Positive and Unlabeled Examples Help Learning*

Francesco De Comité, François Denis, Rémi Gilleron, and Fabien Letouzey

LIFL, URA 369 CNRS, Université de Lille 1
59655 Villeneuve d'Ascq FRANCE
{decomite, denis, gilleron, letouzey}@lifl.fr

Abstract. In many learning problems, labeled examples are rare or expensive while numerous unlabeled and positive examples are available. However, most learning algorithms only use labeled examples. Thus we address the problem of learning with the help of positive and unlabeled data given a small number of labeled examples. We present both theoretical and empirical arguments showing that learning algorithms can be improved by the use of both unlabeled and positive data. As an illustrating problem, we consider the learning algorithm from statistics for monotone conjunctions in the presence of classification noise and give empirical evidence of our assumptions. We give theoretical results for the improvement of Statistical Query learning algorithms from positive and unlabeled data. Lastly, we apply these ideas to tree induction algorithms. We modify the code of C4.5 to get an algorithm which takes as input a set LAB of labeled examples, a set POS of positive examples and a set UNL of unlabeled data and which uses these three sets to construct the decision tree. We provide experimental results based on data taken from UCI repository which confirm the relevance of this approach.

Key words: PAC model, Statistical Queries, Unlabeled Examples, Positive Examples, Decision Trees, Data Mining

1 Introduction

Usual learning algorithms only use labeled examples. But, in many machine learning settings, gathering large sets of unlabeled examples is easy. This remark has been made about text classification tasks and learning algorithms able to classify text from labeled and unlabeled documents have recently been proposed ([BM98], [NMTM98]). We also argue that, for many machine learning problems, a "natural" source of positive examples (that belong to a single class) is available and positive data are abundant and cheap. For example consider a classical domain, such as the diagnosis of diseases: unlabeled data are abundant (all patients); positive data may be numerous (all the patients who have the disease); but, labeled data are rare if detection tests for this disease are expensive. As a

* This research was partially supported by "Motricité et Cognition" : Contrat par objectifs région Nord/Pas-de-Calais

O. Watanabe, T. Yokomori (Eds.): ALT'99, LNAI 1720, pp. 219–230, 1999.

second example, consider mailing for a specific marketing action: unlabeled data are all the clients in the database; positive data are all the clients who asked information about the product concerned by the marketing action before the mailing was done; but, labeled data are rare and expensive because a survey has to be done on a part of the database of all clients. We do not address text classification problems in the present paper, but they are concerned too: for a web-page classification problem, unlabeled web-pages can be inexpensively gathered, a set of web pages you are interested in is available in your bookmarks, labeled web-pages are fairly expensive but a small set of hand labeled web-pages can be designed.

It has been proved in [Den98] that many concepts classes, namely those which are learnable from statistical queries, can be efficiently learned in a PAC framework using positive and unlabeled data only. But the price to pay is an increase in the number of examples needed to achieve learning (although it remains of polynomial size). We consider the problem of learning with a small set of labeled examples, a set of positive examples and a large set of unlabeled examples. We assume that unlabeled examples are drawn according to some hidden distribution D, that labeled examples are drawn according to the standard example oracle $EX(f, D)$, and that positive examples are drawn according to the oracle $EX(f, D_f)$ where D_f is the distribution D restricted to positive examples. The reader should note that our problem is different from the problem of learning with imbalanced training sets (see [KM97]) because we use three sources of examples. In the method we discuss here, labeled examples are only used to estimate the target weight (the proportion of positive examples among all examples); therefore, if an estimate of the target weight is available for the problem, only positive and unlabeled data are needed. We present experimental results showing that unlabeled data and positive data can efficiently boost accuracy of the statistical query learning algorithm for monotone conjunctions in the presence of classification noise. Such boosting can be explained by the fact that SQ algorithms are based on the estimate of probabilities. We prove that these estimates could be replaced by: an estimate of the weight of the target concept with respect to (w.r.t.) the hidden distribution using the (small) set of labeled examples and estimates of probabilities which can be computed from positive and unlabeled data only. If the sets of unlabeled and positive data are large enough, all estimates can be calculated within the accuracy of the estimate of the weight of the target concept. We present theoretical arguments in the PAC framework showing that a gain in the size of the query space (or its VC dimension) can be obtained on the number of labeled examples. But as usual, the results could be better for real problems.

In the last section of the paper, we consider standard methods of decision tree induction and examine the commonly used C4.5 algorithm described in [Qui93]. In this algorithm, when refining a leaf into an internal node, the decision criterion is based on statistical values. Therefore, C4.5 can be seen as a statistical query algorithm and the above ideas can be applied. We adapt the code of C4.5. Our algorithm takes as inputs three sets: a set of labeled examples, a set of positive examples and a set of unlabeled examples. The information gain criterion used by C4.5 is modified such that the three sets are used. The reader should note that labeled examples are used only once for the computation of the weight of the target concept under the hidden distribution. We provide some promising experimental results, but further experiments are needed for an experimental validation of our approach.

2 Preliminaries

2.1 Basic Definitions and Notations

For each $n \geq 1$, X_n denotes an instance space on n attributes. A concept f is a subset of some instance space X_n or equivalently a $\{0,1\}$-valued function defined on X_n. For each $n \geq 1$, let $\mathcal{C}_n \subset 2^{X_n}$ be a set of concepts. Then $\mathcal{C} = \bigcup \mathcal{C}_n$ denotes a concept class over $X = \bigcup X_n$. The *size* of a concept f is the size of a smallest representation for a given representation scheme. An *example* of a concept f is a pair $\langle x, f(x) \rangle$, which is *positive* if $f(x) = 1$ and *negative* otherwise. We denote by $Pos(f)$ the set of all x such that $f(x) = 1$. If D is a distribution defined over X and if A is a subset of the instance space X, we denote by $D(A)$ the probability of the event $[x \in A]$ and we denote by D_A the induced distribution. For instance, if f is a concept over X such that $D(f) \neq 0$, $D_f(x) = D(x)/D(f)$ if $x \in Pos(f)$ and 0 otherwise. We denote by $\overline{f}$ the complement of the set f in X and $f \Delta g$ the symmetric difference between f and g. A *monotone conjunction* is a conjunction of boolean variables . For each $x \in \{0,1\}^n$, we use the notation $x(i)$ to indicate the ith bit of x. If V is a subset of $\{x_1, \ldots, x_n\}$, the conjunction of variables in V is denoted by $\Pi_{x_i \in V} x_i$. If $V = \emptyset$, $\Pi_{x_i \in V} x_i = 1$.

2.2 PAC and SQ Models

Let f be a target concept in some concept class $\mathcal{C}$. Let D be the hidden distribution defined over X. In the PAC model [Val84], the learner is given access to an example oracle $EX(f, D)$ which returns at each call an example $\langle x, f(x) \rangle$ drawn randomly according to D. A concept class $\mathcal{C}$ is *PAC learnable* if there exist a learning algorithm L and a polynomial $p(.,.,.,.)$ with the following property: for any $f \in \mathcal{C}$, for any distribution D on X, and for any $0 < \epsilon < 1$ and $0 < \delta < 1$, if L is given access to $EX(f, D)$ and to inputs ϵ and δ, then with probability at least $1 - \delta$, L outputs a hypothesis concept h satisfying $error(h) = D(f \Delta h) \leq \epsilon$ in time bounded by $p(1/\epsilon, 1/\delta, n, size(f))$.

The SQ-model [Kea93] is a specialization of the PAC model in which the learner forms its hypothesis solely on the basis of estimates of probabilities. A *statistical query* over X_n is a mapping $\chi : X_n \times \{0,1\} \rightarrow \{0,1\}$ associated with a tolerance $0 < \tau \leq 1$. In the SQ-model the learner is given access to a statistics oracle $STAT(f, D)$ which, at each query (χ, τ), returns an estimate of $D(\{x \mid \chi(\langle x, f(x) \rangle) = 1\})$ within accuracy τ. Let $\mathcal{C}$ be a concept class over X. We say that $\mathcal{C}$ is *SQ-learnable* if there exist a learning algorithm L and polynomials

$p(.,.,.), q(.,.,.)$ and $r(.,.,.)$ with the following property: for any $f \in \mathcal{C}$, for any distribution D over X, and for any $0 < \epsilon < 1$, if L is given access to $STAT(f, D)$ and to an input ϵ, then, for every query (χ, τ) made by L, the predicate χ can be evaluated in time $q(1/\epsilon, n, size(f))$, and $1/\tau$ is bounded by $r(1/\epsilon, n, size(f))$, L halts in time bounded by $p(1/\epsilon, n, size(f))$ and L outputs a hypothesis $h \in \mathcal{C}$ satisfying $D(f \Delta h) \leq \epsilon$.

It is clear that given access to the example oracle $EX(f, D)$, it is easy to simulate the statistics oracle $STAT(f, D)$ drawing a sufficiently large set of labeled examples. This is formalized by the following result:

Theorem 1. *[Kea93] Let $\mathcal{C}$ be a class of concepts over X. Suppose that $\mathcal{C}$ is SQ learnable by algorithm L. Then $\mathcal{C}$ is PAC learnable, and furthermore:*

- *If L uses a* finite *query space $\mathcal{Q}$ and α is a lower bound on the allowed approximation error for every query made by L, the number of calls of $EX(f, D)$ is $O(1/\alpha^2 \log |\mathcal{Q}|/\delta)$*
- *If L uses a query space $\mathcal{Q}$ of* finite VC dimension *d and α is a lower bound on the allowed approximation error for every query made by L, the number of calls of $EX(f, D)$ is $O(d/\alpha^2 \log 1/\delta)$*

The reader should note that this result has been extended to white noise PAC models: the Classification Noise model of Angluin and Laird [AL88]; the Constant Partition Classification Noise Model [Dec97]. The proofs may be found in [Kea93] and [Dec97]. Also note that almost all the concept classes known to be PAC learnable are SQ learnable and are therefore PAC learnable with classification noise.

3 Learning Monotone Conjunctions in the Presence of Classification Noise

In this section, the target concept is a monotone conjunction over $\{x_1, \ldots, x_n\}$. In the noise free case, a learning algorithm for monotone conjunctions is:

```
Learning Monotone Conjunctions - Noise Free Case
input: $\epsilon$, $\delta$
  $V = \emptyset$
  Draw a sample $S$ of $m(\epsilon, \delta)$ examples
  for i=1 to n do
    if for every positive example $\langle x, 1 \rangle$, $x(i) = 1$ then $V \leftarrow V \cup \{x_i\}$
output: $h = \Pi_{x_i \in V} x_i$
```

It can be proved that $O\left(1/\epsilon \log 1/\delta + n/\epsilon\right)$ examples are enough to guarantee that the hypothesis h output by the learning algorithm has error less than ϵ with confidence at least $1 - \delta$. The given algorithm is not noise-tolerant. In the presence of classification noise, it is necessary to compute an estimate $\widehat{p_1}(x_i = 0)$ of $p_1(x_i = 0)$ which is the probability that a random example according to the hidden distribution D is positive and satisfies $x(i) = 0$. Then only variables such that this estimate is small enough are included in the output hypothesis. Let us suppose that examples are drawn according to a noisy oracle which, on each call, first draws an instance x according to D together with its correct label and then flips the label with probability $0 \leq \eta < 1/2$. Let us suppose that the noise rate η is known, then we can consider the following learning algorithm of monotone conjunctions from statistics in the presence of classification noise:

```
**Learning Monotone Conjunctions - Noise Tolerant Case**
input: $\epsilon$, $\delta$ , $\eta$
  $V = \emptyset$
  Draw a sample $S$ of $m(\epsilon, \delta, \eta)$ examples
  the size of $S$ is sufficient to ensure that the following estimates
  are accurate to within $\epsilon/(2n)$ with a confidence greater that $1 - \delta$
  for i=1 to n do
    compute an estimate $\widehat{p_1}(x_i = 0)$ of $p_1(x_i = 0)$
    if $\widehat{p_1}(x_i = 0) \leq \epsilon/(2n)$ then $V \leftarrow V \cup \{x_i\}$
output: $h = \Pi_{x_i \in V} x_i$
```

If the noise rate is not known, we can estimate it with techniques described in [AL88]. We do not consider that case because we want to show the best expected gain.

Let $q_1(x_i = 0)$ (resp. $q_0(x_i = 0)$) be the probability that a random example according to the noisy oracle is positive (resp. negative) and satisfies $x(i) = 0$. We have:

$$p_1(x_i = 0) = \frac{(1 - \eta)q_1(x_i = 0) - \eta q_0(x_i = 0)}{1 - 2\eta} \tag{1}$$

Thus we can estimate $p_1(x_i = 0)$ using estimates of $q_1(x_i = 0)$ and $q_0(x_i = 0)$. Then simple algebra and standard Chernoff bound may be used to prove that $O\left[(n^2 \log n)/(\epsilon^2(1 - 2\eta)^2) \log 1/\delta\right]$ examples are sufficient to guarantee that the hypothesis h output by the learning algorithm has error less than ϵ with confidence at least $1 - \delta$. The reader should note that this bound is quite larger than the noise free case one. We now make the assumption that labeled examples are rare, but that sources of unlabeled examples and positive examples are available to the learner. Unlabeled examples are drawn according to D. A noisy positive oracle, on each call, draws examples from the noisy oracle until it gets one with label 1.

We raise the following problems:

- How can we use positive and unlabeled examples in the previous learning algorithm?
- What could be the expected gain?

In the learning algorithm of conjunctions from statistics with noise, an estimate of $p_1(x_i = 0)$ is calculated using (1). From usual formulas for conditional probabilities, $q_1(x_i = 0)$ may be expressed as the probability $q(1)$ that a labeled example is positive according to the noisy oracle times the probability $q_f(x_i = 0)$ that a positive example (drawn according to the positive noisy oracle) satisfies $x(i) = 0$. Now, using the formula for probabilities of disjoint events, $q_0(x_i = 0)$ is equal to the probability $q(x_i = 0)$ that an unlabeled example (drawn according to D) satisfies $x(i) = 0$ minus $q_1(x_i = 0)$. Thus, to compute an estimate of $p_1(x_i = 0)$, we use the following equations: $q_1(x_i = 0) = q(1) \times q_f(x_i = 0)$, $q_0(x_i = 0) = q(x_i = 0) - q_1(x_i = 0)$ and $p_1(x_i = 0) = [(1-\eta)q_1(x_i = 0) - \eta q_0(x_i = 0)]/(1-2\eta)$. Consequently, to compute estimates of $p_1(x_i = 0)$ for all i, we have to compute an estimate of $q(1)$ with labeled examples, estimates of $q_f(x_i = 0)$ for every i using the source of positive examples, and compute estimates of $q_0(x_i = 0)$ for every i using the source of unlabeled examples. The reader should note that labeled examples are used *only once* for the calculation of an estimate of the probability that a labeled example is positive. Thus, we have given a positive answer to our first question: unlabeled examples and positive examples can be used in the learning algorithm of conjunctions from statistics. We now raise the second question, that is: what could be the expected gain? We give below an experimental answer to these questions.

We compare three algorithms:

- The first one is the learning algorithm of conjunctions from statistics where only labeled examples are used
- The second one computes an estimate of $q(1)$ from labeled examples and uses exact values for $q_f(x_i = 0)$ and $q_0(x_i = 0)$. That amounts to say that an *infinite* pool of positive and unlabeled data is available
- The third one computes an estimate of $q(1)$ from labeled examples and estimates of $q_f(x_i = 0)$ and $q_0(x_i = 0)$ from a *finite* number of positive and unlabeled examples

Each of these three algorithms outputs an ordered list $V = (x_{\sigma(1)}, \dots, x_{\sigma(n)})$ of variables such that, for each i, $\widehat{p_1}(x_{\sigma(i)} = 0) \leq \widehat{p_1}(x_{\sigma(i+1)} = 0)$. For a given ordered list V, and for each i, we define $g_i(V) = \Pi_{j \leq i} x_{\sigma(j)}$. The minimal error of an ordered list V is defined as $error_{min}(V) = min\{error(g_i(V)) \mid 0 \leq i \leq n\}$ which is the least error rate we can hope. We compare the minimal errors for the three algorithms. First, let us make more precise these three algorithms. We recall that labeled examples are drawn from a noisy oracle, that positive examples are drawn from the noisy oracle restricted to positive examples, and that the noise rate η is known.

```
Algorithm $L(LAB_N)$
input: a sample $LAB$ of $N$ labeled examples
   for i=1 to n do
      $\widehat{q_1}(x_i = 0) = |\{\langle x, c\rangle \in LAB \mid x(i) = 0 \wedge c = 1\}|/N$
      $\widehat{q_0}(x_i = 0) = |\{\langle x, c\rangle \in LAB \mid x(i) = 0 \wedge c = 0\}|/N$
      $\widehat{p_1}(x_i = 0) = \frac{(1-\eta)\widehat{q_1}(x_i=0)-\eta\widehat{q_0}(x_i=0)}{1-2\eta}$
output: ordered list $V = (x_{\sigma(1)}, \dots, x_{\sigma(n)})$
```

```
Algorithm $L(LAB_N, POS_\infty, UNL_\infty)$
input: a sample $LAB$ of $N$ labeled examples
    $\widehat{q}(1) = |\{\langle x, c\rangle \in LAB \mid c = 1\}|/N$
    for i=1 to n do
        compute exactly $q_f(x_i = 0)$
        compute exactly $q(x_i = 0)$
        $\widehat{q_1}(x_i = 0) = \widehat{q}(1) \times q_f(x_i = 0)$
        $\widehat{q_0}(x_i = 0) = q(x_i = 0) - \widehat{q_1}(x_i = 0)$
        $\widehat{p_1}(x_i = 0) = \frac{(1-\eta)\widehat{q_1}(x_i=0)-\eta\widehat{q_0}(x_i=0)}{1-2\eta}$
output: ordered list $V = (x_{\sigma(1)}, \dots, x_{\sigma(n)})$
```

```
Algorithm $L(LAB_N, POS_M, UNL_M)$
input: a sample $LAB$ of $N$ labeled examples,
        a sample $POS$ of $M$ positive examples
        and a sample $UNL$ of $M$ unlabeled examples
    $\widehat{q}(1) = |\{\langle x, c\rangle \in LAB \mid c = 1\}|/N$
    for i=1 to n do
        $\widehat{q_f}(x_i = 0) = |\{\langle x, c\rangle \in POS \mid x(i) = 0\}|/M$
        $\widehat{q}(x_i = 0) = |\{x \in UNL \mid x(i) = 0\}|/M$
        $\widehat{q_1}(x_i = 0) = \widehat{q}(1) \times \widehat{q_f}(x_i = 0)$
        $\widehat{q_0}(x_i = 0) = \widehat{q}(x_i = 0) - \widehat{q_1}(x_i = 0)$
        $\widehat{p_1}(x_i = 0) = \frac{(1-\eta)\widehat{q_1}(x_i=0)-\eta\widehat{q_0}(x_i=0)}{1-2\eta}$
output: ordered list $V = (x_{\sigma(1)}, \dots, x_{\sigma(n)})$
```

Now, we describe experiments and experimental results[1]. The concept class is the class of monotone conjunctions over n variables $x_1, \dots, x_n$ for some n. The target concept is a conjunction containing five variables. The class $\mathcal{D}$ of distributions is defined as follows: $D \in \mathcal{D}$ is characterized by a tuple $(\rho_1, \dots, \rho_n) \in [0, \rho]^n$ where $\rho = 2(1 - 2^{-\frac{1}{5}}) \simeq 0.26$; for a given $D \in \mathcal{D}$, all values $x(i)$ are selected independently of each other, and $x(i)$ is set to 0 with probability ρ_i and set to 1 with probability $1 - \rho_i$. Note that ρ has been chosen such that, if each ρ_i is drawn randomly and independently in $[0, \rho]$, the average weight of the target concept f w.r.t. D is 0.5. We suppose that examples are drawn accordingly to noisy oracles where the noise rate η is set to 0.2.

Experiment 1. The number n of variables is set to 100. We compare the averages of minimal errors for algorithms $L(LAB_N)$ and $L(LAB_N, POS_\infty, UNL_\infty)$ as functions of the number N of labeled examples. For a given N, averages of minimal errors for an algorithm are obtained doing k times the following:

- f is a randomly chosen conjunction of five variables
- D is chosen randomly in $\mathcal{D}$ by choosing randomly and independently each ρ_i
- N examples are drawn randomly w.r.t. D, they are labeled according to the target concept f, then the correct label is flipped with probability η
- Minimal errors for $L(LAB_N)$ and $L(LAB_N, POS_\infty, UNL_\infty)$ are computed

and then compute the averages over the k iterations. We set k to 100. The results can be seen in Fig. 1. The top plot corresponds to $L(LAB_N)$ and the bottom plot to $L(LAB_N, POS_\infty, UNL_\infty)$.

Experiment 2. We now consider a more realistic case: there are M positive examples and an equal number of unlabeled examples. We show that together with a small number N of labeled examples, these positive and unlabeled examples give about as much information as do M labeled examples alone. We compare the averages of minimal errors for algorithms $L(LAB_N, POS_M, UNL_M)$ and $L(LAB_M)$ as functions of M. The number n of variables is set to 100. The number N of labeled examples is set to 10. The results can be seen in Fig. 1.

[1] Sources and scripts can be found at `ftp://grappa.univ-lille3.fr/pub/Softs/posunlab`.

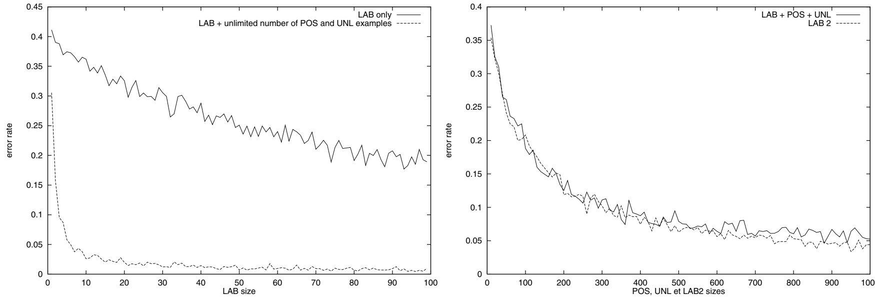

Fig. 1. results of experiments 1 and 2. For experiment 1: target size = 5; 100 variables; 100 iterations. This figure shows the gain we can expect using free positive and unlabeled data. For experiment 2: target size = 5; 100 variables; 100 iterations; $size(LAB) = 10$; $size(POS) = size(UNL) = size(LAB2) = M$; M ranges from 10 to 1000 by step 10. These curves show that with only 10 labeled examples, the learning algorithm performs almost as well with M positive and M unlabeled examples as with M labeled examples.

4 Theoretical Framework

Let $\mathcal{C}$ be a class of concepts over X. Suppose that $\mathcal{C}$ is SQ learnable by some algorithm L. Let f be the target concept and let us consider a statistical query χ made by L. The statistics oracle $STAT(f, D)$ returns an estimate $\widehat{D}_\chi$ of $D_\chi = D(\{x \mid \chi(\langle x, f(x)\rangle) = 1\})$ within some given accuracy. We may write: $D_\chi = D(\{x \mid \chi(\langle x, 1\rangle) = 1 \wedge f(x) = 1\}) + D(\{x \mid \chi(\langle x, 0\rangle) = 1 \wedge f(x) = 0\}) = D(\{x \mid \chi(\langle x, 1\rangle) = 1\} \cap f) + D(\{x \mid \chi(\langle x, 0\rangle) = 1\} \cap \overline{f}) = D(A \cap f) + D(B \cap \overline{f})$ where the sets A and B are defined by: $A = \{x \mid \chi(\langle x, 1\rangle) = 1\}$, $B = \{x \mid \chi(\langle x, 0\rangle) = 1\}$. Furthermore, let A be any subset of the instance space X and f be a concept over X, we have $D(A \cap f) = D_f(A) \times D(f)$ and $D(A \cap \overline{f}) = D(A) - D(A \cap f)$. From the preceding equations, we obtain that $D_\chi = D(f) \times (D_f(A) - D_f(B)) + D(B)$.

Now, in order to estimate D_χ, it is sufficient to estimate $D(f)$, $D_f(A)$, $D_f(B)$ and $D(B)$. If we get an estimate of $D(f)$ within accuracy α and estimates of $D_f(A)$, $D_f(B)$ and $D(B)$ within accuracy β, it can be easily shown that $D(f) \times (D_f(A) - D_f(B)) + D(B)$ is an estimate of $D(f)$ within accuracy $\alpha + \beta(3 + 2\alpha)$.

As usual, $D(f)$ can be estimated using the oracle $EX(f, D)$. We can estimate $D_f(A)$ and $D_f(B)$ using the POS oracle $EX(f, D_f)$. We can estimate $D(B)$ with the UNL oracle $EX(1, D)$. So we can modify any statistical query based algorithm so that it uses the EX, POS and UNL oracles. Furthermore, if the standard algorithm makes N queries, labeled, positive and unlabeled example sources will be used to estimate respectively 1, $2N$ and N queries.

In this paper, we make the assumption that labeled examples are "expensive" and that unlabeled and positive examples are "cheap". If we make the stronger assumption that positive and unlabeled data are free, we can estimate $D_f(A)$,

$D_f(B)$ and $D(B)$ within arbitrary accuracy, i.e. $\beta = 0$. If τ_{min} is the smallest tolerance needed by the learning algorithm L and whatever is the number of queries made by L, we see that we only need labeled examples to estimate **only one** probability, say $D(f)$, within accuracy τ_{min}. Let $\mathcal{Q}$ be the query space used by L. Theorem 1 gives an upper bound of the number of calls of $EX(f, D)$ necessary to simulate the statistical queries needed by L. We see that we can expect to divide this number of calls by $VCDIM(\mathcal{Q})$.

A more precise theoretical study remains to be done. For instance, it should be interesting to estimate the expected improvements of the accuracy when the number of labeled examples is fixed depending on the number of positive and unlabeled examples. This could be done for each usual statistical query learning algorithm.

5 Tree Induction from Labeled, Positive, and Unlabeled Data

C4.5, and more generally decision tree based learning algorithms are SQ-like algorithms because attribute test choices depend on statistical queries. After a brief presentation of C4.5 and as C4.5 is an SQ algorithm, we describe in Sect. 5.2 how to adapt it for the treatment of positive and unlabeled data. We finally discuss experimental results of a modified version of C4.5 which uses positive and unlabeled examples.

5.1 C4.5, a Top-Down Decision Tree Algorithm

Most algorithms for tree induction use a top-down, greedy search through the space of decision trees. The *splitting criterion* used by C4.5 [Qui93] is based on a statistical property, called *information gain*, itself based on a measure from information theory, called *entropy.* Given a sample S of some target concept, the entropy of S is $Entropy(S) = \sum_{i=1}^{c} -p_i \log_2 p_i$ where p_i is the proportion of examples in S belonging to the class i. The information gain is the expected reduction in entropy by partitioning the sample according to an attribute test t. It is defined as

$$Gain(S, t) = Entropy(S) - \sum_{v \in Values(t)} \frac{N_v}{N} Entropy(S_v) \qquad (2)$$

where $Values(t)$ is the set of every possible value for the attribute test t, N_v is the cardinality of the set S_v of examples in S for which t has value v and N is the cardinality of S.

5.2 C4.5 with Positive and Unlabeled Data

Let X be the instance space, we only consider binary classification problems. The classes are denoted by 0 and 1, an example is said to be positive if its label is 1. Let POS be a sample of positive examples of some target concept f, let LAB be a sample of labeled examples and let UNL be a set of unlabeled data. Let D be the hidden distribution which is defined over X. POS is a set of examples $\langle x, f(x) = 1\rangle$ returned by an example oracle $EX(f, D_f)$, LAB is a set of examples $\langle x, f(x)\rangle$ returned by an example oracle $EX(f, D)$ and UNL is a set of instances x drawn according to the distribution D. The entropy of a sample S is defined by $Entropy(S) = -p_0 \log_2 p_0 - p_1 \log_2 p_1$. In this formula, S is the set of training examples associated with the current node n and p_1 is the proportion of positive examples in S. Let D_n be the filtered distribution, that is the hidden distribution D restricted to instances reaching the node n, let X_n be the set of instances reaching the node n: p_1 is an estimation of $D_n(f)$. Now, in the light of the results of Sect. 4, we modify formulas for the calculation of the information gain. We have $D_n(f) = D(X_n \cap f)/D(X_n)$. Using the equation $D(X_n \cap f) = D_f(X_n) \times D(f)$, we obtain $D_n(f) = D_f(X_n) \times D(f) \times 1/D(X_n)$.

We can estimate $D_f(X_n)$ using the set of positive examples associated with the node n, we can estimate $D(f)$ with the complete set of labeled examples, and we can estimate $D(X_n)$ with unlabeled examples. More precisely, let POS^n be the set of positive examples associated with the node n, let UNL^n be the set of unlabeled examples associated with the node n, and let LAB_1 be the set of positive examples in the set of labeled examples LAB, the entropy of the node n is calculated using the following:

$$p_1 = \inf\left(\frac{|POS^n|}{|POS|} \times \frac{|LAB_1|}{|LAB|} \times \frac{|UNL|}{|UNL^n|}, 1\right); p_0 = 1 - p_1 \qquad (3)$$

The reader should note that $\frac{|LAB_1|}{|LAB|}$ is independent of the node n. We now define the information gain of the node n by $Gain(n, t) = Entropy(n) - \sum_{v \in Values(t)} (N_v^n/N^n) Entropy(nv)$ where $Values(t)$ is the set of every possible value for the attribute test t, N^n is the cardinality of UNL^n, N_v^n is the cardinality of the set UNL_v^n of examples in UNL^n for which t has value v, and nv is the node below n corresponding to the value v for the attribute test t.

5.3 Experimental Results

We applied the results of the previous section to C4.5 and called the resulting algorithm C4.5PosUnl. The differences as compared with C4.5 are the following:

- C4.5PosUnl takes as input three sets: LAB, POS and UNL
- $|LAB_1|/|LAB|$ which appears in (3) is calculated only once
- For the current node, entropy and gain are calculated using (3)
- When gain ratio is used, split information is calculated with unlabeled examples
- The majority class is chosen using (3)
- halting criteria during the top-down tree generation are evaluated on UNL
- When pruning the tree, classification errors are estimated with the help of proportions p_0 and p_1 from (3)

We consider two data sets from the *UCI Machine Learning Database* [MM98]: kr-vs-kp and adult. The majority class is chosen as positive. We fix the sizes of the test set, set of positive examples, and set of unlabeled examples. These values are set to:

– For kr-vs-kp: 1000 for the test set, 600 for the set of positive examples, and to 600 for the set of unlabeled examples
– For adult: 15000 for the test set, 10000 for the set of positive examples, and to 10000 for the set of unlabeled examples

We let the number of labeled examples vary, and compare the error rate of C4.5 and C4.5PosUnl. For a given size of LAB, we iterate 100 times the following: all sets are selected randomly (for POS, a larger set is drawn and only the selected number of positive examples are kept), we compute the error rate for C4.5 with input LAB and the error rate for C4.5PosUnl with input LAB, POS and UNL. Then, we average out the error rates over the 1000 experiments.

The results can be seen in Fig. 2. The error rates are promising when the number of labeled examples is small (e.g. less than 100). We think that the better results of C4.5 for higher number of examples is due to our pruning algorithm which does not use in the best way positive and unlabeled examples (C4.5PosUnl trees are consistently larger than C4.5 ones).

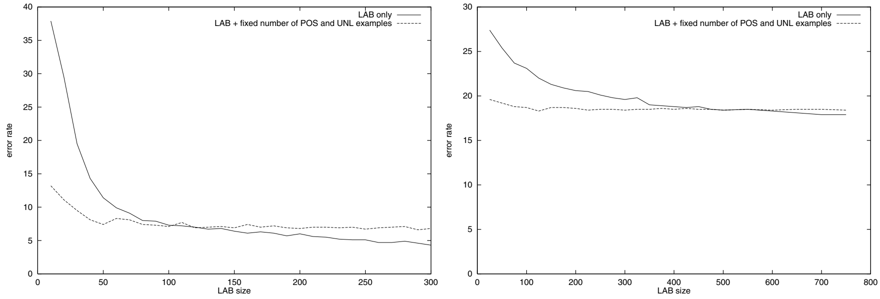

Fig. 2. error rate of C4.5 and C4.5PosUnl averaged over 100 trials. The left figure shows the results on the kr-vs-kp data set and the right one corresponds to the adult data set.

adult and kr-vs-kp were selected in this paper because they are well known and contain many examples. The experiments were run on all other two-class UCI problems (ftp://grappa.univ-lille3.fr/pub/Experiments/C45PosUnl).

6 Conclusion

In many practical learning situations, labeled data are rare or expensive to collect while a great number of positive and unlabeled data are available. In this paper, we have given experimental and theoretical evidence that these kind of examples can efficiently be used to boost statistical query learning algorithms. A lot of work remains to be done, in several directions:

- More precise theoretical results must be stated, at least for specific statistical query learning algorithms
- C4.5 should be modified further, especially the pruning algorithm which must be adapted to the data types presented in this paper
- We intend to collect real data of the kind studied here (labeled, positive and unlabeled) to test this new variant of C4.5
- Our method can be applied to any statistical query algorithm. It would be interesting to know if it can be appropriate elsewhere

References

[AL88] D. Angluin and P. Laird. Learning from noisy examples. *Machine Learning*, 2(4):343–370, 1988.

[BM98] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Proc. 11th Annu. Conf. on Comput. Learning Theory*, pages 92–100. ACM Press, New York, NY, 1998.

[Dec97] S. E. Decatur. Pac learning with constant-partition classification noise and applications to decision tree induction. In *Proceedings of the Fourteenth International Conference on Machine Learning*, 1997.

[Den98] F. Denis. Pac learning from positive statistical queries. In *ALT 98, 9th International Conference on Algorithmic Learning Theory*, volume 1501 of *Lecture Notes in Artificial Intelligence*, pages 112–126. Springer-Verlag, 1998.

[Kea93] M. Kearns. Efficient noise-tolerant learning from statistical queries. In *Proceedings of the 25th ACM Symposium on the Theory of Computing*, pages 392–401. ACM Press, New York, NY, 1993.

[KM97] M. Kubat and S. Matwin. Addressing the curse of imbalanced training sets : One-sided selection. In *Proceedings of the 14th International Conference on Machine Learning*, pages 179–186, 1997.

[MM98] C.J. Merz and P.M. Murphy. UCI repository of machine learning databases, 1998.

[NMTM98] K. Nigam, A. McCallum, S. Thrun, and T. Mitchell. Learning to classify text from labeled and unlabeled documents. In *Proceedings of the 15th National Conference on Artificial Intelligence, AAAI-98*, 1998.

[Qui93] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, CA, 1993.

[Val84] L.G. Valiant. A theory of the learnable. *Commun. ACM*, 27(11):1134–1142, November 1984.