Iterative Machine Teaching

Weiyang Liu [1] **Bo Dai** [1] **Ahmad Humayun** [1] **Charlene Tay** [2] **Chen Yu** [2]
Linda B. Smith [2] **James M. Rehg** [1] **Le Song** [1]

Abstract

In this paper, we consider the problem of machine teaching, the inverse problem of machine learning. Different from traditional machine teaching which views the learners as batch algorithms, we study a new paradigm where the learner uses an iterative algorithm and a teacher can feed examples sequentially and intelligently based on the current performance of the learner. We show that the teaching complexity in the iterative case is very different from that in the batch case. Instead of constructing a minimal training set for learners, our iterative machine teaching focuses on achieving fast convergence in the learner model. Depending on the level of information the teacher has from the learner model, we design teaching algorithms which can provably reduce the number of teaching examples and achieve faster convergence than learning without teachers. We also validate our theoretical findings with extensive experiments on different data distribution and real image datasets.

1. Introduction

Machine teaching is the problem of constructing an optimal (usually minimal) dataset according to a target concept such that a student model can learn the target concept based on this dataset. Recently, there is a surge of interests in machine teaching which has found diverse applications in model compression (Bucila et al., 2006; Han et al., 2015; Ba & Caruana, 2014; Romero et al., 2014), transfer learning (Pan & Yang, 2010) and cyber-security problems (Alfeld et al., 2016; 2017; Mei & Zhu, 2015). Furthermore, machine teaching is also closely related to other subjects of interests, such as curriculum learning (Bengio et al., 2009) and knowledge distilation (Hinton et al., 2015).

[1]Georgia Institute of Technology [2]Indiana University. Correspondence to: Weiyang Liu <wyliu@gatech.edu>, Le Song <lsong@cc.gatech.edu>.

Proceedings of the 34^{th} International Conference on Machine Learning, Sydney, Australia, PMLR 70, 2017.

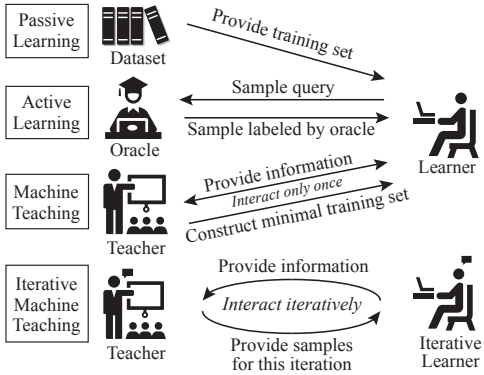

Figure 1. Comparison between iterative machine teaching and the other learning paradigms.

In the traditional machine learning paradigm, a teacher will typically construct a batch set of examples, and provide them to a learning algorithm in one shot; then the learning algorithm will work on this batch dataset trying to learn the target concept. Thus, many research work under this topic try to construct the smallest such dataset, or characterize the size of of such dataset, called the teaching dimension of the student model (Zhu, 2013; 2015). There are also many seminal theory work on analyzing the teaching dimension of different models (Shinohara & Miyano, 1991; Goldman & Kearns, 1995; Doliwa et al., 2014; Liu et al., 2016).

However, in many real world applications, the student model is typically updated via an iterative algorithm, and we get the opportunity to observe the performance of the student model as we feed examples to it. For instance,

- In model compression where we want to transfer a target "teacher model" to a destination "student model", we can constantly observe student model's prediction on current training points. Intuitively, such observations will allow us to get a better estimate where the student model is and pick examples more intelligently to better guide the student model to convergence.
- In cyber-security setting where an attack wants to mislead a recommendation system that learns online, the attacker can constantly generate fake clicks and observe the system's response. Intuitively, such feedback will allow the attacker to figure out the state of the learning system, and design better strategy to mislead the system.

From the aspects of both faster model compression and bet-

ter avoiding hacker attack, we seek to understand some fundamental questions, such as, *what is the sequence of examples that teacher should feed to the student in each iteration in order to achieve fast convergence? And how many such examples or such sequential steps are needed?*

In this paper, we will focus on this new paradigm, called ***iterative machine teaching***, which extends traditional machine teaching from batch setting to iterative setting. In this new setting, the teacher model can communicate with and influence the student model in multiple rounds, but the student model remains passive. More specifically, in each round, the teacher model can observe (potentially different levels of) information about the students to intelligently choose one example, and the student model runs a fixed iterative algorithm using this chosen example.

Furthermore, the smallest number of examples (or rounds) the teacher needs to construct in order for the student to efficiently learn a target model is called the ***iterative teaching dimension*** of the student algorithm. Notice that in this new paradigm, we shift from describing the complexity of a model to the complexity of an algorithm. Therefore, for the same student model, such as logistic regression, the iterative teaching dimension for a teacher model can be different depending on the student's learning algorithms, such as gradient descent versus conjugate gradient descent. In some sense, the teacher in this new setting is becoming active, but not the student. In Fig. 1, we summarize the differences of iterative machine teaching from traditional machine teaching, active learning and passive learning.

Besides introducing the new paradigm, we also propose three iterative teaching algorithms, called omniscient teacher, surrogate teacher and imitation teacher, based on the level of information about the student that the teacher has access to. Furthermore we provide partial theoretical analysis for these algorithms under different example construction schemes. Our analysis shows that under suitable conditions, iterative teachers can always perform better than passive teacher, and achieve exponential improvements. Our analysis also identifies two crucial properties, namely teaching monotonicity and teacher capability, which play critical roles in achieving fast iterative teaching.

To corroborate our theoretical findings, we also conduct extensive experiments on both synthetic data and real image data. In both cases, the experimental results verify our theoretical findings and the effectiveness of our proposed iterative teaching algorithms.

2. Related Work

Machine teaching. Machine teaching problem is to find an optimal training set given a student model and a target. (Zhu, 2015) proposes a general teaching framework. (Zhu, 2013) considers Bayesian learner in exponential family and expresses the machine teaching as an optimization problem over teaching examples that balance the future loss of the learner and the effort of the teacher. (Liu et al., 2016) provides the teaching dimension of several linear learners. The framework has been applied to security (Mei & Zhu, 2015), human computer interaction (Meek et al., 2016) and education (Khan et al., 2011). (Johns et al., 2015) further extends machine teaching to interactive settings. However, these work ignores the fact that a student model is typically learned by an iterative algorithm, and we usually care more about how fast the student can learn from the teacher.

Interactive Machine Learning. (Cakmak & Thomaz, 2014) consider the scenario of a human training an agent to perform a classification task by showing examples. They study how to improve human teacher by giving teaching guidance. (Singla et al., 2014) consider the crowdsourcing problem and propose a sequential teaching algorithm that can teach crowd worker to better classify the query. Both work consider a very different setting where the learner (i.e. human learner) is not iterative and does not have a particular optimization algorithm.

Active learning. Active learning enables a learner to interactively query the oracle to obtain the desired outputs at new samples. Machine teaching is different from active learning in the sense that active learners explore the optimal parameters by itself rather than being guided by the teacher. Therefore they have different sample complexity (Balcan et al., 2010; Zhu, 2013).

Curriculum learning. Curriculum learning (Bengio et al., 2009) is a general training strategy that encourages to input training examples from easy ones to difficult ones. Very interestingly, our iterative teacher model suggests similar training strategy in our experiments.

3. Iterative Machine Teaching

The proposed iterative machine teaching is a general concept, and the paper considers the following settings:

Student's Asset. In general, the asset of a student (learner) includes the initial parameter w_0, loss function, optimization algorithm, representation (feature), model, learning rate η_t over time (and initial η_0) and the trackability of the parameter w^t. The ideal case is that a teacher has access to all of them and can track the parameters and learning rate, while the worst case is that a teacher knows nothing. How practical the teaching is depends on how much the prior knowledge and trackability that a teacher has.

Representation. The teacher represents an example as (x, y) while the student represents the same example as $(\widetilde{x}, \widetilde{y})$ (typically $y = \widetilde{y}$). The representation $x \in \mathcal{X}$ and $\widetilde{x} \in \widetilde{\mathcal{X}}$ can be different but deterministically related. We assume there exists $\widetilde{x} = \mathcal{G}(x)$ for an unknown invertible mapping $\mathcal{G}$.

Model. The teacher uses a linear model $y = \langle v, x \rangle$ with pa-

rameter v^* (w^* for student's space) that is taught to the student. The student also uses a linear model $\widetilde{y}=\langle w, \widetilde{x}\rangle$ with parameter w, i.e., $\widetilde{y}=\langle w, \mathcal{G}(x)\rangle = f(x)$ in general. w and v do not necessarily lie in the same space, but for omniscient teacher, they are equivalent and interchangeably used.

Teaching protocol. In general, the teacher can only communicate with the student via examples. In this paper, the teacher provides one example x^t in one iteration, where t denotes the t-th iteration. The goal of the teacher is to provide examples in each iteration such that the student parameter w converge to its optimum w^* as fast as possible.

Loss function. The teacher and student share the same loss function. We assume this is a convex loss function $\ell(f(x), y)$, and the best model is usually found by minimizing the expected loss below:

$$w^* = \arg\min_w \ \mathbb{E}_{(x,y)} \left[\ell(\langle w, x\rangle, y)\right]. \tag{1}$$

where the sampling distribution $(x, y) \sim \mathbb{P}(x, y)$. Without loss of generality, we only consider typical loss functions, such as square loss $\frac{1}{2}(\langle w, x\rangle - y)^2$, logistic loss $\log(1 + \exp(-y\langle w, x\rangle))$ and hinge loss $\max(1 - y\langle w, x\rangle, 0)$.

Algorithm. The student uses the stochastic gradient descent to optimize the model. The iterative update is

$$w^{t+1} = w^t - \eta_t \frac{\partial \ell(\langle w, x\rangle, y)}{\partial w}. \tag{2}$$

Without teacher's guiding, the student can be viewed as being guided by a random teacher who randomly feed an example to the student in each iteration.

4. Teaching by an Omniscient Teacher

An omniscient teacher has access to the student's feature space, model, loss function and optimization algorithm. In specific, omniscient teacher's (x, y) and student's $(\tilde{x}, \tilde{y})$ share the same representation space, and teacher's optimal model v^* is also the same as student's optimal model w^*.

4.1. Intuition and teaching algorithm

In order to gain intuition on how to make the student model converge faster, we will start with looking into the difference between the current student parameter and the teacher parameter w^* during each iteration:

$$\begin{aligned} \left\|w^{t+1} - w^*\right\|_2^2 &= \left\|w^t - \eta_t \frac{\partial \ell(\langle w, x\rangle, y)}{\partial w} - w^*\right\|_2^2 \\ &= \left\|w^t - w^*\right\|_2^2 + \eta_t^2 \underbrace{\left\|\frac{\partial \ell(\langle w^t, x\rangle, y)}{\partial w^t}\right\|_2^2}_{T_1(x,y|w^t):\text{Difficulty of an example } (x,\,y)} \\ &\quad - 2\eta_t \underbrace{\left\langle w^t - w^*, \frac{\partial \ell(\langle w^t, x\rangle, y)}{\partial w^t}\right\rangle}_{T_2(x,y|w^t):\text{Usefulness of an example } (x,\,y)} \end{aligned} \tag{3}$$

Based on the decomposition of the parameter error, the teacher aims to choose a particular example (x, y) such that $\|w^{t+1} - w^*\|_2^2$ is most reduced compared to $\|w^t - w^*\|_2^2$ from the last iteration. Thus the general strategy for the teacher is to choose an example (x, y), such that $\eta_t^2 T_1 - 2\eta_t T_2$ is minimized in the t-th iteration:

$$\operatorname*{argmin}_{x\in\mathcal{X}, y\in\mathcal{Y}} \eta_t^2 T_1(x, y|w^t) - 2\eta_t T_2(x, y|w^t). \tag{4}$$

The teaching algorithm of omniscient teacher is summarized in Alg.1. The smallest value of $\eta_t^2 T_1 - 2\eta_t T_2$ is $-\|w^t - w^*\|_2^2$. If the teacher achieves this, it means that we have reached the teaching goal after this iteration. However, it usually cannot be done in just one iteration, because of the limitation of teacher's capability to provide examples. T_1 and T_2 have some nice intuitive interpretations:

Difficulty of an example. T_1 quantifies the difficulty level of an example. This interpretation for different loss functions becomes especially clear when the data lives on the surface of a sphere, *i.e.*, $\|x\| = 1$. For instance,

- For linear regression, $T_1 = (\langle w, x\rangle - y)^2$. The larger the norm of gradient is, the more difficult the example is.
- For logistic regression, we have $T_1 = \|\frac{1}{1+\exp(y\langle w,x\rangle)}\|_2^2$. We know that $\frac{1}{1+\exp(y\langle w,x\rangle)}$ is the probability of predicting the wrong label. The larger the number is, the more difficult the example is.
- For support vector machines, we have $T_1 = \frac{1}{2}(\text{sign}(1 - y\langle w, x\rangle) + 1)$. Different from above losses, the hinge loss has a threshold to identify the difficulty of examples. While the example is difficult enough, it will produce 1. Otherwise it is 0.

Interestingly, the difficulty level is not related to the teacher w^*, but is based on the current parameters of the learner w^t. From another perspective, the difficulty level can also be interpreted as the information that an example carries. Essentially, a difficult example is usually more informative. In such sense, our difficulty level has similar interpretation to curriculum learning, but with different expression.

Usefulness of an example. T_2 quantifies the usefulness of an example. Concretely, T_2 is the correlation between discrepancy $w^t - w^*$ and the information (difficulty) of an example. If the information of the example has large correlation with the discrepancy, it means that this example is very useful in this teaching iteration.

Trade-off. Eq.(4) aims to minimize the difficulty level T_1 and maximize the usefulness T_2. In other word, the teacher always prefers easy but useful examples. When the learning rate is large, T_1 term plays a more important role. When learning rate is small, T_2 term plays a more important role. This suggests that initially the teacher should choose easier examples to feed into the student model, and later on the teacher should choose examples to focus more on reducing the discrepancy between $w^t - w^*$. Such examples are very likely the difficult ones. Even if the learning rate is fixed, the gradient $\nabla_w \ell$ is usually large for a convex loss

function at the beginning, so reducing the difficulty level (choosing easy examples) is more important. While near the optimum, the gradient $\nabla_w \ell$ is usually small, so T_2 becomes more important. It is also likely to choose difficult examples. It has nice connection with curriculum learning (easy example first and difficult later) and boosting (gradually focus on difficult examples).

4.2. Teaching monotonicity and universal speedup

Can the omniscient teacher always do better than a teacher who feed random examples to the student (in terms of convergence)? In this section, we identify generic conditions under which we can guarantee that the iterative teaching algorithm always perform better than random teacher.

Definition 1 (Teaching Volume) *For a specific loss function ℓ, we first define a teaching volume function $TV(w)$ with model parameter w as*

$$TV(w) = \max_{x\in\mathcal{X}, y\in\mathcal{Y}} \{-\eta_t^2 T_1(x,y|w) + 2\eta_t T_2(x,y|w)\} \quad (5)$$

Theorem 2 (Teaching Monotonicity) *Given a training set $\mathcal{X}$ and a loss function ℓ, if the inequality*

$$\|w_1 - w^*\|^2 - TV(w_1) \le \|w_2 - w^*\|^2 - TV(w_2) \quad (6)$$

holds for any w_1, w_2 that satisfy $\|w_1 - w^*\|^2 \le \|w_2 - w^*\|^2$, then with the same parameter initialization and learning rate, the omniscient teacher can always converge not slower than random teacher.

The teaching volume represents the teacher's teaching effort in this iteration, so $\|w^t - w^*\|^2 - TV(w^t)$ characterizes the remaining teaching effort needed to achieve the teaching goal after iteration t. Theorem 2 says that for a loss function and a training set, if the remaining teaching effort is monotonically decreasing while the model parameter gets closer to the optimum, we can guarantee that the omniscient teacher can always converge not slower than random teacher. It is a sufficient condition for loss functions to achieve faster convergence than SGD. For example, the square loss satisfies the condition with certain training set:

Proposition 3 *The square loss satisfies the teaching monotonicity condition given the training set $\{x | \|x\| \le R\}$.*

4.3. Teaching capability and exponential speedup

The theorem in previous subsection insures that under certain conditions the omniscient teacher can always lead to faster convergence for the student model, but can there be exponential speedup? To this end, we introduce further assumptions of the "richness" of teaching examples, which we call teaching capability. We start from the ideal case, *i.e.*, the synthesis-based omniscient teacher with hyperspherical feature space, and then, extend to real cases with the restrictions on teacher's knowledge domain, sampling scheme, and student information. We present specific teaching strategies in terms of teaching capability (strong to weak): synthesis, combination and (rescalable) pool.

Synthesis-based teaching. In synthesis-based teaching, the teacher can provide any samples from

$$\mathcal{X} = \{x \in \mathbb{R}^d, \|x\| \le R\}$$
$$\mathcal{Y} = \mathbb{R} \text{ (Regression) or } \{-1, 1\} \text{ (Classification).}$$

Theorem 4 (Exponential Synthesis-based Teaching) *For a synthesis-based omniscient teacher and a student with fixed learning rate $\eta \neq 0$, if the loss function $\ell(\cdot,\cdot)$ satisfies that for any $w \in \mathbb{R}^d$, there exists $\gamma \neq 0$, $|\gamma| \le \frac{R}{\|w - w^*\|}$ such that while $\hat{x} = \gamma(w - w^*)$ and $\hat{y} \in \mathcal{Y}$, we have*

$$0 < \gamma \nabla_{\langle w, \hat{x}\rangle} \ell(\langle w, \hat{x}\rangle, \hat{y}) \le \frac{1}{\eta},$$

then the student can learn an ϵ-approximation of w^* with $\mathcal{O}(C_1^{\gamma,\eta} \log \frac{1}{\epsilon})$ samples. We call such loss function $\ell(\cdot,\cdot)$ exponentially teachable in synthesis-based teaching.

The constant is $C_1^{\gamma,\eta} = (\log \frac{1}{1-\eta\nu(\gamma)})^{-1}$ in which $\nu(\gamma) := \min_{w,y} \gamma \nabla_{\langle w,\hat{x}\rangle} \ell(\langle w, \hat{x}\rangle, y) > 0$. $\nu(\gamma)$ is related to the convergence speed. Note that the sample complexity serves as the iterative teaching dimension corresponding to this particular teacher, student, algorithm and training data.

The sample complexity in iterative teaching is *deterministic*, different from the high probability bounds of traditional sample complexity with random *i.i.d.*samples or actively required samples. This is because the teacher provides the samples deterministically without noise in every iteration.

The radius R for $\mathcal{X}$, which can be interpreted as the knowledge domain of the teacher, will affect the sample complexity by constraining the valid values of γ, and thus $C_1^{\gamma,\eta}$. For example, for absolute loss, if R is large, such that $\frac{1}{\eta} \le \frac{R}{\|w^0 - w^*\|}$, γ can be set to $\frac{1}{\eta}$ and the $\nu(\gamma)$ will be $\frac{1}{\eta}$ in this case. Therefore, we have $C_1^{\gamma,\eta} = 0$, which means the student can learn with only one example (one iteration). However, if $\frac{1}{\eta} > \frac{R}{\|w^0 - w^*\|}$, we have $C_1^{\gamma,\eta} > 0$, and the student can converge exponentially. The similar phenomenon appears in the square loss, hinge loss, and logistic loss. Refer to Appendix A for details.

The exponential synthesis-based teaching is closely related to Lipschitz smoothness and strong convexity of loss functions in the sense that the two regularities provide positive lower and upper bound for $\gamma \nabla_{\langle w,x\rangle} \ell(\langle w, x\rangle, y)$.

Proposition 5 *The Lipschitz smooth and strongly convex loss functions are exponentially teachable in synthesis-based teaching.*

The exponential synthesis-based teachability is a weaker condition compared to the strong convexity and Lipschitz smoothness. We can show that besides the Lipschitz smooth and strongly convex loss, there are some other loss functions, which are not strongly convex, but still are exponentially teachable in synthesis-based scenario, *e.g.*, the hinge loss and logistic loss. Proofs are in Appendix A.

Combination-based teaching. In this scenario, the teacher

Algorithm 1 The omniscient teacher

1: Randomly initialize the student and teacher parameter w^0;
2: Set $t=1$ and the maximal iteration number T;
3: **while** w^t has not converged or $t<T$ **do**
4: Solve the optimization (e.g., pool-based teaching):

$$(x^t,y^t)=\underset{x\in\mathcal{X},y\in\mathcal{Y}}{\operatorname{argmin}}\ \eta_t^2\left\|\frac{\partial\ell\left(\left\langle w^{t-1},x\right\rangle,y\right)}{\partial w^{t-1}}\right\|^2 - 2\eta_t\left\langle w^{t-1}-w^*,\frac{\partial\ell\left(\left\langle w^{t-1},x\right\rangle,y\right)}{\partial w^{t-1}}\right\rangle$$

5: Use the selected example (x^t,y^t) to perform the update:

$$w^t=w^{t-1}-\eta_t\frac{\partial\ell\left(\left\langle w^{t-1},x^t\right\rangle,y^t\right)}{\partial w^{t-1}}.$$

6: $t\leftarrow t+1$
7: **end while**

can provide examples from ($\alpha_i\in\mathbb{R}$)

$$\mathcal{X}=\left\{x|\|x\|\le R, x=\sum_{i=1}^m\alpha_i x_i, x_i\in\mathcal{D}\right\}, \mathcal{D}=\{x_1,\dots,x_m\}$$

$\mathcal{Y}=\mathbb{R}$ (Regression) or $\{-1,1\}$ (Classification)

Corollary 6 *For a combination-based omniscient teacher and a student with fixed learning rate $\eta\neq 0$ and initialization w^0, if the loss function is exponentially synthesis-based teachable and $w^0-w^*\in span\,(\mathcal{D})$, the student can learn an ϵ-approximation of w^* with $\mathcal{O}\left(C_1^{\gamma,\eta}\log\frac{1}{\epsilon}\right)$ samples.*

Although the knowledge pool of teacher is more restricted compared to the synthesis-based scenario, with teacher's extra work to combine samples, the teacher can behave the same as the most knowledgable synthesis-based teacher.

Rescalable pool-based teaching. This scenario is further restricted in both knowledge pool and the effort to prepare samples. The teacher can provide examples from $\mathcal{X}\times\mathcal{Y}$:

$\mathcal{X}=\{x|\|x\|\le R, x=\gamma x_i, x_i\in\mathcal{D},\gamma\in\mathbb{R}\}, \mathcal{D}=\{x_1,\dots\}$
$\mathcal{Y}=\mathbb{R}$ (Regression) or $\{-1,1\}$ (Classification)

In such scenario, we cannot get arbitrary direction rather than the samples from the candidate pool. Therefore, to achieve the exponential improvement, the candidate pool should contain rich enough directions. To characterize the richness in finite case, we define the *pool volume* as

Definition 7 (Pool Volume) *Given the training example pool $\mathcal{X}\in\mathbb{R}^d$, the volume of $\mathcal{X}$ is defined as*

$$\mathcal{V}(\mathcal{X}):=\min_{w\in span(\mathcal{D})}\max_{x\in\mathcal{X}}\frac{\langle w,x\rangle}{\|w\|^2}.$$

Obviously, for the candidate pool of the synthesis-based teacher, we have $\mathcal{V}(\mathcal{X})=1$. In general, for finite candidate pool, the pool volume is $0<\mathcal{V}(\mathcal{X})<1$.

Theorem 8 *For a rescalable pool-based omniscient teacher and a student with fixed learning rate $\eta\neq 0$ and initialization w^0, if for any $w\in\mathbb{R}^d$, $w\not\perp w^*$ and $w^0-w^*\in span\,(\mathcal{D})$, there exists $\{x,y\}\in\mathcal{X}\times\mathcal{Y}$ and γ such that while $\hat{x}=\frac{\gamma\|w-w^*\|}{\|x\|}x, \hat{y}=y$, we have*

$$0<\gamma\nabla_{\langle w,\hat{x}\rangle}\ell\left(\langle w,\hat{x}\rangle,\hat{y}\right)<\frac{2\mathcal{V}(\mathcal{X})}{\eta},$$

then the student can learn an ϵ-approximation of w^* with $\mathcal{O}(C_2^{\eta,\gamma,\mathcal{V}(\mathcal{X})}\log\frac{1}{\epsilon})$ samples. We say such loss function is exponentially teachable in rescalable pool-based teaching.

The pool volume plays a vital role in pool-based teaching. It not only affects the existence of γ and $\{\hat{x},\hat{y}\}$ to satisfy the conditions, but also changes the convergence rate. While $\mathcal{V}(\mathcal{X})$ increases, $C_2^{\eta,\gamma,\mathcal{V}(\mathcal{X})}$ will decrease, yielding smaller sample complexity. With $\mathcal{V}(\mathcal{X})<1$, the rescalable pool-based teaching requires more samples than the synthesis-based teaching. As $\mathcal{V}(\mathcal{X})$ increases to 1, the candidate pool becomes $\left\{x\in\mathbb{R}^d,\|x\|\le R\right\}$ and $C_2^{\eta,\gamma,\mathcal{V}(\mathcal{X})}$ approaches to $C_1^{\gamma,\eta}$. Then the convergence speed of rescalable pool-based teaching approaches to the synthesis/combination-based teaching.

5. Teaching by a less informative teacher

To make the teacher model useful in practice, we further design two less informative teacher model that requires less and less information from the student.

5.1. The surrogate teacher

Suppose we can only query the function output from the learned $\langle w^t,x\rangle$, but we can not directly access w^t. How can we choose the example? In this case we propose to make use of the the convexity of the loss function. That is

$$\left\langle w^t-w^*,\frac{\partial\ell(\langle w^t,x\rangle,y)}{\partial w^t}x\right\rangle\geqslant\ell(\langle w^t,x\rangle,y)-\ell(\langle w^*,x\rangle,y). \tag{7}$$

Taking the pool-based teaching as an example, we can instead optimize the following surrogate loss function:

$$(x^t,y^t)=\underset{\{x,y\}\in\mathcal{X}}{\operatorname{argmin}}\ \eta_t^2\left\|\frac{\partial\ell(\langle w^t,x\rangle,y)}{\partial w^t}\right\|_2^2 - 2\eta_t\left(\ell(\langle w^t,x\rangle,y)-\ell(\langle w^*,x\rangle,y)\right) \tag{8}$$

by replacing $\left\langle w^t-w^*,\frac{\partial\ell(\langle w^t,x\rangle,y)}{\partial w^t}\right\rangle$ with its lower bound. The advantage of this approach is that the teacher only need to query the learner for the function output $\langle w^t,x\rangle$ to choose the example, without the need to access the learner parameter w^t directly. Furthermore, after noticing that in this formulation, the teacher makes prediction via inner products, we find that the surrogate teacher can also be applied to the scenario where the teacher and the student use different feature spaces by further replacing $(\ell(\langle w^t,x\rangle,y)-\ell(\langle w^*,x\rangle,y))$ with $(\ell(\langle w^t,x\rangle,y)-\ell(\langle v^*,\widetilde{x}\rangle,y))$. With this modification, we can provide examples without using information about w^*. The performance of the surrogate teacher largely depends on the tightness of such convexity lower bound.

Algorithm 2 The imitation teacher

1: Randomly initialize the student parameter w^0 and the teacher parameter v^0; Randomly select a training sample (x^0, y^0);
2: Set $t = 1$ and the maximal iteration number T;
3: **while** w^t has not converged or $t < T$ **do**
4: Perform the update:

$$v^t = v^{t-1} - \eta_v \left(\left\langle v^{t-1}, x^{t-1} \right\rangle - \left\langle w^t, x^{t-1} \right\rangle \right) x^{t-1}.$$

5: Solve the optimization (e.g., pool-based teaching):

$$(x^t, y^t) = \underset{x \in \mathcal{X}, y \in \mathcal{Y}}{\operatorname{argmin}} \ \eta_t^2 \left\| \frac{\partial \ell \left(\left\langle w^t, x \right\rangle, y \right)}{\partial v^t} \right\|^2 - 2\eta_t \left\langle v^t - v^*, \frac{\partial \ell \left(\left\langle v^t, x \right\rangle, y \right)}{\partial v^t} \right\rangle.$$

6: Provide the selected example (x^t, y^t) for the student to perform the update ;

$$w^{t+1} = w^t - \eta_t \frac{\partial \ell \left(\left\langle w^t, x \right\rangle, y \right)}{\partial w}.$$

7: $t \leftarrow t + 1$
8: **end while**

5.2. The imitation teacher

When the teacher and the student have different feature spaces, this teaching setting will be much closer to practice than all the previous settings and also more challenging. To this end, we present an imitation teacher who learns to imitate the inner product output $\langle w^t, x \rangle$ of the student model and simultaneously choose examples in teacher's own feature space. The teacher can possibly use active learning to imitate the student's $\langle w^t, x \rangle$. In this imitation, the student model stays unchanged and the teacher model could update itself via multiple queries to the student (input an example and see the inner product output of the student). We present a more simple and straightforward imitation teacher (Alg. 2) which works in a way similar to stochastic mirror descent (Nemirovski et al., 2009; Hall & Willett, 2013). In specific, the teacher first learns to approximate the student's $\langle w^t, x \rangle$ with the following iterative update:

$$v^{t+1} = v^t - \eta_v \left(\left\langle v^t, x \right\rangle - \left\langle w^t, x \right\rangle \right) x \tag{9}$$

where η_v is the learning rate for the update. Then we use v^{t+1} to perform the example synthesis or selection in teacher's own feature space. We summarize this simple yet effective imitation teacher model in Alg. 2.

6. Discussion

Optimality of the teacher model. For arbitary loss function, the optimal teacher model for a student model should find the training example sequence to achieve the fastest possible convergence. Exhaustively finding such example sequence is computational impossible. For example, there are n^T possible training sequences (T is the iteration number) for n-size pool-based teaching. As a results, we need to make use of the properties of loss function to design the teacher model. The proposed teacher models are not necessarily optimal, but they are good enough under some conditions for loss function, student model and training data.

Theoretical aspects of the teacher model. The theoretical study of the teacher model includes finding the conditions for the loss function and training data such that the teacher model is optimal, or achieves provable faster convergence rate, or provably converges faster than the random teacher. We desire these conditions to be sufficient and necessary, but sometimes sufficient conditions suffice in practice. For different student models, the theoretical analysis may be different and we merely consider stochastic gradient learner here. There are still lots of optimization algorithms that can be considered. Besides, our teacher models are not necessarily the best, so it is also important to come up with better teacher models with provable guarantees. Although our paper mainly focuses on the fixed learning rate, our results are still applicable for the dynamic learning rate. However, the teacher should be more powerful in synthesizing or choosing examples (R should be larger than fixed learning rate case). In human teaching, it actually makes sense because while teaching a student who learns knowledge with dynamic speed, the teacher should be more powerful so that the student consistently learn fast.

Practical aspects of the teacher model. In practice, we usually want the teacher model to be less and less informative to the student model, scalable to large datasets, efficient to compute. How to make the teacher model scalable, efficient and less informative remains open challenges.

7. Experiments

7.1. Experimental details

Performance metric. We use three metric to evaluate the convergence performance: objective value w.r.t. the training set, difference between w^t and w^* ($\|w^t - w^*\|_2$), and the classification accuracy on testing set.

Parameters and setup. Detailed experimental setup is given in Appendix B. We mostly evaluate the practical pool-based teaching (without rescaling). We evaluate the different teaching strategies in Appendix C, and give more experiments on spherical data (Appendix E) and infant egocentric visual data (Appendix F). For fairness, learning rates for all methods are the same.

7.2. Teaching linear models on Gaussian data

This experiment explores the convergence of three typical linear models: ridge regression (RR), logistic regression (LR) and support vector machine (SVM) on Gaussian data. Note that SGD on selected set is to run SGD on the union of all samples selected by the omniscient teacher. For the scenario of different feature spaces, we use a random orthogonal projection matrix to generate the teacher's feature space from student's. All teachers use pool-based teaching strategy. For fair comparisons, we use the same random initialization and the same learning rate.

Teaching in the same feature space. The results in Fig.

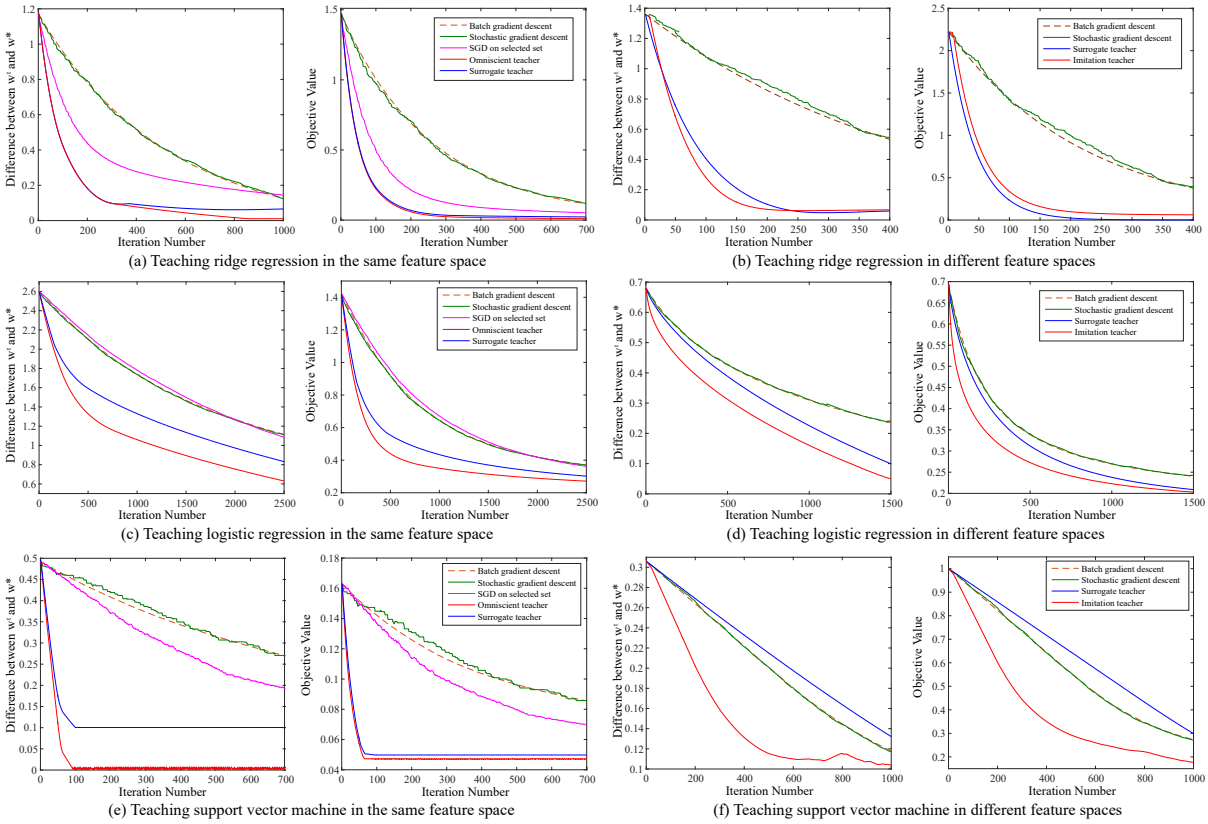

Figure 2. Convergence results on Gaussian distributed data.

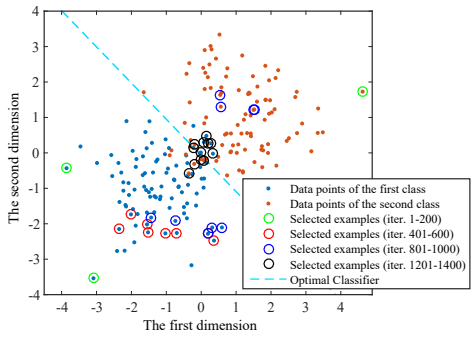

Figure 3. The examples selected by omniscient teacher for logistic regression on 2D binary-class Gaussian data.

2 show that the learner can converge much faster using the example provided by the teacher, showing the effectiveness of our teaching models. As expected, we find that the omniscient teacher consistently achieves faster convergence than the surrogate teacher who has no access to w. It is because the omniscient teacher always has more information about the learner. More interestingly, our guiding algorithms also consistently outperform SGD on the selected set, showing that the order of inputting training samples matters.

Teaching in different feature spaces. It is a more practical scenario that teacher and student use different feature spaces. While the omniscient teacher model is no longer applicable here, we teach the student model using the surrogate teacher and the imitation teacher. While the feature spaces are totally different, it can be expected that there will be a mismatch gap between the teacher model parameter and the student model parameter. Even in such a challenging scenario, the experimental results show that our teacher model still outperforms the conventional SGD and batch GD in most cases. One can observe that the surrogate teacher performs poorly in the SVM, which may be caused by the tightness of the approximated lower bound of the T_2 term. Compared to the surrogate teacher, the imitation teacher is more stable and consistently improves the convergence in all three linear models.

7.3. Teaching Linear Classifiers on MNIST Dataset

We further evaluate our teacher models on MNIST dataset. We use 24D random features to classify the digits (0/1, 3/5 as examples). We generate the teacher's features using a random projection matrix from the original 24D student's features. Note that, omniscient teacher and surrogate teacher (same space) assume the teacher uses the student's feature space, while surrogate teacher (different space) and imitation teacher assume the teacher uses its own space. From Fig. 4, one can observe that all these teacher model produces significant convergence speedup. We can see that the omniscient teacher converges fastest as expected. Interestingly, our imitation teacher achieves very similar con-

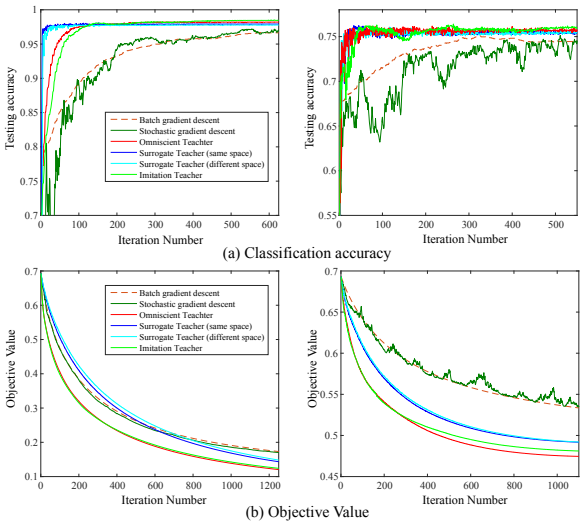

Figure 4. Teaching logistic regression on MNIST dataset. Left column: 0/1 classification. Right column: 3/5 classification

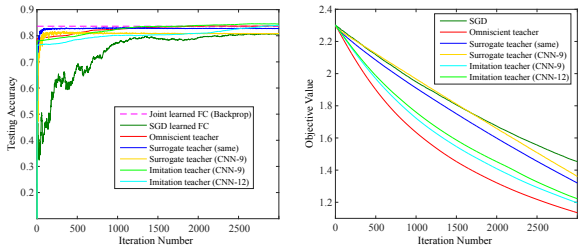

Figure 5. Teaching fully connected layers of CNNs on CIFAR-10. Left: testing accuracy. Right: training objective value.

vergence speedup to the omniscient teacher under the condition that the teacher does not know the student's feature space. In Fig.6, we also show some examples of teacher's selected digit images (0/1 as examples) and find that the teacher tends to select easy example at the beginning and gradually shift the focus to difficult examples. This also has the intrinsic connections with the curriculum learning.

7.4. Teaching Fully Connected Layers in CNNs

We extend our teacher models from binary classification to multi-class classification. The teacher models are used to teach the final fully connected (FC) layers in convolutional neural network on CIFAR-10. We first train three baseline CNNs (6/9/12 convolution layers, detailed configuration is in Appendix B) on CIFAR-10 without data augmentation and obtain the 83.5%, 86.1%, 87.2% accuracy. First, we applied the omniscient teacher and the surrogate teacher to the CNN-6 student using the optimal FC layer from the joint backprop training. It is essentially to teach the FC layer in the same feature space. Second, we applied the surrogate teacher and the imitation teacher to the CNN-6 student using the parameters of optimal FC layers from CNN-9 and CNN-12. It is to teach the FC layer in different feature spaces. More interestingly, this different feature space may not necessarily have an invertible one-to-one mapping, but we could still observe convergence speedup using our teacher models. From Fig. 5, we can see that all the teacher models produces very fast convergence in terms of testing accuracy. Our teacher models can even produces better testing accuracy than the backprop-learned FC layer. For objective value, the omniscient teacher shows the largest convergence speedup, and the imitation teacher performs slightly worse but still much better than the SGD.

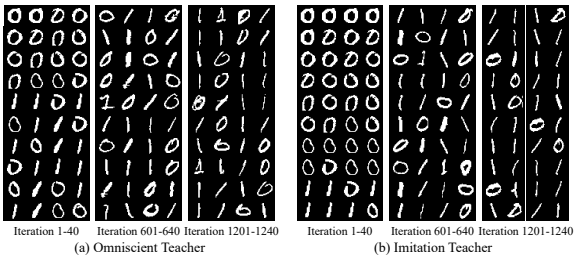

Figure 6. Some selected training examples on MNIST.

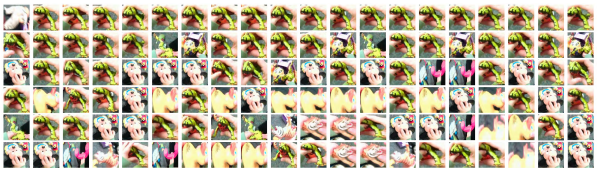

Figure 7. Selected training examples by the omniscient teacher on ego-centric data of infants. (The examples are visualized every 100 iteration, with left-to-right and top-to-bottom ordering)

7.5. Teaching on ego-centric visual data of infants

Using our teaching model, we analyze cropped object instances obtained from ego-centric video of an infant playing with toys (Yurovsky et al., 2013). Full detailed settings and results are in Appendix F. The results in Fig. 7 demonstrate a strong qualitative agreement between the training examples selected by the omniscient teacher and the order of examples received by a child in a naturalistic play environment. In both cases, the learner experiences extended bouts of viewing the same object. In contrast, the standard SGD learner receives random inputs. Our convergence results demonstrate that the learner converges significantly faster when receiving similar inputs to the child. Previous works have documented the unique temporal structure of the image examples that a child receives during object play (Bambach et al., 2016; Pereira et al., 2014). We believe these are the first results demonstrating that similar orderings can be obtained via a machine teaching approach.

8. Concluding Remarks

The paper proposes an iterative machine teaching framework. We elaborate the settings of the framework, and then study two important properties: teaching monotonicity and teaching capability. Based on the framework, we propose three teacher models for gradient learners, and give theoretical analysis for the learner to provably achieve fast convergence. Our theoretical findings are verified by experiments.

Acknowledgement

We would like to sincerely thank all the reviewers and Prof. Xiaojin Zhu for the valuable suggestions to improve the paper, Dan Yurovsky and Charlotte Wozniak for their help in collecting the dataset of children's visual inputs during object learning, and Qian Shao for help with the annotations. This project was supported in part by NSF IIS-1218749, NIH BIGDATA 1R01GM108341, NSF CAREER IIS-1350983, NSF IIS-1639792 EAGER, ONR N00014-15-1-2340, NSF Awards (BCS-1524565, BCS-1523982, and IIS-1320348) Nvidia and Intel. In addition, this work was partially supported by the Indiana University Areas of Emergent Research initiative in Learning: Brains, Machines, Children.

References

Alfeld, Scott, Zhu, Xiaojin, and Barford, Paul. Data poisoning attacks against autoregressive models. In *AAAI*, pp. 1452–1458, 2016.

Alfeld, Scott, Zhu, Xiaojin, and Barford, Paul. Explicit defense actions against test-set attacks. In *AAAI*, 2017.

Ba, Jimmy and Caruana, Rich. Do deep nets really need to be deep? In *Advances in neural information processing systems*, pp. 2654–2662, 2014.

Balcan, Maria-Florina, Hanneke, Steve, and Vaughan, Jennifer Wortman. The true sample complexity of active learning. *Machine learning*, 80(2-3):111–139, 2010.

Bambach, Sven, Crandall, David J, Smith, Linda B, and Yu, Chen. Active Viewing in Toddlers Facilitates Visual Object Learning: An Egocentric Vision Approach. *Proceedings of the 38th Annual Meeting of the Cognitive Science Society*, 2016.

Bengio, Yoshua, Louradour, Jerome, Collobert, Ronan, and Weston, Jason. Curriculum learning. In *ICML*, 2009.

Bucila, Cristian, Caruana, Rich, and Niculescu-Mizil, Alexandru. Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 535–541. ACM, 2006.

Cakmak, Maya and Thomaz, Andrea L. Eliciting good teaching from humans for machine learners. *Artificial Intelligence*, 217:198–215, 2014.

Doliwa, Thorsten, Fan, Gaojian, Simon, Hans Ulrich, and Zilles, Sandra. Recursive teaching dimension, vc-dimension and sample compression. *Journal of Machine Learning Research*, 15(1):3107–3131, 2014.

Goldman, Sally A and Kearns, Michael J. On the complexity of teaching. *Journal of Computer and System Sciences*, 50(1):20–31, 1995.

Hall, Eric C and Willett, Rebecca M. Online optimization in dynamic environments. *arXiv preprint arXiv:1307.5944*, 2013.

Han, Song, Mao, Huizi, and Dally, William J. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015.

Hinton, Geoffrey, Vinyals, Oriol, and Dean, Jeff. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Johns, Edward, Mac Aodha, Oisin, and Brostow, Gabriel J. Becoming the expert - interactive multi-class machine teaching. In *CVPR*, 2015.

Khan, Faisal, Mutlu, Bilge, and Zhu, Xiaojin. How do humans teach: On curriculum learning and teaching dimension. In *NIPS*, 2011.

Liu, Ji, Zhu, Xiaojin, and Ohannessian, H Gorune. The teaching dimension of linear learners. In *ICML*, 2016.

Meek, Christopher, Simard, Patrice, and Zhu, Xiaojin. Analysis of a design pattern for teaching with features and labels. *arXiv preprint arXiv:1611.05950*, 2016.

Mei, Shike and Zhu, Xiaojin. Using machine teaching to identify optimal training-set attacks on machine learners. In *AAAI*, 2015.

Nemirovski, Arkadi, Juditsky, Anatoli, Lan, Guanghui, and Shapiro, Alexander. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on optimization*, 19(4):1574–1609, 2009.

Pan, Sinno Jialin and Yang, Qiang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2010.

Pereira, Alfredo F, Smith, Linda B, and Yu, Chen. A Bottom-up View of Toddler Word Learning. *Psychonomic bulletin & review*, 21(1):178–185, 2014.

Romero, Adriana, Ballas, Nicolas, Kahou, Samira Ebrahimi, Chassang, Antoine, Gatta, Carlo, and Bengio, Yoshua. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*, 2014.

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

Singla, Adish, Bogunovic, Ilija, Bartok, Gabor, Karbasi, Amin, and Krause, Andreas. Near-optimally teaching the crowd to classify. In *ICML*, pp. 154–162, 2014.

Yurovsky, Daniel, Smith, Linda B, and Yu, Chen. Statistical Word Learning at Scale: The Baby's View is Better Developmental Science. *Developmental Science*, 16(6): 959–966, 2013.

Zhu, Xiaojin. Machine teaching for bayesian learners in the exponential family. In *NIPS*, 2013.

Zhu, Xiaojin. Machine teaching: An inverse problem to machine learning and an approach toward optimal education. In *AAAI*, pp. 4083–4087, 2015.