the first hypothesis it encounters that correctly classifies a sequence of at least $\frac{1}{\epsilon}\log\frac{k}{\delta}$ i.i.d. examples from D. If $\mathsf{err}(h) > \epsilon$ for some hypothesis h, then the probability that h correctly classifies such a sequence of examples is at most $(1-\epsilon)^{\frac{1}{\epsilon}\log\frac{k}{\delta}} \leq \frac{\delta}{k}$. Assuming Algorithm 3 updates its hypothesis at most k times, a union bound over the k hypotheses considered guarantees that with probability at least $1 - \delta$, the hypothesis output by Algorithm 3 has error rate at most ϵ. In the remainder of the proof, we will bound the total number of samples required by Algorithm 3 and show that it makes at most k updates to its hypothesis.

Queries per Stage: In the loops over connected components of G, one label is queried for each component. The components are those formed by removing from G a subset of the non-indicators, so the total number of queries made in these loops is at most T, the number of components after removing all non-indicators.

Now suppose the hypothesis h misclassifies an example x. Let x' be the nearest labeled example to x, and let

x_i and x_j be the endpoints of the shortest path from x to x' in G. If each variable along the path appears in examples of only one label, then there could be no path between x_i and x_j, which appear in examples with different labels. Thus, there must exist a variable along the path from x_i to x_j that appears in both positive and negative examples, i.e. a non-indicator. Since the commonality graph was constructed using the examples in U, we can query the labels of examples that contain variables between x_i and x_j in order to find the non-indicator. Using binary search, the number of queries is logarithmic in the path length, which is at most n.

Query Complexity and Running Time: Each mistake results in removing a non-indicator from the G, so at most k mistakes can be made. For each mistake, $O(\log n)$ queries are needed to find a non-indicator to remove and at most $\frac{1}{\epsilon}\log\frac{k}{\delta}$ more queries are used before another mistake is made. Combined with the queries for the connected components, we can bound the total number of queries by $O\left(T + k\left[\log n + \frac{1}{\epsilon}\log\frac{k}{\delta}\right]\right)$. We can further bound T by $\log|C_{D,\chi}(\epsilon)|$ via Lemma 1, and pay the price of an additional δ probability of failure. The running time is clearly polynomial. □

5. Random Classification Noise

We also consider the more challenging problem of learning two-sided disjunctions under random classification noise, where the observed label of each example is reversed from that given by the target concept independently with probability $\alpha \in [0, 1/2)$. The key modification we make to extend our algorithms is to determine the indicator type of each variable by taking a majority vote of the labels of the containing examples. To guarantee success with high probability this scheme is used only for variables that are *significant*, that is, they appear in at least $\tilde{O}(\frac{1}{(1-2\alpha)^2})$ labeled examples. We discuss some details below for the extensions to Algorithms 2 and 3, and provide complete proofs for all three algorithms in the supplementary material.

Semi-supervised Learning: We first draw enough examples to ensure that all non-indicators are significant and then build a hypothesis in a similar manner to the noise-free setting by deciding indicator types via majority voting. We then test the hypothesis on a set of labeled examples and output it if the empirical error is small. Otherwise, the high error may be caused either by a component without any labeled variables (which can be corrected by a majority vote of the examples in that component) or a non-indicator (which can be identified by using enough labeled examples). As in the noise-free setting, this allows us to bound the number of updates to the hypothesis.

Active Learning: Besides using majority voting to determine indicator types, another alteration is to perform binary search over edges (instead of vertices) in order to distinguish disagreement caused by non-indicators from that caused by noise. If an edge contains an indicator, a majority vote of examples containing the two variables in the edge will agree with the type of the indicator, so any variable contained in edges of different labels must be a non-indicator.

6. Discussion

One drawback of our semi-supervised algorithms is that their dependence on the minimum non-indicator probability restricts the class of distributions they can efficiently learn. Additionally, the class of target concepts for which Algorithm 1 can efficiently learn a consistent and compatible hypothesis is restricted, and the reduction given in the supplementary material proves that some such restriction is necessary as the general problem is NP-hard. One surprising result of our work is that both restrictions can be lifted entirely in the active learning setting while improving label complexity at the same time. The ability to adaptively query the labels of examples allows us to execute a strategy for identifying non-indicators that would require too many labeled examples in the semi-supervised setting. While this represents the first known example of how active learning can be used to avert computational difficulties present in semi-supervised learning, it would be worthwhile to give more such examples and to understand more generally when active learning provides this type of advantage.

It is important to note that the problem of learning two-sided disjunctions can be viewed as learning under a large-margin assumption. We can represent a two-sided disjunction h as a linear threshold function $h(x) = \text{sign}(w^\mathsf{T}x)$ where $w_i = +1$ for positive indicators, $w_i = -1$ for negative indicators, and $w_i = 0$ for each of the k non-indicators. If h is fully compatible with the distribution D, it is straightforward to show that for every example $x \sim D$, $\frac{|w^\mathsf{T}x|}{\|w\|_\infty\|x\|_1} \geq \frac{1}{k+1}$, which corresponds to an $L_\infty L_1$ margin of $O(1/k)$. This is a different notion of margin than the L_2L_2 margin appearing in the mistake bounds for the Perceptron algorithm (Rosenblatt, 1958) and the L_1L_∞ margin in the bounds for Winnow (Littlestone, 1988). One interesting area of future work is to provide generic algorithms with bounds depending on the $L_\infty L_1$ margin.

Acknowledgements
This work was supported in part by NSF grant CCF-0953192, AFOSR grant FA9550-09-1-0538, and a Microsoft Research Faculty Fellowship.

References

Balcan, M.-F. and Blum, A. A PAC-style model for learning from labeled and unlabeled data. In *Proceedings of the 18th Annual Conference on Computational Learning Theory (COLT)*, 2005.

Balcan, M. F., Beygelzimer, A., and Langford, J. Agnostic active learning. In *Proceedings of the 23rd International Conference on Machine Learning (ICML)*, 2006.

Balcan, Maria-Florina and Blum, Avrim. A discriminative model for semi-supervised learning. *Journal of the ACM*, 57(3), 2010.

Beygelzimer, A., Hsu, D., Langford, J., and Zhang, T. Agnostic active learning without constraints. In *Advances in Neural Information Processing Systems (NIPS)*, 2010.

Blum, Avrim and Balcan, Maria-Florina. Open problems in efficient semi-supervised PAC learning. In *Proceedings of the 20th Annual Conference on Computational Learning Theory (COLT)*, 2007.

Blum, Avrim and Mitchell, Tom. Combining labeled and unlabeled data with co-training. In *Proceedings of the 11th Annual Conference on Computational Learning Theory (COLT)*, 1998.

Chapelle, O., Schlkopf, B., and Zien, A. *Semi-Supervised Learning*. MIT press, 2006.

Dasgupta, S. Coarse sample complexity bounds for active learning. In *Advances in Neural Information Processing Systems (NIPS)*, 2005.

Dasgupta, S. Active learning. *Encyclopedia of Machine Learning*, 2011.

Dasgupta, S., Littman, M. L., and McAllester, D. PAC generalization bounds for co-training. In *Advances in Neural Information Processing Systems (NIPS)*, 2001.

Dasgupta, Sanjoy, Hsu, Daniel, and Monteleoni, Claire. A general agnostic active learning algorithm. In *Advances in Neural Information Processing Systems (NIPS)*, 2007.

Hanneke, S. A bound on the label complexity of agnostic active learning. In *Proceedings of the 24th International Conference on Machine Learning (ICML)*, 2007a.

Hanneke, S. Teaching dimension and the complexity of active learning. In *Proceedings of the 20th Annual Conference on Computational Learning Theory (COLT)*, 2007b.

Joachims, T. Transductive inference for text classification using support vector machines. In *Proceedings of the 16th International Conference on Machine Learning (ICML)*, 1999a.

Joachims, Thorsten. Transductive inference for text classification using support vector machines. In *Proceedings of the 16th International Conference on Machine Learning (ICML)*, 1999b.

Kaariainen, Matti. Generalization error bounds using unlabeled data. In *Proceedings of the 18th Annual Conference on Computational Learning Theory (COLT)*, 2005.

Koltchinskii, V. Rademacher complexities and bounding the excess risk in active learning. *Journal of Machine Learning*, 11:2457–2485, 2010.

Littlestone, Nick. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning*, 2(4):285–318, 1988.

Rigollet, Philippe. Generalized error bounds in semi-supervised classification under the cluster assumption. *Journal of Machine Learning Research*, 8, 2007.

Rosenberg, David S. and Bartlett, Peter L. The rademacher complexity of co-regularized kernel classes. In *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2007.

Rosenblatt, Frank. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65:386–407, 1958.

Zhu, X., Ghahramani, Z., and Lafferty, J. Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of the 20th International Conference on Machine Learning (ICML)*, 2003.

Zhu, Xiaojin. Semi-supervised learning. *Encyclopedia of Machine Learning*, 2011.