Teaching with Examples in A Real Environment

Gourab Kundu
Department of Computer Science
University of Illinois at Urbana Champaign
Urbana, IL
kundu2@illinois.edu

Dan Roth
Department of Computer Science
University of Illinois at Urbana Champaign
Urbana, IL
danr@illinois.edu

Abstract

Teaching is challenging in a real environment. One problem is that not all examples may be available to teach. We show how to teach several important concept classes namely conjunction, disjunction and linear threshold functions under different characterizations of the domain of available examples. We show that a monotone linear threshold function is teachable using a polynomial number of examples when the accessible domain is defined by the intersection of multiple monotone linear threshold functions. Also, a teacher may not be smart enough to know the target concept exactly but he may be able to provide better examples from available examples. We show how to teach without knowing the target concept exactly and using only available examples. Our experiments on the benchmark data sets of text categorization and movie review classification show that the algorithm **Partial Instance Feedback (PIF)** results in $8-11\%$ error reduction over active learning and $16-18\%$ error reduction over random sampling.

1 Introduction

All teaching models proposed in the computational learning theory literature [1, 2, 3, 4, 5, 6, 7] assume that any example is available to the teacher and the teacher knows the target concept exactly. But these two assumptions are unrealistic in real world. We ask two research questions:

- Is teaching a target concept tractable when the teacher is limited to using a subset of the example space?
- How to teach without knowing the target concept and using only examples from a limited subset of the example space?

To answer question 1, we study two variants of a teaching model where the teacher can draw examples only from a subset of the example space. Our final results show that a monotone linear threshold function is teachable using a polynomial number of examples when the accessible domain is defined by the intersection of multiple monotone linear threshold functions.

To answer question 2, we present an algorithm referred as *Partial Instance Feedback (PIF)* for the teacher to teach using available examples for the tasks of text categorization and movie review classification. Our experimental results demonstrate that PIF outperforms random sampling, compares favorably with the state of the art and when used together with active learning, significantly improves over active learning alone.

There are some works in information retrieval to incorporate feedback from the teacher [8, 9, 10, 11, 12, 13, 14]. Discussions of these methods and comparisons with ours is provided later in Section 6.

1.1 Preliminaries

We consider the task of learning functions of the form $f : \{0,1\}^n \rightarrow \{0,1\}$ where the example space X is the set of all examples with n binary attributes and the function/hypothesis f maps each example to either one of $\{0,1\}$. The learner is a *consistent* learner with the hypothesis space as the space of conjunctions or disjunctions or linear threshold functions. A linear threshold function is characterized by the weight vector $w = (w_1, w_2, \ldots, w_n)$ and a threshold θ and it classifies an example $x \in \{0,1\}^n$ as positive if $(w, x) >= \theta$ where (w, x) is the dot product between w and x. We call a linear threshold function MLTF (monotone linear threshold function) if it has no negative weight, i.e., $\forall i\, w_i >= 0$. A linear threshold function with both positive and negative weights will be referred to as NLTF (Non-monotone linear threshold function).

Definition 1. *If $x \in \{0,1\}^n$, $A(x) = \{i : x_i = 1\}$, that is, $A(x)$ is the set of indices of active variables in x. Similarly we can consider conjunction or disjunction such that if h is a conjunction or disjunction, $A(h)$ is the set of indices of variables in h.*

Let x be an example. If $x \in \{0,1\}^4$ and $x = (1,0,0,1)$, we have $A(x) = \{1,4\}$. Again, given $D = \{1,4\}$, we can use the function $A^{-1}(D)$ to get $x = (1,0,0,1)$. Now consider a conjunction $h = l_1 \wedge l_3 \wedge l_5$, $A(h) = \{1,3,5\}$.

Definition 2. *For Concept Class H and Target Concept h, a Teaching Set is a set of examples S such that the only hypothesis from H consistent with S is h. A minimal teaching set S is a set such that there exists no other teaching set S' with $|S'| < |S|$.*

Definition 3. *An Access Function $f : \{0,1\}^n \rightarrow \{0,1\}$ is a function such that an example x is available to the teacher if and only if $f(x)$ is true.*

Definition 4. *A minimal true instance (resp. maximal false instance) for f is an instance a if there is no true (resp. false) instance b ($b \neq a$) such that $b <= a$ (resp. $a <= b$) where $a <= b$ implies $a_i <= b_i$ for $i = 1, 2, \ldots, n$ where n is the number of variables.*

To avoid confusion, when we say an example e is positive (negative), it will always be w.r.t the target concept h. If an example is positive w.r.t f, we will say it is accessible by f.

2 Is teaching a target concept tractable when the teacher is limited to using a subset of the example space?

We consider two variants of constrained teaching. In one variant referred as *Exact Teaching*, the learner has to learn a concept that is consistent with the target concept over the entire space of examples. For example, consider the case where a teacher is teaching a concept to some kids using examples but wants to avoid inappropriate examples during teaching although the learner might encounter these examples in future. In another variant referred as *Domain Exact Teaching*, the learner has to learn a concept that needs to be consistent with the target concept only over the subset of the example space from where the teacher is drawing the examples. For example, consider the natural language processing domain where each training example comes from a valid sentence and the learner will never see an invalid example, neither during training nor testing.

Let $T(f)$ be the set of true instances for f and $F(f)$ be the set of false instances for f. Let $min\ T(f)$ denote the set of minimal true instances for f and $max\ F(f)$ denote the set of maximal false instances for f. There are existing algorithms in the literature for computing the sets of $min\ T(f)$ and $max\ F(f)$. In [15], the authors presented two algorithms for this task with time $O(nm^2+n^2m)$ and $O(n^3m)$ where $m = |min\ T(f)| + |max\ F(f)|$ and n is the number of variables. Later in [16], a relatively simpler and faster algorithm was presented that took time $O(n^2m)$. All these algorithms were proposed for 2-monotonic positive functions which is a superset of the class of monotonic linear threshold functions.

2.1 Exact Teaching a Monotone Conjunction h when the access function is a MLTF f

The algorithm for producing a teaching set is given in Algorithm 1[1].

[1] Omitted proofs for theorems or lemmas are in the supplementary material

Algorithm 1 Compute Teaching Set for a Monotone Conjunction h when the access function is a MLTF f

1: Compute the set min $T(f)$ for f
2: Present a sequence of positive examples $e_1, e_2, \ldots, e_m$ such that $\cap_{i=1}^{m} A(e_i) = A(h)$ and $A(e_i) = A(D_i) \cup A(h)$ ($\exists D_i \in min\ T(f)$
3: **for** each $g \in A(h)$ **do**
4: $C = \{1, 2, \ldots, n\} - \{g\}$
5: Present $A^{-1}(C)$ as a negative example
6: **end for**

Theorem 2.1. *Algorithm 1 produces a teaching set.*

Lemma 2.2. *If* $(\cap_{D \in min\ T(f)} A(D)) \neq \Phi$*, teaching is not possible.*

2.2 Exact Teaching a Monotone Disjunction h when the access function is a MLTF f

The algorithm for this case is given in Algorithm 2.

Algorithm 2 Compute Teaching Set for a Monotone Disjunction h when the access function is a MLTF f

1: $D = \{1, 2, \ldots, n\} - A(h)$
2: Present $A^{-1}(D)$ as a negative example
3: **for** each $g \in A(h)$ **do**
4: $C = \{g\} \cup D$
5: Present $A^{-1}(C)$ as a positive example
6: **end for**

Theorem 2.3. *Algorithm* 2 *produces a teaching set.*

Lemma 2.4. *If* $(\cap_{D \in min\ T(f)} A(D)) \cap A(h) \neq \Phi$*, teaching is not possible.*

2.3 Exact Teaching a MLTF h when the access function is a MLTF f

To teach a MLTF h, the teacher has to present the sets $min\ T(h)$ and $max\ F(h)$. Exact teaching of h is possible if for every $b \in max\ F(h) \cup min\ T(h)$, b is accessible by f.

3 Domain Exact Teaching a MLTF h when the access function is the intersection of multiple MLTFs $f_1, f_2, \ldots, f_m$

$T(f_1, f_2, \ldots, f_m, h) = \{a \in \{0,1\}^n | h(a) = true\ and\ f_1(a) = true\ and\ f_2(a) = true\ and \ldots f_m(a) = true\}$

$F(f_1, f_2, \ldots, f_m, h) = \{a \in \{0,1\}^n | h(a) = false\ and\ f_1(a) = true\ and\ f_2(a) = true\ and \ldots f_m(a) = true\}$

To teach h by examples that are accessible by the intersection of $f_1, f_2, \ldots, f_m$, the teacher has to provide the sets $min\ T(f_1, f_2, \ldots, f_m, h)$ and $max\ F(f_1, f_2, \ldots, f_m, h)$ since these two sets delineate the boundary between positive and negative examples for h that are accessible by the intersection of $f_1, f_2, \ldots, f_m$.

Lemma 3.1. $max\ F(f_1, f_2, \ldots, f_m, h) = (max\ F(h)) \cap T(f_1) \cap T(f_2) \cap \ldots T(f_m)$.

The algorithm for computing the teaching set is given in Algorithm 3. For input example x positive for h and accessible by the intersection of $f_1, f_2, \ldots, f_m$, Algorithm 4 returns an example y such that y is positive for h and accessible by the intersection of $f_1, f_2, \ldots, f_m$ and y is minimal. This algorithm is a modification of the algorithm for finding prime implicant in [17].

Theorem 3.2. *Algorithm 3 produces a teaching set.*

Algorithm 3 Compute Teaching Set for a MLTF h when the access function is the intersection of m MLTFs $f_1, f_2, \ldots, f_m$

Compute $min\ T(h)$ and $max\ F(h)$
for $i \leftarrow 1\ to\ m$ **do**
  Compute $min\ T(f_i)$ and $max\ F(f_i)$
end for
$P \leftarrow \phi\ \{P = min\ T(f_1, f_2, \ldots, f_m, h)\}$
$Q \leftarrow \phi\ \{Q = max\ F(f_1, f_2, \ldots, f_m, h)\}$
for each $a \in max\ F(h)$ **do**
  if a is accessible by the intersection of $f_1, f_2, \ldots, f_m$ **then**
    $Q \leftarrow Q \cup \{a\}$
  end if
end for
for each tuple $(a, b_1, b_2, \ldots, b_m) \in min\ T(h) \times min\ T(f_1) \times min\ T(f_2) \times \ldots min\ T(f_m)$ **do**
  $c \leftarrow A^{-1}(A(a) \cup A(b_1) \cup A(b_2) \cup \ldots A(b_m))$
  $c \leftarrow Findminimal(c, h, f_1, f_2, \ldots, f_m)$
  $P \leftarrow P \cup \{c\}$
end for

Algorithm 4 $Findminimal(x, h, f_1, f_2, \ldots, f_m)$

for $i \leftarrow 1\ to\ n$ **do**
  if $x_i = 1$ **then**
    $y \leftarrow A^{-1}(A(x) - \{i\})$
    if $h(y) = true$ and $f_1(y) = f_2(y) = \ldots = f_m(y) = true$ **then**
      $x \leftarrow y$
    end if
  end if
end for

Lemma 3.3. *An upper bound of the size of the teaching set is* $|min\ T(h)| * |min\ T(f_1)| * |min\ T(f_2)| * \ldots * |min\ T(f_n)| + |max\ F(h)|$. *This is a tight upper bound.*

The algorithm for computing $min\ T(h)$ and $max\ F(h)$ from [16] has complexity $O(n^2 * |min\ T(h) \cup max\ F(h)|)$. The complexity for computing $min\ T(f_i)$ and $max\ F(f_i)$ is $O(n^2 * |min\ T(f_i) \cup max\ F(f_i)|)$. The for loop in Lines $12 - 16$ of Algorithm 3 runs for $|min\ T(h)| * |min\ T(f_1)| * |min\ T(f_2)| * \ldots |min\ T(f_m)|$ iterations. In each iteration, it invokes Algorithm 4 which takes $O(n^2 m)$ time. So the complexity of the above algorithm is $O(n^2 * (|min\ T(h) \cup max\ F(h))| + |min\ T(f_1) \cup max\ F(f_1)| + |min\ T(f_2) \cup max\ F(f_2)| + \ldots + |min\ T(f_n) \cup max\ F(f_n)|) + n^2 m * (|min\ T(h)| * |min\ T(f_1)| * |min\ T(f_2)| * \ldots * |min\ T(f_n)|) + nm * (|max\ F(h)|))$.

3.1 Interactive Domain Exact Teaching of a MLTF/NLTF when the access function is the intersection of multiple MLTFs

Previous variants of the teaching models assume that the teacher knows the target function exactly which is rarely the case. In reality, teacher may incorporate supervision by creating new informative examples absent in the corpus or by modifying an example to make it more informative to accelerate the learning.

Lemma 3.4. *If the concept class is the class of MLTFs, a positive example q is more informative than a positive example p ($p \neq q$) if $A(q) \subseteq A(p)$ and a negative example q is more informative than a negative example p ($p \neq q$) if $A(q) \supseteq A(p)$ assuming p and q are both accessible to the teacher.*

Definition 5. *If $t \in \{0,1\}^n$ and the target concept w is a NLTF, $A^+(t) = \{i : t_i = 1$ and $w_i > 0\}$, that is, $A^+(t)$ is the set of indices of active features in t that have positive weights in w. Similarly let $A^-(t) = \{i : t_i = 1$ and $w_i < 0\}$, that is, $A^-(t)$ is the set of indices of active features in t that have negative weights in w.*

Lemma 3.5. *If the concept class is the class of NLTFs and if the learner knows the polarity of weight for each feature, a positive example q is more informative than a positive example p ($p \neq q$) if $A^+(q) \subseteq A^+(p)$ and $A^-(q) \supseteq A^-(p)$. A negative example q is more informative than a negative example p ($p \neq q$) if $A^+(q) \supseteq A^+(p)$ and $A^-(q) \subseteq A^-(p)$ assuming p and q are both accessible to the teacher.*

4 Teaching Text Categorization Tasks

A teacher can incorporate supervision by creating a more informative example from a given example following Lemma 3.5. Unfortunately in real setting, the polarity of weight for each feature will not be known. If we assume that positive (resp. negative) examples mostly contain positive (resp. negative) weighted features, then from an example p labeled as positive (resp. negative), a more informative example q can be constructed by taking a subset of the active features in p such that the label of q remains the same as the label of p. For the case when p is a document with label positive (negative), a more informative q will be a fragment of the document p such that q has the same label as p. Moreover, q has to be a chunk of valid sentences, so, the teacher is constrained. We take q to consist of only one sentence, thus simplifying the job of the teacher. Then the algorithm for the teacher (**Partial Instance Feedback (PIF)**) is given in Algorithm 5. Although the teacher does not change the learner, he **assumes** that the learners hypothesis class is the class of linear threshold functions and the feature functions used by the learner are known to him. The assumptions are not unrealistic at least in document classification tasks where linear classifiers are the *de facto* standard and traditional features used are mostly words, bi-grams etc.

5 Experimental Results

We report our experiments on the tasks of text categorization and movie review classification. The standard text categorization data set is the Reuters21578 data set[1] where the task is to assign categories to news articles. We use the ModApte split and evaluate on the 10 most frequent classes as in the previous studies. This split has 9603 training and 3299 testing instances. Since each document

[1] http://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html

Algorithm 5 Partial Instance Feedback

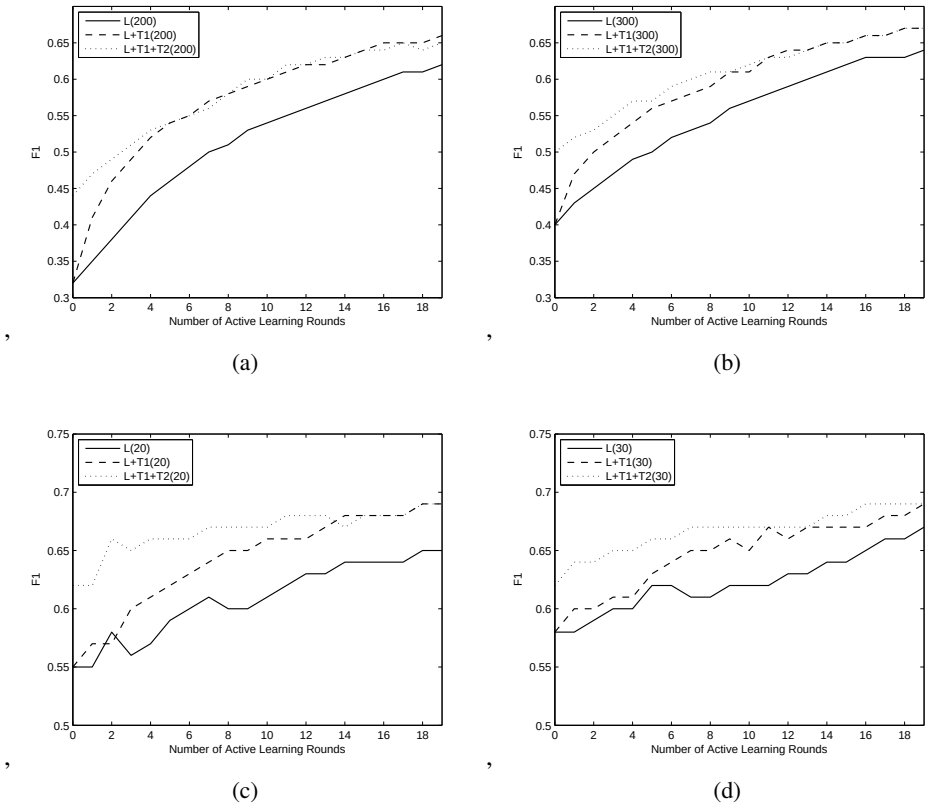
Input: document d
Output: A set of documents D
$D = \phi$
Partial Instance Feedback - Positive Feedback (PIF-PF):
if the document d is labeled as positive **then**
  for each sentence s in d **do**
    if A new document d' containing only s should be labeled as positive **then**
      $D = D \cup \{(d', positive)\}$
    end if
  end for
  $D = D \cup \{(d, positive)\}$
end if
Partial Instance Feedback - Negative Feedback (PIF-NF):
if the document d is labeled as negative **then**
  for each sentence s in d **do**
    if A new document d' containing only s should be labeled as negative **then**
      $D = D \cup \{(d', negative)\}$
    end if
  end for
  $D = D \cup \{(d, negative)\}$
end if

has been assigned multiple categories, we treat this as 10 independent binary classification problems (whether a document is labeled by a particular category or not) and report the macro F1 score. We also experiment on a movie review data set, the Polarity v2.0 data set[2]. Originally this data set has 1000 positive and 1000 negative movie reviews. This data is divided in the format of 10 folds cross validation. We use the first 5 folds for training and the last 5 folds for testing. For both tasks, we treat the documents as bag-of-words and our features are frequencies of the words in a document. As the learning algorithm, we use support vector machine with C value of 10. In our experiments, we found that incorporating supervision for negative examples (PIF-NF) was not beneficial. This might be due to the fact that for text categorization tasks, a document labeled as a category contains relevant words for that category and so these words will be positively correlated to the category or have positive weights. But for a document not labeled as that category, it contains irrelevant words which are not essentially negatively correlated with the category, i.e., these words are not bound to have negative weights. So in our results, we only report results for supervision for positive examples (PIF-PF).

For each data set, we train an oracle weight vector over all training and test documents. This oracle weight vector simulates the teacher. We compare against active learning. For active learning, we randomly select some examples I including at least one positive and one negative example to form the initial classifier. Then a batch of examples B is selected in each of the next n rounds using uncertainty sampling. Our teacher aided active learning also proceeds in the same way except, new documents are created from I and B and provided to the learner. We vary I across experiments. Size of B is taken to be 5 and n is taken to be 20 in all experiments. We run 10 random iterations and average the results.

The experimental results are plotted in Figure 1. As can be seen in both data sets with different size of I, incorporating supervision helps significantly. Supervision helps when the teacher adds new examples from the examples selected by random sampling to form the initial classifier before active learning starts (The difference between the L+T1+T2 curve and the L curve at round 0). Also supervision helps when the teacher adds new examples from the examples selected by uncertainty sampling during active learning rounds (The difference between the L+T1 curve and the L curve). As for example, in Figure $1(a)$, improvement over random sampling is 12% absolute F1. In Figure $1(c)$, improvement over active learning is 4% absolute F1.

[2]http://www.cs.cornell.edu/People/pabo/movie-review-data/



(a) (b)

(c) (d)

Figure 1: Active Learning (L) and Active Learning with feedback only on the examples selected by uncertainty sampling (L + T1) and Active Learning with feedback on both the examples selected by uncertainty sampling and the examples I selected by random sampling for initial formation of the classifier (L + T1 + T2). a) Reuters21578 data set, $|I| = 200$. b) Reuters21578 data set, $|I| = 300$. c) Polarity data set, $|I| = 20$. d) Polarity data set, $|I| = 30$.

Table 1: Results of User Study

Active Learning Round	User1	User2	Uncertainty Sampling	Oracle Teacher
0	8.0	6.0	0.0	0.0
1	7.9	5.0	1.0	0.0
2	9.7	4.0	1.0	27.0
3	33.6	29.0	3.0	32.0
4	25.0	27.0	7.0	27.0

We also compare with [8] which has better performance compared to [18, 13, 9]. On the movie review data set, [8] reported 62% after examining 50 features, PIF reaches 62% after labeling 20 documents. According to [11], labeling a feature is 5 times faster than labeling a document. So [8] has advantage in terms of annotation time. But [8] changed the learning problem to incorporate the prior knowledge whereas our problem is teaching a learner with examples only. Also, as reported in [11], labeling documents is easier for annotators than labeling features. So PIF is supposed to be more annotator friendly. However, this is an informal comparison since training and test splits are not the same and their splits are not available.

In Box 1, we show a document labeled by the category *corn*. Sentence boundaries are marked by square brackets. The teacher (oracle) creates a new document containing the italicized sentence, labels it with the category *corn* and presents it along with the original document.

> [The contract provides for the delivery of 48,000 lbs, plus or minus two pct, of bulk HFCS-55 meeting specified standards regarding its physical and chemical properties.] [CFTC said the exchange plans to begin trading a July 1987 HFCS-55 contract on April 6.] [*CFTC said the soft drink industry currently buys at least 95 pct of all U.S.-produced HFCS-55, a liquid food and beverage sweetener produced through the processing of corn starch by corn refiners.]*

Box 1: Example of Supervision

A small scale user study on the category *crude oil* was conducted to see if humans can emulate the oracle teacher. A interface was created that did active selection of documents, split each positive document into sentences and presented one sentence at a time. It was possible to ignore some sentences or even documents and proceed. One author of the paper and a undergraduate student of the computer science department marked the sentences that were eligible to be positive documents of the category *crude oil*. The F1 scores are shown in Table 1.

6 Related Work

In [1], the authors introduce the classical teaching model. They define the teaching dimension of a hypothesis class as the size of the smallest set of examples that can uniquely identify any target hypothesis in the class. They provide the upper and lower bounds of teaching dimension for several concept classes. In that work, they outlined procedures for teaching conjunction or monomial in the unconstrained setting. In [2], the authors analyze the concept of specification number which is the minimal number of examples needed to specify a particular target concept. The concept class studied in [2] is the class of linearly separable boolean functions. They find the highest, lowest and average specification numbers for this concept class. In [3], the authors argue that teachability and learnability are very related. They present an algorithm to construct a specifying set of examples for a linear threshold function with boolean weights, i.e., weights can by only 0 or 1. These works are related to our work since we also study the teachability of the concept classes of conjunctions or linear threshold functions. The difference is the constrained teacher assumption. In [4], the authors assume that the learning algorithm is known to the teacher. Specifically they consider the cases where the learner is using the nearest-neighbor classification algorithm and the teacher is trying to teach various geometric concepts. We assume that the teacher does not know the algorithm used by

the learner. In [5], several variants of the classical teaching models are studied. In one variant, the learner chooses the consistent hypothesis with the least complexity. In another variant, the learner assumes that the teacher is optimal and eliminates hypothesis that have teaching sets with smaller size than the number of examples presented by the teacher. This idea is further extended in [6] where a learner eliminates hypotheses whose teaching sets do not include an example provided by the teacher even if the example is consistent with the hypothesis. We do not make any such assumption of the learner and treat it as a black box. In [7], the authors study how to teach by a limited number of examples, specially less than the teaching dimension. Our work differs since we do not assume that the number of examples that can be drawn is limited, rather we assume that examples drawn must come from a limited domain.

In information retrieval, there have been a lot of work on incorporating supervision from the teacher to help the learner learn quickly. In [11], the authors propose an interactive active learning algorithm where the active learner asks for a label on a document, the teacher provides the label and shows the relevant features in the document, so that these features get scaled differently than other features, resulting in higher weights for these features. Compared to them, our teacher provides supervision at the level of the sentence, not at the level of the feature. As found in [11], annotators find labeling features more difficult than labeling documents. Humans tend to operate in the level of sentences instead of the level of features that machines operate on. Moreover, annotators need not be familiar with machine learning, in this case, they will find it difficult to label features. In [8], the authors use generalized expectation criteria to design a learning algorithm where the active learner asks for labels on features rather than on instances. In [13], the authors use hand crafted rules to soft label data and then modified AdaBoost to choose weak learners that both fit the labeled training data and the soft labeled data. In [9], a generalization of support vector machine is trained over soft labeled data and additional labeled data. In [14], a support vector machine formulation is presented that can incorporate knowledge about ranked features (a feature to get higher weight than another feature) and relationships about between sets of features. Our work differs from them since they changed the learning problem which may not be possible in all scenarios. It may happen that the learner is a black box to the teacher and it is possible to teach only by examples. In [10], a teacher is asked for a set of representative words for each class and then EM algorithm is used to build a classifier from these words and the unlabeled data set. Our work differs since supervision is at the level of the sentences and it is incorporated in an interactive manner during learning rounds instead of specifying it in the outset in the form of keywords or rules.

The psychological aspects of learning and teaching has been studied in cognitive science, for example, in [19, 20, 21, 22]. In [20], the author studies learning from only positive examples under the assumption that out of all consistent hypotheses, smaller hypothesis quickly become more probable than larger hypotheses. Compared to his probabilistic framework, our teaching framework is based on an exact identification of a particular target concept and so both positive and negative examples are needed. In [19], the authors show that in pedagogical situations, the learner assumes that the teacher will provide helpful examples and the teacher is assumed to be aware of this fact. Compared to them, our learner and teacher do not make any such assumption. From the cognitive perspective, our theoretical results show that when the teacher is constrained to use only a subset of examples, teaching becomes harder but not intractable since the number of examples to teach remain bounded by polynomial in terms of sizes of the minimal true and maximal false examples of the target concept and the access function.

Active learning is a hot research area [23, 24, 25] (see [26] for a good survey). Pool based active learning remedies the problem that queries generated by the learner may be difficult to label. Our method augments this pool based active learning by incorporating teacher supervision. A teacher can give a partial example which an active learner cannot query since it is not in the corpus. As shown in the experimental section, this supervision can give significant improvements.

7 Conclusion

In this paper, we tried to bridge a connection of the computational teaching theory to the real application scenarios. To this end, we show that teaching is possible theoretically when the teacher is constrained so that he can draw only examples from a limited domain. Then using insights from the theory, we devise an algorithm for incorporating supervision in the task of text categorization.

The new algorithm outperforms active learning on two benchmark data sets. As pointed in [27], the label complexity of active learning can be bounded by the extended teaching dimension. It can be investigated in future if incorporating supervision from the teacher can reduce the label complexity of active learning. Another interesting direction of study can be made to see if combining the term or feature level feedback studied in the information retrieval literature with partial instance feedback can further reduce the complexity of active learning in text categorization tasks.

References

[1] Sally A. Goldman and Michael J. Kearns. On the complexity of teaching. *Journal of Computer and System Sciences*, 50:20–31, 1992.

[2] Martin Anthony, Graham Brightwell, and John Shawe-Taylor. On specifying boolean funcions by labelled examples. *Discrete Applied Math*, 61:1–25, 1991.

[3] Ayumi Shinohara and Satoru Miyano. Teachability in computational learning. *New Generation Computing*, 8:337–347, 1991.

[4] Steven Salzberg, Arthur Delcher, David Heath, and Simon Kasif. Learning with a helpful teacher. In *IJCAI*, pages 705–711, 1991.

[5] Frank J. Balbach. Measuring teachability using variants of the teaching dimension. *Theoretical Computer Science*, 397:94–113, 2008.

[6] Sandra Zilles, Steffen Lange, Robert C. Holte, and Martin Zinkevich. Models of cooperative teaching and learning. *Journal of Machine Learning Research*, 12:349–384, 2011.

[7] Hayato Kobayashi and Ayumi Shinohara. Complexity of teaching by a restricted number of examples. In *COLT*, 2009.

[8] Gregory Druck, Gideon Mann, and Andrew McCallum. Learning from labeled features using generalized expectation criteria. In *SIGIR*, 2008.

[9] Xiaoyun Wu and Rohini Srihari. Incorporating prior knowledge with weighted margin support vector machines. In *SIGKDD*, 2004.

[10] Bing Liu, Xiaoli Li, Wee Sun Lee, and Philip S. Yu. Text classification by labeling words. In *AAAI*, 2004.

[11] Hema Raghavan, Omid Madani, and Rosie Jones. Active learning on both features and instances. *Journal of Machine Learning Research*, 7:16551686, 2006.

[12] Gregory Druck, Burr Settles, and Andrew McCallum. Active learning by labeling features. In *EMNLP*, 2009.

[13] Robert E. Schapire, Marie Rochery, Mazin Rahim, and Narendra Gupta. Incorporating prior knowledge into boosting. In *ICML*, 2002.

[14] Kevin Small, Byron C. Wallace, Carla E. Brodly, and Thomas A. Trikalinos. The constrained weight space svm: Learning with ranked features. In *ICML*, 2011.

[15] Endre Boros, Peter L. Hammer, Toshihide Ibaraki, and Kazuhiko Kawakami. Identifying 2-monotonic positive boolean functions in polynomial time. *SIAM Journal of Computing*, 26:93–109, 1997.

[16] Kazuhisa Makino and Toshihide Ibaraki. A fast and simple algorithm for identifying 2-monotonic positive boolean functions. *Journal of Algorithms*, 26:291–305, 1998.

[17] Dana Angluin. Queries and concept learning. *Machine Learning*, 1988.

[18] Hema Raghavan and James Allan. An interactive algorithm for asking and incorporating feature feedback into support vector machines. In *SIGIR*, 2007.

[19] Patrick Shafto and Noah Goodman. Teaching games: Statistical sampling assumptions for pedagogical situations. In *Proceedings of the 30th annual conference of the Cognitive Science Society*, 2008.

[20] Joshua B. Tenenbaum. Bayesian modeling of human concept learning. In *NIPS*, 1998.

[21] György Gergely, Katalin Egyed, and Ildikó Király. On pedagogy. *Developmental Science*, 10:139–146, 2007.

[22] Gergely Csibra. Teachers in the wild. *Trends in Cognitive Sciences*, 11:95–96.

[23] Maria-Florina Balcan, Alina Beygelzimer, and John Langford. Agnostic active learning. *Journal of Computer and System Sciences*, 75:78–89, 2009.

[24] Sanjoy Dasgupta, Daniel Hsu, and Claire Monteleoni. A general agnostic active learning algorithm. In *NIPS*, 2007.

[25] Simon Tong and Daphne Koller. Support vector machine active learning with applications to text classification. *Journal of Machine Learning Research*, pages 45–66, 2001.

[26] Burr Settles. Active learning literature survey. Technical Report 1648, University of WisconsinMadison Computer Sciences Department, January 2010.

[27] Steven Hanneke. Teaching dimension and the complexity of active learning. In *COLT*, 2007.