An Overview of Machine Teaching

Xiaojin Zhu
UW-Madison
Madison, WI, USA
jerryzhu@cs.wisc.edu

Adish Singla
MPI-SWS
Saarbrücken, Germany
adishs@mpi-sws.org

Sandra Zilles
Univ. of Regina
Canada
zilles@uregina.ca

Anna N. Rafferty
Carleton College
Northfield, MN, USA
arafferty@carleton.edu

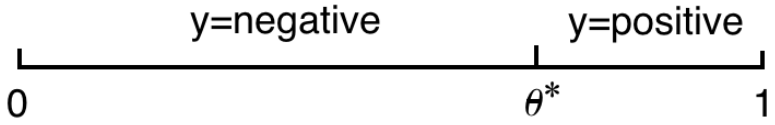
January 19, 2018

Abstract

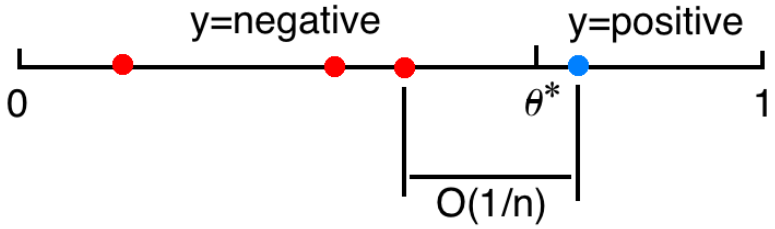
In this paper we try to organize machine teaching as a coherent set of ideas. Each idea is presented as varying along a dimension. The collection of dimensions then form the problem space of machine teaching, such that existing teaching problems can be characterized in this space. We hope this organization allows us to gain deeper understanding of individual teaching problems, discover connections among them, and identify gaps in the field.

1 Introduction

We start with several examples of machine teaching, with the goal of contrasting machine teaching with machine learning, in particular passive learning and active learning in supervised learning. Consider learning a 1D threshold classifier where the input distribution P_X is uniform over the interval $[0, 1]$, the true threshold is θ^*, and the binary label is noiseless: $y := \theta^*(x) = \mathbf{1}_{\{x \geq \theta^*\}}$:

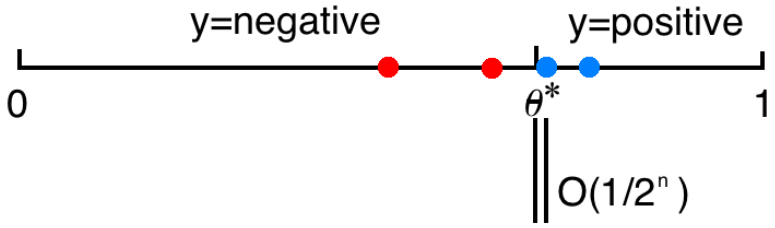


Passive learning receives n training items $x_1, \ldots, x_n \overset{i.i.d.}{\sim} P_X$ with $y_i = \theta^*(x_i)$. It can be shown that with large probability a consistent learner (one that makes zero training error) incurs a generalization error $|\hat{\theta} - \theta^*| = O(n^{-1})$. Intuitively with n uniform training items the average spacing is $1/n$ which is the uncertainty of the decision boundary, as defined by the inner-most pair of negative, positive training items:



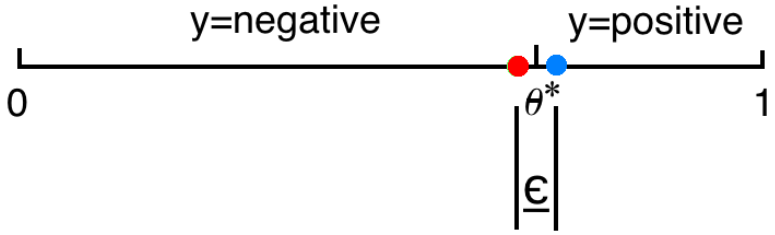
Equivalently, to achieve an ϵ generalization error one would need $n \geq O(\epsilon^{-1})$ training items. For example, if the desired generalization error is 0.001 the training set needs to be on the order of 1000.

Active learning allows the learner to adaptively pick queries x, and an oracle answers the label $\theta^*(x)$. For our problem, active learning is equivalent to binary search on the interval $[0, 1]$. With each query, the learner can remove half of the remaining interval since it can deduce that the threshold cannot be in that half:



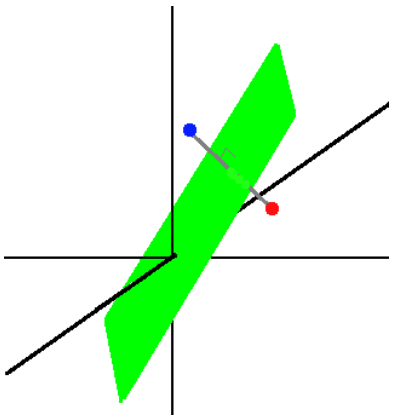
This results in a geometric reduction of generalization error $|\hat{\theta} - \theta^*| = O(2^{-n})$. Equivalently, to achieve an ϵ error the number of active learning queries is $n \geq O(\log(\epsilon^{-1}))$. For example, for the same 0.001 error an active learner only needs around 10 queries.

Machine teaching involves a teacher who knows θ^* and designs an optimal training set (also called a teaching set) for the learner (also called the student). For any consistent learner, it is easy to see that the teacher can construct such a teaching set by judiciously picking two training items, one negative and the other positive, such that they are at most ϵ apart and contain θ^* in the middle. The learner trained on this teaching set by definition achieves ϵ generalization error. Importantly, the teaching set size is always two regardless of the magnitude of ϵ:



Let us look at a second example, where the teacher teaches a hard-margin SVM a target hyperplane decision boundary in d-dimensional space. It turns out the teacher can again construct a teaching set consisting of two d-dimensional points. In fact there are infinitely many such teaching

sets, as long as the line segment between the two points is bisected by, and perpendicular to, the target hyperplane:

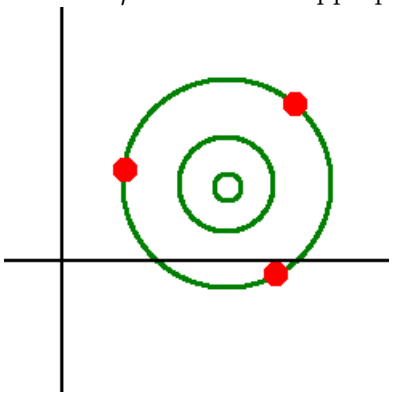


The so-called Teaching Dimension – the smallest teaching set size – of this problem is TD = 2. This is to be contrasted with the VC dimension of d-dimensional hyperplanes, which is VCD = $d+1$.

As a third example, consider a teacher who wants to teach a d-dimensional Gaussian density $N(\mu^*, \Sigma^*)$ to a learner. The learner learns by computing the sample mean and sample covariance matrix of given data:

$$\begin{aligned} \hat{\mu} &= \frac{1}{n}\sum_{i=1}^{n} x_i \\ \hat{\Sigma} &= \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \hat{\mu})(x_i - \hat{\mu})^\top. \end{aligned}$$

One can show that the teacher can construct a teaching set with $d+1$ points, which are the vertices of a d-dimensional tetrahedron centered at μ^* and scaled appropriately:



Let us contrast machine learning vs. machine teaching more formally, but still use passive learning as the problem setting. Machine learning takes a given training set D and learns a model $\hat{\theta}$. The learner can take a variety of forms, including version space learners for the theoretical study of Teaching Dimension (to be discussed later) [18], Bayesian learners [47], deep neural networks, or cognitive models which model how humans learn [33]. For now, we will use the popular regularized empirical risk minimization framework as an example:

$$\hat{\theta} = \text{argmin}_{\theta\in\Theta} \sum_{(x,y)\in D} \ell(x, y, \theta) + \lambda\|\theta\|^2 \quad (1)$$

where Θ is the hypothesis space, ℓ is the loss function, D is the training set, and λ is the regularization weight.

In contrast, in machine teaching the target model θ^* is *given*, and the teacher finds a teaching set D – not necessarily *i.i.d.* – such that a machine learner trained on D will approximately learn θ^*. One special instance of machine teaching can be written as a bilevel optimization problem:

$$\min_{D,\hat{\theta}} \quad \|\hat{\theta} - \theta^*\|^2 + \eta\|D\|_0 \tag{2}$$

$$\text{s.t.} \quad \hat{\theta} = \text{argmin}_{\theta\in\Theta} \sum_{(x,y)\in D} \ell(x, y, \theta) + \lambda\|\theta\|^2. \tag{3}$$

Remarks:

- The upper optimization is the teacher's problem. The teacher aims to bring the student model $\hat{\theta}$ close to the target θ^* while also to use a small teaching set ($\|D\|_0$ is the cardinality of the teaching set).
- The teacher is typically optimizing over a discrete space of teaching sets.
- The lower optimization is the learner's machine learning problem.
- The teacher needs to know the learning algorithm to formulate this optimization.

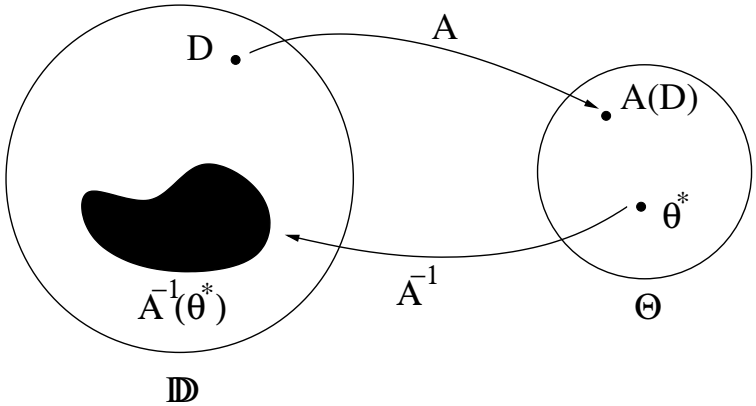
2 Why bother if the teacher already knows θ^*?

At this point we must address an important question: if the target model θ^* is known, what is the point of machine teaching? One needs to go beyond the machine learning mentality. There are applications where the teacher needs to convey the target model θ^* to a learner via training data. For example:

- In certain education problems (with human students) the teacher has an educational goal that can be formulated as θ^*. For example, a geologist may want to teach students to categorize rocks into igneous, sedimentary, and metamorphic. The geologist has the correct decision boundary θ^* in her mind, but she cannot telepathize θ^* into the student's mind. Instead, she teaches by picking informative rock samples to show the students. If the geologist has a good *cognitive model* on how the students learn from samples, she can use machine teaching to optimize the choice of rock samples.
- In one type of adversarial attack known as training-set poisoning, an attacker manipulates the behavior of a machine learning system by maliciously modifying the training data [2,30,31]. For example, consider a spam filter which constantly adapts its threshold in order to accommodate the changing legitimate content over time. An attacker knowing the algorithm may send specially designed emails to the spam filter to manipulate the threshold, such that certain spam emails can get pass the filter. Here the attacker plays the role of the teacher, and the victim is the unsuspecting student.

In fact, it may be useful to understand machine teaching from a coding perspective [48]. The teacher has a message which is the target model θ^*. The decoder is a fixed machine learning algorithm

A which accepts a teaching set D and decodes it into a model $A(D) := \hat{\theta}$. The teacher must encode θ^* using the code words consisting of teaching sets. This is depicted in the following figure where each dot on the left is a teaching set. Here the feasible code words are the preimage of θ^* under A, namely the black set of teaching sets. The teacher would select among the preimage the smallest teaching set.



To illustrate the central role of the decoder which is the learning algorithm A, consider the task of teaching a homogeneous (no offset) linear regression function $y = \theta^* x$ for $x \in \mathbb{R}$. One may think that a teaching set consisting of a single labeled example "on the line" $D_1 = \{(x_1, \theta^* x_1)\}, x_1 \neq 0$ is sufficient to teach. This is indeed true, but only if the learner A is the ordinary least squares (OLS) estimator: $A(D) = \text{argmin}_\theta \|X\theta - y\|^2 = (X^\top X)^{-1} X^\top y$. If, instead, the learner A is ridge regression with regularization weight λ: $A(D) = \text{argmin}_\theta \|X\theta - y\|^2 + \lambda\|\theta\|^2 = (X^\top X + \lambda I)^{-1} X^\top y$, D_1 will not teach θ^* due to the learner shrinking the estimation toward zero. The teacher can still teach with a single labeled example, but must nudge the y value upward off the line in anticipation of learner shrinkage [28]: $D_2 = \{(x_1, \theta^* x_1 + \frac{\lambda \theta^*}{x_1})\}, x_1 \neq 0$. Note the amount of nudging depends on the property of the learner (decoder), specifically λ. This also assumes that the teacher knows the decoder – ways to relax this assumption are discussed in sections 3.4 and 3.6.

Therefore, machine teaching is not about learning a model (although that is used as a subroutine), but rather about generating data in order to transmit a model, control a learner, shape reinforcement learning, persuade an agent, influence vertices, or even attack an adaptive system. It may be said that whenever one is optimizing data it is machine teaching; while if one is optimizing a model it is machine learning. This leads us to a slightly more general formal definition of machine teaching:

$$\min_{D,\hat{\theta}} \quad \text{TeachingRisk}(\hat{\theta}) + \eta \text{TeachingCost}(D) \tag{4}$$

$$\text{s.t.} \quad \hat{\theta} = \text{MachineLearning}(D). \tag{5}$$

In the above, we define a generic function TeachingRisk($\hat{\theta}$) for how unsatisfied the teacher is. The target model θ^* is folded into the teaching risk function. Alternatively, the teaching risk can be defined, e.g., by $\hat{\theta}$'s generalization error on an evaluation set and no target parameter θ^* is needed.

We also generalized the teaching cost function beyond the number of teaching items. For example, different items may have different cognitive burden for a human student to absorb.

We can alternatively consider two constrained forms of the machine teaching problem. One

constrained form attempts to minimize the teaching cost subject to sufficient learning:

$$\min_{D,\hat{\theta}} \quad \text{TeachingCost}(D) \tag{6}$$

$$\text{s.t.} \quad \text{TeachingRisk}(\hat{\theta}) \leq \epsilon \tag{7}$$

$$\hat{\theta} = \text{MachineLearning}(D). \tag{8}$$

This form allows either approximate teaching or exact teaching (student must learn exactly θ^*). In fact, it includes classic Teaching Dimension as a special case of exact teaching. Specifically, let $\text{TeachingCost}(D) = \|D\|_0$ the cardinality of the teaching set, let $\text{MachineLearning}(D)$ be the version space of the learner after seeing D, and let $\text{TeachingRisk}(\hat{\theta}) = 0$ if $\hat{\theta}$ is the singleton set $\{\theta^*\}$ made from the target concept, ∞ for any other version space. The optimization objective is then precisely the Teaching Dimension of θ^* with respect to the hypothesis space Θ, namely the minimum teaching set size to uniquely specify θ^*. More discussions are in section 4.1.

The other constrained form allows a teaching budget and optimizes learning:

$$\min_{D,\hat{\theta}} \quad \text{TeachingRisk}(\hat{\theta}) \tag{9}$$

$$\text{s.t.} \quad \text{TeachingCost}(D) \leq B \tag{10}$$

$$\hat{\theta} = \text{MachineLearning}(D). \tag{11}$$

It should be noted that research on machine teaching is also relevant to scenarios in which the teacher has full information about the target, but does not have a succinct representation for it, i.e., the teacher knows everything about the behavior of the model but cannot formulate this knowledge in terms of model parameters θ^*. In such cases, the teacher might still be able to provide the learning algorithm with carefully selected training examples, while the goal of the learning process would be to infer the model parameters. For instance, consider a scenario in which a human needs to extract certain information from web documents and would like to automate the extraction process. While the human may know exactly what type of information they are looking for, and can specify various highly representative examples of the desired information extraction, they may be incapable of formulating the machine-interpretable model that would be required for automating the information extraction process.

3 Characterizing the machine teaching space

We now organize the machine teaching space by introducing different dimensions. Each teaching problem can then be thought of as a point or a region in this space. This organization is by no means complete or prescriptive, and should be updated as the field moves along.

3.1 The human vs. machine dimension: Who teaches whom?

Machine teaching has the form "teacher T teaches student S." While the mathematics behind machine teaching is unified, the applications can look quite different depending on who T, S are:

- T=machine, S=machine: A good example is data poisoning attacks. S is a computer victim who runs a standard machine learning algorithm A that learns model $A(D)$ from training data D. T is an attacker – a malicious program – who can change (poison) D and wants S to learn

some nefarious θ^* instead. T also wants the poisoning to be subtle to avoid detection. The attacking problem is a special case of machine teaching:

$$\min_{\delta,\hat{\theta}} \quad \|\hat{\theta} - \theta^*\| + \eta\|\delta\| \tag{12}$$

$$\text{s.t.} \quad \hat{\theta} = A(D + \delta) \tag{13}$$

with appropriate norms. More discussions in section 4.4.

- T=machine, S=human: An example is computer tutoring systems. The key is to assume a cognitive model A of the human student. In the geology example earlier, A may be the Generalized Context Model for human categorization. T's goal is for the student to have high test scores in categorizing rocks, and T must teach by choosing training rock samples. The computer tutor can pose this problem as (4). More discussions in section 4.3.

- T=human, S=machine: One good example is to utilize human domain experts to quickly train a text classifier. As the earlier 1D threshold classifier example demonstrates, a human teacher able to produce an optimal teaching set – by either selecting documents from a corpus or even writing some new ones – will vastly outperform active learning where the human is merely used by the machine as a label oracle.

  Of course, the human teacher may not be optimal. In that case, machine teaching allows the machine student to "teach the human how to teach" [41]. Take the same 1D example. While the machine does not know the human teacher's target θ^*, it knows the *structure* of the optimal teaching set. Therefore, it can teach the human with analogues: "If you want to teach $\theta^* = 0.3$, you can show me a negative example at $x = 0.29$ and a positive example at $x = 0.31$."

  More broadly, human teachers afford the opportunity to teach beyond labeled examples: they can teach by features, pairwise comparisons, rules, etc. provided that the learning algorithm is equipped to accept such teaching signals. This is an active research direction in machine teaching.

- T=human, S=human: While traditional education is not the focus of machine teaching, insights gained from machine teaching have the potential to enhance pedagogy.

3.2 The teaching signal dimension: What can the teacher use?

The teaching signal depends on the learner.

For supervised learning, the standard setting is to teach with labeled training items. We distinguish the following settings:

- synthetic / constructive teaching: The teacher can use any item x in the (typically continuous) feature space $\mathcal{X}$. For an $x \in \mathcal{X}$ the teacher typically needs to synthesize / construct an artificial item that has feature x, hence the name. We may further distinguish between an honest teacher, who is constrained to use the correct label according to the target model $(x \in \mathcal{X}, y = \theta^*(x))$, and a lying teacher who is free to pair any label with x: $(x, y) \in \mathcal{X} \times \mathcal{Y}$. In education the latter may raise ethical questions.

- pool-based teaching: For many applications it is desirable to teach with actual items, e.g. real images or documents. Artificial items may look nonsensical to a (human) student. It

is therefore reasonable to assume that the teacher has a finite pool of candidate items, and must choose teaching items from the pool. This is the same consideration as pool-based active learning.

- Of course, there can be hybrid settings. For example, the teacher may start with a pool of candidate images. But for any image x in the pool, the teacher can perform mild synthesis by scaling, rotating, and cropping the image. As another example, in training set poisoning the attacker often alters an existing training set D_0 by perturbing some training item (x, y) into $(x + \delta_x, y + \delta_y)$.

Newer teaching signals for supervised learners are being studied. Examples include teaching by features or pairwise comparisons. Of course, this requires the ability of the learning algorithm to accept such signals.

For reinforcement learning, the teaching signal can be a demonstration in the form of a state trajectory. Or it can be artificial rewards in reward shaping. Both aim to teach a policy to the reinforcement learning agent.

For maximizing memory recall in humans one teaching signal is "spacing": choosing when to prompt the human student to review an item.

3.3 The batch vs. sequential dimension: Teaching with a set or a sequence?

In batch teaching the teacher gives a training set to the student, which is a batch learner. The order of training items does not matter. Depending on the problem setting, the training set may or may not contain repeated items.

In sequential teaching the teacher must also optimize the order of training items, as the student is a sequential learner. For example, some argue that it is important for the cognitive model of human students to be sequential, as humans are sensitive to the sequential effect. In machine learning, sequential learners include Stochastic Gradient Descent (SGD) algorithms, multi-armed bandits (MAB), and reinforcement learners (RL). Teaching sequential learners is most commonly used in teaching robots, as in the "teaching by demonstration" approach [4]. Curriculum learning [6] is a general approach in which training examples are presented to the learner in a sequence that is tailored to learning "from simple to difficult." One can also show that perceptron-type learning has much better guarantees in terms of the convergence rate when examples far away from the decision boundary are presented before examples that are closer to the decision boundary [44]. Sequential teaching can also benefit the teacher if the teacher can obtain information about the learner's state during the course of teaching (as discussed further in sections 3.4 and 3.4).

3.4 The model-based vs. model-free dimension: How much does the teacher know about the student?

Machine teaching critically depends on the amount of knowledge the teacher has about the student.

On one extreme the teacher takes a model-based approach. The student is a clearbox where the teacher has full knowledge of the learning algorithm. For example,

- If the student is a Support Vector Machine (SVM), the teacher knows in (1) that $\ell()$ is the hinge loss and also the value of the regularization weight λ.

- If the student is an SGD algorithm, the teacher knows the initial parameter w_0 of SGD, the loss function $\ell()$ or its gradient, and the learning rate parameter η.

- In classic Teaching Dimension, the teacher knows that the student is a version space learner, who maintains a version space VS consisting of all hypotheses that are consistent with the training data D:
$$\text{VS}(D) = \{\theta \in \Theta : \theta \text{ is consistent with } D\}. \tag{14}$$
This also implies that the teacher knows the learner's hypothesis space Θ to start with.

Such full knowledge of the student allows the teacher to specify teaching as the bilevel optimization problem (4).

On the other extreme the teacher takes a model-free approach. The teacher does not assume any learning algorithm used by the student. Instead, the student is a blackbox to the teacher: the teacher gives it training data D and only observes TeachingRisk as output. This pointwise function evaluation view motivates the teacher to perform derivative free optimization on the student blackbox.

In between the two extremes is a graybox student, where the teacher assumes partial knowledge of the learning algorithm. For example, the teaching may assume that the student is running ridge regression, i.e. the loss function ℓ in (1) is the squared loss; but the teacher does not know the value of regularization weight λ. It is easy to see that, with uncertainty in λ, the teacher cannot exactly teach θ^* using any finite training set. However, it may be possible for the teacher to "probe" the student. For example, after teaching with a few items, the teacher may ask the student to predict on a new item $\tilde{x}$. The teacher may know that the student uses its current estimate $\hat{\theta}$ to make a noisy prediction:
$$\tilde{y} = \hat{\theta}^\top \tilde{x} + \epsilon$$
where ϵ is a zero-mean noise with known variance. The teacher, upon receiving $\tilde{y}$ can then update its belief of the student's λ. Such teaching / probing activities can be interleaved. The teacher may benefit from applying active learning to such probing.

3.5 The student awareness dimension: Does the learner know it is being taught?

The vast majority of teaching settings involve a student that does not anticipate teaching. The student may employ standard learning algorithms such as deep neural networks or reinforcement learning. Typically (especially for supervised learning) the student assumes the training data is *i.i.d.*, oblivious of the fact that a teaching set may in fact be specially constructed and non-*i.i.d.* The student simply applies the learning algorithm to that data and produces a model. This is the assumption behind the machine teaching optimization problem (4). In security applications, this means the victim does not anticipate attacks.

However, an increasing number of teaching settings now involve a student who is aware that it is being taught by a teacher.

- In computational learning theory, such anticipatory student enables various models of teaching, see, e.g., [5, 12, 16, 50]. For example, the notions of Recursive Teaching Dimension (RTD) and Preference-based Teaching Dimension (PBTD) assume that the teacher and learner share a preference order over the class of all possible target concepts (i.e., the hypothesis space) and the learner expects the teacher to present examples that distinguish the target concept

from all those that are ranked higher in that preference order. These models outperform the classical model of Teaching Dimension (TD) in terms of the number of examples required for teaching [12, 16]. Intuitively, the learner can identify the target faster if it knows the strategy according to which the given examples were chosen.

- The student may understand that the teacher is a human who is suboptimal at teaching. As discussed earlier, the student can educate the teacher on the structure of the optimal teaching set, thus improving the quality of the teaching set it receives. The student may also decide to switch to active learning if it decides that the teacher is hopelessly suboptimal [9, 10, 41].

- The student may be aware that it is running learning algorithm A, but the teacher is teaching for a different learning algorithm B. For example, A may be an SVM while the human teacher treats the computer learner more like a human child, where B is the implicit cognitive model of the child assumed by the teacher. In this case, even if the teaching set is optimal for B it is in general suboptimal for A. However, the student can "translate" the teaching set with the knowledge of A, B. To illustrate, let A be the ridge regression estimator with regularization weight $\lambda_A = 2$:

$$A(D) = \text{argmin}_{\theta\in\mathbb{R}} \sum_{(x,y)\in D} \frac{1}{2}(\theta x - y)^2 + \frac{\lambda_A}{2}\theta^2.$$

  The teacher wants to teach the target $\theta^* = 1$, but it assumes a slightly different learner B with $\lambda_B = 1$. The teacher could construct a singleton teaching set $(x = \theta^*, y = \lambda_B + x^2) = (1, 2)$. If A were to take this teaching set directly, it will learn a wrong $\hat{\theta} = \frac{xy}{x^2+\lambda_A} = \frac{2}{3}$. Instead, if A is aware of the teacher's student model B, it could perform the following translation of the teaching set:

$$\tilde{x} = \frac{xy}{x^2 + \lambda_B}, \quad \tilde{y} = \lambda_A + \tilde{x}^2$$

  The translated teaching set is $(1, 3)$, making A learn the correct $\theta = 1$.

- In security applications, this means the victim may employ defense mechanisms to resist attacks [3].

- In educational applications, a human student may expect that the teacher is selecting examples specifically to aid learning (as discussed further in section 4.3)

3.6 The one vs. many dimension: how many students are simultaneously taught?

Most teaching settings have one teacher and one student.

In some settings, however, there may be many students with different learning algorithms. This is the case, for example, in classroom teaching. The key constraint is that the teacher must use the *same* teaching set on all students. Even if the teacher has perfect knowledge of each student's learning algorithm, it is in general impossible to perfectly teach all students [49]. The teacher may choose to optimize teaching for the worst student, which leads to a minimax risk formulation:

$$\min_{D,\{\hat{\theta}_\lambda\}} \quad \max_{\lambda} \text{TeachingRisk}(\hat{\theta}_\lambda) + \eta\text{TeachingCost}(D) \tag{15}$$

$$\text{s.t.} \quad \hat{\theta}_\lambda = \text{MachineLearning}_\lambda(D) \tag{16}$$

where the subscript λ denotes different students. Alternatively, the teacher may choose to optimize teaching for the average student, which requires a prior distribution $f(\lambda)$ (e.g. uniform) over the students and leads to the Bayes risk formulation:

$$\min_{D,\{\hat{\theta}_\lambda\}} \quad \int_\lambda f(\lambda)\text{TeachingRisk}(\hat{\theta}_\lambda)d\lambda + \eta\text{TeachingCost}(D) \tag{17}$$

$$\text{s.t.} \quad \hat{\theta}_\lambda = \text{MachineLearning}_\lambda(D). \tag{18}$$

3.7 The angelic vs. adversarial dimension: Is the teacher a friend or foe?

Machine teaching applications can be characterized by their intention. This ranges from angelic (optimized, personalized education; improving cognitive models; fast classifier training; debugging machine learners [7, 17, 46], etc.) to adversarial (training-set poisoning attacks).

3.8 The theoretical vs. empirical dimension: What is the work style?

While most teaching problems have both theoretical and empirical components, usually one of them is emphasized that dictates the approach toward the problems. On one extreme, there is pure theoretical research centered on understanding the Teaching Dimension and its variants such as Recursive Teaching Dimension and Preference-based Teaching Dimension etc. On the other extreme, many computer tutoring systems employ heuristic teaching methods in order to improve human student performance.

4 Some research directions in machine teaching

4.1 Algorithmic teaching theory

A teaching setting where the most theoretical advances have to be made is the study of Teaching Dimension. One way to cast Teaching Dimension in the machine teaching framework is constrained optimization (6), where the student is the version space learner (14), the teaching risk constrains the version space to be the singleton set $\{\theta^*\}$, and the teaching cost is the cardinality of the teaching set.

In some cases, this notion of teaching seems intuitively rather weak, but when designing different teaching models, one faces the issue of not trivializing the learner's role by introducing "unfair coding tricks" or "collusion." For instance, for a countable hypothesis space over a countable domain, it is not desirable to teach the ith concept in the hypothesis space with the ith element of the domain (irrespective of its label), which would result in a teaching dimension of 1 for all such hypothesis spaces, assuming teacher and learner agree on specific enumerations of the hypothesis space and the domain. There is no generally adopted notion of collusion, see [19, 50] for some examples. The Recursive Teaching Dimension (RTD) and the Preference-Based Teaching Dimension (PBTD) correspond to two notions of teaching that are collusion-free in the most stringent sense defined in the literature and that overcome many of the shortcomings of the classical TD approach. A systematic study of the effect of various notions of collusion is an interesting direction for future research.

One of the fundamental theoretical questions is whether there is any relationship between the information complexity of teaching (such as the TD or its variants) and the sample complexity of passive learning (particularly the VC Dimension (VCD)) for a given concept class. While there is no

general relationship between TD and VCD, i.e., TD can be arbitrarily smaller and arbitrarily bigger than VCD, recent results on the parameters PBTD and RTD suggest that teaching complexity is indeed related to the complexity of passive learning. While RTD can be arbitrarily smaller than VCD, it was shown that RTD $\in O(\text{VCD}^2)$ [22], and for a few special cases RTD is known to be equal to VCD [13]. These results immediately transfer to PBTD, since PBTD $\leq$ RTD holds in general [16]. An open question is whether RTD $\in O(\text{VCD})$, i.e., whether RTD is upper-bounded by a function linear in the VCD [38]. One reason why this question is of interest to a wider audience within the computational learning theory community is that it appears to be related to the *sample compression conjecture*, which has been open for 30 years and which states that every concept class $\mathcal{C}$ has a "sample compression scheme" in which each sample set S consistent with a concept in $\mathcal{C}$ can be compressed to a subset of size at most the VCD of $\mathcal{C}$ without losing any label information [14, 27]. Due to a number of results that show how to use teaching sets as compression sets and vice versa [11, 13], resolving the question whether or not RTD $\in O(\text{VCD})$ would shed some light on the sample compression conjecture.

Teaching Dimension can be generalized to non-version space learners, such as ridge regression, logistic regression, and SVMs [28], or Bayesian learners [47].

The current notions of Teaching Dimension are based on a setting where teacher only provides labeled examples as input. It would be interesting to consider settings with different teaching signals. For instance, let us say a teacher can provide features of an example in addition to the label. Also, it would be interesting to consider new query modalities, for instance, teaching based on pairwise comparison queries instead of labels, cf. [23].

Currently, most of the teaching models are studied in a batch setting where the teacher provides a set of examples in a batch to the learner. However, in real-world scenarios (e.g. personalized education), the teaching process is often interactive where the teacher can feed examples sequentially to the student, and can adapt these examples based on the current performance and feedback received from the student. One of the primary research questions is to understand how much speed up can be gained by adaptivity, cf. [20, 29, 39, 40].

Another interesting algorithmic question inspired from real-world scenarios is to model students with limited memory and/or limited computational power—current models usually do not put any such constraint on the student. For instance, it would be important to understand the Teaching Dimension of different concept classes when teaching such limited-capacity students. It would also be interesting to consider more realistic scenarios of how to model this limited capacity of students, cf. [33].

4.2 Human Robot/Computer Interaction

In the context of improving the efficiency of human robot interaction, an exciting new direction of research is to study machine teaching formulation for reinforcement learning agents. This, in turn, can be useful in two ways: (i) insights gained here can be used to teach human users on how to optimally interact with and provide instructions to robots (e.g. for personalized robotic devices like Siri or Alexa) [1, 37, 43], (ii) we could develop smarter learning algorithms for robots that can anticipate being taught by a human teacher (see discussions in section 3.5).

One concrete setting is to study the problem of teaching a student that uses an inverse reinforcement learning (IRL) algorithm. In IRL, the student has access to demonstrations provided by an expert (*e.g.*, trajectories when executing an optimal policy in an MDP), and the goal is to learn a reward function using these demonstrations. In classic IRL, these trajectories are usually provided

in a random fashion where the intention is not to teach (see the discussion of "doing" vs. "showing" in [21]). Here, one could study the optimization problem from the teacher's point of view on how to generate the best set of demonstrations for the student, cf. [8] for initial results in this direction. Furthermore, there are several more challenging real-world scenarios, e.g. when there is a mismatch of the state-space in the underlying MDP (e.g. human and robot have a different view of the world given different sensory inputs), or when there is a mismatch in the perceived rewards.

Another concrete setting is to study optimal reward shaping. Reward shaping refers to a teacher manipulating the rewards delivered to a reinforcement learning agent, with the goal to quickly teach a target policy to the agent [32]. Optimal reward shaping aims to minimize the learning time and the total amount of reward manipulation.

4.3 Education

One exciting application domain for machine teaching is personalized education; here, machine teaching formulations could enable us further development of rigorous algorithms for intelligent tutoring systems. These settings frequently provide the machine teacher with information about the learner and/or the progress of learning via the learner's responses to a problem; this makes a sequential approach to machine teaching appropriate both because humans are often sensitive to ordering, and because the teacher gains more information over time. While simple, potentially suboptimal, ways of deciding what data to present to the human learner are commonly used in automated teaching systems (e.g., ask the learner to solve a problem about any unmastered concept), there has been increasing recent interest in treating the problem as one of optimal control, often with approximate solutions that take into account learner responses [34, 45]. These approaches thus employ a teaching policy rather than seeking a fixed teaching set. One challenge in this area is because most approaches are model-based, accurate cognitive models must be identified for different learning tasks relevant to education. This has been an ongoing enterprise within cognitive and educational psychology, with increasing interest in creating or refining cognitive models based on existing datasets (e.g., [24, 33]). Behavioral experiments have also been employed within large scale MOOCs and intelligent tutoring systems; see [48] and [25] for more discussion.

One concrete learning task that has gained a lot of recent interest is that of reviewing content via flashcards (e.g. for teaching vocabulary of a foreign language in Duolingo) [35]. Given a set of vocabulary words that a student aims to learn, software tools (online websites or smartphone apps) for flashcards would shuffle through these cards with a goal to improve the recall probability of these words. Here, the key is to properly model the forgetting aspect of learning, and many software tools are using spaced-repetition models based on a cognitive model that uses a parametric decay function on a recall probability variable. This is then an optimal control problem, where the control is when/how often the algorithm should perform a review/test action. There are several exciting questions here, for instance, how to learn parameters of the cognitive model on the fly (e.g. via exploration-exploitation techniques) in order to provide personalized reviewing schedule. One approach has been to leverage a probabilistic model to make inferences about the unknown difficulty of words and unknown individual differences among students based on distributions over the populations, enabling personalized review that improves as more students interact with the system [26]. Another approach poses the spaced-repetition problem as stochastic differential equations and solves the optimal control problem [42]. Further exploration about the best ways to improve models while teaching and balance performance for an individual and for a population is needed.

While the most common approaches in education focus on a single learner and assume the learner

is not sensitive to being taught, there have been several interesting explorations of other parts of the space delineated by the machine teaching dimensions above. For example, consideration of the best teaching set for a group of learners, each with somewhat varying parameters in their cognitive model, can lead to predictions about the best type of student groupings and sizes for instruction (e.g., are small class sizes or similar ability students best for helping all children learn?) [15, 49]. Developmental psychology has also identified many cases where children are sensitive to whether the setting is pedagogical - that is, whether the informant is attempting to teach them. This has lead to automated teaching approaches that select examples for a learner who assumes a helpful teacher [36]. Examining how sensitive people are to helpful teachers across a range of domains is likely to lead to new approaches for a range of learning algorithms.

4.4 Trustworthy AI

As discussed above, machine teaching can also be applied to an adversarial attack setting where the teacher is an attacker. To elaborate, we distinguish two levels of adversarial attacks. Level 1 adversarial attacks are test-time attacks. They manipulate a test item $(\tilde{x}, \tilde{y})$ such that a *fixed, deployed* model $\hat{\theta} : X \mapsto Y$ would misclassify it. Specifically, level 1 attacks find a small perceptual perturbation to change $\tilde{x}$ (e.g. a stop sign image) into x (the stop sign image will a few pixels changed) so that x is classified differently (e.g. into a yield sign) by $\hat{\theta}$. This can be expressed as the following optimization problem:

$$\min_{x} \quad \|\tilde{x} - x\|_p \tag{19}$$

$$\text{s.t.} \quad \hat{\theta}(x) \neq \tilde{y}, \tag{20}$$

where the p-norm is a surrogate to perceptual distance.

In contrast, level 2 adversarial attacks are training-set poisoning attacks. The inputs are D_0 the original training set, $A : \mathcal{D} \mapsto \Theta$ the learning algorithm, and postcondition $\Psi : \Theta \mapsto \mathit{Boolean}$. Level 2 attacks find small perceptual perturbation to the training set D_0 so the trained model satisfies the postcondition:

$$\min_{D} \quad \|D_0 - D\|_p \tag{21}$$

$$\text{s.t.} \quad \Psi(A(D)). \tag{22}$$

For instance, the postcondition can be $\Psi(\hat{\theta}) = [\hat{\theta}(\tilde{x}) = \bar{y}]$ which plants an attacker-desired classification $\bar{y}$ for test item $\tilde{x}$ implicitly through the poisoned training data and the training process. Because the learning algorithm A is involved, level 2 attacks are more challenging to solve than level 1 attacks.

Understanding the optimal training-set poisoning attacks can, in turn, help us design optimal defenses against attackers. For instance, we can develop an automated defense system that could flag the parts of training data which are likely to be attacked (based on our model of the teacher) and focus human analysts' attention on those parts. This could drastically increase the chance of detecting such attacks by analysts once they know where to look, see [31, 46] for more discussions.

An important line of research here would be to model interactions between an attacker (the teacher) and a learning algorithm (the student) as a repeated game. Here, we would like to design a robust learning algorithm that can anticipate about the teacher's actions (i.e. the future attacks) and develop an optimal forward-looking defenses against such attacks.

4.5 Efficiently finding optimal teaching solutions

The optimization problem for machine teaching (4) is intrinsically hard due to its combinatorial, bilevel nature. Even simple instances are NP-hard: one can show that simple teaching problems include the set-cover and subset sum problems. While some special cases have closed-form solution, many more require careful formulation. For problems when the teaching set size is small, it is possible to formulate teaching as a mixed integer nonlinear programming (MINLP) problem and use existing solvers. Some other problems can benefit from approximate algorithms with guarantees, for example, by utilizing the submodularity properties of the problems.

References

[1] Baris Akgun, Maya Cakmak, Jae Wook Yoo, and Andrea Lockerd Thomaz. Trajectories and keyframes for kinesthetic teaching: A human-robot interaction perspective. In *Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction*, pages 391–398. ACM, 2012.

[2] Scott Alfeld, Xiaojin Zhu, and Paul Barford. Data poisoning attacks against autoregressive models. In *The Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16)*, 2016.

[3] Scott Alfeld, Xiaojin Zhu, and Paul Barford. Explicit defense actions against test-set attacks. In *The Thirty-First AAAI Conference on Artificial Intelligence (AAAI)*, 2017.

[4] Brenna Argall, Sonia Chernova, Manuela M. Veloso, and Brett Browning. A survey of robot learning from demonstration. *Robotics and Autonomous Systems*, 57(5):469–483, 2009.

[5] Frank J. Balbach. Measuring teachability using variants of the teaching dimension. *Theor. Comput. Sci.*, 397(1-3):94–113, 2008.

[6] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning, ICML*, pages 41–48, 2009.

[7] Gabriel Cadamuro, Ran Gilad-Bachrach, and Xiaojin Zhu. Debugging machine learning models. In *ICML Workshop on Reliable Machine Learning in the Wild*, 2016.

[8] Maya Cakmak and Manuel Lopes. Algorithmic and human teaching of sequential decision tasks. In *AAAI*, 2012.

[9] Maya Cakmak and Leila Takayama. Teaching people how to teach robots: The effect of instructional materials and dialog design. In *Proceedings of the 2014 ACM/IEEE international conference on Human-robot interaction*, pages 431–438. ACM, 2014.

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

[11] Malte Darnstädt, Thorsten Kiss, Hans Ulrich Simon, and Sandra Zilles. Order compression schemes. *Theor. Comput. Sci.*, 620:73–90, 2016.

[12] T. Doliwa, G. Fan, H. U. Simon, and S. Zilles. Recursive teaching dimension, VC-dimension and sample compression. *Journal of Machine Learning Research*, 15:3107–3131, 2014.

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

[14] S. Floyd and M. K. Warmuth. Sample compression, learnability, and the Vapnik-Chervonenkis dimension. *Machine Learning*, 21(3):269–304, 1995.

[15] Michael Frank. Modeling the dynamics of classroom education using teaching games. In *Proceedings of the Cognitive Science Society*, volume 36, 2014.

[16] Ziyuan Gao, Christoph Ries, Hans U Simon, and Sandra Zilles. Preference-based teaching. *Journal of Machine Learning Research*, 18(31):1–32, 2017.

[17] Shalini Ghosh, Patrick Lincoln, Ashish Tiwari, and Xiaojin Zhu. Trusted machine learning for probabilistic models. In *ICML Workshop on Reliable Machine Learning in the Wild*, 2016.

[18] S. Goldman and M. Kearns. On the complexity of teaching. *Journal of Computer and Systems Sciences*, 50(1):20–31, 1995.

[19] Sally A. Goldman and H. David Mathias. Teaching a smarter learner. *J. Comput. Syst. Sci.*, 52(2):255–267, 1996.

[20] Daniel Golovin and Andreas Krause. Adaptive submodularity: Theory and applications in active learning and stochastic optimization. *Journal of Artificial Intelligence Research*, 42:427–486, 2011.

[21] Mark K Ho, Michael Littman, James MacGlashan, Fiery Cushman, and Joseph L Austerweil. Showing versus doing: Teaching by demonstration. In *Advances in Neural Information Processing Systems*, pages 3027–3035, 2016.

[22] Lunjia Hu, Ruihan Wu, Tianhong Li, and Liwei Wang. Quadratic upper bound for recursive teaching dimension of finite VC classes. In *Proceedings of the 30th Conference on Learning Theory, COLT*, pages 1147–1156, 2017.

[23] Daniel M. Kane, Shachar Lovett, Shay Moran, and Jiapeng Zhang. Active classification with comparison queries. In *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS*, pages 355–366, 2017.

[24] Robert V Lindsey, Mohammad Khajah, and Michael C Mozer. Automatic discovery of cognitive skills to improve the prediction of student learning. In *Advances in neural information processing systems*, pages 1386–1394, 2014.

[25] Robert V Lindsey, Michael C Mozer, William J Huggins, and Harold Pashler. Optimizing instructional policies. In *Advances in Neural Information Processing Systems*, pages 2778–2786, 2013.

[26] Robert V Lindsey, Jeffery D Shroyer, Harold Pashler, and Michael C Mozer. Improving students' long-term knowledge retention through personalized review. *Psychological science*, 25(3):639–647, 2014.

[27] N. Littlestone and M. Warmuth. Relating data compression and learnability. Unpublished notes, 1986.

[28] Ji Liu and Xiaojin Zhu. The teaching dimension of linear learners. *Journal of Machine Learning Research*, 17(162):1–25, 2016.

[29] Weiyang Liu, Bo Dai, Ahmad Humayun, Charlene Tay, Chen Yu, Linda B. Smith, James M. Rehg, and Le Song. Iterative machine teaching. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017*, pages 2149–2158, 2017.

[30] Shike Mei and Xiaojin Zhu. The security of latent Dirichlet allocation. In *The Eighteenth International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2015.

[31] Shike Mei and Xiaojin Zhu. Using machine teaching to identify optimal training-set attacks on machine learners. In *The Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[32] Andrew Y Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *ICML*, volume 99, pages 278–287, 1999.

[33] K. Patil, X. Zhu, L. Kopec, and B. C. Love. Optimal teaching for limited-capacity human learners. *Advances in Neural Information Processing Systems (NIPS)*, 2014.

[34] Anna N Rafferty, Emma Brunskill, Thomas L Griffiths, and Patrick Shafto. Faster teaching via pomdp planning. *Cognitive science*, 40(6):1290–1332, 2016.

[35] Burr Settles and Brendan Meeder. A trainable spaced repetition model for language learning. In *ACL (1)*, 2016.

[36] Patrick Shafto, Noah D Goodman, and Thomas L Griffiths. A rational account of pedagogical reasoning: Teaching by, and learning from, examples. *Cognitive psychology*, 71:55–89, 2014.

[37] Patrice Y. Simard, Saleema Amershi, David Maxwell Chickering, Alicia Edelman Pelton, Soroush Ghorashi, Christopher Meek, Gonzalo Ramos, Jina Suh, Johan Verwey, Mo Wang, and John Wernsing. Machine teaching: A new paradigm for building machine learning systems. *CoRR*, abs/1707.06742, 2017.

[38] Hans Ulrich Simon and Sandra Zilles. Open problem: Recursive teaching dimension versus VC dimension. In *Proceedings of the 28th Conference on Learning Theory (COLT)*, pages 1770–1772, 2015.

[39] Adish Singla, Ilija Bogunovic, G Bartók, A Karbasi, and A Krause. On actively teaching the crowd to classify. In *NIPS Workshop on Data Driven Education*, 2013.

[40] Adish Singla, Ilija Bogunovic, Gábor Bartók, Amin Karbasi, and Andreas Krause. Near-optimally teaching the crowd to classify. In *Proceedings of the 31th International Conference on Machine Learning, ICML*, pages 154–162, 2014.

[41] J. Suh, X. Zhu, and S. Amershi. The label complexity of mixed-initiative classifier training. *International Conference on Machine Learning (ICML)*, 2016.

[42] B. Tabibian, U. Upadhyay, A. De, A. Zarezade, B. Schoelkopf, and M. Gomez-Rodriguez. Optimizing Human Learning. *ArXiv e-prints*, December 2017.

[43] Andrea L Thomaz and Maya Cakmak. Learning about objects with human teachers. In *Proceedings of the 4th ACM/IEEE international conference on Human robot interaction*, pages 15–22. ACM, 2009.

[44] Shankar Vembu and Sandra Zilles. Interactive learning from multiple noisy labels. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases, ECML PKDD*, pages 493–508, 2016.

[45] Jacob Whitehill and Javier Movellan. Approximately optimal teaching of approximately optimal learners. *IEEE Transactions on Learning Technologies*, 2017.

[46] Xuezhou Zhang, Xiaojin Zhu, and Stephen Wright. Training set debugging using trusted items. In *The Thirty-Second AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

[47] Xiaojin Zhu. Machine teaching for Bayesian learners in the exponential family. In *Advances in Neural Information Processing Systems*, 2013.

[48] Xiaojin Zhu. Machine teaching: an inverse problem to machine learning and an approach toward optimal education. In *The Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI "Blue Sky" Senior Member Presentation Track)*, 2015.

[49] Xiaojin Zhu, Ji Liu, and Manuel Lopes. No learner left behind: On the complexity of teaching multiple learners simultaneously. In *The 26th International Joint Conference on Artificial Intelligence (IJCAI)*, 2017.

[50] Sandra Zilles, Steffen Lange, Robert Holte, and Martin Zinkevich. Models of cooperative teaching and learning. *J. Mach. Learn. Res.*, 12:349–384, 2011.