Fine-Grained Machine Teaching with Attention Modeling

Jiacheng Liu, Xiaofeng Hou, Feilong Tang*
Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China
{liujiacheng, xfhelen}@sjtu.edu.cn, tang-fl@cs.sjtu.edu.cn

Abstract

The state-of-the-art machine teaching techniques overestimate the ability of learners in grasping a complex concept. On one side, since a complicated concept always contains multiple fine-grained concepts, students can only grasp parts of them during a practical teaching process. On the other side, because a single teaching sample contains unequal information in terms of various fine-grained concepts, learners accept them at different levels. Thus, with more and more complicated dataset, it is challenging for us to rethink the machine teaching frameworks. In this work, we propose a new machine teaching framework called Attentive Machine Teaching (AMT). Specifically, we argue that a complicated concept always consists of multiple features, which we call fine-grained concepts. We define *attention* to represent the learning level of a learner in studying a fine-grained concept. Afterwards, we propose AMT, an adaptive teaching framework to construct the personalized optimal teaching dataset for learners. During each iteration, we estimate the workers' ability with Graph Neural Network (GNN) and select the best sample using a pool-based searching approach. For corroborating our theoretical findings, we conduct extensive experiments with both synthetic datasets and real datasets. Our experimental results verify the effectiveness of AMT algorithms.

1 Introduction

In the last decades, the increasingly sophisticated dataset for training machine learning (ML) models necessitates the need for cultivating more specialized annotators. Namely, more annotators must acquire domain-specific knowledge to improve the performance of ML algorithms. Taking computer vision application as an example, At the early stage, image datasets of vision applications like CIFAR (Krizhevsky and Hinton 2010) are small and simple. Therefore, most of the annotators can complete the annotation tasks merely based on their knowledge of the classes from their everyday life (Daniel et al. 2018). With the advent of larger and more informative dataset like ImageNet (Russakovsky et al. 2015), annotators with specialized training facilitate the performance of ML models with more accurate classification. Nowadays, for more complicated datasets like Visual Genome (Krishna et al. 2017), the demand for high-performance machine learning (ML) models far exceeds the supply of specialized annotators. *Thus, extensively training annotators to acquire domain specific knowledge is needed*.

Recently, machine teaching emerges as a promising approach to enhance the specialized skills of normal annotators (Chen et al. 2018; Doroudi et al. 2016). There are mainly two categories of teaching techniques, the batch machine teaching and interactive machine teaching. The batch machine teaching assumes the learner using a hypothesis transition model based on observed feedback (Singla et al. 2014). Representing a new machine teaching paradigm, iterative machine teaching (IMT) selects teaching examples adaptive to the learning progress of the crowd (Liu et al. 2017). Sequentially, more intelligent machine teaching frameworks (Zhou, Nelakurthi, and He 2018) enable the teaching process to achieve better converge rate in theory and in practice. There are also massive applications (Dontcheva et al. 2014; Honnibal and Montani 2017), which adopt machine teaching to facilitate the annotation tasks.

However, the state-of-the-art machine teaching techniques are too optimistic, they overestimate the ability of learners in grasping a complex concept. In education systems (Ambrose et al. 2010; Alkhatlan and Kalita 2018), students cannot concentrate on the overall concept of a complicated item, which often contains multiple fine-grained concepts. Instead, they increasingly memorize partial concepts of this item. Taking the flower classification as an example, learners always notice a few features with every example rather than remember all the features including shape, texture, etc. Nevertheless, most of the existing machine teaching schemes overlook the fact that human beings can only obtain parts of a concept, which involves multiple features. Meanwhile, a single sample contains unequal information in terms of various features during the teaching process. Thus, learners accept the features of the whole concept at different levels. Summarily, current techniques are insufficient for modeling the teaching process of sophisticated concepts. With more and more complicated dataset, it is challenging for us to rethink the machine teaching frameworks.

To meet the requirement of increasingly complicated an-

*Feilong Tang is the corresponding author of this paper.

notation tasks, we first indicate that a complicated concept always consists of multiple features, which we call fine-grained concepts. Then, We define *attention*, which represents the learning level of a learner in studying a fine-grained concept. As discussed above, the *attention* of each fine-grained concept is different for the sake of learners' ability and teaching examples. According to education principle, human focus more on concept which is rarely learned than common concept (Ambrose et al. 2010). Thus, we model *attention* with the cumulative learning progress of each fine-grained concept. For optimization, we extend it to the information diminishing setting. Based on *attention* , we propose Attentive Machine Teaching (AMT), an adaptive teaching framework to construct the personalized optimal teaching set for learners in each iteration. We estimate the worker prediction with Graph Neural Network (GNN) and select the best sample using a pool-based searching approach. To corroborate our theoretical findings, we conduct extensive experiments with real dataset as well as synthetic dataset sample from Gaussian distribution. Our experimental results verify the effectiveness of AMT algorithms. It also shows that AMT can significantly improve the teaching quality through considering that the learners pay attention to the different fine-grained concepts.

Overall, we make the following contributions:

- We argue that a complicated concept always consists of multiple fine-grained concepts. We leverage *attention* to represent the cumulative learning progress of a learner in grasping each fine-grained concept.
- Based on *attention*, we propose Attentive Machine Teaching (AMT). AMT is an adaptive teaching framework for constructing the optimal teaching dataset considering the learners' attention on different fine-grained concepts.
- During each teaching iteration, we introduce a Graph Neural Network (GNN) based approach to estimate the learner's ability. Then select the best sample through using a pool-based searching approach.
- We conduct extensive experiments with real dataset as well as synthetic dataset sample from Gaussian distribution. Our experimental results show that AMT can significantly improve the teaching quality.

2 Related Work

2.1 Machine Teaching

Existing research works on machine teaching (Alfeld, Zhu, and Barford 2016; 2017) can be classified into two representative categories - batch machine teaching (Singla et al. 2014; Mac Aodha et al. 2018) and interactive machine teaching (Liu et al. 2017; Zhou, Nelakurthi, and He 2018). Typically, batch-based techniques focus on teaching the target concept to learners with the selected dataset in one shot. Thus, interaction-driven teaching presents a more practical process. Some of these works (Patil et al. 2014; Zhou, Nelakurthi, and He 2018) emphasize constructing the personalized optimal teaching set with consideration of the ability of different learners, such as Zhou et al. proposes JEDI teaching framework (Zhou, Nelakurthi, and He 2018), where each learner has an exponentially decayed memory. Other works like (Simard et al. 2017; Melo, Guerra, and Lopes 2018) concern the mismatch between teachers' assumption on the learners' performance and the actual performance of learners. The rest of works aim at improving the learning performance with different approaches (Mac Aodha et al. 2018; Chen et al. 2018). E.g., Chen et al. analyze the adaptivity with version space learner (Chen et al. 2018). Differently, we investigate a newly machine teaching framework which facilitates the learning of increasingly complicated items.

2.2 Graph-based Semi-Supervised Learning

As one of the most effective semi-supervised learning algorithms, graph-based semi-supervised learning often gather the information of labels throughout a graph via the form of explicit graph-based regularization (Zhu, Ghahramani, and Lafferty 2003). Recently, the graph neural network (GNN) becomes a promising approach in this field (Wu et al. 2019; Zhang, Cui, and Zhu 2018; Zhou et al. 2018; Bronstein et al. 2017). This approach has adopted in many applications (Yao, Mao, and Luo 2019; Tang, Zhang, and Yang 2019; Tang, Zhang, and Li 2018; Tang 2019), such as GNN is used to encode the syntactic structure of sentences in machine translation (Bastings et al. 2017). Our work differs from them in that we first combine GNN with machine teaching for maintain the consistency of learners' hypothesis and teachers'.

3 The Preliminaries of AMT

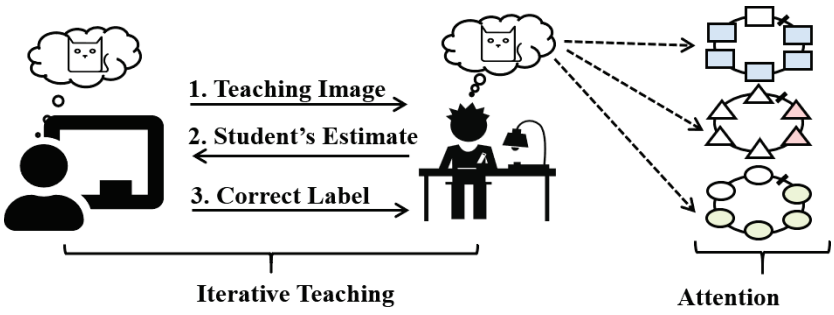
3.1 Attention on Fine-grained Concepts

Generally, a complicated concept often contains multiple fine-grained concepts. As shown in Figure 1(a), the teacher teaches the learners one image, i.e., one concept at a time. It begins by showing them an image from the larger labeled image dataset while concealing the true class label. The learners/students respond with their estimate of the image's class. In an actual teaching process, students cannot grasp the overall concept. Instead, they increasingly memorize its fine-grained concepts presented as the *Attention* part in Figure 1(a). After obtaining the student's answer, the teacher updates his estimation of the student. Finally, the student will be given the true label and learn from it. This process is repeated with more images until teaching ends.

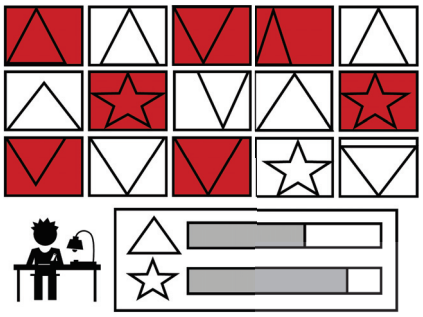
Meanwhile, as we can see in Figure 1(b), There are two classes of fine-grained concepts respectively represented as triangle and pentagram. A learner will see several selected examples marked with red color and continuously update his inner concept. The progress bar presents the degree of a learner's attention on different fine-grained concepts. In this case, the pentagram fine-grained concept attracts more attention than the triangle since human focus more on the concept which is rarely learned than common concept (Ambrose et al. 2010). All the above phenomena are ignored by the existing machine teaching literature.

3.2 Problem Formulation

Without loss of generality, we consider the teaching in a binary classification task. All the formulations in our proposal can be easily adapted to other settings. Our attentive



(a) Different learning progress of fine-grained concepts.



(b) Unequal information in training example.

Figure 1: Different attentions on fine-grained concepts. In an actual teaching process, students cannot grasp the overall concept. Instead, they increasingly memorize its fine-grained concepts due to: (a) a learner accepts different knowledge of various fine-grained concepts; (b) the training example contains unequal information on various fine-grained concepts.

machine teaching problem is formulated as follows. We leverage $\mathbf{X} \in \mathbb{R}^m$ to represent the m dimensional features of a complicated item. These features are related to the m fine-grained concepts of the target concept $\mathbf{w}$. The target concept is also a m dimensional vector, namely $\mathbf{w} \in \mathbb{R}^m$.

Model of Teachers: In our teaching framework, we assume the teacher knows the target concept $\mathbf{w}^*$. But it is impossible for him to directly teach it to the students. Thus, the target concept $\mathbf{w}^*$ is taught with a set of teaching samples from a *teaching pool* denoted by $\mathcal{T}$. The selected teaching sample in iteration t is expressed with a pair of $(\mathbf{x}^t, y^t)$ from $\mathcal{T}$, where $\mathbf{x}^t \in \mathbf{X}$ and $y^t \in \{-1, +1\}$. Thus, the mission of the teacher is to properly select the teaching sample in each iteration that can minimize the following objective function.

$$\underset{(\mathbf{x}^t, y^t) \in \mathcal{T}}{\operatorname{argmin}} \left\| \mathbf{w}^t - \mathbf{w}^* \right\|_2^2 . \tag{1}$$

Model of Learners: We assume the student use a linear inference model $y = \langle \mathbf{w}, \mathbf{x} \rangle$, where parameter $\mathbf{w}$ is similar to (Liu et al. 2017). And we assume that the annotator has a convex loss function $\mathcal{L}(f(x), y)$. The goal of the teaching is to find a concept $\mathbf{w}^*$ which minimizes this loss function in a given distribution $\mathbb{P}(\mathbf{x}, y)$ as shown in Eq. (2),

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \mathbb{E}_{(\mathbf{x},y)\sim\mathbb{P}(\mathbf{x},y)} \left[\mathcal{L}(\langle \mathbf{w}, \mathbf{x} \rangle, y) \right] . \tag{2}$$

Without loss of generality, we consider the logistic loss (Hosmer Jr, Lemeshow, and Sturdivant 2013) expressed as Eq. (3) in the following. It is one of the typical convex loss functions.

$$\mathcal{L} = \log(1 + \exp(-y \langle \mathbf{w}, \mathbf{x} \rangle)) , \tag{3}$$

Meanwhile, similar to previous works, we assume that the student use a learning algorithm based on the gradient descent algorithm (Ruder 2016). In iteration $t+1$, the update rule is,

$$\mathbf{w}^{t+1} = \mathbf{w}^t - \eta \frac{\partial \ell(\langle \mathbf{w}^t, \mathbf{x}^{t+1} \rangle, y^{t+1})}{\partial \mathbf{w}^t} . \tag{4}$$

4 Attention Machine Teaching

4.1 Formalizing the Attention Factor

The attention factor reflects the learner's learning progress of massive fine-grained concepts, which is defined as follows.

Definition 1 (Attention Factor)**.** *In iteration t, the attention factor $\boldsymbol{\alpha}_t$ is defined as the inverse of accumulated information in previous iterations,*

$$\boldsymbol{\alpha}_t = \frac{Z^t}{\sqrt{\boldsymbol{INFO}^t}} . \tag{5}$$

In the above definition, the square root is element-wise square root of the vector. Z^t is the normalized value of interpretation since the interpretation values are different under various scenarios. **INFO** is the accumulated information, which is computed in accordance with the following process.

Generally, the annotator is trained with a teaching example $(\mathbf{x}^t, y^t)$ at the t-th iteration. He reviews his preconceived concept $\mathbf{w}_{t-1}$ and compares it with what learned from this teaching example $(\mathbf{x}^t, y^t)$. If there exists a mismatch between $\mathbf{w}_{t-1}$ and $(\mathbf{x}^t, y^t)$, he update his concept with a gradient descent algorithm. Therefore, the information received by the student in iteration t is defined as,

$$\mathbf{info}^t = \frac{\partial \ell(\langle \mathbf{w}^{t-1}, \mathbf{x}^t \rangle, y^t)}{\partial \mathbf{w}^{t-1}} \odot \frac{\partial \ell(\langle \mathbf{w}^{t-1}, \mathbf{x}^t \rangle, y^t)}{\partial \mathbf{w}^{t-1}} , \tag{6}$$

where the $\odot$ is the element-wise product operation. We use this representation to make the information always positive. These are also similar to some optimization techniques (Duchi, Hazan, and Singer 2011; Kingma and Ba 2015). Then, we formulate the accumulated information until iteration t as,

$$\mathbf{INFO}^t = \sum_{p=1}^{t-1} \mathbf{info}^p , \tag{7}$$

It is remarkable that the learner often pays more attention to rare concepts in each teaching iteration. Therefore, we use the inverse value of the maximum $\mathbf{INFO}^{t-1}$ to computed the attention factor in Eq. (5). Based on the above analysis, formula (4) is updated to

$$\mathbf{w}^{t+1} = \mathbf{w}^t - \eta\,\boldsymbol{\alpha}^t \frac{\partial \ell(\langle \mathbf{w}^t, \mathbf{x}^{t+1}\rangle, y^{t+1})}{\partial w^t}\,. \tag{8}$$

In the following of the paper, to simplify the notation, we use ∇_t to denote the gradient of data sample $(\mathbf{x}^t, y^t)$, namely,

$$\nabla_t = \frac{\partial \ell(\langle \mathbf{w}^{t-1}, \mathbf{x}^t\rangle, y^t)}{\partial \mathbf{w}^{t-1}} = \frac{-y^t \mathbf{x}^t}{1+\exp\left(y^t \langle \mathbf{w}^{t-1}, \mathbf{x}^t\rangle\right)} \tag{9}$$

4.2 Information-Diminishing AMT

According to (Zhou, Nelakurthi, and He 2018), a human learner will forget partial information learned from the previous teaching examples because of the memory decay. The forgotten rate approximately follows an exponential curve (Loftus 1985). Therefore, it is more practical to model the accumulated information with consideration of the forgotten information. It is a combination of previous accumulated information and information gotten in iteration t,

$$\mathbf{INFO}^t = \beta \mathbf{INFO}^{t-1} + \mathbf{info}^t\,, \tag{10}$$

where $\beta \in (0,1)$ is the forget parameters for memory decay. Expanding this formula, we can get the following equation,

$$\mathbf{INFO}^t = \beta^t \mathbf{INFO}^0 + \sum_{p=1}^{t} \beta^{t-p} \mathbf{info}^t\,, \tag{11}$$

where the $\mathbf{INFO}^0$ is always zero for an annotator with no background knowledge. Meanwhile, the learning procedure is guide by the concept momentum (Zhou, Nelakurthi, and He 2018).

$$\boldsymbol{v}^t = \gamma^t \boldsymbol{v}^0 + \sum_{p=1}^{t} \gamma^{t-p} \frac{\partial \ell(\langle \mathbf{w}^{p-1}, \mathbf{x}\rangle, y^p)}{\partial \mathbf{w}^{p-1}}\,.$$

In the above equations, the γ is the individual memory decay rate. The concept momentum in iteration 0 is always set to 0, namely, $\boldsymbol{v}^0 = \mathbf{0}$. Then the learning algorithm turns into

$$\mathbf{w}^t = \mathbf{w}^{t-1} - \eta \boldsymbol{\alpha}^t \boldsymbol{v}^t\,. \tag{12}$$

4.3 Objective Analysis

Then in each iteration, we want to minimize the objective function, the decomposition is expressed as follows,

$$\begin{aligned}
&\left\|\mathbf{w}^{t+1} - \mathbf{w}^*\right\|_2^2 = \left\|\mathbf{w}^t - \eta \boldsymbol{\alpha}_t\, \boldsymbol{v}^{t+1} - \mathbf{w}^*\right\|_2^2 \\
&= \underbrace{\left\|\mathbf{w}^t - \mathbf{w}^*\right\|_2^2}_{T_1:\text{Discrepancy in iter. } t} + \eta^2 \underbrace{\left\| \frac{Z^t \sum_{p=1}^t \gamma^{t-p} \nabla_p}{\sqrt{\sum_{p=1}^t \beta^{t-p} \langle \nabla_p, \nabla_p\rangle}} \right\|_2^2}_{T_2:\text{Attentive Hardness in iter. } t} \\
&\quad \underbrace{- 2\eta \left\langle \mathbf{w}^t - \mathbf{w}^*, \frac{Z^t \sum_{p=1}^t \gamma^{t-p} \nabla_p}{\sqrt{\sum_{p=1}^t \beta^{t-p} \langle \nabla_p, \nabla_p\rangle}} \right\rangle}_{T_3:\text{Attentive Diversity in iter. } t}.
\end{aligned} \tag{13}$$

The first part is the discrepancy between the target concept $\mathbf{w}^*$ and the learned previous concept $\mathbf{w}^t$. The second part models the attentive hardness of the teaching sequence. The third part is the attentive diversity of the teaching sequence.

Algorithm 1 Attentive Machine Teaching Algorithm (AMT)

Input: Initial concept $\mathbf{w}^0$, target concept $\mathbf{w}^*$, learning rate η, teaching pool $\mathcal{T}$, maximum teaching iteration maxIter, converge threshold ϵ
Set $t \leftarrow 0$;
while $\|\mathbf{w}^t - \mathbf{w}^*\| \geq \epsilon$ and $t <$ maxIter **do**
  Select the teaching sample $(\mathbf{x}^t, y^t)$ by solving Eq. (14)
  Student annotate teaching sample $\mathbf{x}^t$;
  Teacher reveal the label y^t and student perform learning based on Eq. (12)
  $t \leftarrow t+1$
end while

4.4 Teaching Algorithm

Since the T_1 part is a constant in iteration t, final object is,

$$\operatorname*{argmin}_{(x,y)\in\mathcal{T}} \eta^2 T_2 - 2\eta T_3\,. \tag{14}$$

Based on these observations, we propose a information-diminishing AMT as shown in Algorithm 1. By giving the annotator's initial concept, the target concept and the learning rate, this algorithm will select the teaching sample to optimize the objective function in Eq. (14) iteratively.

Aforementioned teaching is based on the omniscient teacher who knows everything about the learner. However, it is difficult for the teacher to obtain every information of the learners. Therefore we propose the blackbox AMT algorithm for this more challenging scenario.

5 Blackbox AMT

In this section, we propose to solve blackbox AMT problem. Firstly, we use the convexity to bypass the unreachable concept, then a GNN-based learning algorithm is adopt to estimate the learner's master degree of the target concept.

5.1 Unreachable Concept

In annotator teaching, we cannot get the annotator's concept, while we can get the output of prediction $\langle \mathbf{w}^t, \mathbf{x}\rangle$. Thus we use the convexity of the loss function to help this process (Liu et al. 2017). That is,

$$\left\langle \mathbf{w}^t - \mathbf{w}^*, \frac{\partial \ell(\langle \mathbf{w}^t, \mathbf{x}\rangle, y)}{\partial \mathbf{w}^t} \right\rangle \geqslant \mathcal{L}(\langle \mathbf{w}^t, \mathbf{x}\rangle, y) - \mathcal{L}(\langle \mathbf{w}^*, \mathbf{x}\rangle, y)\,.$$

By using this equation, we can replace it with its lower bound. Thus, we can only need the output $\langle \mathbf{w}^t, \mathbf{x}\rangle$ from the student instead of directly accessing the concept $\mathbf{w}^t$. In addition, when teaching continues, $\|\mathbf{w}^t - \mathbf{w}^*\|$ is becoming smaller, thus this approximation becomes much more accurate.

5.2 Estimation of Student's Performance

Since it is impossible to get student's prediction $\langle \mathbf{w}, \mathbf{x}\rangle$ for every data sample in $\mathcal{T}$, The teacher need estimate the student's prediction based on his past answers. Due to the limit of the labeled examples training a supervised model to imitate the student is difficult, so there are some researches (Zhou, Nelakurthi, and He 2018; Johns, Mac Aodha, and Brostow

2015) adopting the semi-supervised learning method in Gaussian fields and Harmonic functions proposed in (Zhu, Ghahramani, and Lafferty 2003) to estimate this. However, these estimations only use the similarity of data. It discards most of the information in the feature space. Also, it's hard to scale to a larger dataset due to the computational expensive matrix manipulate operations. These all limit its effectiveness in recent teaching tasks.

Recently, the GNN-based approach has gained promising result in semi-supervised learning task. It has shown its ability on large scale dataset (Ying et al. 2018). We firstly construct the graph using the teaching set in the KNN graph way. The distance is defined using the RBF kernel.

$$\text{distance} = \exp\left(-\gamma \left\|\mathbf{x}_i - \mathbf{x}_j\right\|_2^2\right) . \tag{15}$$

Based on this distance measurement, we can construct a graph $G = (\mathbf{V}, \mathbf{E})$ where each node $v \in \mathbf{V}(|\mathbf{V}| = n)$ represents a teaching sample in $\mathcal{T}$, and each edge $e \in \mathbf{E}$ represents the similarity relationships. Also all the features of the teaching dataset compose a feature matrix $\mathbf{X} \in R^{n\times m}$, where m is the dimension of feature vector. Then, we construct a adjacent matrix $\mathbf{A}$, and add self-loop to it $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}_N$. we use a two-layer GCN network to predict the label of unseen data (Kipf and Welling 2017), where the second layer shares the same dimension as the first one. We lastly add a $\mathrm{softmax}$ to get the probabilistic result.

$$\mathbf{Z} = \mathrm{softmax}\left(\mathbf{L}\,\mathrm{ReLU}\left(\mathbf{L}\mathbf{X}\mathbf{W}_0\right)\mathbf{W}_1\right) , \tag{16}$$

where $\mathbf{L}$ is the symmetric graph Laplacian. It can be calculated as $\mathbf{L} = \mathbf{D}^{-\frac{1}{2}}\tilde{\mathbf{A}}\mathbf{D}^{\frac{1}{2}}$, where $\mathbf{D}$ is the degree matrix with $\mathbf{D}_{ii} = \sum_j \tilde{\mathbf{A}}_{ij}$. The $\mathrm{softmax}$ is applied in row wise. Next, we use the cross-entropy error over all labeled teaching samples $\mathcal{T}_l$ as the loss function,

$$\mathcal{L} = -\sum_{l\in\mathcal{T}_L}\sum_{f=1}^{D}\mathbf{Y}_{lf}\ln\mathbf{Z}_{lf} , \tag{17}$$

where $\mathcal{T}_L$ is the set of teaching samples already answered by students. D is the dimension of the output features. It equals to the number of classes (as discussed in Section 1). $\mathbf{Y}$ is the label indicator matrix. The weights $\mathbf{W}_0$ and $\mathbf{W}_1$ can be trained via gradient descent. After the training, we can get the prediction from the network as $p(y|\mathbf{x})$. Then, based on the error prediction rate of probability, we can estimate the prediction on unseen data as,

$$\langle \mathbf{w}, \mathbf{x}\rangle = \frac{1}{y}\log\frac{1-p(y|\mathbf{x})}{p(y|\mathbf{x})} . \tag{18}$$

5.3 Teaching Algorithm

The detail of the teaching algorithm in is proposed in Algorithm 2. This algorithm works as follows, Firstly, it set initial value for the parameters and randomly select sample to start the teaching process (Line 1-3). Then, we construct a KNN graph, and estimate the prediction of the student on unseen data using GNN (Line 4-7). After that, teaching sample is selected by solving the objective function or randomly (to escape from the bad estimation in the beginning) (Line 8-12). Finally, we release the teaching sample to student and perform the update of labels (Line 13-16).

Algorithm 2 Blackbox AMT Algorithm

1: **Input:** Initial concept $\mathbf{w}^0$, target concept $\mathbf{w}^*$, learning rate η, teaching pool $\mathcal{T}$, maximum teaching iteration maxIter, converge threshold ϵ, Random selection threshold α and T.
2: Randomly select one training sample $(\mathbf{x}^1, y^1)$ and get the prediction from student.
3: Set $t \leftarrow 1$;
4: **while** $\|\mathbf{w}^t - \mathbf{w}^*\| \geq \epsilon$ or $t <$ maxIter **do**
5: Construct the KNN graph of the data in $\mathcal{T}$ with Eq. (15)
6: Train a GNN defined in Eq. (16) with loss function defined in Eq. (17)
7: Estimate the prediction of the student using Eq. (18)
8: **if** random variable $\sigma \leq \alpha^t$ and $t \leq T$ **then**
9: Random select teaching sample with probability p.
10: **else**
11: Select the teaching sample $(\mathbf{x}^{t+1}, y^{t+1})$ by solving Eq. (14) with the estimate values
12: **end if**
13: Student annotate teaching sample $\mathbf{x}^{t+1}$;
14: Teacher reveal the label y^{t+1} and student perform learning based on Eq. (12)
15: $t \leftarrow t + 1$
16: **end while**

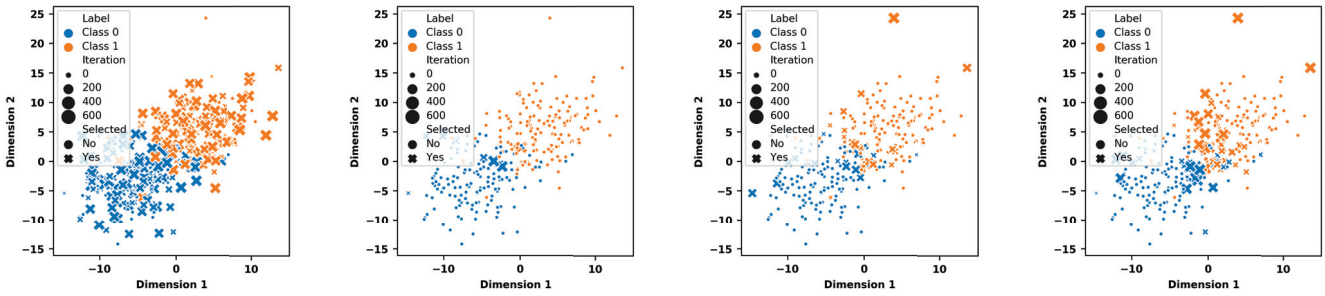
6 Performance Evaluation

6.1 Experiment Setup

Baseline teaching methods: In order to analyze the proposed AMT strategy, we compare AMT with benchmark and the state-of-the-art machine teaching methods. Firstly, We consider SGD, a naive case that the student learns without the teacher's guidance. In this case, the student can be viewed as being guided by a random teacher who randomly feeds an example to the student in each iteration. Secondly, we consider Iterative Machine Teaching (IMT) (Liu et al. 2017), which models the learning process with gradient descent, and only consider basic teaching settings. Lastly, we compare with Adaptive Crowd Teaching with Exponentially Decayed Memory Learners (JEDI) (Zhou, Nelakurthi, and He 2018) which models the learning process with concept momentum. To guarantee a fair comparison, all these methods are tested using the same initial concept and learning rate.

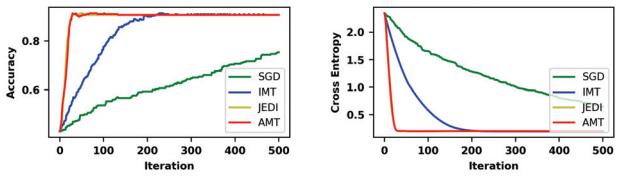
Teaching tasks: In this paper we consider the following teaching tasks to validate the teaching methods. The first is 2D Gaussian dataset, which is a synthetic two-dimensional dataset drawn from Gaussian distribution for selected sample visualization. The second is 10D and 100D Gaussian dataset, which is a synthetic dataset for validating the effectiveness of the proposed method in medium and high dimensions. The third is the hate speech detection dataset (Davidson et al. 2017), which contains three categories including "hate speech", "offensive language" and "neither". We select the first two to form a binary classification problem.

Evaluation metrics: To measure the effectiveness, we use two evaluation metrics: (1) The accuracy of the student model. (2) The value of the objective function. We use the



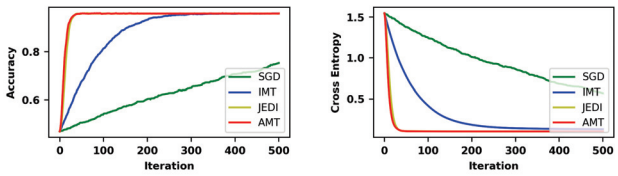
(a) SGD (b) IMT (c) JEDI (d) AMT

Figure 2: The unique examples in the teaching sequences of SGD, IMT, JEDI and AMT.



(a) Comparison of performance (b) Comparison of obj. value

Figure 3: Performance on 2D Gaussian.



(a) Comparison of performance (b) Comparison of obj. value

Figure 4: Performance on 10D Gaussian

cross entropy as a measurement of the learning degree.

6.2 Selection of Teaching Samples

In this paragraph, we firstly show the selected teaching samples of different algorithms. We test these teaching schemes with 2D Gaussian mixture dataset. The data is generated from the following distribution.

$$\begin{aligned} p_+(\mathbf{x}) &= \frac{2}{3}\mathcal{N}\left(\mathbf{x}|\mu_1,\Sigma_1\right) + \frac{1}{3}\mathcal{N}\left(\mathbf{x}|\mu_2,\Sigma_2\right) \\ p_-(\mathbf{x}) &= \frac{2}{3}\mathcal{N}\left(\mathbf{x}|\mu_3,\Sigma_1\right) + \frac{1}{3}\mathcal{N}\left(\mathbf{x}|\mu_4,\Sigma_2\right) \ , \end{aligned} \tag{19}$$

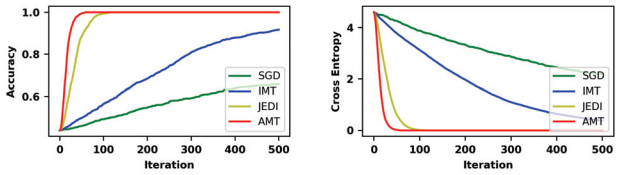
The parameters are, $\mu_1 = [0, 8]$, $\mu_2 = [8, 0]$, $\mu_3 = [-8, 0]$, $\mu_4 = [0, -8]$, $\Sigma_1 = [[36, 8], [8, 36]]$,$\Sigma_2 = [[10, 5], [5, 10]]$. For each class we generate 150 data samples. The learning rate is set to 0.004 and the initial concept w^0 for all the schemes is set with the standard normal distribution. In order to get the target concept, we train a logistic regression model and get the weight as the target concept w^*.For clarity, we remove the redundant teaching examples.

As shown in Figure 2, the two color types mean two classes (class 1 and class 2). The dimension 1 and 2 represent two fine-grained concepts considered in this teaching task. The cross and circle shape respectively present whether the sample is chosen by the teaching algorithm. For example, orange cross means that the chosen data is class 1. Besides, we use different size of the shape to represent the example is selected in which iteration, the larger size of a cross or a circle presents it is selected in later iterations. According to Figure 2(a), it is obvious that the SGD algorithm selects plethoric teaching samples, which almost covers the overall dataset. Additionally, the order of the selection is randomly thus it will cause the learning process hard to converge. The IMT algorithm chooses much less teaching samples than the other methods as shown in Figure 2(b). It always selects the same training examples (the results of prior iteration is hidden by the latter one) at each iteration. This can severely overlook the learners' attention on various fine-grained concepts. Figure 2(c) demonstrates that JEDI can select more informative and diverse teaching samples. This is mainly due to the use of concept momentum of learners. Reversely, our proposal can adaptively adjust the teaching example according to the learners' attention on various fine-grained concepts.

6.3 Evaluation on Extensive Datasets

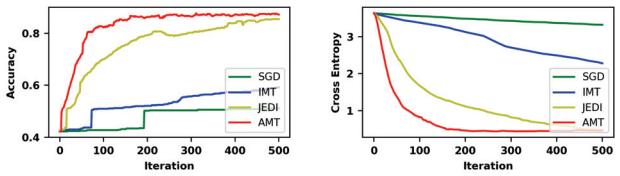
The above observation has shown that different teaching algorithms leverage different teaching samples to teach the learners. It is hard to conclude a larger and more complicate set of teaching samples is better or not. Therefore, we compare the accuracy and convergence speed of these algorithms with their selected dataset. In this section, we consider 4 datasets as shown in Section *6.1*. The AMT algorithm leverages the protocol as shown in Algorithm 1. We conduct the experiment with the 2D, 10D and 100D Gaussian dataset. The 10D and 100D Gaussian dataset in this section has a distribution of (0.5,$\cdots$,0.5) (label: $+1$) and (-0.5,$\cdots$,-0.5) (label: -1) as mean. Its covariance matrix is the identity matrix. We generate 1000 training data points for each label. The results is shown in Figure 3, 4 and 5.

For 2D Gaussian case, we find that AMT teaching algorithm obviously converges fastest within 30 iterations. In Figure 3(a), we observe that the students learn faster with a



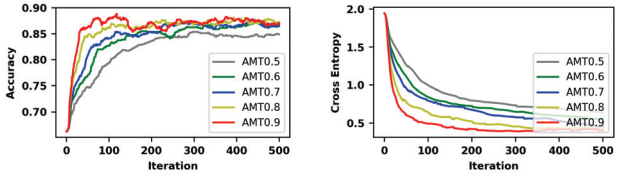
(a) Comparison of performance (b) Comparison of obj. value

Figure 5: Performance on 100D Gaussian



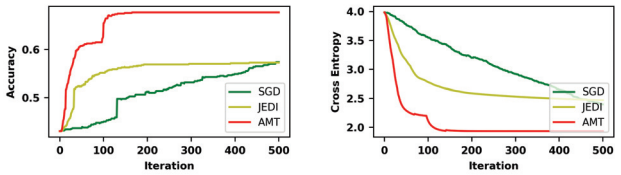
(a) Comparison of performance (b) Comparison of obj. value

Figure 6: Performance on Hate Speech Detection.



(a) Comparison of performance (b) Comparison of obj. value

Figure 7: Impact of Memory Parameters.



(a) Comparison of performance (b) Comparison of obj. value

Figure 8: Blackbox AMT in Hate Speech Detection.

teacher than without the teacher (SGD). It also shows that the accuracy of AMT is highest among the compared methods. Although the result of JEDI algorithm is similar to the AMT's, it takes more iteration to converge. Figure 3(b) illustrates that AMT has the smallest value of the objective function, which means that it teaches the target concept to the learners most accurately. Furthermore, the teaching procedure is more steady for AMT, that means the teaching sample selection is more suitable.

For 10D case, we observe the similar trends in 2D Gaussian data, however the differences between JEDI and AMT is larger. Then in 100D Gaussian data, we observe that AMT converges fastest with no more than 50 iterations with the accuracy over 98%. While JEDI needs more than 100 iterations to reach the same performance. Summarily, with the classification tasks becoming more and more complicated, AMT always performs better through considering the attention of the learners on various fine-grained concepts, while others overlook the actual learning progress of the learners.

6.4 Hate Speech Classification

To test our proposed method in a harder text classification problem, we choose hate speech detection. As a major threat on the web, hate speech has been especially prevalent in online forums, chatrooms, and social media. The difficulty of hate speech detection largely depends on the domain and the context, and it's quite a subject, which means a seemingly neutral sentence can be offensive for one person and not bother another. We use 20% of the data as the testing dataset. And we use the common techniques to preprocess the dataset including: remove the stop words and convert all characters to lowercase before tokenizing. Finally, we build a vocabulary that only considers the top 1000 features ordered by term frequency across the corpus. The final feature value is calculated through the TF-IDF. As we can see in Figure 6, AMT performs better than the other schemes in both accuracy and objective value, and the gaps of different teaching strategy is significant.

6.5 Memory Size Analysis

In the previous experiments, we set the memory parameters β and γ to 0.9 for simplicity. In this section we analyze the impact of multiple memory parameters, namely different memory size. The selected memory parameters in this part contains 0.5, 0.6, 0.7, 0.8 and 0.9. As seen in Figure 7, we plot the results of the proposed AMT algorithm with different memory size using the hate speech detection dataset. As we can see in these results, the learner with larger memory will learn faster which is similar to the result in (Zhou, Nelakurthi, and He 2018).

6.6 Blackbox AMT

In this section, we conduct experiments on the more challenging teaching in the blackbox scenario. The algorithm used for AMT is proposed in Algorithm 2. Since the original IMT algorithm is not supported for estimating the prediction of students, we omit it in this comparison. As we can see in Figure 8, the proposed AMT algorithm performs well at the latter time. It converges the fastest. In the beginning, due to the lack of the labeled data, the estimation of student's performance in different data sample is inaccurate. Sometimes this may lead to the bad choice (maybe even worse than the random teacher). Besides, we find that the GNN-based estimation is fast enough in this task. For larger graph it is also faster than the label propagation approach (3-10x faster in our experiments). Also we found that the blackbox setting is much more challenging, the teaching will sometimes fall into the bad direction due to inaccurate estimation.

7 Conclusion

In this paper, we propose the Attentive Machine Teaching (AMT) framework that models the annotator's attention of

learning a complicated concept. Base on the concept modeling, we propose a teaching algorithm that addresses the teaching problem in real world teaching scenario. The work in this paper is promising in modeling the real learning ability of human learners, that help the machine teaching techniques apply to real world teaching.

8 Acknowledgements

This work was supported in part by the National Natural Science Foundation of China projects (Nos. 61832013 and 61672351), in part by STCSM (Science and Technology Commission of Shanghai Municipality) AI project (No. 19511120300), in part by the National Key Research and Development Program of China (No. 2019YFB2102200), and in part by the Huawei Technologies Co., Ltd projects (Nos. YBN201905075 and YBN2019075061). Feilong Tang is the corresponding author of this paper.

References

Alfeld, S.; Zhu, X.; and Barford, P. 2016. Data poisoning attacks against autoregressive models. In *AAAI*.

Alfeld, S.; Zhu, X.; and Barford, P. 2017. Explicit defense actions against test-set attacks. In *AAAI*.

Alkhatlan, A., and Kalita, J. K. 2018. Intelligent tutoring systems: A comprehensive historical survey with recent developments. *CoRR*.

Ambrose, S. A.; Bridges, M. W.; DiPietro, M.; Lovett, M. C.; and Norman, M. K. 2010. *How learning works: Seven research-based principles for smart teaching*. John Wiley & Sons.

Bastings, J.; Titov, I.; Aziz, W.; Marcheggiani, D.; and Sima'an, K. 2017. Graph convolutional encoders for syntax-aware neural machine translation. In *EMNLP*.

Bronstein, M. M.; Bruna, J.; LeCun, Y.; Szlam, A.; and Vandergheynst, P. 2017. Geometric deep learning: Going beyond euclidean data. *IEEE Signal Processing Magazine* 34(4):18–42.

Chen, Y.; Singla, A.; Aodha, O. M.; Perona, P.; and Yue, Y. 2018. Understanding the role of adaptivity in machine teaching: The case of version space learners. In *NeurIPS*.

Daniel, F.; Kucherbaev, P.; Cappiello, C.; Benatallah, B.; and Allahbakhsh, M. 2018. Quality control in crowdsourcing: A survey of quality attributes, assessment techniques, and assurance actions. *CSUR* 51(1):7.

Davidson, T.; Warmsley, D.; Macy, M.; and Weber, I. 2017. Automated hate speech detection and the problem of offensive language. In *ICWSM*.

Dontcheva, M.; Morris, R. R.; Brandt, J. R.; and Gerber, E. M. 2014. Combining crowdsourcing and learning to improve engagement and performance. In *CHI*.

Doroudi, S.; Kamar, E.; Brunskill, E.; and Horvitz, E. 2016. Toward a learning science for complex crowdsourcing tasks. In *CHI*.

Duchi, J. C.; Hazan, E.; and Singer, Y. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *JMLR* 12(Jul):2121–2159.

Honnibal, M., and Montani, I. 2017. Prodigy: A new tool for radically efficient machine teaching. *https://explosion.ai/blog/prodigy-annotation-tool-active-learning*.

Hosmer Jr, D. W.; Lemeshow, S.; and Sturdivant, R. X. 2013. *Applied logistic regression*. John Wiley & Sons.

Johns, E.; Mac Aodha, O.; and Brostow, G. J. 2015. Becoming the Expert - Interactive Multi-Class Machine Teaching. In *CVPR*.

Kingma, D. P., and Ba, J. 2015. Adam: A method for stochastic optimization. In *ICLR*.

Kipf, T. N., and Welling, M. 2017. Semi-supervised classification with graph convolutional networks. In *ICLR*.

Krishna, R.; Zhu, Y.; Groth, O.; Johnson, J.; Hata, K.; Kravitz, J.; Chen, S.; Kalantidis, Y.; Li, L.-J.; Shamma, D. A.; Bernstein, M.; and Fei-Fei, L. 2017. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *IJCV* 123(1):32–73.

Krizhevsky, A., and Hinton, G. 2010. Convolutional deep belief networks on cifar-10.

Liu, W.; Dai, B.; Rehg, J. M.; and Song, L. 2017. Iterative machine teaching. In *ICML*.

Loftus, G. R. 1985. Evaluating forgetting curves. *Journal of Experimental Psychology: Learning, Memory, and Cognition* 11(2):397.

Mac Aodha, O.; Su, S.; Chen, Y.; Perona, P.; and Yue, Y. 2018. Teaching categories to human learners with visual explanations. In *CVPR*.

Melo, F. S.; Guerra, C.; and Lopes, M. 2018. Interactive optimal teaching with unknown learners. In *IJCAI*.

Patil, K. R.; Zhu, J.; Kopeć, Ł.; and Love, B. C. 2014. Optimal teaching for limited-capacity human learners. In *NIPS*.

Ruder, S. 2016. An overview of gradient descent optimization algorithms. *CoRR*.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; Berg, A. C.; and Fei-Fei, L. 2015. ImageNet Large Scale Visual Recognition Challenge. *IJCV* 115(3):211–252.

Simard, P. Y.; Amershi, S.; Chickering, D. M.; Pelton, A. E.; Ghorashi, S.; Meek, C.; Ramos, G.; Suh, J.; Verwey, J.; Wang, M.; and Wernsing, J. R. 2017. Machine teaching: A new paradigm for building machine learning systems. *CoRR*.

Singla, A.; Bogunovic, I.; Bartók, G.; Karbasi, A.; and Krause, A. 2014. Near-optimally teaching the crowd to classify. In *ICML*.

Tang, F.; Zhang, H.; and Li, J. 2018. Joint topology control and stable routing based on pu prediction for multihop mobile cognitive networks. *TWC* 17:1713–1726.

Tang, F.; Zhang, H.; and Yang, L. T. 2019. Multipath cooperative routing with efficient acknowledgement for leo satellite networks. *TMC* 18:179–192.

Tang, F. 2019. Bidirectional active learning with gold-instance-based human training. In *IJCAI*.

Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; and Yu, P. S. 2019. A comprehensive survey on graph neural networks. *CoRR*.

Yao, L.; Mao, C.; and Luo, Y. 2019. Graph convolutional networks for text classification. In *AAAI*.

Ying, Z.; He, R.; Chen, K.; Eksombatchai, P.; Hamilton, W. L.; and Leskovec, J. 2018. Graph convolutional neural networks for web-scale recommender systems. In *KDD*.

Zhang, Z.; Cui, P.; and Zhu, W. 2018. Deep learning on graphs: A survey. *CoRR*.

Zhou, J.; Cui, G.; Zhang, Z.; Yang, C.; Liu, Z.; and Sun, M. 2018. Graph neural networks: A review of methods and applications. *CoRR*.

Zhou, Y.; Nelakurthi, A. R.; and He, J. 2018. Unlearn what you have learned: Adaptive crowd teaching with exponentially decayed memory learners. In *KDD*.

Zhu, X.; Ghahramani, Z.; and Lafferty, J. D. 2003. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*.