Recent Developments in Algorithmic Teaching

Frank J. Balbach[1] and Thomas Zeugmann[2]

[1] Neuhofen, Germany
frank-balbach@gmx.de
[2] Division of Computer Science
Hokkaido University, Sapporo 060-0814, Japan
thomas@ist.hokudai.ac.jp

Abstract. The present paper surveys recent developments in algorithmic teaching. First, the traditional teaching dimension model is recalled.

Starting from the observation that the teaching dimension model sometimes leads to counterintuitive results, recently developed approaches are presented. Here, main emphasis is put on the following aspects derived from human teaching/learning behavior: the order in which examples are presented should matter; teaching should become harder when the memory size of the learners decreases; teaching should become easier if the learners provide feedback; and it should be possible to teach infinite concepts and/or finite and infinite concept classes.

Recent developments in the algorithmic teaching achieving (some) of these aspects are presented and compared.

1 Introduction

When preparing a lecture, a good teacher is carefully selecting *informative examples*. Additionally, a good teacher is taking into account that students do not memorize everything previously taught. And usually we make a couple of assumptions about the learners. They should neither be ignorant, lazy, nor should they be tricky. Thus, it is only natural to ask whether or not such human behavior is at least partially reflected in some algorithmic learning and/or teaching models studied so far in the literature.

Learning concepts from examples has attracted considerable attention in learning theory and machine learning. Typically, a learner does not know much about the source of these examples. Usually the learner is required to learn from all such sources, regardless of their quality. This is even true for the query learning model introduced by Angluin [1,2], since the teacher or oracle, though answering truthfully, is assumed to behave adversarially whenever possible. Therefore, it was only natural to ask whether or not one can also model scenarios in which a helpful teacher is honestly interested in the learner's success.

Perhaps the first approach was proposed by Freivalds, Kinber, and Wiehagen [3,4]. They developed a learning model in the inductive inference paradigm of identifying recursive functions in which the learner is provided with *good* examples chosen by an implicitly given teacher. Jain, Lange, and Nessel [5] adopted

A.H. Dediu, A.M. Ionescu, and C. Martín-Vide (Eds.): LATA 2009, LNCS 5457, pp. 1–18, 2009.

this model to learn recursively enumerable languages from *good* examples in the inductive inference paradigm.

The next step was to consider teaching as the natural counterpart of learning. Teaching has been modeled and investigated in various ways within algorithmic learning theory. However, the more classical models studied so far all follow one of two basically different approaches.

In the first approach, the goal is to find a teacher *and* a learner such that a given learning task can be carried out by them. Jackson and Tomkins [6] as well as Goldman and Mathias [7,8] defined models of teacher/learner pairs where teachers and learners are constructed explicitly. In all these models, some kind of adversary disturbing the teaching process is necessary to avoid collusion between the teacher and the learner. That is, when modeling teaching, a major problem consists in avoiding coding tricks. Though there is no generally accepted definition of coding tricks, it will be clear from our exposition that *no* form of coding tricks is used and thus no collusion occurs.

Angluin and Kriķis' [9,10] model prevents collusion by giving incompatible hypothesis spaces to teacher and learner. This makes simple encoding of the target impossible.

In the second approach, a teacher has to be found that teaches *all* deterministic consistent learners. Here a learner is said to be consistent if its hypothesis is correctly reflecting all examples received. This prevents collusion, since teaching happens the same way for all learners and cannot be tailored to a specific one. Goldman, Rivest, and Shapire [11] and Goldman and Kearns [12] substitute the adversarial teacher in the online learning model by a helpful one selecting good examples. They investigate how many mistakes a consistent learner can make in the worst case. In Shinohara and Miyano's [13] model the teacher produces a set of examples for the target concept such that it is the only consistent one in the concept class. The size of this set is the same as the worst case number of mistakes in the online model. This number is termed the *teaching dimension* of the target. Because of this similarity we shall from now on refer to both models as the *teaching dimension model* (abbr. TD model).

One difficulty of teaching in the TD model results from the fact that the teacher is not knowing anything about the learners besides them being consistent. In reality a teacher can benefit a lot from knowing the learners' behavior or their current hypotheses. It is therefore natural to ask how teaching can be improved if the teacher may observe the learners' hypotheses after each example. We refer to this scenario as to teaching with *feedback*.

After translating this question into the TD model, one sees that there is no gain in sample size at all. The current hypothesis of a consistent learner reveals nothing about its following hypothesis. Even if the teacher knew the hypothesis and provided a special example in response, he can only be sure that the learner's next hypothesis will be consistent. But this was already known to the teacher. So, in the TD model, feedback is useless.

There are also several other deficiencies in the teaching models studied so far. These deficiencies include that the order in which the teacher presents examples does not matter, and that teaching infinite concepts or infinite concept classes is severely limited. Another drawback is the rather counterintuitive dependence on the memory size of the learner. If the learner's memory size is large enough to store all examples provided by the teacher, then successful teaching is possible. Otherwise, it immediately becomes impossible. Another problem is that there are concept classes which are intuitively easy to teach that have a large teaching dimension.

Therefore, our goal has been to devise teaching models that remedy the above mentioned flaws. In particular, our aim has been to develop teaching models such that the following aspects do matter.

(1) The order in which the teacher presents the information should have an influence on the performance of the teacher.
(2) Teaching should get harder when the memory size of the learners decreases, but it should not become impossible for small memory.
(3) Teaching should get easier when the learners give feedback to the teacher.
(4) Concepts that are more complex should be harder to teach.
(5) The teaching model should work for both finite and infinite concepts and/or finite and infinite concept classes.

We studied and developed several models of algorithmic teaching to overcome these flaws to a different extent (cf. [14,15,16,17,18]). Within the present paper, we shortly summarize our and the related results obtained.

The paper is organized as follows. Section 2 shortly recalls the TD model and fundamental definitions needed subsequently. Then we discuss more recent approaches. In Section 3 we summarize results concerning teaching learners that have to obey restrictions on possible mind changes. Next, we turn our attention to a randomized model of teaching (see Section 4). Finally, we shortly touch teaching dimensions for complexity based learners and for cooperative learning.

2 The Teaching Dimension Model

We start by introducing the necessary notions and definitions. Let $\mathbb{N} = \{0, 1, \ldots\}$ denote the set of all natural numbers. For any set S we write $|S|$ to denote its cardinality. Let X be any (finite) set of *instances* also called *instance space*. A *concept* c is a subset of X and a *concept class* $\mathcal{C}$ is a set of concepts over X. It is convenient to identify every concept c with its characteristic function, i.e., for all $x \in X$ we have $c(x) = 1$ if $x \in c$ and $c(x) = 0$ otherwise. We consider mainly three instance spaces: $\{0, 1\}^n$ for Boolean functions, Σ^* for languages over a finite and non-empty alphabet Σ, and $X_n = \{x_1, \ldots, x_n\}$ for having any fixed instance space of cardinality n.

By $\mathcal{X} = X \times \{0, 1\}$ we denote the set of *examples* over X. An example (x, b) is either *positive*, if $b = 1$, or *negative*, if $b = 0$.

A concept c is *consistent* with a set $S = \{(x_1, b_1), \ldots, (x_n, b_n)\}$ of examples iff $c(x_i) = b_i$ for all $i = 1, \ldots, n$.

In the TD model, a *learning algorithm* takes as input a set S of examples for a concept $c \in \mathcal{C}$ and computes a hypothesis h. As mentioned in the Introduction, we have to restrict the set of admissible learners. A *consistent* and *class preserving* learning algorithm is only allowed to choose the hypotheses from the set

$$\mathcal{H}(S) = \{h \in \mathcal{C} \mid h \text{ is consistent with } S\} .$$

A *teaching set*[1] for a concept c with respect to $\mathcal{C}$ is a set S of examples such that c is the only concept in $\mathcal{C}$ consistent with S, i.e., $\mathcal{H}(S) = \{c\}$ (cf. [12,11]). The *teaching dimension* $\mathrm{TD}(c)$ is the size of the smallest teaching set for c, the teaching dimension of $\mathcal{C}$ is

$$\mathrm{TD}(\mathcal{C}) = \max\{\mathrm{TD}(c) \mid c \in \mathcal{C}\} . \tag{1}$$

Consequently, the teaching dimension of a concept c determines the number of examples needed by an optimal teacher for teaching c to all consistent and class preserving learning algorithms. So in the TD model the information theoretic complexity of teaching is reduced to a combinatorial parameter. Note that the teaching dimension has been calculated for many natural concept classes such as (monotone) monomials, monotone k-term DNFs, k-term μ-DNFs, monotone decision lists and rectangles in $\{0, 1, \ldots, n-1\}^d$ (cf. [12]); for linearly separable Boolean functions (cf. [21,22]); for threshold functions (cf. [13]); and for k-juntas and sparse GF_2 polynomials (cf. [23]).

Since the teaching dimension does depend exclusively on the concept class, it has also been compared to other combinatorial parameters studied in learning theory. These parameters comprise the query complexity in Angluin's [2] query learning model, the VC-dimension and parameters studied in the online learning model (cf. Hegedűs [24,25], Ben-David and Eiron [26], and Rivest and Yin [27]).

Despite its succinctness and elegance, the teaching dimension has also drawbacks. For seeing this, let us consider the following example. Fix any natural number $n \geq 2$ and define the concept class $\mathcal{S}_n = \{c_0, c_1, \ldots, c_n\}$ over X_n as follows: $c_0 = \emptyset$ as well as $c_i = \{x_i\}$ for all $i = 1, \ldots, n$. Then we have $\mathrm{TD}(c_i) = 1$ for all $i = 1, \ldots, n$, since the single positive example $(x_i, 1)$ is sufficient for teaching c_i. Nevertheless, $\mathrm{TD}(c_0) = n$, since there are at least two consistent hypotheses until all n negative examples have been presented to the learners. Thus, $\mathrm{TD}(\mathcal{S}_n) = n$ despite the fact that the class $\mathcal{S}_n$ seems rather simple. However, the teaching dimension is the maximum possible.

Similar effects can be observed for the class of all monomials, all 2-term DNFs, all 1-decision lists, and all Boolean functions (over $\{0,1\}^n$), since all these classes have the same teaching dimension, i.e., 2^n.

2.1 The Average Teaching Dimension

As we have shortly outlined, the teaching dimension does not always capture our intuition about the difficulty to teach concepts. One reason for the implausibility of the results sometimes obtained is due to the fact that the teaching dimension of the class is determined by the worst case teaching dimension over all concepts.

[1] Note that a teaching set is also called *key* [13], *discriminant* [19] and *witness set* [20].

Thus, all easily learnable concepts are not taken into account. So a natural remedy is to consider the *average teaching dimension* instead of the worst case teaching dimension.

Definition 1. *Let $\mathcal{C}$ be a concept class. The* average teaching dimension *of $\mathcal{C}$ is defined as* $\overline{\mathrm{TD}}(\mathcal{C}) = \frac{1}{|\mathcal{C}|}\sum_{c\in\mathcal{C}} \mathrm{TD}(c)$.

Looking again at the class $\mathcal{S}_n$ defined above, we directly see that

$$\overline{\mathrm{TD}}(\mathcal{S}_n) = \frac{n + n\cdot 1}{n+1} < 2 \quad \text{for all } n \geq 2$$

and thus much smaller than the (worst case) teaching dimension $\mathrm{TD}(\mathcal{S}_n) = n$.

Anthony, Brightwell and Shawe-Taylor [22] showed that the average teaching dimension for the class of linearly separable Boolean functions is $O(n^2)$ and Kuhlmann [28] proved that all classes of VC-dimension 1 have an average teaching dimension of less than 2 and that balls of radius d in $\{0,1\}^n$ have an average teaching dimension of at most $2d$.

A more general result was shown by Kushilevitz, Linial, Rabinovich, and Saks [20]. They showed an upper bound of $O(\sqrt{|\mathcal{C}|})$ for the average teaching dimension of any concept class $\mathcal{C}$. Additionally, in [20] a family of classes is defined for which the average teaching dimension is $\Omega(\sqrt{|\mathcal{C}|})$.

Naturally, determining the average teaching dimension for classes that are more complex than $\mathcal{S}_n$ is often much harder than calculating their worst case teaching dimension. However, recently progress has been made. Balbach [14] succeeded in showing that 2-term DNFs and 1-decision lists have an average teaching dimension of $O(n)$ nicely contrasting their teaching dimension which is 2^n.

Based on Balbach's [14] results, Lee, Servedio, and Wan [23] have shown that the class of DNFs with at most $s \leq 2^{\Theta(n)}$ terms has an average teaching dimension of $O(ns)$. Furthermore, they proved that the class of k-juntas has an average teaching dimension of at most $2^k + o(1)$ and that the average teaching dimension of the class of GF_2 polynomials with $s \leq (1-\varepsilon)\log_2 n$ monomials is at most $ns + 2s$.

Nevertheless, there are still points of concern when comparing the TD model and the average teaching dimension model to a scenario where we have a machine teacher and human learners. Such scenarios are of interest for *intelligent tutoring systems* (abbr. ITS), see e.g., `www.aaai.org/AITopics/html/tutor.html`.

Human learners are not necessarily consistent, they do not remember all examples, they are sensitive to the order of examples, and they usually provide feedback about their learning progress.

Clearly, the order of examples does not matter in the TD model and as mentioned in the Introduction, in the TD model feedback is useless. Learners not being consistent with all examples are excluded by the definition of the TD model. There is, however, a dependence on the memory of the learners. As long as the learners can memorize at least $\mathrm{TD}(c)$ many examples, teaching the concept c is possible. If less than $\mathrm{TD}(c)$ many examples can be memorized then teaching becomes *impossible*.

Last but not least, the applicability to infinite concepts and classes is limited. Even a rather simple class, like the class of all finite languages over a fixed alphabet Σ yields an infinite teaching dimension. Therefore, we continue with different approaches to model algorithmic teaching.

3 Teaching Learners with Restricted Mind Changes

In this section we summarize some of the results from Balbach and Zeugmann [16]. We modify the TD model by introducing a neighborhood relation over all possible hypotheses. The learners are then restricted to choose a new hypothesis from the neighborhood of their current one. This may reflect human behavior, since humans tend to modify their hypotheses instead of creating completely new ones.

We then compare basically two variants: In the first, the teacher receives the learner's hypothesis after every example taught. In the second, the teacher has no feedback available. As a matter of fact, in this new model feedback can really make a difference. Some concept classes can be taught much faster with feedback than without and some cannot be taught unless feedback is available to the teacher.

Some additional notation is necessary. Let R be a set of strings. We say that R *represents* the class $\mathcal{C}$ iff there is a function $\gamma\colon R \times X \to \{0,1\}$ such that $\mathcal{C} = \{\mathcal{C}_r \mid r \in R\}$, where $\mathcal{C}_r = \{x \mid \gamma(r,x) = 1\}$. The length of r is denoted by $|r|$ and $\mathit{size}(c) := \min\{|r| \mid \mathcal{C}_r = c\}$ for every $c \in \mathcal{C}$. For any set S, we denote by S^* the set of all finite tuples over S. We use the symbols $\circ$ for concatenation of tuples and $\triangle$ for the symmetric difference of two sets. Let c be a concept and let $\boldsymbol{x} \in \mathcal{X}^*$ be a list of examples, then $\mathit{err}(\boldsymbol{x}, c)$ is the set of all examples in $\boldsymbol{x}$ that are inconsistent with c.

For studying feedback, the learners in our model have to evolve over time. We adopt the online learning model and divide the teaching process into rounds. In each round the teacher provides an example to the learner who then computes a hypothesis from R. At the end of the round the teacher observes this hypothesis.

Thus, we describe a teacher by a function $T\colon R \times R^* \to \mathcal{X}$ receiving a concept's representation and a sequence of previously observed hypotheses as input and outputting an example.

A learner can be described by a function $L\colon \mathcal{X}^* \to R$ receiving a sequence of examples as input and outputting a hypothesis. Let $\nu \subseteq R \times R$ be a relation over R. Then L is called *restricted* to ν iff $\forall \boldsymbol{x} \in \mathcal{X}^*\ \forall z \in \mathcal{X}\ [(L(\boldsymbol{x}), L(\boldsymbol{x} \circ z)) \in \nu]$, that is ν defines the admissible mind changes of L. Now, (R, ν) is a directed graph and we define the neighborhood of $r \in R$ as $\mathit{Nb}(r) := \{s \in R \mid (r,s) \in \nu\} \cup \{r\}$ and denote by $\mathit{dist}(r,s)$ the length of a shortest path from r to s.

As we have seen, in the TD model, the learner is required to always output a consistent hypothesis. Since in the restricted model all admissible hypotheses might be inconsistent, we have to modify this demand. We require that L chooses only among the admissible hypotheses with least error with respect to the known examples. Moreover, we require a form of *conservativeness*: L may only change its hypothesis if the new one has a smaller error. This ensures that L will not change its mind after reaching a correct hypothesis. On the other hand, we also *require* L to search for a better hypothesis if it receives an inconsistent example. Otherwise, L could stay at the initial hypothesis forever and teaching were impossible.

Definition 2. *Let R be a representation language for a concept class $\mathcal{C}$ and let $\nu \subseteq R \times R$ be a relation over R and $h_0 \in R$ a starting hypothesis. A* ν-learner *is a function $L\colon \mathcal{X}^* \to R$ with $L(\emptyset) = h_0$ and for all $\boldsymbol{x} \in \mathcal{X}^*$ and for all $z \in \mathcal{X}$:*

(1) $(L(\boldsymbol{x}), L(\boldsymbol{x} \circ z)) \in \nu$,
(2) *if $L(\boldsymbol{x}) \neq L(\boldsymbol{x} \circ z)$ then z is inconsistent with $\mathcal{C}_{L(\boldsymbol{x})}$,*
(3) *if z is inconsistent with $\mathcal{C}_{L(\boldsymbol{x})}$ then*
$L(\boldsymbol{x} \circ z) \in \arg\min_{s \in Nb(L(\boldsymbol{x}))} |err(\boldsymbol{x} \circ z, \mathcal{C}_s)|$.

We briefly remark that one can think of many plausible variants of the above definition. For instance, the learner could be allowed to change its mind on a consistent example if its hypothesis is inconsistent with an example received earlier. In this section, however, all learners follow Definition 2.

The teaching process for a concept $c = \mathcal{C}_r$ is fully described by a teacher T and a learner L together with an initial hypothesis h_0. Such a process will result in a series $(h_i)_{i \in \mathbb{N}}$ of hypotheses and a series $(z_i)_{i \in \mathbb{N}}$ of examples: $h_{i+1} = L(z_0, \ldots, z_i)$ and $z_i = T(r, (h_0, \ldots, h_i))$.

Definition 3. *Let $\mathcal{C}$ be a concept class with representation R and let $\nu \subseteq R \times R$. We call $\mathcal{C}$* teachable to ν-learners in the limit with feedback *iff there is a teacher T such that for all representations $r \in R$ and all ν-learners L the series $(h_i)_{i \in \mathbb{N}}$ of hypotheses converges to an h with $\mathcal{C}_h = \mathcal{C}_r$.*

The teaching time of T on r *is the maximum i such that there is a ν-learner L that reaches a representation of $\mathcal{C}_r$ at round i for the first time.*

Note that an infinite teaching time does not imply unteachability of a concept. For studying the influence of feedback, we also have to define teaching without feedback. In this situation the teacher is modeled as a function $T\colon R \times \mathbb{N} \to \mathcal{X}$, where the second argument specifies the round. The series of hypotheses is then given by $h_{i+1} = L(T(r, 0), \ldots, T(r, i))$. With this notation the definition of teaching *in the limit without feedback* is literally the same as Definition 3.

In the situation with feedback the teacher can stop teaching as soon as the learner has reached the goal. If there is no feedback, the teacher may or may not know when to stop. A teacher stopping after finitely many examples and still ensuring the learning success is said to teach *finitely without feedback*. More formally we consider $T\colon R \times \mathbb{N} \to \mathcal{X} \cup \{\perp\}$ where $\perp$ means "teaching has stopped."

With feedback we do not need to distinguish teaching finitely from teaching in the limit and we shall call this kind of teaching simply *teaching with feedback*.

Definition 4. *Let $\mathcal{C}$ be a concept class with representation R and let $\nu \subseteq R \times R$. We call $\mathcal{C}$* finitely teachable to ν-learners without feedback *iff there is a teacher T such that for all representations $r \in R$ and all ν-learners L the hypothesis h_j with $j = \min\{i \mid T(r, i) = \perp\}$ satisfies $\mathcal{C}_{h_j} = \mathcal{C}_r$.*

Setting $\nu = R \times R$ in Definition 4 gives the teacher-directed learning model having no restriction on hypothesis changes (cf. [11]). Theorem 5 justifies the use of *arbitrary* ν's for studying the impact of feedback on the teaching process.

Theorem 5. *Let $\mathcal{C}$ be a concept class with representations R and let $\nu = R \times R$. Then the following statements are equivalent:*

(1) *$\mathcal{C}$ is finitely teachable to ν-learners without feedback,*
(2) *$\mathcal{C}$ is teachable in the limit to ν-learners without feedback,*
(3) *$\mathcal{C}$ is teachable to ν-learners with feedback.*

Furthermore in all three cases the same teacher can be used to obtain minimum teaching time which for all $c \in \mathcal{C}$ equals $\mathrm{TD}(c)$ with respect to $\mathcal{C}$.

Note that Theorem 5 relies on the fact that neither the teacher nor the learners nor the function γ are required to be recursive. Adding these requirements leads to new questions which we skip here due to space constraints.

Next, we apply the new framework to the class $\mathcal{C}_{\mathit{fin}}$ of all finite languages over an alphabet Σ. This class cannot be taught in the TD model. By using different ν-restrictions we demonstrate various effects.

We fix any total ordering on all strings over Σ and use as representation language R the set of all comma-separated ordered lists of strings over Σ, i.e., $r = w_1, \ldots, w_m \in R$ represents the language $\{w_1, \ldots, w_m\}$. We define the allowed transitions from r to s by $(r, s) \in \nu$ iff $|\mathcal{C}_r \triangle \mathcal{C}_s| \leq 1$. The initial hypothesis is the empty string ε representing the empty concept. Now we have:

Fact 6. *$\mathcal{C}_{\mathit{fin}}$ is finitely teachable to ν-learners without feedback.*

Feedback can be utilized when the restriction is modified. We define $(r, s) \in \nu'$ iff $\mathcal{C}_s = \mathcal{C}_r \cup \{w_1, w_2\}$ for some $w_1, w_2 \in \Sigma^*$ or $\mathcal{C}_s = \mathcal{C}_r \setminus \{w_1\}$. In both cases, we require that the size of the hypotheses may at most double each round: $|s| \leq 2|r|$. In the special case $r = \varepsilon$ we allow every singleton concept as neighbor: $(\varepsilon, s) \in \nu$ for all s with $|\mathcal{C}_s| = 1$. For ν'-learners there is a big difference in teaching time between teaching with and without feedback.

Fact 7. *$\mathcal{C}_{\mathit{fin}}$ is teachable to ν'-learners with feedback such that for all $c \in \mathcal{C}$ the number of examples is $O(|c|) \leq O(\mathit{size}(c))$.*

If we remove the size restriction from ν' we obtain ν''.

Fact 8. *$\mathcal{C}_{\mathit{fin}}$ is not finitely teachable to ν''-learners without feedback, but it is finitely teachable with feedback as well as in the limit without feedback.*

Finally we define ν'''. It differs from ν'' in that a string may only be removed from the hypothesis if neither its predecessor nor its successor (with respect to the fixed ordering on Σ^*) is contained in the hypothesis.

Fact 9. *$\mathcal{C}_{\mathit{fin}}$ is not teachable to ν'''-learners in the limit without feedback, but it is finitely teachable with feedback.*

If we denote by $\mathit{TFIN}, \mathit{TFB}, \mathit{TLIM}$ the set of all $(\mathcal{C}, R, \nu, h_0)$ such that $\mathcal{C}$ is finitely teachable without feedback, with feedback or in the limit, respectively, we have just proved the following theorem.

Theorem 10. *$\mathit{TFIN} \subset \mathit{TLIM} \subset \mathit{TFB}$.*

The teaching times in our model can hardly be compared to the teaching dimension, since the latter depends only on $\mathcal{C}$, whereas different choices of ν can lead to different teaching times for the same $\mathcal{C}$. The problem of finding an optimal teacher (with or without feedback) for ν-learners is $\mathcal{NP}$-hard, since it is a generalization of finding an optimal teaching set, namely if $\nu = R \times R$ (cf. [13,12,21]). More precisely, we have the following theorem.

Theorem 11. *For all notions of teaching, the following problem is $\mathcal{NP}$-hard:*

Instance*: $\mathcal{C}, R, \nu$, and a concept c^* as 0-1-vector of length $|X|$.*
Question*: Can c^* be taught to ν-learners?*

For infinite instance spaces or classes (and infinite ν) the next theorem applies.

Theorem 12. *The following function is not computable:*

Input*: Algorithms computing total functions deciding $\mathcal{C}$ and ν.*
Output*: 1, if $\mathcal{C}$ can be taught to ν-learners; 0 otherwise.*

We finish this section by looking at teaching *without* feedback. A teacher T without feedback knows all learners' initial hypotheses h_0, but can quickly lose track of them during teaching. On the other hand, T can rule out neighbors r of h_0 by giving examples consistent with h_0, but inconsistent with r. If in such a way T can eliminate all but one neighbor r', he effectively forces all learners to switch to r'. By continuing in this manner, T always knows all learners' hypotheses even without feedback. If the enforced hypotheses approach the target, T will be successful. Figure 1 describes this strategy more formally.

1 $r := h_0$;
2 **while** $\mathcal{C}_r \neq c^*$ **do**:
  2.1 Find $s \in \mathit{Nb}(r)$, $S \subseteq \mathcal{X}$, and $z \in \mathcal{X}$ such that (1) $\mathcal{C}_r$ is consistent with S, but not with z, (2) s is the only neighbor of r consistent with $S \cup \{z\}$, and (3) $\mathit{dist}(s, r^*) < \mathit{dist}(r, r^*)$;
  2.2 Teach S in arbitrary order and then z;
  2.3 $r := s$;

Fig. 1. A simple general strategy for teaching without feedback by forcing all learners to make the same mind changes. The initial hypothesis is h_0, r^* represents the target.

The feasibility of this strategy depends on Step 2.1. If teaching does not need to be finite, the condition in Step 2 does not need to be checked. Albeit simple, the strategy works surprisingly often for natural concept classes and ν-restrictions. In the following we give some examples.

First, we consider the class of all monomials over n variables. Let $R = \{0, 1, *\}^n$ and define $(r, s) \in \nu$ iff r and s differ only in one "bit." As initial hypothesis $h_0 = *^n$ is used.

Fact 13. *Monomials are finitely teachable without feedback. The teaching time for each concept equals its teaching dimension.*

Next, we look at decision trees. Each learner starts at the tree consisting of only one negative leaf. In each round one leaf may be substituted by an internal node that has two differently labeled leaves as children. This specifies a relation ν_{DT} over all decision trees. Then, we have:

Fact 14. *The class of Boolean functions represented as decision trees can be taught without feedback to ν_{DT}-learners. The teaching time is linear in the size of the tree representation.*

Note that the teaching dimension with respect to all Boolean functions is 2^n for all concepts. As we have seen, for ν-learners based on decision trees, teaching can often be successful with much fewer examples. Finally, we have been a bit surprised to obtain:

Fact 15. *The class of monotone 1-decision lists can be taught to ν_{DL}-learners with feedback using $m + 1$ examples for a list of length m. It cannot be taught without feedback.*

In our model of teaching learners with restricted mind changes several effects regarding feedback can be observed. Feedback can be useless, helpful, or even indispensable for teaching. In addition, natural infinite concept classes can be taught in this model. However, the main drawback is that one has to define suitably a neighborhood relation for every concept class. Our next model avoids this difficulty. It will also allows us to study teaching learners with limited memory.

4 The Randomized Teaching Model

For the sake of motivation, let us consider the concept class of all Boolean functions over $\{0, 1\}^n$. To teach a concept to all consistent learning algorithms, i.e., in the TD model, the teacher must present all 2^n examples. Teaching a concept to all consistent learners that can memorize less than 2^n examples is impossible; there is always a learner with a consistent, but wrong hypothesis. So teaching gets harder, but in a rather abrupt way.

It seems that the worst case analysis style makes it impossible to investigate the influence of memory limitations or learner's feedback. A common remedy for this is to perform an average case analysis instead (cf. Subsection 2.1). In this section we look at a rather radical approach, i.e., we replace the set of learners by a single one that is intended to represent an "average learner."

We achieve this goal by substituting the set of deterministic learners by a single randomized one. Basically, such a learner picks a hypothesis at random from all hypotheses consistent with the known examples. Teaching is successful as soon as the learner hypothesizes the target concept. For ensuring that the learner maintains this correct hypothesis, we additionally require the learner to be *conservative*, i.e., it can change its hypotheses only on examples that are inconsistent with its current hypothesis. The complexity of teaching is measured by the *expected* teaching time.

Next, we explain why this model should work. Since at every round there is a chance to reach the target, the target will eventually be reached even if, for instance, the randomized learner can only memorize few examples. The ability of the teacher to observe the learner's current hypothesis should be advantageous, since it enables the teacher to teach an inconsistent example in every round. Recall that only these examples can cause a hypothesis change. Below we show these intuitions to be valid.

Note that for randomized learners the complexity of the teaching process does not only depend on the examples, but also on the *order* in which they are given to the learner.

The randomized teaching model can be regarded as a *Markov Decision Process* (abbr. MDP). Such processes have been studied for several decades and we shall make use of some results from this theory (cf. [29,30]). An MDP is a probabilistic system whose state transitions can be influenced during the process by actions which incur costs. Let $\mathbb{R}$ denote the set of all real numbers. Formally, an MDP consists of a finite set S of states, an initial state $s_0 \in S$, a finite set A of actions, a function $cost\colon S \times A \to \mathbb{R}$, and a function $p\colon S \times A \times S \to [0,1]$; $cost(s,a)$ is the cost incurred by action a in state s; $p(s,a,s')$ is the probability for the MDP to change from state s to s' under action a.

In the *total cost infinite horizon* setting, the goal is to choose actions such that the expected total cost, when the MDP runs forever, is minimal. This makes sense only if there is a costless absorbing state $s^* \in S$. In the *finite horizon* setting the MDP is only run for finitely many rounds.

The actions chosen at each point in time are described by a *policy*. This is a function depending on the observed history of the MDP and the current state. A basic result says that there is a minimum-cost policy that is *stationary*, i.e., that depends only on the current state. A stationary policy $\pi\colon S \to A$ defines a Markov chain over S and for all $s \in S$ an expected time $H(s)$ to reach s^* from s. Such a policy is optimal iff for all $s \in S$:

$$\pi(s) \in \operatorname*{argmin}_{a \in A} \left(cost(s,a) + \sum_{s' \in S} p(s,a,s') \cdot H(s') \right) .$$

Finding optimal policies can be phrased as a linear programming problem and can thus be done in polynomial time in the representation size of the MDP.

For the following, we need a bit more notation. For numbers a, b with $a < b$ we write $[a,b]$ to denote the set $\{a, a+1, \ldots, b\}$ or $\{a, a+1, \ldots\}$ if $b = \infty$. As above, for any set S, we denote by S^* the set of all finite lists of elements from S. Furthermore, by S^m and $S^{\leq m}$ we denote the set of all lists with length m and at most length m, respectively. The operator $\circ_\mu$ concatenates a list of length at most

μ with a single element resulting in a list of length at most μ: $\langle x_1, \ldots, x_\ell \rangle \circ_\mu \langle y \rangle$ equals $\langle x_1, \ldots, x_\ell, y \rangle$ if $\ell < \mu$ and $\langle x_2, \ldots, x_\ell, y \rangle$ if $\ell = \mu$. We regard $\circ_\infty$ as the usual list concatenation. For a list $\boldsymbol{x}$ of examples, we set

$$\mathcal{C}(\boldsymbol{x}) = \{c \in \mathcal{C} \mid \boldsymbol{x} \text{ is consistent with } c\} \ .$$

We denote by $\mathcal{M}_n$ the concept class of monomials over $\{0,1\}^n$. We exclude the empty concept from $\mathcal{M}_n$ and can thus identify each monomial with a string from $\{0,1,*\}^n$ and vice versa. We use $\mathcal{D}_n$ to denote the set of all 2^n concepts over $[1,n]$. Thus, there are 2^n many concepts in $\mathcal{D}_n$.

Next, we define the randomized teaching model. The teaching process is divided into rounds. In each round the teacher gives the learner an example of a target concept. The learner memorizes this example and computes a new hypothesis based on its last hypothesis and the known examples.

The Learner. In a sense, consistency is a minimum requirement for a learner. We thus require our learners to be consistent with all examples they know. However, the hypothesis is chosen at *random* from all consistent ones.

The memory of our learners may be limited to $\mu \geq 1$ examples. If the memory is full and a new example arrives, the oldest example is erased. In other words, the memory works like a queue. Setting $\mu = \infty$ models unlimited memory.

The goal of teaching is making the learner to hypothesize the target *and to maintain it.* Consistency alone cannot guarantee this behavior if the memory is too small. In this case, there is more than one consistent hypothesis at every round and the learner would oscillate between them rather than maintaining a single one. To avoid this, *conservativeness* is required, i.e., the learner can change its hypothesis only when taught an example inconsistent with its current one.

To study the influence of the learners' feedback to the teacher, we distinguish between private and public output of the learner. The private output is the result of the calculation during a round (i.e., new memory content and hypothesis), the public output is that part of the private one observable by the teacher. So, if the learner gives feedback, the teacher can observe in every round the complete hypothesis computed by the learner. If the learner does not give feedback, the teacher can observe nothing. The following algorithm describes the behavior of the *μ-memory learner with/without feedback* (short: L_μ^+ / L_μ^-) during one round of the teaching process.

Input: memory $\boldsymbol{x} \in \mathcal{X}^{\leq \mu}$, hypothesis $h \in \mathcal{C}$, example $z \in \mathcal{X}$.
Private Output: memory $\boldsymbol{x}'$, hypothesis h'.
Public Output: hypothesis h' / nothing.

$\boldsymbol{x}' := \boldsymbol{x} \circ_\mu \langle z \rangle$;
if $z \notin \mathcal{X}(h)$ **then** pick h' uniformly at random from $\mathcal{C}(\boldsymbol{x}')$;
else $h' := h$;

For making our results dependent on $\mathcal{C}$ alone, rather than on an arbitrary initial state of the learner, we stipulate a special initial hypothesis, called *init.* We assume every example inconsistent with *init.* Thus, *init* is left after the first example and cannot be reached again. Moreover, the initial memory is empty.

The Teacher. A teacher is an algorithm taking initially a given target concept c^* as input. Then, in each round, it receives the public output of the learner (if any) and outputs an example for c^*.

Definition 16. *Let $\mathcal{C}$ be a concept class and $c^* \in \mathcal{C}$. Let L_μ^σ be a learner, where $\sigma \in \{+,-\}$, let T be a teacher and let $(h_i)_{i\in\mathbb{N}}$ be the series of random variables for the hypothesis at round i. The event* "teaching success in round t," *denoted by G_t, is defined as*

$$h_{t-1} \neq c^* \quad \wedge \quad \forall t' \geq t\colon h_{t'} = c^* \ .$$

The success probability *of T is* $\Pr\left[\bigcup_{t\geq 1} G_t\right]$. *A teaching process is* successful *iff the success probability equals* 1. *A successful teaching process is called* finite *iff there is a t' such that* $\Pr\left[\bigcup_{1\leq t\leq t'} G_t\right] = 1$, *otherwise it is called* infinite. *For a successful teaching process we define the* expected teaching time *as* $\mathbb{E}[T, L_\mu^\sigma, c^*, \mathcal{C}] := \sum_{t\geq 1} t \cdot \Pr[G_t]$.

Definition 17. *Let $\mathcal{C}$ be a concept class, $c^* \in \mathcal{C}$ and L_μ^σ a learner. We call c^** teachable *to L_μ^σ iff there is a successful teacher T. The* optimal teaching time for c^* *is*

$$E_\mu^\sigma(c^*) := \inf_T \mathbb{E}[T, L_\mu^\sigma, c^*, \mathcal{C}]$$

and the optimal teaching time for $\mathcal{C}$ is denoted by $E_\mu^\sigma(\mathcal{C}) := \max_{c\in\mathcal{C}} E_\mu^\sigma(c)$.

For exemplifying our model, we compute the optimal teaching times for $\mathcal{D}_n$. To the learner L_μ^+ $(1 \leq \mu \leq n)$ the teacher gives an example inconsistent with the current hypothesis in each round. For all such examples, there are $2^{n-\mu}$ hypotheses consistent with the μ examples in the learner's memory and it chooses one of them. So the probability of choosing the target concept is $2^{-(n-\mu)}$. Since in the first $\mu - 1$ rounds the memory contains less then μ examples, $E_\mu^+(\mathcal{D}_n)$ is, for constant μ, asymptotically equal to $2^{n-\mu}$. Clearly, teaching becomes faster with growing μ. Moreover the teaching speed increases continuously with μ and not abruptly as in the classical deterministic model. In particular, teaching is possible even with the smallest memory size ($\mu = 1$), although it takes very long (2^{n-1} rounds).

Teaching is more difficult when feedback is unavailable. In this situation the teacher can merely guess examples hoping that they are inconsistent with the current hypothesis. Roughly speaking, when teaching $\mathcal{D}_n$, the teacher needs two guesses on average to find such an example. Hence, the expected teaching time E_μ^- is about two times E_μ^+. Thus feedback doubles the teaching speed for $\mathcal{D}_n$.

Fact 18. *For all $\mathcal{C}$ and $\mu \in [1,\infty]$ all $c^* \in \mathcal{C}$ and $\sigma \in \{+,-\}$:*

(1) $E_\mu^+(c^*) \leq E_\mu^-(c^*)$, (2) $E_\infty^\sigma(c^*) \leq E_{\mu+1}^\sigma(c^*) \leq E_\mu^\sigma(c^*)$.

Proper inequality holds for the concepts in $\mathcal{D}_n$.

Next, we relate the TD model, i.e., the teaching dimension, to the randomized model (in terms of the expected teaching time).

Lemma 19 ([18]). *Let $\mathcal{C}$ be a concept class and let $c^* \in \mathcal{C}$ be a target. For all $\mu \in [1, \mathrm{TD}(c^*)]$,*

$$E^-_\mu(c^*) \;\geq\; E^+_\mu(c^*) \;\geq\; \frac{\mu(\mu-1)}{2\mathrm{TD}(c^*)} + \mathrm{TD}(c^*) + 1 - \mu,$$

and for all $\mu > \mathrm{TD}(c^*)$, $E^-_\mu(c^*) \;\geq\; E^+_\mu(c^*) \;\geq\; \mathrm{TD}(c^*)/2$.

Now, we take a closer look at learners with feedback. For the sake of presentation we start with learners with 1-memory. A teaching process involving L^+_1 can be modeled as an MDP with $S = \mathcal{C} \cup \{init\}$, $A = \mathcal{X}(c^*)$, $cost(h,z) = 1$ for $h \neq c^*$ and $cost(c^*, z) = 0$. Furthermore, for $h \neq c^*$, $p(h,z,h') = 1/|\mathcal{C}(z)|$ if $z \in \mathcal{X}(h') \setminus \mathcal{X}(h)$ and $p(h,z,h') = 0$ otherwise; finally $p(c^*, z, c^*) = 1$ (see [30,29]). The initial state is *init* and the state c^* is costless and absorbing. The memory is not part of the state, since the next hypothesis only depends on the newly given example which is modeled as an action.

An example $z \in \mathcal{X}(h)$ does not change the learner's state h and is therefore useless. An optimal teacher refrains from teaching such examples and thus we can derive the following criterion.

Lemma 20. *Let $\mathcal{C}$ be a concept class over X and let c^* be a target. A teacher $T\colon \mathcal{C} \cup \{init\} \to \mathcal{X}(c^*)$ with expectations $H\colon \mathcal{C} \cup \{init\} \to \mathbb{R}$ is optimal iff for all $h \in \mathcal{C} \cup \{init\}$:*

$$T(h) \in \operatorname*{argmin}_{\substack{z \in \mathcal{X}(c^*) \\ z \notin \mathcal{X}(h)}} \left(1 + \frac{1}{|\mathcal{C}(z)|} \sum_{h' \in \mathcal{C}(z)} H(h') \right) .$$

This criterion can be used to prove optimality for teaching algorithms.

We compare E^+_1 with other dimensions. The comparison of E^+_1 with the number *MQ* of membership queries (see Angluin [1]) is interesting because *MQ* and E^+_1 are both lower bounded by the teaching dimension.

Fact 21

(1) *For all $\mathcal{C}$ and $c^* \in \mathcal{C}$: $E^+_1(c^*) \geq \mathrm{TD}(c^*)$.*
(2) *There is no function of* TD *upper bounding $E^+_1(c)$.*
(3) *There is no function of E^+_1 upper bounding MQ.*
(4) *There is a concept class $\mathcal{C}$ with $E^+_1(\mathcal{C}) > MQ(\mathcal{C})$.*
(5) *For all concept classes $\mathcal{C}$, $E^+_1(\mathcal{C}) \leq 2^{MQ(\mathcal{C})}$.*

Roughly speaking, teaching L^+_1 can take arbitrarily longer than teaching in the classical model, but is still incomparable with membership query learning.

We finish this subsection by looking at learners with ∞-memory. A straightforward MDP for teaching c^* to L^+_∞ has states $S = (\mathcal{C} \cup \{init\}) \times \mathcal{X}(c^*)^{\leq |X|}$. The number of states can be reduced because two states (h,m) and (h,m') with $\mathcal{C}(m) = \mathcal{C}(m')$ are equivalent from a teacher's perspective, but in general the size of the resulting MDP will not be polynomial in the size of the matrix representation of $\mathcal{C}$. Therefore, optimal teachers cannot be computed efficiently by the known general MDP algorithms.

A similar criterion as Lemma 20 can be stated for the L_∞^+ learner, too, and used to prove optimality of algorithms. Note that there is always a teacher that needs at most $\mathrm{TD}(c^*)$ rounds by giving a minimal teaching set, hence $E_\infty^+(c^*) \leq \mathrm{TD}(c^*)$. Second, it follows from Lemma 19 that $E_\infty^+(c^*) \geq \mathrm{TD}(c^*)/2$. This means that every algorithm computing $E_\infty^+(c^*)$ also computes a factor 2 approximation of the teaching dimension.

As it has often been noted [13,21,12], the problem of computing the teaching dimension is essentially equivalent to the `SET-COVER` (or `HITTING-SET`) problem which is a difficult approximation problem. Raz and Safra [31] have shown that there is no polynomial time constant-factor approximation (unless $\mathcal{P} = \mathcal{NP}$). Moreover, Feige [32] proved that `SET-COVER` cannot be approximated better than within a logarithmic factor (unless $\mathcal{NP} \subseteq DTime(n^{\log\log n})$).

Corollary 22. *Unless $\mathcal{NP} \subseteq DTime(n^{\log\log n})$, computing E_∞^+ is $\mathcal{NP}$-hard and cannot be approximated with a factor of $(1-\epsilon)\log(|\mathcal{C}|)$ for any $\epsilon > 0$.*

However, we have:

Fact 23. *Let $\mathcal{C}$ be a concept class and $c^* \in \mathcal{C}$ a target. Then there is a successful teacher for the learner L_∞^+ halting after at most $|X|$ rounds that is also optimal.*

Note that our model of teaching a randomized learner also allows for studying teaching from *positive data* only. The interested reader is referred to [18] for details. Here we only mention the following topological characterization.

Theorem 24. *Let $\mathcal{C}$ be a concept class and $c^* \in \mathcal{C}$ a target concept. Then for all learners L_μ^σ with $\mu \in [1,\infty]$, $\sigma \in \{+,-\}$: The concept c^* is teachable from positive data iff there is no $c \in \mathcal{C}$ with $c \supset c^*$.*

Studying the teachability of randomized learners without feedback is much harder, since the problem of finding the optimal cost in an MDP whose states are not observable is much more difficult. We refer the interested reader to Balbach and Zeugmann [17] for results in this regard.

5 Further Directions

A number of variations of the teaching dimension have been studied in which the learner is assumed to act smarter than just choosing a consistent hypothesis. One such model (see Balbach [14,15]) assumes the learner picks hypotheses that are not only consistent but of minimal complexity. This model is inspired by the *Occam's razor* principle. A teacher that exploits this learning behavior can make do with fewer examples than in the original teaching dimension model. For example, 2-term DNFs and 1-decision lists can be taught with $O(n)$ examples, as opposed to 2^n examples (n being the number of variables). Note that this model presupposes a measure of complexity for all concepts in the class, such as the length of a decision list or the number of terms in a DNF. But there might not be a unique, natural complexity measure for a given class.

Another variant devised by Balbach [15] assumes that the learners know the teaching dimensions of all concepts in the class and choose their hypotheses only from the consistent ones with a teaching dimension at least as large as the sample given by the teacher. In other words, they assume the teacher does not give more examples than necessary for the concept to be taught. This *optimal teacher teaching dimension (OTTD)* model demands more of the learners, as they need to know all the teaching dimensions, but reduces the number of examples compared to the plain teaching dimension. For example, the OTTD of monomials is linear in the number of variables, and the OTTD of 1-decision lists is bounded by $\Omega(\sqrt{n} \cdot 2^{n/2})$ and $O(n\sqrt{\log n} \cdot 2^{n/2})$.

If the learners in the OTTD model base their reasoning on the OTTD rather than the teaching dimension, one obtains yet another dimensionality measure. Intuitively, the learners assume that the teacher knows how they think and adjusts the sample he gives. Now the teacher can exploit this new behavior of the students, again reducing the number of examples needed. This gives rise to a series of decreasing dimensionality notions which, for finite concept classes, will eventually converge.

In a similar vein, Zilles, Lange, Holte, and Zinkevich [33] defined a teaching model which is based on cooperative learners. Here the learners are assumed to know all minimal teaching sets for all concepts in the class, and they always choose from the consistent hypotheses a minimal teaching set which contains all examples given so far. In other words they assume that at all times the sample given by the teacher can be extended to a minimal teaching set for the concept to be taught. This demands even more of the learners than the OTTD, as they now have to know all minimal teaching sets of all concepts. In a similar way as above, this idea can be iterated, yielding a series of dimensionality notions that converge to one called the *subset teaching dimension (STD)*. The STD is a lower bound for the iterated OTTDs. Zilles *et al.* [33] show a number of surprising properties of the STD. For example, the STD of the class of monomials is two, independent of the number of variables. A surprising property of the STD is its nonmonotonicity, that is, the STD of some classes is less than that of one of their subclasses.

Finally, Zilles *et al.* [33] devise a dimensionality notion called the *recursive teaching dimension (RTD)*, which combines the easy teachability of the monomials in the STD model with the property of monotonicity in the TD and OTTD models. However, it is not based on refined assumptions about the learners' behavior and only measures the teachability of an entire concept class, not that of individual concepts. The basic idea is to order the concepts in a class and determine the teaching dimension of each concept with respect only to the class of all following concepts. The maximum teaching dimension for any concept minimized over all possible orderings determines the RTD of the class. The RTD is a lower bound for all iterated OTTDs, but its precise relationship to the STD is unknown. The STD is conjectured to be a lower bound for the RTD.

References

1. Angluin, D.: Queries and concept learning. Machine Learning 2, 319–342 (1988)
2. Angluin, D.: Queries revisited. Theoret. Comput. Sci. 313, 175–194 (2004)
3. Freivalds, R., Kinber, E.B., Wiehagen, R.: Inductive inference from good examples. In: Jantke, K.P. (ed.) AII 1989. LNCS (LNAI), vol. 397, pp. 1–17. Springer, Heidelberg (1989)
4. Freivalds, R., Kinber, E.B., Wiehagen, R.: On the power of inductive inference from good examples. Theoret. Comput. Sci. 110, 131–144 (1993)
5. Jain, S., Lange, S., Nessel, J.: On the learnability of recursively enumerable languages from good examples. Theoret. Comput. Sci. 261, 3–29 (2001)
6. Jackson, J., Tomkins, A.: A computational model of teaching. In: Proc. 5th Annual ACM Workshop on Computational Learning Theory, pp. 319–326. ACM Press, New York (1992)
7. Goldman, S.A., Mathias, H.D.: Teaching a smarter learner. J. of Comput. Syst. Sci. 52, 255–267 (1996)
8. Mathias, H.D.: A model of interactive teaching. J. of Comput. Syst. Sci. 54, 487–501 (1997)
9. Angluin, D., Kriķis, M.: Teachers, learners and black boxes. In: Proceedings of the Tenth Annual Conference on Computational Learning Theory, pp. 285–297. ACM Press, New York (1997)
10. Angluin, D., Kriķis, M.: Learning from different teachers. Machine Learning 51, 137–163 (2003)
11. Goldman, S.A., Rivest, R.L., Schapire, R.E.: Learning binary relations and total orders. SIAM J. Comput. 22, 1006–1034 (1993)
12. Goldman, S.A., Kearns, M.J.: On the complexity of teaching. J. of Comput. Syst. Sci. 50, 20–31 (1995)
13. Shinohara, A., Miyano, S.: Teachability in computational learning. New Generation Computing 8, 337–348 (1991)
14. Balbach, F.J.: Teaching classes with high teaching dimension using few examples. In: Auer, P., Meir, R. (eds.) COLT 2005. LNCS (LNAI), vol. 3559, pp. 668–683. Springer, Heidelberg (2005)
15. Balbach, F.J.: Measuring teachability using variants of the teaching dimension. Theoret. Comput. Sci. 397, 94–113 (2008)
16. Balbach, F.J., Zeugmann, T.: Teaching learners with restricted mind changes. In: Jain, S., Simon, H.U., Tomita, E. (eds.) ALT 2005. LNCS (LNAI), vol. 3734, pp. 474–489. Springer, Heidelberg (2005)
17. Balbach, F.J., Zeugmann, T.: Teaching memoryless randomized learners without feedback. In: Balcázar, J.L., Long, P.M., Stephan, F. (eds.) ALT 2006. LNCS (LNAI), vol. 4264, pp. 93–108. Springer, Heidelberg (2006)
18. Balbach, F.J., Zeugmann, T.: Teaching randomized learners. In: Lugosi, G., Simon, H.U. (eds.) COLT 2006. LNCS (LNAI), vol. 4005, pp. 229–243. Springer, Heidelberg (2006)
19. Natarajan, B.K.: Machine Learning: A Theoretical Approach. Morgan Kaufmann, San Mateo (1991)
20. Kushilevitz, E., Linial, N., Rabinovich, Y., Saks, M.: Witness sets for families of binary vectors. Journal of Combinatorial Theory Series A 73, 376–380 (1996)
21. Anthony, M., Brightwell, G., Cohen, D., Shawe-Taylor, J.: On exact specification by examples. In: Proc. 5th Annual ACM Workshop on Computational Learning Theory, pp. 311–318. ACM Press, New York (1992)

22. Anthony, M., Brightwell, G., Shawe-Taylor, J.: On specifying Boolean functions by labelled examples. Discrete Applied Mathematics 61, 1–25 (1995)
23. Lee, H., Servedio, R.A., Wan, A.: DNF are teachable in the average case. In: Lugosi, G., Simon, H.U. (eds.) COLT 2006. LNCS, vol. 4005, pp. 214–228. Springer, Heidelberg (2006)
24. Hegedűs, T.: Combinatorial results on the complexity of teaching and learning. In: Privara, I., Ružička, P., Rovan, B. (eds.) MFCS 1994. LNCS, vol. 841, pp. 393–402. Springer, Heidelberg (1994)
25. Hegedűs, T.: Generalized teaching dimensions and the query complexity of learning. In: Proc. 8th Annual Conference on Computational Learning Theory, pp. 108–117. ACM Press, New York (1995)
26. Ben-David, S., Eiron, N.: Self-directed learning and its relation to the VC-dimension and to teacher-directed learning. Machine Learning 33, 87–104 (1998)
27. Rivest, R.L., Yin, Y.L.: Being taught can be faster than asking questions. In: Proc. 8th Annual Conference on Computational Learning Theory, pp. 144–151. ACM Press, New York (1995)
28. Kuhlmann, C.: On teaching and learning intersection-closed concept classes. In: Fischer, P., Simon, H.U. (eds.) EuroCOLT 1999. LNCS, vol. 1572, pp. 168–182. Springer, Heidelberg (1999)
29. Puterman, M.L.: Markov Decision Processes: Discrete Stochastic Dynamic Programming. John Wiley & Sons, Chichester (1994)
30. Bertsekas, D.P.: Dynamic Programming and Optimal Control. Athena Scientific (2001)
31. Raz, R., Safra, S.: A sub-constant error-probability low-degree test, and a sub-constant error-probability PCP characterization of NP. In: Proc. of the 29th ACM Symposium on Theory of Computing, pp. 475–484. ACM Press, New York (1997)
32. Feige, U.: A threshold of ln n for approximating set cover. J. of the ACM 45, 634–652 (1998)
33. Zilles, S., Lange, S., Holte, R., Zinkevich, M.: Teaching dimensions based on cooperative learning. In: 21st Annual Conference on Learning Theory - COLT 2008, Helsinki, Finland, pp. 135–146. Omnipress (2008)