Teaching Active Human Learners

Zizhe Wang[1,3], Hailong Sun*[1,2,3]

[1]SKLSDE Lab, School of Computer Science and Engineering, Beihang University, Beijing, China 100191
[2]School of Software, Beihang University, Beijing, China 100191
[3]Beijing Advanced Innovation Center for Big Data and Brain Computing, Beihang University, Beijing, China 100191
wangzz@act.buaa.edu.cn, sunhl@buaa.edu.cn

Abstract

Teaching humans is an important topic under the umbrella of machine teaching, and its core problem is to design an algorithm for selecting teaching examples. Existing work typically regards humans as passive learners, where an ordered set of teaching examples are generated and fed to learners sequentially. However, such a mechanism is inconsistent with the behavior of human learners in practice. A real human learner can actively choose whether to review a historical example or to receive a new example depending on the belief of her learning states. In this work, we propose a model of active learners and design an efficient teaching algorithm accordingly. Experimental results with both simulated learners and real crowdsourcing workers demonstrate that our teaching algorithm has better teaching performance compared to existing methods.

1 Introduction

Machine teaching studies the problem of finding an optimal training set to guide a "student" to learn a target model (Zhu 2015; Zhu et al. 2018). And the students usually fall into two categories, including humans and machines. As for humans, machine teaching helps reduce the cost of education and improve the learning experience. For example, teaching algorithms can be used to select a curriculum of learning materials for students in intelligent tutoring systems (ITS), or to train crowdsourcing workers for doing complex tasks.

However, conventional researches of teaching humans typically regard a human learner passively as a machine (Piech et al. 2015; Zhang et al. 2017; Aodha et al. 2018), where learners are assumed to receive the examples one by one in the order given by certain algorithms. Thus, these approaches cannot fulfill the potentials of humans because there are three salient differences between humans and machines. First, machines never forget and they remember every single input, while humans may forget what they have learned (Zhou, Nelakurthi, and He 2018). Second, the learning process of humans is hard to model. We may know all the details of a machine learning algorithm, but we cannot perfectly model how humans learn. That means it is hard to find an optimal teaching algorithm to teach a human learner. Third, human learners know what they need. After learning a certain amount of examples, a human learner can realize which knowledge she has not learnt well. Although we cannot perfectly model human learners, we can take full advantage of their initiative to improve teaching results by allowing them to review the previous teaching examples.

In this paper, we focus on the problem of teaching crowdsourcing workers to improve their skills. We propose a new teaching form that stimulates the learner's initiative and introduce a review mechanism into the new teaching form. Different from traditional teaching form, a learner in our teaching form can have two options during the process of learning: reviewing a historical example or receiving a new example. Then we design a learner model to characterize the learning behavior of the learners. As existing learner models do not consider the initiative of the learners (Singla et al. 2014; Aodha et al. 2018), we propose an active learner model with three factors that influence human learners' cognition: (1) the consistency between a hypothesis and a teaching example; (2) the differences between the current teaching example and previous ones; (3) the times that an example has been reviewed. With the learner model, we then design a greedy algorithm to select the examples and we show the theoretical approximation guarantee on the convergence to certain error rate.

Our main contributions are summarized as follows:

- We design a new teaching form with a review mechanism that takes advantage of human learners' initiative.
- We model the active learner under the new teaching form and propose a teaching algorithm called Active Learner Teaching Algorithm (ALTA).
- We conducted both simulations and experiments on a well-known crowdsourcing platform, and the results show the superiority of our approach.

The rest of this paper is organized as follows: Section 2 gives a brief introduction of the related works on machine teaching and teaching humans. Section 3 provides the problem description and introduces a classic learner model STRICT. Section 4 defines the active learner model and designs a teaching algorithm ALTA. Section 5 presents the experimental evaluation results. Finally, in Section 6, we conclude this work.

*Corresponding author

2 Related Work

2.1 Machine Teaching

Machine teaching is a problem of finding an optimal training set given a student and a target model (Zhu 2015; Zhu et al. 2018). For example, consider a student who is a Support Vector Machine (SVM) and a teacher who wants the student to learn a target model θ^*. In this case, θ^* is a hyperplane. Machine teaching aims to design a minimal training set A through which a student can learn θ^*. The students and teachers in machine teaching can be any intelligent entities, such as humans and machine learning algorithms.

The studies of teaching complexity lay the foundation for machine teaching, which aims to find the lower bound of the size of a training set for obtaining the optimal training results with a machine learning algorithm (Goldman and Kearns 1995; Shinohara and Miyano 1991; Anthony et al. 1992). But in machine teaching, the "student" can be a machine learning algorithm, a robot (Kulick et al. 2013) and even a human (Corbett and Anderson 1994; Piech et al. 2015; Zhang et al. 2017).

Teaching humans is usually harder than teaching machine learning algorithms because we do not know the hypothesis space of humans and how they learn knowledge from teaching examples. Dasgupta et al. (2019) study a situation that a teacher do not know anything about students and propose a teaching algorithm that teaching students by interacting a lot with them. But the algorithm is not suitable for human learners, since it needs a lot of interactions to obtain the optimized teaching set, which is time-consuming.

2.2 Teaching Humans

In the studies of intelligent tutoring systems (ITS), there is a method called knowledge tracing (KT), which selects a curriculum of learning materials for students. Knowledge tracing traces the knowledge state of students based on their past exercise performance. By doing this, it can predict students' performance on future exercise and recommend suitable exercise for them. Bayesian knowledge tracing (Corbett and Anderson 1994) is one of the most popular KT models. It uses a Hidden Markov Model to model the changing process of students' knowledge states. As the development of deep learning, Long Short-Term Memory network, and Memory Augmented Neural Network has been used in Knowledge Tracing (Piech et al. 2015; Zhang et al. 2017), and have excellent performance. Besides education, KT can also be utilized in crowdsourcing tasks (Wang, Sun, and Han 2020) for designing task allocation strategies on the basis of tracing workers' knowledge states.

However, knowledge tracing methods require a lot of historical data to ensure the accuracy of the model. When publishing annotation tasks to crowdsourcing platforms, requesters usually do not have sufficient historical data of workers. In this case, we need to deal with the cold start problem and design a learner model that does not need historical data. Singla et al. (2014) models learners via a Markov chain and uses a greedy algorithm to select teaching examples for learners. Based on this strategy, Aodha et al. (2018) introduces explanation into the teaching materials by highlighting the parts of an image that are responsible for the class label to help the student learn. Liu, Hou, and Tang (2020) considers the fine-grained concepts of a learner and uses graph neural network to estimate the learner's ability. But these work feeds the learners with examples of fixed order, which wastes the learners' initiative. By considering initiative of human learners, Peltola et al. (2019) studies a sequential machine teaching problem, where the learner actively chooses queries and the teacher can only provides responses to them. However, when the number of teaching examples increases, it will be hard for a human learner to select examples. So we introduce the reviewing mechanism, which aims to utilize the initiative of learners while selecting an optimal teaching set.

3 The Preliminaries

3.1 Problem Description

Let $\mathcal{X}$ denote a set of examples (e.g. images), $x \in \mathcal{X}$. Each x has its label $y(x) \in \{-1, 1\}$. We use $\mathcal{H}$ to denote a finite set of hypotheses. Each element of $\mathcal{H}$ is a function $h : \mathcal{X} \longmapsto \mathbb{R}$. $|h(x)|$ indicates how much confidence hypothesis h has in the label of x. The label assigned to x by hypothesis h is $sgn(h(x))$. Our target is to teach an active human learner a hypothesis h^*, where for each $x \in \mathcal{X}$, $sgn(h^*(x)) = y(x)$. The way to teach human learners is by showing them examples. We call $A \subset \mathcal{X}$ a teaching set if a human learner learns h^* after being taught with A. And our goal is to find such a teaching set with minimal size. To solve the problem, there are two major issues we need to handle. First, since we do not know the learning process of human learners, we need to model it. Second, with the learner model, we need to design a teaching algorithm for it.

3.2 The STRICT Learner Model

The STRICT algorithm (Singla et al. 2014) is a classic teaching algorithm for teaching humans and our learner model is modified on it. Learners are modeled to carry out a random walk in a finite hypothesis space $\mathcal{H}$. At the beginning of the teaching, a learner randomly chooses a hypothesis h_1, drawn from the prior distribution of $P_0(h)$. During teaching, she will be presented with a sequence of examples along with the corresponding true labels. After showing the learner the t^{th} example x_t and its label y_t, there could be two possible results. If (x_t, y_t) agrees with the prediction of the learner's current hypothesis h_t (i.e., $sgn(h_t(x_t)) = y_t$), the learner keeps the hypothesis (i.e., $h_{t+1} = h_t$). Otherwise, if $sgn(h_t(x_t)) \neq y_t$, the learner draws a new hypothesis h_{t+1} from the distribution $P_t(h)$. Through the updating process of the distribution, those hypotheses that disagree with the ground truth of examples in the previous steps are less likely to be chosen. The formula of calculating $P_t(h)$ is given as the following:

$$P_t(h) = \frac{1}{Z_t} P_0(h) \prod_{\substack{s=1 \\ y_s \neq sgn(h(x_s))}}^{t} P(y_s|h, x_s) \qquad (1)$$

with normalization factor

$$Z_t = \sum_{h \in \mathcal{H}} P_0(h) \prod_{\substack{s=1 \\ y_s \neq sgn(h(x_s))}}^{t} P(y_s|h, x_s) \quad (2)$$

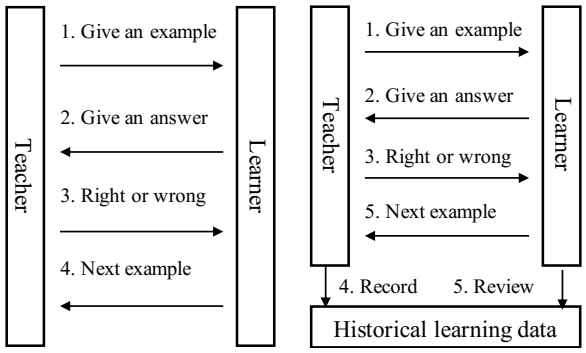
$P(y_s|h, x_s)$ describes how much the prediction of hypothesis h agrees with (x_s, y_s):

$$P(y_s|h, x_s) = \frac{1}{1 + e^{-\alpha h(x_s) y_s}}, \alpha > 0 \quad (3)$$

When (x_t, y_t) strongly disagrees with the prediction $sgn(h_t(x_t))$ (i.e., $h(x_s)y_s$ takes a large negative value), $P(y_s|h, x_s)$ will be very small. Then the hypothesis h will be more unlikely to be chosen. The parameter $\alpha > 0$ controls the effect of observing inconsistent examples. When $\alpha \to \infty$, the hypotheses that disagree with (x_t, y_t) will be completely abandoned.

4 Our Approach

4.1 A Teaching Form With Review Mechanism



(a) Traditional teaching form (b) Our teaching form

Figure 1: The traditional teaching form and our teaching form with review mechanism. In Step 5, by adding a historical examples pool, the learner can decide whether to ask for a new teaching example or review an old one.

As shown in Figure 1(a), in traditional work, the sequence of examples is decided by the teacher (Singla et al. 2014; Aodha et al. 2018). The teacher shows the learner one example. Then the learner answers the teacher and receives the correct answer from the teacher. Finally, the learner finishes this example and asks for the next. In this teaching form, human leaners can only learn the examples one by one without reviewing the historical examples.

We designs a new teaching form that can utilize the initiative of human learners. Figure 1(b) shows the workflow of our teaching form. In the fourth step, the teacher records the example to the historical data pool. The learner also has access to historical data. So in the fifth step, she can choose to ask for a new example or review a historical one. We call the human learners in our new teaching form the *active learners* since they can actively choose which example to learn.

4.2 Active Learner Model

We now present the active learner model. At the beginning of the teaching, the learner randomly picks a hypothesis $h_1 \in \mathcal{H}$ according to the prior distribution $P_0(h)$. During teaching, she will be presented with a sequence of examples along with the correct class label. Let S_A denote an example sequence generated from $A \subset \mathcal{X}$ by the learner, the examples in S_A can be duplicated ($|S_A| = \eta|A|$). At round t, after receiving a new example or reviewing a historical example s_t, the learner will switch her hypothesis h_t to a new one h_{t+1} based on the distribution $P_t(h)$:

$$P_t(h) = \frac{1}{Z_t} P_0(h) \prod_{i=1}^{t} (\mathcal{A}(s_i)\mathcal{B}(s_i)\mathcal{C}(s_i))^{\frac{1}{\eta}} \quad (4)$$

with normalization factor

$$Z_t = \sum_{h \in \mathcal{H}} P_0(h) \prod_{i=1}^{t} (\mathcal{A}(s_i)\mathcal{B}(s_i)\mathcal{C}(s_i))^{\frac{1}{\eta}}. \quad (5)$$

The three functions $\mathcal{A}, \mathcal{B}, \mathcal{C}$ respectively represent three factors that influence the updating of the hypotheses distribution. The explanation is given as follows.

$\mathcal{A}$: We use $a(s) = -h(s)y$ to represent how much the label $y(s)$ disagrees with the learner's prediction $sgn(h(s))$. Then we can define

$$\mathcal{A}(s_i) = \frac{1}{1 + e^{\alpha a(s_i)}}. \quad (6)$$

Different from the traditional learner model (Singla et al. 2014; Aodha et al. 2018), the learner in our model always draws a new hypothesis according to the distribution $P_t(h)$ after learning an example, even if the label $y(s)$ agrees with the prediction $sgn(h(s))$. That is because a hypothesis with a correct prediction on s could also be a wrong hypothesis. This small change improves the computing efficiency.

$\mathcal{B}$: This function describes the influence of the diversity between two adjacent examples. We suppose the learner will get more information in the following two situations:

- The two adjacent examples are in different classes but seems similar.
- The two adjacent examples are in the same classes but seems different.

Let $d(s_i, s_{i-1})$ be the distance between s_i and s_{i-1}, since the distance belongs to $[0, +\infty)$, we need to normalize it first:

$$b(s_i) = \frac{2}{1 + e^{\beta d(s_i, s_{i-1})}}, \beta > 0 \quad (7)$$

where y_i is the label of s_i. We can see that $b(s_i) \in (0, 1]$ Then we can give the definition of $\mathcal{B}$

$$\mathcal{B}(s_i) = \begin{cases} 1 & i = 1 \\ b(s_i) & i > 1, y_i y_{i-1} = 1 \\ 1 - b(s_i) & i > 1, y_i y_{i-1} = -1. \end{cases} \quad (8)$$

$\mathcal{C}$: Studies in industry engineering indicate that the quality of workers improve when they complete repetitive

work (Adler and Clark 1991; Vits and Gelders 2002). And the learning curve is a mature theory for describing such improvements (Fioretti 2007). Our prior work shows that the ability of crowdsourcing workers also follows a certain learning curve (Wang et al. 2017).

We use exponential learning curve $lc(k) = 1 - e^{-\gamma k} (\gamma > 0)$ to represent how much a learner comprehends an example when she sees the example k times. Then we can calculate the benefits she obtains from the k^{th} time by $c(s_i) = e^{\gamma - \gamma k(s_i)} - e^{-\gamma k(s_i)}$, where $k(s_i)$ is the number of examples that are the same to s_i in $(s_1, \ldots, s_i)$. Since $c(x_i) \in [0, 1]$, and a bigger $c(s_i)$ makes more benefits to the decreasing of the expected rate, which means smaller multipliers to distribution $P_{t-1}(h)$. Now we have:

$$\mathcal{C}(s_i) = 1 - c(s_i). \tag{9}$$

η: Given a teaching set A of size $|A|$, the distribution in Equation (1) updates $|A|$ times. However, in our learner model, the distribution of h updates $|S_A|$ times. We set the length of S_A to $\eta|A|$, which means each example is reviewed for η times on average. As for the probability distribution of a hypothesis after the learner seeing an example, the effect of our teaching form is $\frac{1}{\eta}$ of the traditional form. Therefore we adjust the multiplier $\mathcal{A}(s_i)\mathcal{B}(s_i)\mathcal{C}(s_i)$ to $(\mathcal{A}(s_i)\mathcal{B}(s_i)\mathcal{C}(s_i))^{\frac{1}{\eta}}$. This adjustment reduces the benefits our learners gain from one example. Then we can compare our method to traditional methods more fairly.

Besides defining the updating methods of the distribution $P_t(h)$, we need to model the process that a learner selects examples. In other words, we should declare how S_A is generated from A. Since the learner does not know what the following instances are, she will not make a decision that benefits her in the future. We suppose the learner's strategy follows a greedy algorithm, which always selects the example that benefits the learner the most at that time.

4.3 Teaching Algorithm

Algorithm 1 ALTA

Input: $\mathcal{X}$, $\mathcal{H}$, P_0, ϵ
Output: A

```
A = ∅
while F(A) < E[err|∅] − P_0(h*)ε do
    x = argmax_{x∈X} F(A ∪ {x})
    A = A ∪ {x}
end while
```

Given the learner model, how should a teacher choose examples to teach the learner the hypothesis h^*? In our learner model, "teaching the learner h^*" is actually "increasing the weight of h^* in the distribution $P_t(h)$", which also means reducing the expected error-rate of the learner. Before solving this problem, we need to define some new notations. A is a subset of $\mathcal{X}$. The learning sequence of a learner who learns A is S_A. We set the length of S_A as $\eta|A|$. Then we can write the learner's posterior after showing A as:

$$P(h|S_A) = \frac{1}{Z(S_A)} P_0(h) \prod_{i=1}^{\eta|A|} (\mathcal{A}(s_i)\mathcal{B}(s_i)\mathcal{C}(s_i))^{\frac{1}{\eta}}. \tag{10}$$

The expected error-rate of the learner after seeing A together with the labels can be expressed as

$$\mathbb{E}[err|S_A] = \sum_{h \in \mathcal{H}} P(h|S_A) err(h, h^*), \tag{11}$$

where

$$err(h, h^*) = \frac{|\{x \in \mathcal{X} : sgn(h(x)) \neq sgn(h^*(x))\}|}{|\mathcal{X}|} \tag{12}$$

describes the distance between h and h^*.

Given a limitation of error-rate $\epsilon > 0$, our objective is to find the smallest learning sequence S_A^* achieving this error-rate as below:

$$S_{A,\epsilon}^* = arg\min |S_A|, s.t. \mathbb{E}[err|S_A] \leq \epsilon. \tag{13}$$

Since $|S_A|$ is determined by $|A|$ ($|S_A| = \eta|A|$), the optimal S_A must be generated from the smallest teaching set A. The objective can be written as:

$$A_\epsilon^* = arg\min_{A \subset \mathcal{X}} |A|, s.t. \mathbb{E}[err_L|S_A^*] \leq \epsilon. \tag{14}$$

However, the computation complexity of solving Equation (14) is NP-hard, which is intractable. We convert this problem to a set cover problem. To solve this problem, we refer to the methods of solving set cover problem (Nemhauser, Wolsey, and Fisher 1978). The greedy algorithm is an efficient approximation algorithm for the set cover problem, and it also works for our problem.

At first, we need to define a new function F:

$$F(A) = \sum_{h \in \mathcal{H}} (G_0(h) - G(h|S_A)) err(h, h^*), \tag{15}$$

where

$$G(h|S_A) = P_0(h) \prod_{i=1}^{|S_A|} (\mathcal{A}(s_i)\mathcal{B}(s_i)\mathcal{C}(s_i))^{\frac{1}{\eta}}. \tag{16}$$

$G(h|S_A)$ is the unnormalized posterior of the learner, $G_0(h) = P_0(h)$. We can prove the efficiency of the greedy algorithm using the submodularity of $F(A)$. Maximizing $F(A)$ helps us to approximate the optimal result. We call the greedy algorithm Active Learner Teaching Algorithm (ALTA) and describe it in Algorithm 1.

Let θ denote $2P_0(h^*)\epsilon$, Algorithm 1 terminates after at most $|A_\theta^*| log\frac{2}{\theta}$ steps with a set A such that $\mathbb{E}[err|S_A] \leq \epsilon$.

4.4 Hypothesis Space

Now we give a discussion about the hypothesis space $\mathcal{H}$. From the definition of the learner model, we can easily find that $\mathcal{H}$ is an essential element in our model. Since we do not know the $\mathcal{H}$ of a human learner, we need to simulate it. In traditional researches, $\mathcal{H}$ is usually set as a set of linear functions (Singla et al. 2014). For instance, if a teaching example $x \in \mathcal{X}$ is represented as a vector of features

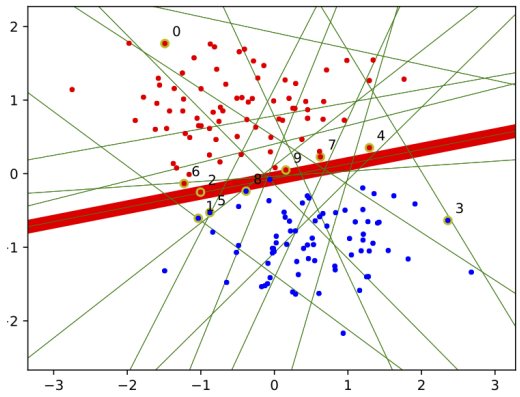


Figure 2: The hypotheses and teaching examples. The lines are randomly generated hypotheses. The points are the teaching examples. Each hypothesis divides the points into two categories. The red line is the best hypothesis. The marked points are chosen to teach the student.

$x = (f_1, f_2, \ldots, f_n)$, the hypothesis $h \in \mathcal{H}$ is a vector of the same length. The hypothesis h maps x to $h \cdot x$, while $sgn(h \cdot x)$ is the class of x under hypothesis h.

Besides setting $\mathcal{H}$ as a set of linear functions, we also need to decide what functions should be in the set. Previous work chooses the functions manually. And there is no criterion for choosing the elements. In other words, they rely on their intuition to construct the hypothesis space. However, a manual method results in lower efficiency, and relying on intuition means less robust. Our method avoids the problems above. We only need to generate a large set of random hypotheses. Figure 2 is an instance of the hypothesis space. The lines are the hypotheses that are randomly generated. Each of them divides the data points (examples $x \in \mathcal{X}$) into two categories. And the marked points are the teaching examples chosen by our algorithm.

5 Experiments

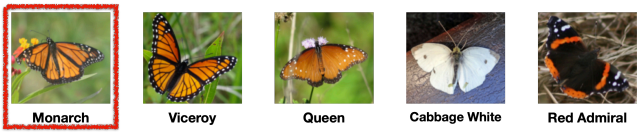
The section presents the experimental evaluation of our approach. The source code and data are publicly available[1].

5.1 Datasets

We conducted experiments on both simulated learners and real human learners with four datasets including *Butterfly*, *Chinese Character*, *Woodpecker* and *Breast Cancer*, which are widely used in existing work (Singla et al. 2014; Aodha et al. 2018).

Butterfly Our first dataset is sampled from the iNaturalist species classification and detection dataset (Horn et al. 2017). It contains images of five different species of butterflies, i.e. "Monarch", "Viceroy", "Queen", "Red Admiral", and "Cabbage White". Figure 3(a) shows the examples of the five categories of butterflies, where we can observe that the first three species have many common features and the last two are distinct in appearance. In total, there are

[1]https://github.com/Brickkkkkk/ALTA_AAAI21



(a) Butterfly

(b) Chinese character

(c) Woodpecker

Figure 3: Examples of butterfly, Chinese character, woodpecker datasets. The category in the red box is the class we chose to classify.

2,224 images in the dataset, which uniformly distribute on the five species. For each image, a task is designed to classify whether the butterfly shown in the image is "Monarch".

Chinese Character Our second dataset is extracted from CASIA Online and Offline Chinese Handwriting Databases (Liu et al. 2011). It contains 717 images, each of which contains one of three Chinese characters (i.e. "grass", "mound" and "stem") handwritten with different styles. Figure 3(b) shows the three categories of Chinese characters. These characters look very similar to each others, which increases the classification difficulty. Similarly, for each image in the dataset, a binary classification task is designed to recognize if the character is "grass".

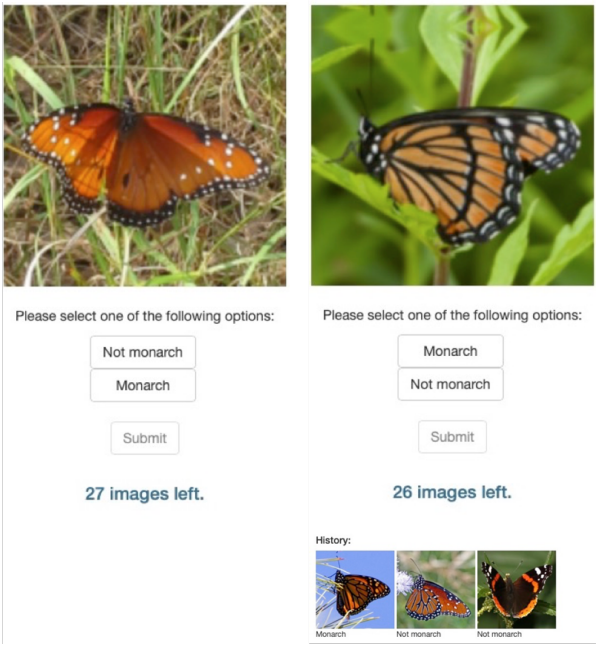
Woodpecker The images in our third dataset belong to a publicly available dataset (Wah et al. 2011). There are 176 real images of three species of woodpeckers, i.e. "Red-Bellied Woodpecker", "Red-Cockaded Woodpecker" and "Red-Headed Woodpecker". And there are about 60 images for each species. Three examples of them are shown in Figure 3(c). In this dataset, the task is to classify whether a given image contains a "Red-Bellied Woodpecker" or not.

Breast Cancer The last one is the breast cancer dataset (Dua and Graff 2017). There are 569 samples in this dataset and each sample has 30 dimensions. Since there are only feature vectors in this dataset and no images available, we only conducted simulated experiments on this dataset.

5.2 Experimental Setup

In our experiments, we hope to answer the following four questions:

- Can our method outperform the state-of-the-art methods?
- Is our method robust for any hypothesis space?
- Are $\mathcal{B}(x)$ and $\mathcal{C}(x)$ reasonable and effective?
- Is our new teaching form more time-consuming?



(a) Without review (b) With review

Figure 4: The teaching interfaces of ALTA and traditional methods. We introduce the review mechanism in the teaching interface. Workers can see the examples they have learned and their corresponding labels.

To answer the four questions, we compared our ALTA algorithm with other algorithms for different types of hypothesis space $\mathcal{H}$. We also studied the performance of ALTA with different parameters α, β, γ.

We conducted experiments with both simulated learners and Amazon Mechanical Turk workers. In the simulated experiments, we simulated 240 learners with two kinds of hypothesis space $\mathcal{H}$: random, cluster-random. The random hypothesis space contains hypotheses generated randomly. The cluster-random hypothesis space consists of two kinds of hypotheses. One is randomly generated hypotheses, the other is clusters of hypotheses. Each hypothesis cluster can bracket images of the same category together (the annotations can be wrong). It is worth mentioning that there is always an optimal hypothesis that can classify all examples correctly in all hypothesis spaces. And the optimal hypothesis is generated by SVM. We used Euclidean distance in $\mathcal{B}(x)$. For each dataset, we sampled 80% of the examples in each category as the teaching examples set $\mathcal{X}$. We also created a test set with the rest examples. The parameters of our learner model was set as $\alpha = 0.5$, $\beta = 0.001$, $\gamma = 1$, $\eta = 3$. We used the teaching algorithm to select teaching examples from $\mathcal{X}$. After getting the teaching set A, we used A to teach the 240 learners and calculated their expected error on the test set. We ran the algorithm under different teaching set size $|A|$ and plot the changing trend of the expected error. We also studied the effectiveness of $\mathcal{B}(x)$ and $\mathcal{C}(x)$ by conducting an ablation experiment.

In the experiments on AMT, since we do not know the hypothesis space of AMT workers, we need to simulate a learner first. We simulated a learner with random hypothesis space and utilized the teaching algorithms for her. Then we got different teaching sets from different algorithms. After that, we released tasks on AMT. Figure 4 shows the different interfaces between our method and traditional methods. Each worker will answer a questionnaire of 30 tasks. The first 10 tasks are teaching examples, and the worker will get the right answers after answering them. The last 20 tasks are testing tasks, which are used for estimating the expected error rate. The interfaces of traditional methods are like Figure 4(a), where workers have no chance to review the previous images. And the interface of our method is showed in Figure 4(b), in which we added a "history" module to help workers review the historical examples.

We compared our ALTA method with two traditional teaching algorithms and one baseline method.

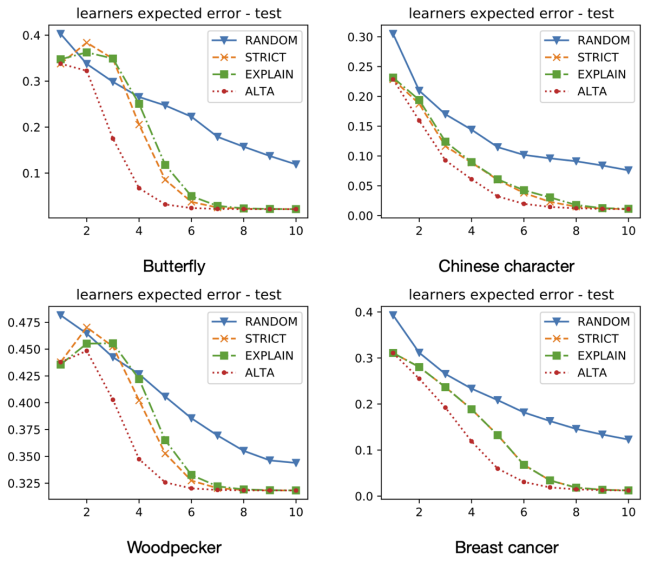
- STRICT (Singla et al. 2014): A classic teaching algorithm. This algorithm is a greedy algorithm used for teaching the learner model introduced in Section 3.2.
- EXPLAIN (Aodha et al. 2018): The state-of-the-art algorithm. The paper that proposed this algorithm introduced the explanation of teaching examples into the teaching interface. This algorithm is used for teaching the learners who are taught with both images and their explanations.
- RANDOM: The baseline method, which randomly selects the teaching examples.

5.3 Results

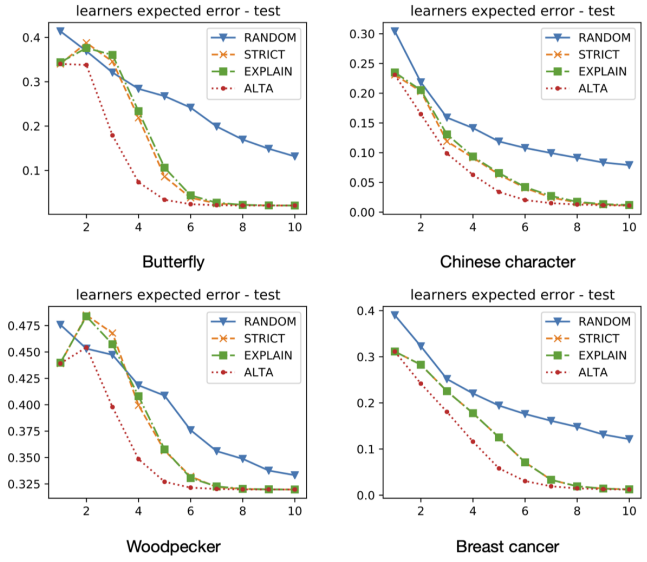
Results on Simulated Learners Figure 5(a) shows the results of the 120 simulated learners with random hypothesis spaces. We calculated the average expected error on the test set and our method outperforms all the other algorithms on the four datasets. With the expansion of the teaching set, the expected error gradually decreases to zero. Figure 5(b) shows the results on cluster-random hypothesis spaces, in which our ALTA algorithm demonstrates excellent performance as well. The results can answer the first question.

Is our method robust in any kind of hypothesis space? Since the hypothesis spaces are randomly generated, Figure 5(a) is enough to show the robustness of our algorithm. To be more convincing, we also experimented on cluster-random hypothesis space. The results show that our ALTA algorithm is effective on the two automatically generated hypothesis spaces.

Is it effective to introduce the $\mathcal{B}(x)$ and $\mathcal{C}(x)$ into the learner model? To answer this question, we conducted an ablation experiment. At first, we deleted $\mathcal{B}(x)$ and $\mathcal{C}(x)$ in Equation 4 and found that our method can still outperform the other methods. This proved the effectiveness of our teaching form. For example, suppose a learner receives a teaching set with two examples A and B. Let $\eta = 3$. If she is taught with a traditional teaching form, her learning sequence can only be (A, A, A, B, B, B) or (B, B, B, A, A, A). But in our teaching form, she has more choices (e.g., (A, B, A, A, B, B)). That is why our method could outperform the other comparing methods even we



(a) Random



(b) Cluster Random

Figure 5: Expected error of four algorithms on test sets. Figure (a) is the results of random hypothesis spaces. Figure (b) is the results of cluster-random hypothesis spaces. The horizontal axes represent the size of the teaching set. The vertical axes are the expected error on the test set.

deleted $\mathcal{B}(x)$ and $\mathcal{C}(x)$. Then we studied the effectiveness of $\mathcal{B}(x)$ and $\mathcal{C}(x)$ by adjusting one's parameter while fixing the other parameters. The results show that both of them can improve the performance of our method. Finally we set the parameters to $\alpha = 0.5$, $\beta = 0.001$, $\gamma = 1$, $\eta = 3$.

It is worth noting that the EXPLAIN algorithm does not outperform the STRICT algorithm in Figure 5. Maybe it is because the hypotheses of learners are generated randomly. Since Aodha et al.(2018) did not conduct experiments on simulated learners, we cannot compare our results with theirs. But the effectiveness of EXPLAIN is verified in the real human experiments.

	RANDOM	STRICT	EXPLAIN	ALTA
Butterfly	67.50	64.82	59.56	**70.18**
Chinese	53.72	56.43	58.63	**61.32**
Woodpecker	70.14	74.32	73.45	**78.40**

Table 1: The accuracy of MTurk workers on the test images.

	RANDOM	STRICT	EXPLAIN	ALTA
Time (seconds)	118.63	118.32	127.68	120.33

Table 2: The average time that each strategy cost to teach MTurk workers with 10 images.

Results on MTurk Workers We generated the teaching set by teaching a simulated learner with random hypothesis space. Then we conducted three tasks on AMT. Each task contains 120 questionnaires of annotating images from one of the three datasets in Figure 3. There are four kinds of questionnaires (30 questionnaires each) generated from the four algorithms. We paid $ 0.3 for each questionnaire for the first task and $ 0.1 for each questionnaire for the other two tasks. The results are shown in Table 1.

From Figure 1 we can see that ALTA has the best performance in all the tasks. In the experiment on the butterfly dataset, there is a weird result that the random algorithm outperforms the other two algorithms. We guess it is because the random algorithm had chosen a good teaching set by chance. From the results of the woodpecker task, we can see that STRICT outperforms EXPLAIN. This is because the woodpecker dataset is too small, the explanations generated by CNN has low quality. The results show that our review mechanism and ALTA teaching algorithm are efficient in teaching human learners.

Is our new teaching form more time-consuming? We calculated the average time we cost to teach a learner. Table 2 shows the average time that the MTurk workers take to learn the teaching set with 10 images. There is no significant difference between our strategy and other strategies.

6 Conclusion

In this work, we propose a new form of teaching human learners by adding a review mechanism which helps learners to choose the examples they need to learn further. With this teaching form, we model the learning process of active learners and design a greedy teaching algorithm ALTA. Our experiments on both simulated learners and Amazon Mechanical Turk workers demonstrate the effectiveness and robustness of our approach. Our approach can be used to boost the performance of human learners in a wide range of applications, e.g. online crowdsourcing and MOOCs.

This work mainly concerns binary classification tasks. In future, we plan to extend our leaner model to support more complex learning tasks, e.g. multi-classification tasks.

Acknowledgements

This work was supported by National Natural Science Foundation under Grant No.(61932007,61972013). We thank our colleague Lei Chai for his help on the proofreading.

References

Adler, P.; and Clark, K. 1991. Behind the Learning Curve: A Sketch of the Learning Process. *Management Science* 37: 267–281.

Anthony, M.; Brightwell, G.; Cohen, D.; and Shawe-Taylor, J. 1992. On Exact Specification by Examples. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, COLT '92, 311–318. New York, NY, USA: Association for Computing Machinery. ISBN 089791497X.

Aodha, O. M.; Su, S.; Chen, Y.; Perona, P.; and Yue, Y. 2018. Teaching Categories to Human Learners With Visual Explanations. In *CVPR*, 3820–3828. IEEE Computer Society.

Corbett, A. T.; and Anderson, J. R. 1994. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction* 4(4): 253–278.

Dasgupta, S.; Hsu, D.; Poulis, S.; and Zhu, X. 2019. Teaching a black-box learner. volume 97 of *Proceedings of Machine Learning Research*, 1547–1555. Long Beach, California, USA: PMLR.

Dua, D.; and Graff, C. 2017. UCI Machine Learning Repository. URL http://archive.ics.uci.edu/ml. Accessed on 2020-06-20.

Fioretti, G. 2007. The organizational learning curve. *European Journal of Operational Research* 177(3): 1375 – 1384. ISSN 0377-2217.

Goldman, S.; and Kearns, M. 1995. On the Complexity of Teaching. *J. Comput. Syst. Sci.* 50(1): 20–31. ISSN 0022-0000.

Horn, G. V.; Aodha, O. M.; Song, Y.; Shepard, A.; Adam, H.; Perona, P.; and Belongie, S. J. 2017. The iNaturalist Challenge 2017 Dataset. *CoRR* abs/1707.06642. URL http://arxiv.org/abs/1707.06642.

Kulick, J.; Toussaint, M.; Lang, T.; and Lopes, M. 2013. Active Learning for Teaching a Robot Grounded Relational Symbols. In *IJCAI*, 1451–1457. ISBN 9781577356332.

Liu, C.; Yin, F.; Wang, D.; and Wang, Q. 2011. CASIA Online and Offline Chinese Handwriting Databases. In *2011 International Conference on Document Analysis and Recognition*, 37–41.

Liu, J.; Hou, X.; and Tang, F. 2020. Fine-Grained Machine Teaching with Attention Modeling. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020*, 2585–2592. AAAI Press.

Nemhauser, G.; Wolsey, L.; and Fisher, M. 1978. An Analysis of Approximations for Maximizing Submodular Set Functions—I. *Mathematical Programming* 14: 265–294.

Peltola, T.; Çelikok, M. M.; Daee, P.; and Kaski, S. 2019. Machine Teaching of Active Sequential Learners. In *Neural Information Processing Systems 2019, NeurIPS 2019*, 11202–11213.

Piech, C.; Bassen, J.; Huang, J.; Ganguli, S.; Sahami, M.; Guibas, L. J.; and Sohl-Dickstein, J. 2015. Deep Knowledge Tracing. In Cortes, C.; Lawrence, N. D.; Lee, D. D.; Sugiyama, M.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 28*, 505–513. Curran Associates, Inc.

Shinohara, A.; and Miyano, S. 1991. Teachability in computational learning. *New Generation Computing* 8(4): 337–347.

Singla, A.; Bogunovic, I.; Bartók, G.; Karbasi, A.; and Krause, A. 2014. Near-Optimally Teaching the Crowd to Classify. In *ICML*, volume 32 of *JMLR Workshop and Conference Proceedings*, 154–162. JMLR.org.

Vits, J.; and Gelders, L. 2002. Performance improvement theory. *International Journal of Production Economics* 77(3): 285 – 298. ISSN 0925-5273.

Wah, C.; Branson, S.; Welinder, P.; Perona, P.; and Belongie, S. 2011. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology.

Wang, Z.; Sun, H.; Fu, Y.; and Ye, L. 2017. Recommending crowdsourced software developers in consideration of skill improvement. In *ASE*, 717–722. IEEE Computer Society.

Wang, Z.; Sun, H.; and Han, T. 2020. Predicting Crowdsourcing Worker Performance with Knowledge Tracing. In *KSEM (2)*, volume 12275 of *Lecture Notes in Computer Science*, 352–359. Springer.

Zhang, J.; Shi, X.; King, I.; and Yeung, D. 2017. Dynamic Key-Value Memory Networks for Knowledge Tracing. In *WWW*, 765–774. ACM.

Zhou, Y.; Nelakurthi, A. R.; and He, J. 2018. Unlearn What You Have Learned: Adaptive Crowd Teaching with Exponentially Decayed Memory Learners. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018*, 2817–2826. ACM.

Zhu, X. 2015. Machine Teaching: An Inverse Problem to Machine Learning and an Approach Toward Optimal Education. In *AAAI*.

Zhu, X.; Singla, A.; Zilles, S.; and Rafferty, A. N. 2018. An Overview of Machine Teaching. *arXiv e-prints* arXiv:1801.05927.