Teaching a black-box learner

Sanjoy Dasgupta [1] **Daniel Hsu** [2] **Stefanos Poulis** [1 3] **Xiaojin Zhu** [4]

Abstract

One widely-studied model of *teaching* (Goldman & Kearns, 1995; Shinohara & Miyano, 1991; Anthony et al., 1992) calls for a teacher to provide the minimal set of labeled examples that uniquely specifies a target concept. The assumption is that the teacher knows the learner's hypothesis class, which is often not true of real-life teaching scenarios. We consider the problem of teaching a learner whose representation and hypothesis class are *unknown*: that is, the learner is a black box.

We find that a teacher who does not interact with the learner can do no better than providing random examples. However, by interacting with the black-box learner, a teacher can efficiently find a set of teaching examples that is a provably good approximation to the optimal set.

As an illustration, we show how this scheme can be used to *shrink* training sets for any family of classifiers: that is, to find an approximately-minimal subset of training instances that yields the same classifier as the entire set.

1. Introduction

The theory of machine learning has focused primarily on situations where training data consists of random samples from an underlying distribution, as in the statistical learning framework (Valiant, 1984), or is chosen in an arbitrary and possibly adversarial manner, as in online learning (Littlestone, 1988). In real life, however, data is often chosen by a teacher who wishes to help the learner. This is likely to be the case, for instance, when a human is personalizing an electronic assistant; or when an intelligent tutoring system is explaining concepts to a human student; or when one machine is communicating a classifier to another machine with a different architecture or representation. To model such scenarios, several notions of *teaching* have been developed.

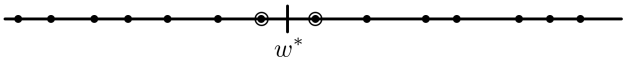
One influential model, introduced independently by Goldman & Kearns (1995), Shinohara & Miyano (1991), and Anthony et al. (1992), is based on the notion of a *teaching set*. To define this, let $\mathcal{X}$ be any finite instance space and $\mathcal{H}$ any finite set of concepts on $\mathcal{X}$, so that each $h \in \mathcal{H}$ is of the form $h : \mathcal{X} \to \{0, 1\}$. Let $h^* \in \mathcal{H}$ denote a target concept. We say $S \subset \mathcal{X}$ is a *teaching set* for $(h^*, \mathcal{H})$ if h^* is the only concept in $\mathcal{H}$ that is consistent with the labeled examples $\{(x, h^*(x)) : x \in S\}$. An *optimal teacher* is then one who provides the learner with the smallest possible teaching set.

This notion of teaching is related to *compression*. A simple learner—human or machine—might have difficulty remembering a large set of examples and finding a hypothesis consistent with them. The teacher helps by identifying a concise set of examples that conveys the desired concept.

Consider, for instance, the case of thresholds on the line. Here data points are real numbers, so $\mathcal{X} \subset \mathbb{R}$, and the hypothesis class is $\mathcal{H} = \{h_w : w \in \mathbb{R}\}$, where

$$h_w(x) = \begin{cases} 1 & \text{if } x \geq w \\ 0 & \text{otherwise} \end{cases}$$

For finite $\mathcal{X}$, we need only consider $|\mathcal{X}| + 1$ distinct hypotheses. If the target concept is given by threshold w^*, the optimal teaching set consists of the two points in $\mathcal{X}$ nearest w^*, on either side of it:



In this case, no matter how large $\mathcal{X}$ might be, two teaching examples suffice. Thus the teacher makes learning easier.

This example also illustrates a significant issue with this notion of teaching: it requires the teacher to know $\mathcal{H}$, the learner's hypothesis class. This can be unrealistic in many scenarios. When teaching a human, one generally has no idea what the underlying hypotheses might be. And when teaching a machine, the general type of concept might be known (a neural net, for instance), but the specifics (number of layers, number of nodes per layers, other parameter settings) may be opaque; and even if they were known, it is unclear how they would be used in choosing a teaching set.

[1]University of California, San Diego [2]Columbia University [3]NTENT [4]University of Wisconsin–Madison. Correspondence to: Sanjoy Dasgupta <dasgupta@eng.ucsd.edu>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019.

Under the strong assumption that $\mathcal{H}$ is known, teaching does not need to be *adaptive*: the teacher merely serves up the relevant examples beforehand, and does not need to see what the learner does with them. We will call this *oblivious* teaching. This paper studies two basic questions.

- How well can oblivious teachers perform in situations where they are not aware of the learner's concept class?
- Can interaction help in such cases?

1.1. Contributions

We begin with a negative result for *oblivious* teaching.

We show that an oblivious teacher who does not know the learner's concept class must, in general, provide labels on all of $\mathcal{X}$. This holds even if the teacher knows that the unknown class $\mathcal{H}$ has low VC dimension and small teaching set size.

We then provide two positive results for *interactive* teaching.

We consider an interactive protocol in which the teacher provides one teaching example at a time, and in the interim is allowed to probe the predictions of the learner's current model, rather like giving the learner a quiz. We show that without knowing $\mathcal{H}$, such a teacher can efficiently pick a teaching set of size at most $O(t \cdot \log|\mathcal{X}| \cdot \log|\mathcal{H}|)$, where t is the optimal teaching set size for $\mathcal{H}$.

We also look at a setting in which the teacher not only probes the learner's prediction, but also the learner's uncertainty levels. We show bounds on the number of teaching examples needed that depend only on the VC dimension of $\mathcal{H}$ and its *disagreement coefficient* (Hanneke, 2011).

One interesting use of our teaching algorithm is in shrinking a training set T: finding a subset $S \subset T$ that yields the same final classifier. This can be useful in situations where the computational complexity of training scales poorly (e.g. quadratically) with the number of training instances. Our method can be used with any black-box learner. It constructs S incrementally by adding a few examples at a time, assessing the resulting classifier, and then deciding whether more examples are needed. In this way, it never needs to train on more than $|S|$ instances. We illustrate its behavior in experiments with kernel machines and neural nets.

1.2. From finite to infinite

The notion of teaching set is defined for *finite* instance spaces and concept classes. To see what this means for more general spaces, suppose we have an arbitrary instance space $\mathcal{X}^o$ and that there is some distribution P on this space. Suppose also that the possibly-infinite concept class $\mathcal{H}^o$ has VC dimension d. Then, we can draw $m = O(d/\epsilon)$ examples at random from P and treat these as a finite instance space $\mathcal{X}$; by standard generalization bounds, with high probability over the choice of examples, any $h \in \mathcal{H}$ that agrees with the target concept h^* on all of $\mathcal{X}$ will have error $\leq \epsilon$ on distribution P. Moreover, since $|\mathcal{X}| = m$, we know by VC bounds (Sauer, 1972) that the effective size of the hypothesis class is at most $N = O(m^d)$, and we can set $\mathcal{H}$ to this finite set of candidate concepts.

These equivalences, $|\mathcal{X}| = O(d/\epsilon)$ and $\mathcal{H} = O(|\mathcal{X}|^d)$, are useful in interpreting the bounds we obtain. Our lower bound then says that an oblivious teacher will need to use a teaching set of size $\Omega(d/\epsilon)$, which could be very large for small ϵ. On the other hand, an interactive teacher can find a teaching set of size $O(td\log^2(d/\epsilon))$, which has only a logarithmic dependence on $1/\epsilon$.

1.3. Related work: models of teaching

The literature on teaching can be organized along two main dimensions: whether the learner is required to be consistent with all teaching examples and whether the teacher has full knowledge of the learner (Zhu et al., 2018).

Earlier theoretical work on teaching assumes both, such as the classic teaching dimension (Goldman & Kearns, 1995; Shinohara & Miyano, 1991), the recursive teaching dimension (Zilles et al., 2011; Hu et al., 2017) and the preference-based teaching dimension (Gao et al., 2017). Recently, there has been growing interest in settings where both dimensions are negative: for instance, the learner is a convex empirical risk minimizer (e.g., Liu & Zhu, 2016), or the teacher does not target a specific learner (Zhu et al., 2017), or the teacher does not know the learner's hyper-parameters or hypothesis space (Liu et al., 2017). Of particular relevance is recent work by Liu et al. (2018), which assumes the teacher and the learner use different linear feature spaces. The teacher cannot fully observe the learner's linear model but knows the learner's algorithm and can employ active querying to learn the mapping between feature spaces.

In contrast, the present work assumes the learner is consistent with teaching examples but does not require knowledge of its concept class or learning algorithm. This setting is closer to that of classical learning theory and offers a crisp characterization of teaching black-box learners.

1.4. Related work: sample compression

The notion of *sample compression* was introduced by Littlestone & Warmuth (1986) and has been the subject of much further work (e.g., Floyd & Warmuth, 1995; Moran & Yehudayoff, 2016). It is centered on an intriguing question: for a given concept class $\mathcal{H}$, is it possible to design (1) a learning algorithm $\mathcal{A}$ that operates on labeled samples of some fixed size k, and (2) a procedure that, given any labeled data set, chooses a subset of size k such that when $\mathcal{A}$ is applied to this subset, it produces a classifier consistent with the full

data set? A recent result of Moran & Yehudayoff (2016) showed that if $\mathcal{H}$ has VC dimension d, then $k = d2^d$ is always achievable. The results in our paper can be thought of as a form of *adaptive sample compression*, where the concept class $\mathcal{H}$ is unknown and the learning algorithm $\mathcal{A}$ is fixed in advance and also unknown.

2. Lower bounds for oblivious teaching

Suppose the teacher knows only that the learner's concept class is one of $\mathcal{H}_1, \ldots, \mathcal{H}_k$, but not which one. They all contain the target h^*, and suppose they each have teaching sets of size t. What can an *oblivious* teacher do?

One option is to provide a teaching set for the union of the concept classes, $\mathcal{H}_1 \cup \cdots \cup \mathcal{H}_k$. This could have size up to tk. We'll see that, in general, an oblivious teacher can do no better than this. Thus, if k is large, the teacher simply has to provide labels for all of $\mathcal{X}$.

We will demonstrate this lower bound through two examples that illustrate different problems with oblivious teaching.

2.1. Example 1: Decision stumps

It might seem that, even without knowing $\mathcal{H}$ exactly, the teacher would still be able to select examples near the true decision boundary. We'll see that this is not the case. The different hypothesis classes could effectively deal with very different representations of the data, so that the identities of the "boundary examples" change dramatically from one hypothesis class to the next.

The instance space will be a finite set $\mathcal{X} \subset \mathbb{R}^k$, to be specified shortly. The concept classes $\mathcal{H}_1, \ldots, \mathcal{H}_k$ consist of thresholds along individual coordinates: $\mathcal{H}_i$ consists of all functions $h_{i,w} : \mathcal{X} \to \{0, 1\}$ of the form

$$h_{i,w}(x) = \begin{cases} 1 & \text{if } x_i > w; \\ 0 & \text{otherwise.} \end{cases}$$

where $w \in \mathbb{R}$. That is, the hypotheses in $\mathcal{H}_i$ only use the ith coordinate of the data. Each $\mathcal{H}_i$ has VC dimension 1 and has a teaching set of size 2.

We select $\mathcal{X}$ so that every point in it has either *all positive coordinates* or *all negative coordinates*. The target concept h^* is 1 if the coordinates are all positive and 0 if all negative. Thus h^* lies in every $\mathcal{H}_i$: in particular, $h^* = h_{i,0}$ for all i.

Set $\mathcal{X} = \{x^{(1)}, x^{(2)}, \ldots, x^{(k)}, -x^{(1)}, \ldots, -x^{(k)}\}$, where the $x^{(i)} \in \mathbb{R}^k_+$ are defined as follows:

- The values of the k features of $x^{(i)}$ are $2, 3, 4, \ldots, k$, in that order, with a 1 inserted in the ith position.
- Thus $x^{(1)} = (1, 2, 3, \ldots, k)$, $x^{(2)} = (2, 1, 3, \ldots, k)$, $x^{(3)} = (2, 3, 1, \ldots, k)$, and $x^{(k)} = (2, 3, 4, \ldots, k, 1)$.

Along any coordinate i, the correct threshold is 0, and the minimal teaching set consists of the two examples closest to 0, on either side of it. These are $-x^{(i)}, x^{(i)}$, whose ith coordinates have values $-1, 1$ respectively. In other words: for $\mathcal{H}_i$, the optimal teaching set consists of $-x^{(i)}$ and $x^{(i)}$.

However, the only teaching set that works for every $\mathcal{H}_i$ simultaneously is all of $\mathcal{X}$.

Theorem 1 *In the construction above, the concept classes $\mathcal{H}_1, \ldots, \mathcal{H}_k$ each have VC dimension 1 and teaching set size 2. If an oblivious teacher does not know which of these concept classes is being used by the learner, the smallest possible teaching set it can provide is all of $\mathcal{X}$, of size $2k$.*

PROOF: Consider any teaching set that leaves out some point in $\mathcal{X}$, say $x^{(i)}$. Then, if the learner happens to have concept class $\mathcal{H}_i$, it can consistently set the threshold to be 1.5 along the ith coordinate, since the $k - 1$ positive instances it has seen all have ith coordinate ≥ 2. Thus it will get $x^{(i)}$ wrong. □

Thus in this situation, the minimal teaching set has size tk, where t is the teaching set size of each individual $\mathcal{H}_i$.

2.2. Example 2: All-but-one rules

In the previous example, the different concept classes $\mathcal{H}_i$ dealt with different representations of the input, and the resulting differences in boundary regions created problems for the teacher. But even if the different hypothesis classes work with the same representation of the input, they might focus on different regions of the input space, in the sense of allowing a more detailed boundary in those regions. In such cases, a teaching set would need to provide the necessary level of detail in *all* of these regions.

To see a situation of this type, let $\mathcal{X}$ be a finite instance space. Define the target hypothesis h^* to be identically zero. For any $x' \in \mathcal{X}$, define hypothesis $h_{x'} : \mathcal{X} \to \{0, 1\}$ to be zero everywhere except x', that is,

$$h_{x'}(x) = \begin{cases} 1 & \text{if } x = x'; \\ 0 & \text{otherwise.} \end{cases}$$

For any subset $S \subseteq \mathcal{X}$, define hypothesis class $\mathcal{H}(S) = \{h^*\} \cup \{h_x : x \in S\}$. Now, partition $\mathcal{X}$ into k subsets $S_1, \ldots, S_k$ of size $|\mathcal{X}|/k$ and define $\mathcal{H}_j = \mathcal{H}(S_j)$.

Theorem 2 *In the construction above, each $\mathcal{H}_j$ has VC dimension 1 and teaching set size $t = |\mathcal{X}|/k$. If an oblivious teacher does not know which of these concept classes is being used by the learner, the smallest teaching set it can provide is all of $\mathcal{X}$, of size tk.*

PROOF: Each hypothesis class $\mathcal{H}_j = \mathcal{H}(S_j)$ has a teaching set of size $t = |S_j|$, consisting of $\{(x, 0) : x \in S_j\}$. There

is no smaller teaching set: if $x \in S_j$ is left out of the set, then the concepts h^* and h_x will both be consistent.

Likewise, if the teacher does not know which of $\mathcal{H}_1, \ldots, \mathcal{H}_k$ is being used by the learner, its minimal teaching set must consist of *all* points in $\mathcal{X}$. Suppose, on the contrary, that it leaves out a particular x, and that $x \in S_j$. Then a learner using $\mathcal{H}_j$ could choose h_x as a consistent hypothesis. □

3. Teaching by probing the learner's predictions

We now consider scenarios in which the teacher has *no knowledge* of the learner's concept class other than an upper bound on its size or VC dimension. It does not, for instance, have a shortlist of possibilities $\mathcal{H}_1, \ldots, \mathcal{H}_k$, as in the previous section. Nevertheless, if the teacher is allowed to interact with the learner, it can come up with a teaching set that is provably within a factor $\log|\mathcal{X}| \cdot \log|\mathcal{H}|$ of optimal.

Here is a model in which the teacher and learner communicate in rounds of interaction.

On each round,

- The teacher supplies one or more teaching examples $(x, y) \in \mathcal{X} \times \{0, 1\}$ to the learner.
- The learner gives the teacher a black-box classifier $h : \mathcal{X} \to \{0, 1\}$ that is consistent with all the teaching examples it has seen so far.

The idea here is that the teacher cannot look inside the black box classifier, but can test it on examples to get a sense of where its mistakes lie.

If the learner is a machine, this is a natural setup. If the learner is a human, probing the black box corresponds to giving the learner a quiz. In either case, we distinguish *teaching examples* from *probes* to the black box. It is of primary interest to keep the former small: they constitute a summary of what is important, and the learner is required to be consistent with them. But we also want the number of probes to be polynomially bounded.

We will say that an interactive teaching strategy of the type above is a (t, p)-*protocol* if it ultimately yields a teaching set of size t after making at most p probes.

3.1. A generic interactive teaching procedure

Suppose the learner's hypothesis class $\mathcal{H}$ has optimal teaching set size t. The teacher has no information about $\mathcal{H}$, except that it contains h^*. We will see that there is an efficient interactive protocol of the type above in which the teacher needs to provide at most $O(t \cdot \log|\mathcal{X}| \cdot \log|\mathcal{H}|)$ teaching examples before the learner converges to h^*.

The key idea is as follows. Teaching is essentially a *set cover* problem: each teaching example eliminates some sub-optimal hypotheses in $\mathcal{H}$, and a teaching set is a collection of examples that eliminate, or "cover", *all* sub-optimal hypotheses. By this view, optimal teaching is equivalent to minimum set cover. However, in our setting, the set to be covered—the set of sub-optimal hypotheses—is unknown, since $\mathcal{H}$ is unknown, and this would seem to be a major problem. But there is an alternative *online* formulation of the set cover problem, in which the elements to be covered are not provided beforehand but appear one at a time, and must be covered immediately. An elegant algorithm is known for online set cover (Alon et al., 2009), and we will see that it can be simulated in our setting.

The resulting learning algorithm is shown in Figure 1. It is a randomized procedure that begins by drawing values T_x, one for each $x \in \mathcal{X}$, from a suitable exponential distribution. Then the interaction loop begins. A key quantity computed by the algorithm, for any learner-supplied black-box classifier h, is the set of misclassified points,

$$\Delta(h) = \{x \in \mathcal{X} : h(x) \neq h^*(x)\}.$$

Roughly speaking, the points x that are most likely to be chosen as teaching examples are those that have been misclassified multiple times by the learner's models, and for which T_x happens to be small.

Theorem 3 *Let t be the size of an optimal teaching set for $\mathcal{H}$. Pick any $0 < \delta < 1$. With probability at least $1 - \delta$, the algorithm of Figure 1 halts after at most $t \log(2|\mathcal{X}|)$ iterations. The number of teaching examples it provides is in expectation at most*

$$(1 + t\lg(2|\mathcal{X}|)) \cdot \left(\ln|\mathcal{H}| + \ln\frac{1}{\delta}\right).$$

The algorithm of Figure 1 is efficient and yields a teaching set of size $O(t \cdot \log|\mathcal{X}| \cdot \log|\mathcal{H}|)$, despite having no knowledge of the concept class $\mathcal{H}$. This can be significantly better than a teaching set of all $|\mathcal{X}|$ points, as we have seen would be needed by an oblivious teacher.

Along the way, the teacher makes $O(t \cdot |\mathcal{X}| \cdot \log|\mathcal{X}|)$ probes to intermediate classifiers provided by the learner. This is polynomial, but is nonetheless significant, and consequently this algorithm is better for teaching machines than humans. In fact, we will later see (Section 5) that probing all of $\mathcal{X}$ is inevitable when the teacher does not know $\mathcal{H}$.

1. Let $S = \emptyset$ (teaching set)
2. For each $x \in \mathcal{X}$:
   - Initialize weight $w(x) = 1/m$
   - Choose threshold T_x from an exponential distribution with rate $\lambda = \ln(N/\delta)$
3. Repeat until done:
   - Learner provides some $h : \mathcal{X} \to \{0, 1\}$ as a black box
   - By probing the black box, determine $\Delta(h) = \{x \in \mathcal{X} : h(x) \neq h^*(x)\}$
   - If $\Delta(h) = \emptyset$: halt and accept h
   - While $w(\Delta(h)) < 1$:
     - Double each $w(x)$, for $x \in \Delta(h)$
     - If this doubling causes $w(x)$ to exceed T_x for the first time, add x to S and provide $(x, h^*(x))$ as a teaching example to the learner

Figure 1. The teacher's algorithm. Here $m = |\mathcal{X}|$ and $N = |\mathcal{H}|$. For $S \subset \mathcal{X}$, we define $w(S) = \sum_{x \in S} w(x)$.

3.2. Proof of Theorem 3

The proof follows that of the original online set cover algorithm (Alon et al., 2009), with some additional subtleties. A proof sketch is provided here, with details in the appendix.

Lemma 4 *Let t be the size of an optimal teaching set for $\mathcal{H}$. Then the total number of doubling steps performed by the algorithm is at most $t \cdot \lg(2m)$, and at any point in time,*

$$\sum_{x \in \mathcal{X}} w(x) \leq 1 + t \cdot \lg(2m).$$

PROOF: First, $w(x) \leq 2$ for all x, always. This is because $w(x)$ increases only during a doubling step, which happens only if x belongs to a subset of $\mathcal{X}$ of total weight < 1.

Let $T^* \subset \mathcal{X}$ denote an optimal teaching set, of size t. By definition, T^* must intersect $\Delta(h)$ for all $h \neq h^*$. Now, a doubling step doubles the weight of each $x \in \Delta(h)$, and thus some element of T^*. And since the weight of an individual point begins at $1/m$ and never exceeds 2, the total number of doubling steps cannot exceed $t \cdot \lg(2m)$.

During each doubling step, $w(\Delta(h))$, and thus $\sum_x w(x)$, increases by at most 1. The lemma follows by noting that the initial value of this summation is 1, and there are at most $t \cdot \lg(2m)$ doubling steps. □

Lemma 5 *With probability at least $1 - \delta$, at the end of any iteration of the main loop, any hypothesis $h \neq h^*$ with $w(\Delta(h)) \geq 1$ is invalidated by the teaching examples.*

PROOF IDEA: Fix any $h \neq h^*$ and consider the first time at which $w(\Delta(h)) \geq 1$. Recall that the thresholds T_x are drawn from an exponential distribution with rate $\lambda = \ln(N/\delta)$. The probability, over random choice of thresholds, that no point in $\Delta(h)$ is chosen as a teaching example is

$$\begin{aligned}\prod_{x \in \Delta(h)} \Pr(w(x) \leq T_x) &= \prod_{x \in \Delta(h)} \exp(-\lambda w(x)) \\ &= \exp(-\lambda w(\Delta(h))) \\ &\leq \exp(-\lambda) = \frac{\delta}{N}.\end{aligned}$$

Now take a union bound over all N hypotheses in $\mathcal{H}$. □

Lemma 6 *The expected total number of teaching examples provided is at most $(1 + t \lg(2m)) \ln(N/\delta)$.*

PROOF IDEA: Pick any $x \in \mathcal{X}$; suppose that during a particular round of doubling its weight increases from w to w'. The probability it is chosen as a teaching example during that doubling is

$$\begin{aligned}\Pr(T_x \leq w' \mid T_x > w) &= 1 - \Pr(T_x > w' \mid T_x > w) \\ &= 1 - \exp(-\lambda(w' - w)) \\ &\leq \lambda(w' - w).\end{aligned}$$

Thus the expected number of teaching examples chosen during a round of doubling is at most λ times the increase in total weight during the doubling. The result then follows by applying Lemma 4. □

4. Teaching by probing the learner's uncertainty

Suppose that the learner communicates not just a classifier but also its *uncertainty*. We model this by allowing the

teacher two forms of communication with the learner:

- *Teaching example.* The teacher provides a labeled example $(x, h^*(x))$. The learner must subsequently adopt this as a hard constraint.
- *Uncertainty-rated probe.* The teacher makes a query x, and the learner answers according to its current version space:
  - 0 or 1 if everything in the version space agrees on this label
  - ? if there is disagreement within the version space

We will use the notation (t, p)-*protocol* if a teaching set of size t is produced after at most p uncertainty-rated probes.

To make the notion of disagreement precise, let V denote the current version space of the learner: that is, the set of hypotheses in $\mathcal{H}$ that are consistent with all teaching examples seen so far. We define $\text{DIS}(V) = \{x \in \mathcal{X} : \text{there exist } h, h' \in V \text{ for which } h(x) \neq h'(x)\}$. The uncertainty-rated probe then returns ? if $x \in \text{DIS}(V)$.

If the learner is a machine, it can easily return a black box that computes uncertainty probes. This is no harder than learning: if S is the set of labeled teaching examples so far, an uncertainty probe on $x \in \mathcal{X}$ can be performed efficiently, as follows:

- Let $S_0 = S \cup \{(x, 0)\}$. Check if there is a hypothesis in $\mathcal{H}$ that perfectly fits S_0.
- Let $S_1 = S \cup \{(x, 1)\}$. Check if there is a hypothesis in $\mathcal{H}$ that perfectly fits S_1.
- If both succeed: return ?. If only S_0 succeeds: return 0. If only S_1 succeeds: return 1.

If the learner is human, the uncertainty probe model is less reasonable, and it would make sense to consider a version with weaker requirements, for instance where the learner answers ? if there is a substantial amount of disagreement on the label, with respect to its (unknown) prior over concepts.

4.1. Teaching by simulating the Cohn-Atlas-Ladner active learner

Uncertainty-rated probes permit a teaching strategy that simulates the active learning algorithm of Cohn et al. (1994):

1. Teacher randomly permutes instance space $\mathcal{X}$
2. For each $x \in \mathcal{X}$, in this order:
   - Teacher probes the learner on x
   - If learner returns ?: teacher provides $(x, h^*(x))$ as a teaching example.

Using an analysis by Hanneke (2011), we show that this method constructs a teaching set of expected size $\theta \cdot (\log^2 |\mathcal{X}| + \log |\mathcal{X}| \cdot \log |\mathcal{H}|)$, where

$$\theta = \sup_{r>0} \frac{\left|\bigcup_{h \in \mathcal{H}: |\Delta(h)| \leq r} \Delta(h)\right|}{r}$$

is the *disagreement coefficient* of h^* within $\mathcal{H}$. (Recall $\Delta(h) = \{x \in \mathcal{X} : h(x) \neq h^*(x)\}$.)

Hanneke introduced the disagreement coefficient to bound the label complexity of active learning. To get some intuition for it in our context, consider k hypotheses $h_1, \ldots, h_k$ that all differ from h^* on the same number of points, and suppose the hypothesis class is just $\mathcal{H} = \{h_1, \ldots, h_k, h^*\}$. If $\Delta(h_i) \cap \Delta(h_j) = \emptyset$ for all $i \neq j$ (so $\theta = k$), then a teaching set must contain at least one point from each $\Delta(h_i)$. If, on the other hand, the $\Delta(h_i)$ overlap substantially (so $\theta \ll k$), then just a few random points in $\Delta(h_1) \cup \cdots \cup \Delta(h_k)$ are likely to constitute a teaching set.

In many cases, θ has a bound that is independent of the cardinality of $\mathcal{X}$; see Hanneke (2014) for several examples. For instance, for thresholds on the line, $\theta = 2$.

Theorem 7 *The teaching strategy that simulates the Cohn-Atlas-Ladner algorithm constructs a teaching set for $(h^*, \mathcal{H})$ with expected cardinality $O(\theta \cdot (\log^2 |\mathcal{X}| + \log |\mathcal{X}| \cdot \log |\mathcal{H}|))$.*

Thus, a $(\theta \cdot (\log^2 |\mathcal{X}| + \log |\mathcal{X}| \cdot \log |\mathcal{H}|), |\mathcal{X}|)$-protocol is always achievable. A more careful analysis of Hanneke (2011) replaces the $\log^2 |\mathcal{X}|$ with $\log |\mathcal{X}| \cdot \log \log |\mathcal{X}|$.

4.2. Proof of Theorem 7

Let $x_1, x_2, \ldots, x_m$ be the random ordering of $\mathcal{X}$ used by the teacher (with $m = |\mathcal{X}|$), and for any $1 \leq k \leq m$, let $\mathcal{H}_k = \{h \in \mathcal{H} : \Delta(h) \cap \{x_1, \ldots, x_k\} \neq \emptyset\}$ be the hypotheses in $\mathcal{H}$ that disagree with h^* on at least one of the first k points.

Lemma 8 *The set of teaching examples provided through time k is a teaching set for $(h^*, \mathcal{H}_k)$.*

PROOF: Take any $h \in \mathcal{H}_k$, and consider the first $x_i \in \Delta(h) \cap \{x_1, \ldots, x_k\}$. When the teacher probes the learner on x_i, the learner must return ?, as $h(x_i) \neq h^*(x_i)$. So $(x_i, h^*(x_i))$ is provided as a teaching example. □

Lemma 8 implies that the final set of teaching examples is indeed a teaching set for $(h^*, \mathcal{H})$. It remains to bound the (expected) number of teaching examples.

Define $r_{k,\delta} = (m/k) \ln(|\mathcal{H}|/\delta)$.

Lemma 9 *Pick any $\delta \in (0,1)$ and any $1 \leq k \leq m$. With probability at least $1-\delta$, every hypothesis $h \in \mathcal{H}$ that agrees with h^* on $x_1, \ldots, x_k$ has $|\Delta(h)| \leq r_{k,\delta}$.*

PROOF: Pick a $h \in \mathcal{H}$ with $|\Delta(h)| > r_{k,\delta}$. The probability that it agrees with $x_1, \ldots, x_k$ is at most $(1-|\Delta(h)|/m)^k \leq \exp(-|\Delta(h)|k/m) < \delta/|\mathcal{H}|$. Now apply a union bound over all such h. □

Lemma 10 *The expected number of teaching examples is at most $2 + \theta \cdot \ln(m) \cdot \ln(|\mathcal{H}|m)$.*

PROOF: Let E_k be the $1-\delta$ probability event of Lemma 9, and let Q_k be the event that the learner returns ? when probed on x_k. We'll show that $\Pr(Q_k)$ is at most the probability that: either E_{k-1} does not happen, or $x_k \in \mathcal{X}_{k,\delta} = \cup_{h \in \mathcal{H}: |\Delta(h)| \leq r_{k-1,\delta}} \Delta(h)$. Indeed, if Q_k happens, then by Lemma 8, there is some $h \in \mathcal{H}$ that agrees with h^* on $x_1, \ldots, x_{k-1}$, but $h(x_k) \neq h^*(x_k)$. If E_{k-1} holds, then any such hypothesis $h \in \mathcal{H}$ must have $|\Delta(h)| \leq r_{k-1,\delta}$. This line of reasoning gives the bound

$$\begin{aligned}\Pr(Q_k) &\leq \Pr(\neg E_{k-1} \vee x_k \in \mathcal{X}_{k,\delta}) \\ &\leq \Pr(\neg E_{k-1}) + \frac{|\mathcal{X}_{k,\delta}|}{m} \;\leq\; \delta + \theta \cdot \frac{r_{k-1,\delta}}{m}.\end{aligned}$$

Summing $\Pr(Q_k)$ from $k = 1, \ldots, m$ and choosing $\delta = 1/m$ proves the claim. □

5. A lower bound on the number of probes

In both our teaching procedures, all of $\mathcal{X}$ is probed. To see why this is necessary, recall the hypothesis classes $\mathcal{H}(S)$ of Section 2.2 and consider $|\mathcal{X}|$ classes of the form $\mathcal{H}(\{x\}) = \{h^*, h_x\}$, where x is a single element of $\mathcal{X}$. Each such class has a teaching set of size 1, consisting of the labeled example $(x, 0)$. If the teacher does not know which of these concept classes is used by the learner, then every x must either be probed or supplied as a teaching example; otherwise, the teacher cannot be sure that the learner is not using h_x.

6. Experimental illustration

In this section, we use Algorithm 1 to *shrink* several synthetic and real datasets, that is, to find subsets (teaching sets) of the data that yield the same final classifier. This can be useful for reducing storage/transmission costs of training data, or in situations where the computational complexity of training scales poorly with the number of samples.

Suppose the learning algorithm has running time $T(n)$, where n is the size of the training set. Algorithm 1 builds a teaching set incrementally, in iterations that involve adding a few points, invoking the learning algorithm, and evaluating the classifier that results. If the teaching set sizes along the way are $t_1 < t_2 < \cdots < t_k$, the total training time is $T(t_1) + \cdots + T(t_k)$, which can be much smaller than $T(n)$.

Synthetic data We looked at synthetic data in the form of *moons*, *circles*, and *mixtures*. For each, we generated two-dimensional *separable* and *non-separable* datasets of 4000 points each, by varying the level of noise. We then tested Algorithm 1 using SVM learners with linear, quadratic, and RBF kernels. For each simulation we report: (1) the support vectors (SVs) of each learner; (2) the teaching points (TPs), as decided by the algorithm; (3) the points that are both support vectors and teaching points (TPs AND SVs); and (4) teaching curves.

For a support vector machine, it is always possible to create a teaching set of size two by choosing the points so that their perpendicular bisector is the boundary; the maximum-margin objective function will then yield exactly the target classifier. However, any given data set is unlikely to contain such a pair of points. Thus in our examples, the size of the optimal teaching set is not known, although it is certainly upper-bounded by the number of support vectors.

Some of the results are shown in Figure 2. For instance, the top left-hand panel shows the result of the teaching algorithm on the moon-shaped data. There are 123 support vectors in the full data set, but a teaching set of just 19 points is found. As can be seen on the right, these points are picked in five batches: the first batch has two points and already brings the accuracy above 75%. Overall, the learning algorithm is called five times, on data sets of size 2, 10, 13, 17, 19; and we get the same effect as calling it on the entire set of 4000 points.

The full range of experiments on synthetic data can be seen in Figures 3 to 20 in the appendix.

Real datasets We also looked at the MNIST and fashion MNIST (Xiao et al., 2017) datasets, both with 60K points.

1. On MNIST, we used an SVM with a quadratic kernel. This data has 32,320 support vectors, and a teaching set of 4,445 points is found (almost all support vectors).
2. On fashion MNIST, we used a convolutional network with 4 different layers of 2d convolutions (32, 64, 128, 128) each followed by a ReLU and a max pooling layer.

The bottom panel of Figure 2 shows the teaching curves for these two data sets. In either case, the accuracy achieved on the full training set is below 100%.

For all experiments we used the same termination criterion: the algorithm terminated when it got within .01 of the accuracy of the learner that was trained using the full data. Also, to initialize the weight T_x of each data point we set the confidence parameter δ of Algorithm 1 to .1.

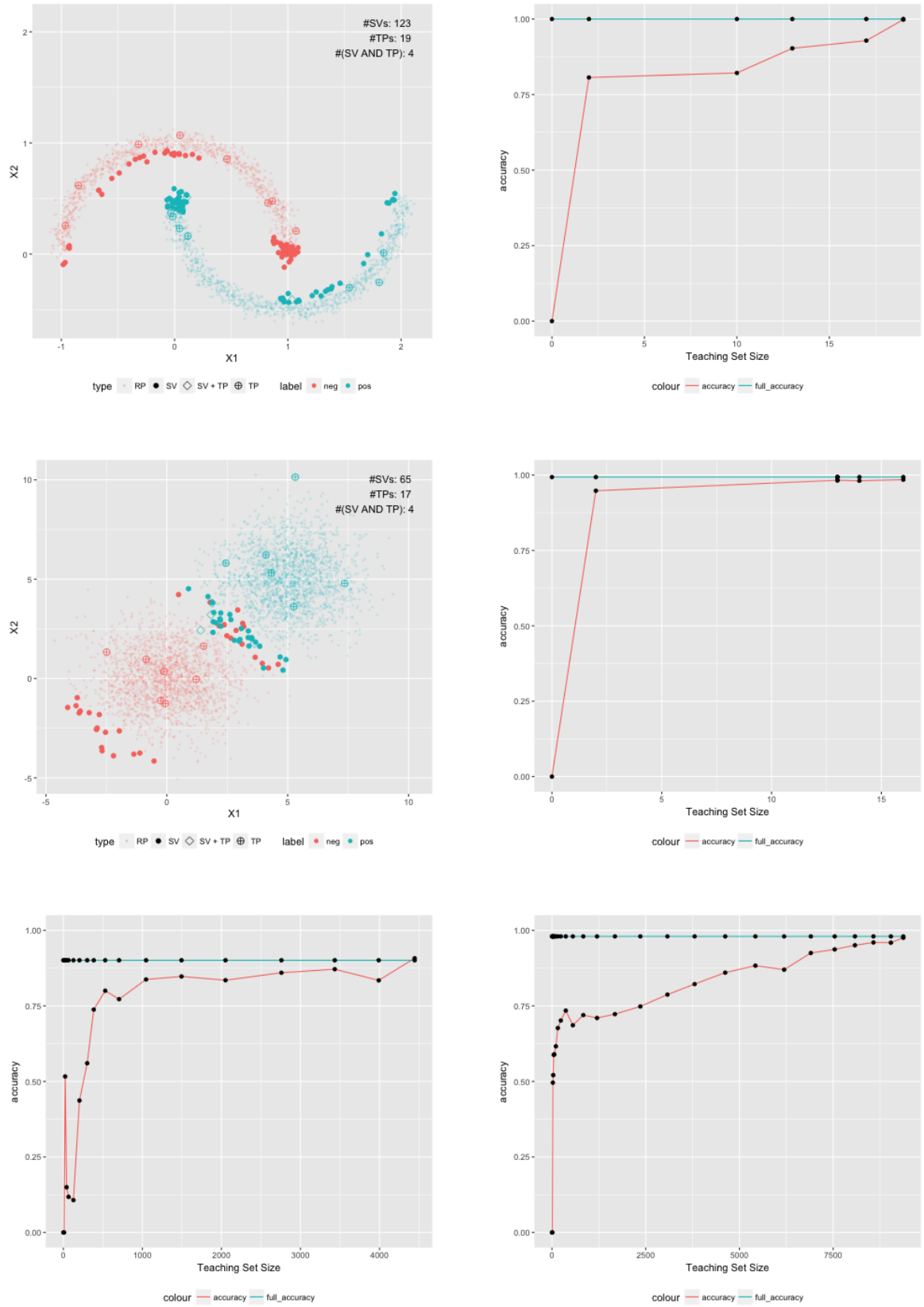


Figure 2. Top: 'Moon' data with RBF kernel SVM; Middle: 'Mixtures' data with quadratic kernel; Bottom: MNIST (quadratic SVM) and Fashion MNIST (convolutional neural net). Shown: support vectors (SV), teaching points (TP), regular points (RP).

Acknowledgements

This collaboration began during the "Foundations of Machine Learning" program at the Simons Institute for the Theory of Computing, Berkeley. Dasgupta is grateful to the NSF for support under grant CCF-1813160. Hsu was supported in part by NSF grant CCF-1740833. Zhu was supported in part by NSF 1545481, 1704117, 1836978. Poulis is grateful for support from NTENT. The authors thank Daniel Kane and Phil Long for helpful conversations and Eduardo Laber for pointing out the sloppiness in the original proof of Lemma 5.

References

Alon, N., Awerbuch, B., Azar, Y., Buchbinder, N., and Naor, S. The online set cover problem. *SIAM Journal on Computing*, 39(2):361–370, 2009.

Anthony, M., Brightwell, G., Cohen, D., and Shawe-Taylor, J. On exact specification by examples. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, pp. 311–318, 1992.

Cohn, D., Atlas, L., and Ladner, R. Improving generalization with active learning. *Machine Learning*, 15(2): 201–221, 1994.

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

Gao, Z., Ries, C., Simon, H. U., and Zilles, S. Preference-based teaching. *Journal of Machine Learning Research*, 18(31):1–32, 2017.

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

Hanneke, S. Rates of convergence in active learning. *Annals of Statistics*, 39(1):333–361, 2011.

Hanneke, S. Theory of disagreement-based active learning. *Foundations and Trends in Machine Learning*, 7(2-3):131–309, 2014. ISSN 1935-8237. doi: 10.1561/2200000037. URL http://dx.doi.org/10.1561/2200000037.

Hu, L., Wu, R., Li, T., and Wang, L. Quadratic upper bound for recursive teaching dimension of finite VC classes. In *Proceedings of the 30th Conference on Learning Theory, COLT*, pp. 1147–1156, 2017.

Littlestone, N. Learning quickly when irrelevant attributes abound: a new linear-threshold algorithm. *Machine Learning*, 2:285–318, 1988.

Littlestone, N. and Warmuth, M. Relating data compression and learnability. *Unpublished*, 1986.

Liu, J. and Zhu, X. The teaching dimension of linear learners. *Journal of Machine Learning Research*, 17(162):1–25, 2016. URL http://jmlr.org/papers/v17/15-630.html.

Liu, W., Dai, B., Humayun, A., Tay, C., Yu, C., Smith, L. B., Rehg, J. M., and Song, L. Iterative machine teaching. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017*, pp. 2149–2158, 2017.

Liu, W., Dai, B., Li, X., Liu, Z., Rehg, J., and Song, L. Towards black-box iterative machine teaching. In *International Conference on Machine Learning*, pp. 3147–3155, 2018.

Moran, S. and Yehudayoff, A. Sample compression schemes for VC classes. *Journal of the ACM*, 63(3), 2016.

Sauer, N. On the density of families of sets. *Journal of Combinatorial Theory, Series A*, 13:145–147, 1972.

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

Valiant, L. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

Xiao, H., Rasul, K., and Vollgraf, R. Fashion MNIST: a novel image dataset for benchmarking machine learning algorithms. *arXiv ePrints*, 2017.

Zhu, X., Liu, J., and Lopes, M. No learner left behind: On the complexity of teaching multiple learners simultaneously. In *The 26th International Joint Conference on Artificial Intelligence (IJCAI)*, 2017.

Zhu, X., Singla, A., Zilles, S., and Rafferty, A. An overview of machine teaching. *ArXiv e-prints*, January 2018. https://arxiv.org/abs/1801.05927.

Zilles, S., Lange, S., Holte, R., and Zinkevich, M. Models of cooperative teaching and learning. *J. Mach. Learn. Res.*, 12:349–384, 2011.

A. Proof details for Theorem 3

In what follows, let $W_i(x)$ denote the weight of point $x \in X$ at the end of round i of the main loop. If there are I rounds in all, then $W_o(x) = 1/m$ and the final weight of x is $W(x) = W_I(x)$. For any set $X_o \subset \mathcal{X}$, let $W_i(X_o) = \sum_{x \in X_o} W_i(x)$.

Each x has an associated exponentially-distributed threshold T_x. Given the memory-less property of the exponential, we can decide on its value gradually: Is it more than 0.05? If so, is it more than 0.1? And so on. In particular, it is only when the weight of x increases from $W_{i-1}(x)$ to $W_i(x)$ that we will ask, is $T_x > W_i(x)$? Accordingly, let $\mathcal{F}_i$ be the sigma-field of all indicator events $\{\mathbf{1}(T_x > W_{i'}(x)) : x \in X, 1 \le i' \le i\}$. This captures the information about the thresholds that has been revealed up to and including the end of round i.

Note that $W_i(x) \in \mathcal{F}_{i-1}$: by the end of round $i-1$, the weight of x at the end of round i is fully determined.

A.1. Proof of Lemma 5

Pick any $h \in \mathcal{H}$ and let $X_o = \Delta(h)$. Define Z_i to be 1 if no point in X_o is chosen during rounds $1, 2, \ldots, i$. Note that $Z_i \in \mathcal{F}_i$ and that $W_i(x) \in \mathcal{F}_{i-1}$ for $x \in X$. Thus

$$\begin{aligned}
\mathbb{E}[Z_i|\mathcal{F}_{i-1}] &= Z_{i-1} \cdot \Pr(\text{no point in } X_o \text{ chosen in round } i | Z_{i-1}, \mathcal{F}_{i-1}) \\
&= Z_{i-1} \cdot \prod_{x \in X_o} \Pr(T_x > W_i(x) | T_x > W_{i-1}(x)) \\
&= Z_{i-1} \cdot \prod_{x \in X_o} e^{-\lambda(W_i(x) - W_{i-1}(x))} \\
&= Z_{i-1} e^{-\lambda(W_i(X_o) - W_{i-1}(X_o))}.
\end{aligned}$$

This implies that $Y_i = e^{\lambda W_i(X_o)} Z_i$ is a martingale with respect to $(\mathcal{F}_i)$:

$$\mathbb{E}[e^{\lambda W_i(X_o)} Z_i | \mathcal{F}_{i-1}] = e^{\lambda W_i(X_o)} \mathbb{E}[Z_i | \mathcal{F}_{i-1}] = e^{\lambda W_{i-1}(X_o)} Z_{i-1}.$$

Now, for the very first round,

$$\mathbb{E}[Z_1|\mathcal{F}_o] = \prod_{x \in X_o} e^{-\lambda W_1(x)} = e^{-\lambda W_1(X_o)},$$

and thus $\mathbb{E}[Y_1] = 1$. Therefore, $\mathbb{E}[Y_i] = 1$ for all i.

Let M denote the first round i in which $W_i(X_o) \ge 1$. Since $\{M \le i\} \in \mathcal{F}_{i-1}$, it is a stopping time and we have $\mathbb{E}[Y_M] = 1$, so that

$$1 = \mathbb{E}[Y_M] = \mathbb{E}[e^{\lambda W_M(X_o)} Z_M] \ge \mathbb{E}[e^{\lambda} Z_M] = e^{\lambda} \Pr(Z_M = 1).$$

Thus $\Pr(Z_M = 1) \le e^{-\lambda} = \delta/N$. We finish by taking a union bound over all $h \in \mathcal{H}$.

A.2. Proof of Lemma 6

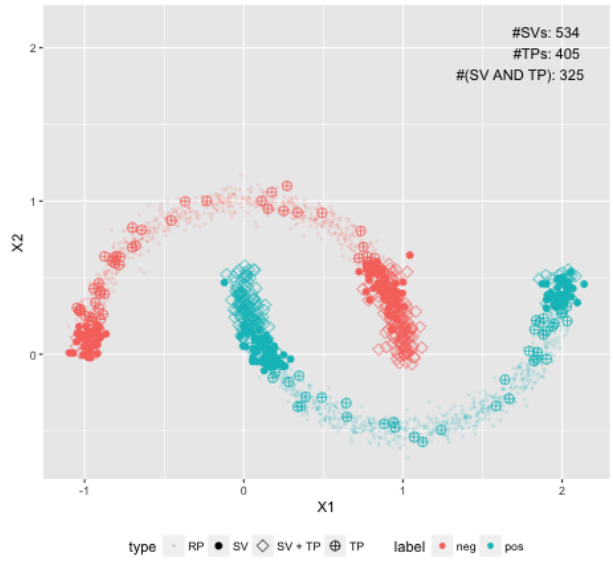
Let M_i denote the number of teaching examples selected in the ith round of doubling. Then

$$\begin{aligned}
\mathbb{E}[M_i|\mathcal{F}_{i-1}] &= \sum_{x \in \mathcal{X}} \Pr(x \text{ chosen in round } i | x \text{ not chosen before}, \mathcal{F}_{i-1}) \\
&\le \sum_{x \in \mathcal{X}} (1 - e^{-\lambda(W_i(x) - W_{i-1}(x))}) \\
&\le \sum_{x \in \mathcal{X}} \lambda(W_i(x) - W_{i-1}(x)) \;=\; \lambda(W_i(\mathcal{X}) - W_{i-1}(\mathcal{X})).
\end{aligned}$$

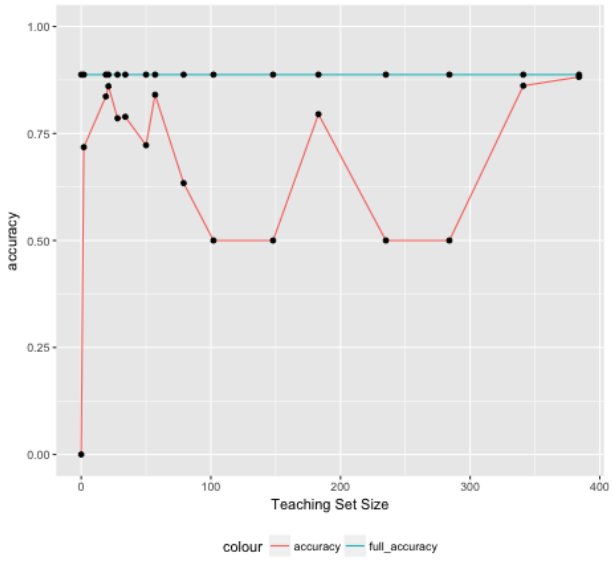
Since the total weight of $\mathcal{X}$ increases by at most 2 during any round, this is $\le 2\lambda$. The conclusion then follows from the bound on the number of rounds (Lemma 4). To shave off a factor of two, consider rounds of doubling, in which the total weight increases by at most 1, rather than rounds of the algorithm.

B. Experimental results

Below, we give the full set of experimental results on synthetic and real datasets.

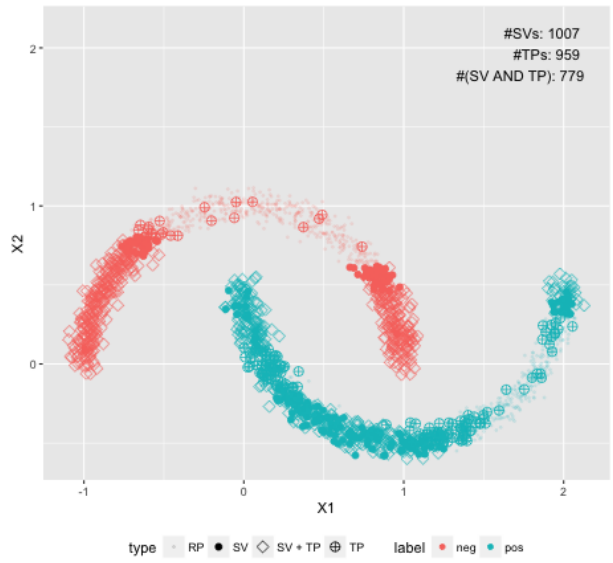


(a)

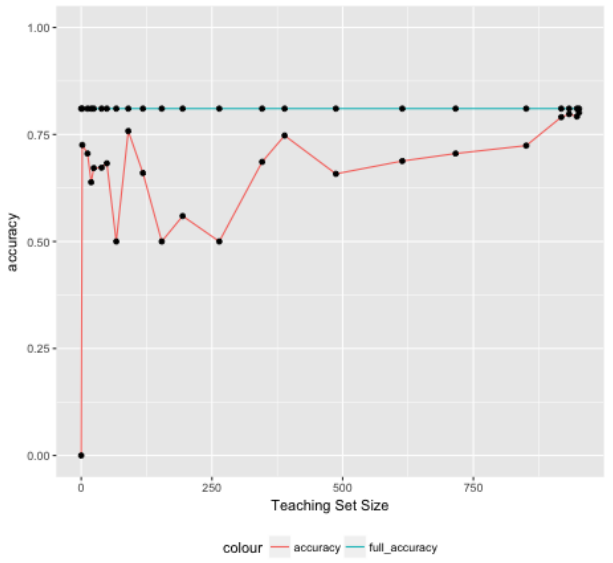


(b)

Figure 3. Moon-shaped dataset (separable), Linear kernel

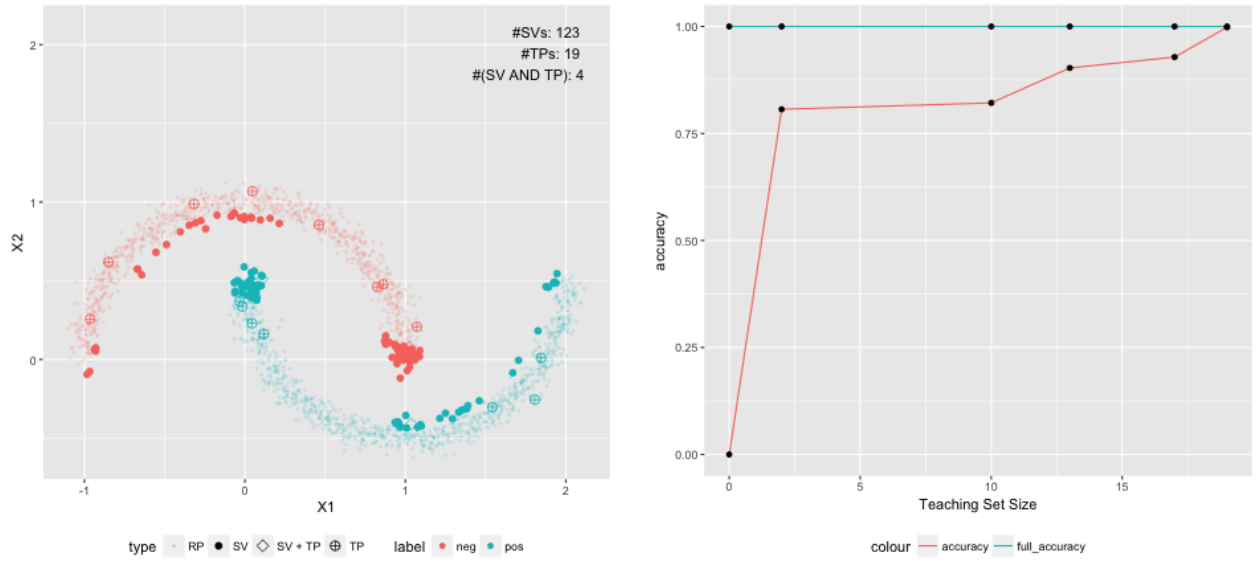


(a)



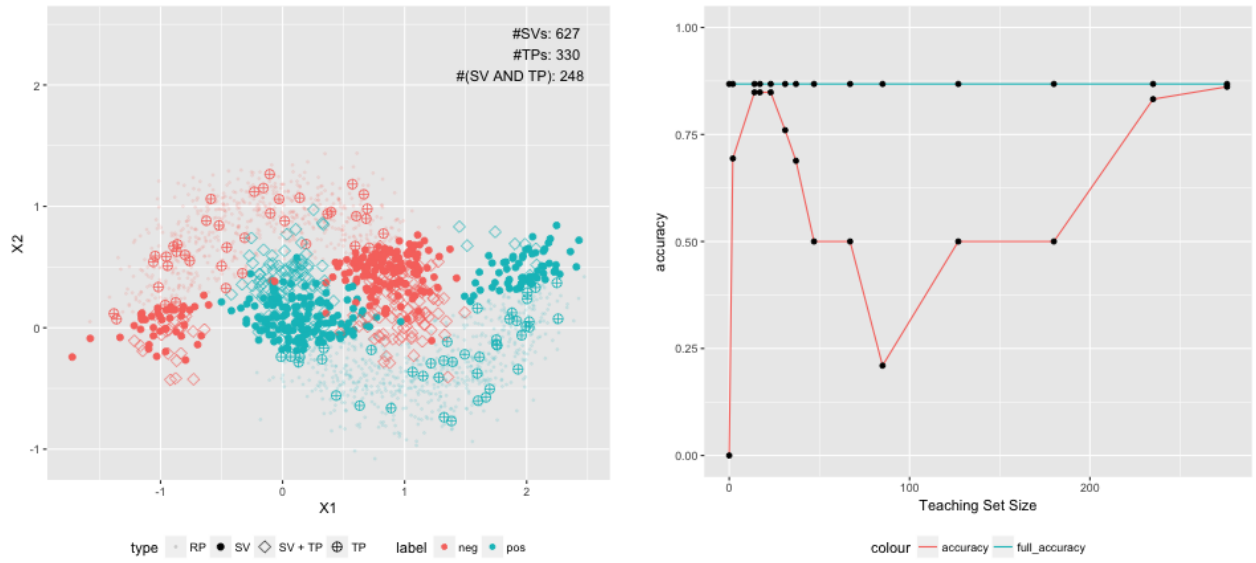
(b)

Figure 4. Moon-shaped dataset (separable), Quadratic kernel



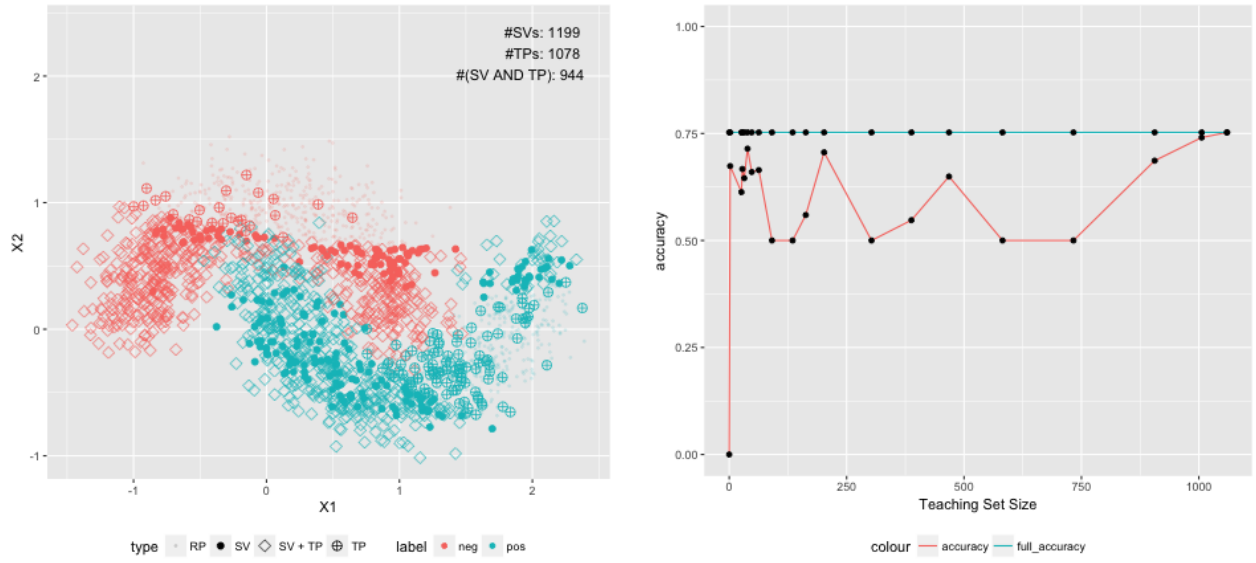
(a) (b)

Figure 5. Moon-shaped dataset (separable), RBF kernel



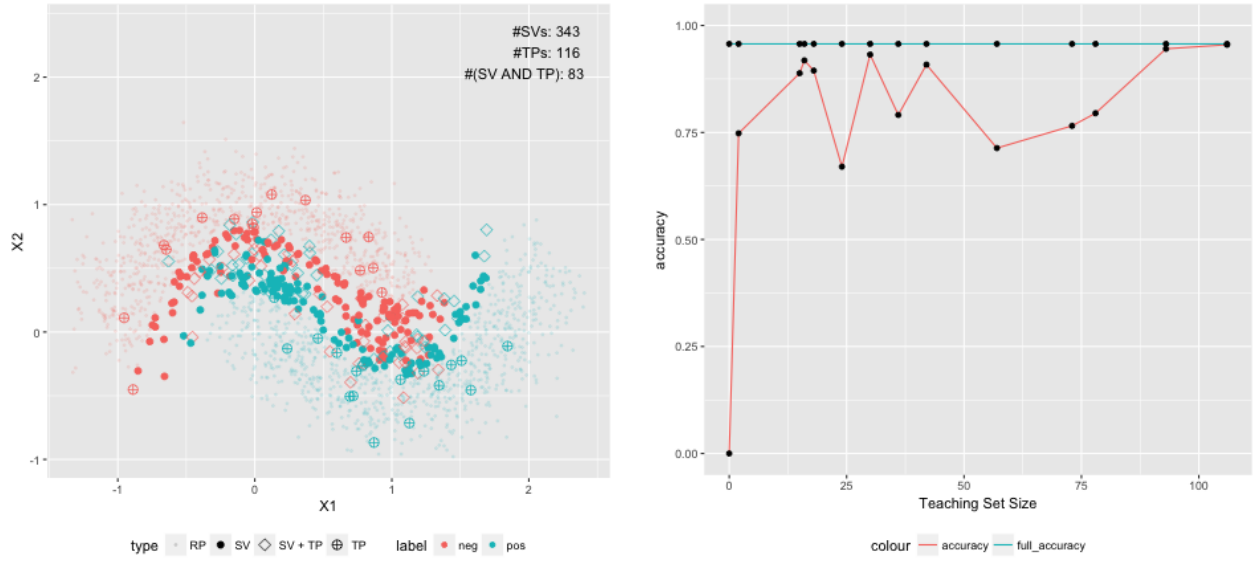
(a) (b)

Figure 6. Moon-shaped dataset (non-separable), Linear kernel



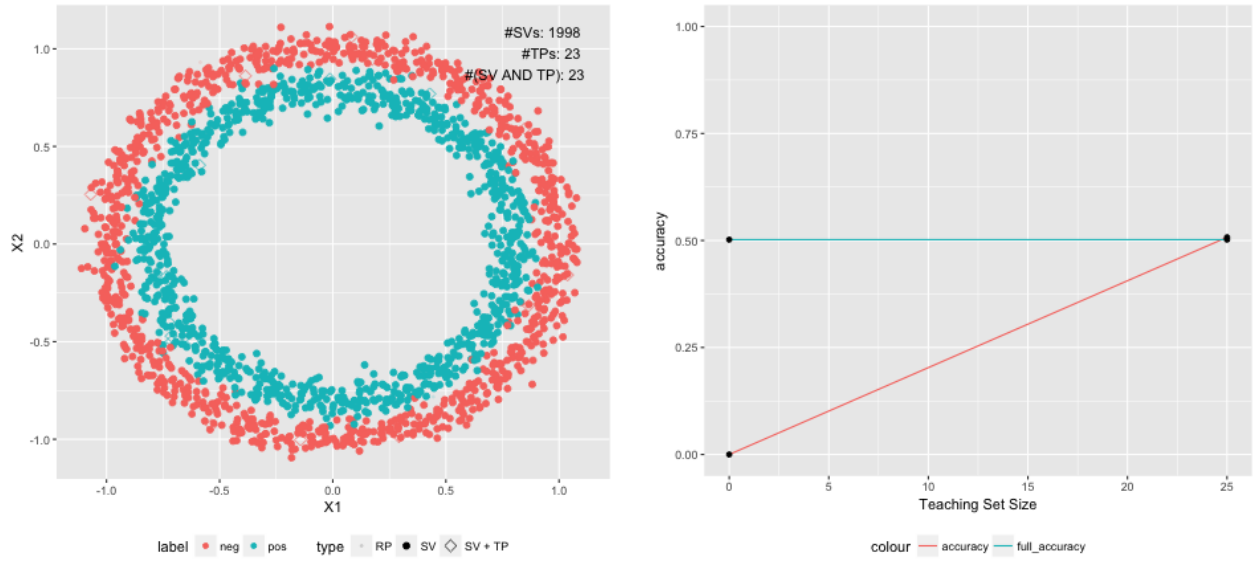
(a) (b)

Figure 7. Moon-shaped dataset (non-separable), Quadratic kernel



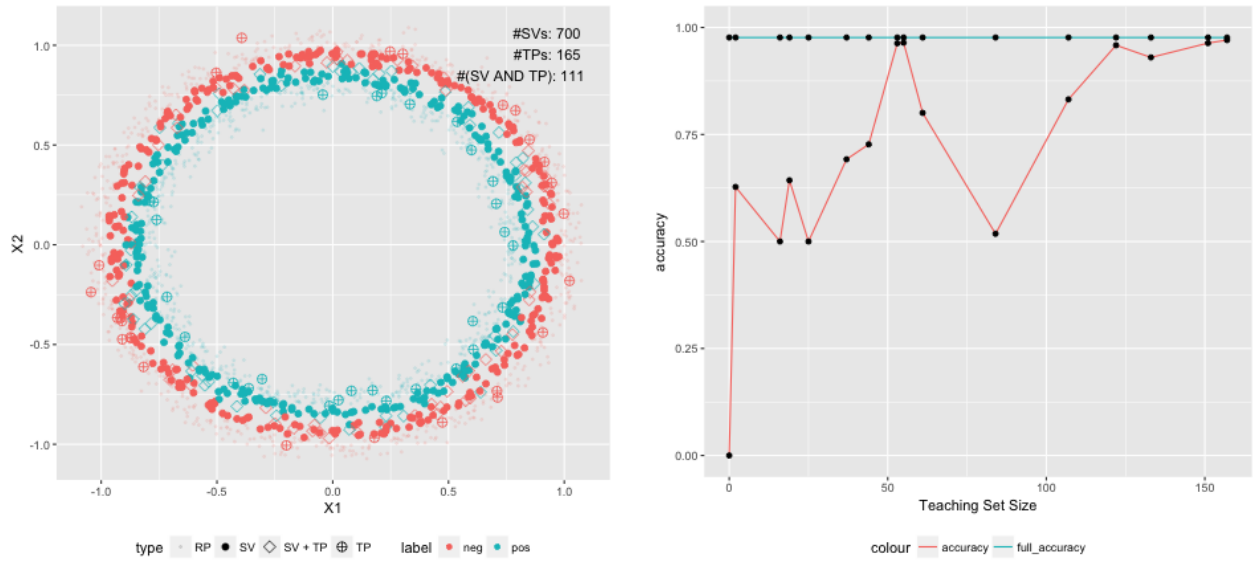
(a) (b)

Figure 8. Moon-shaped dataset (non-separable), RBF kernel



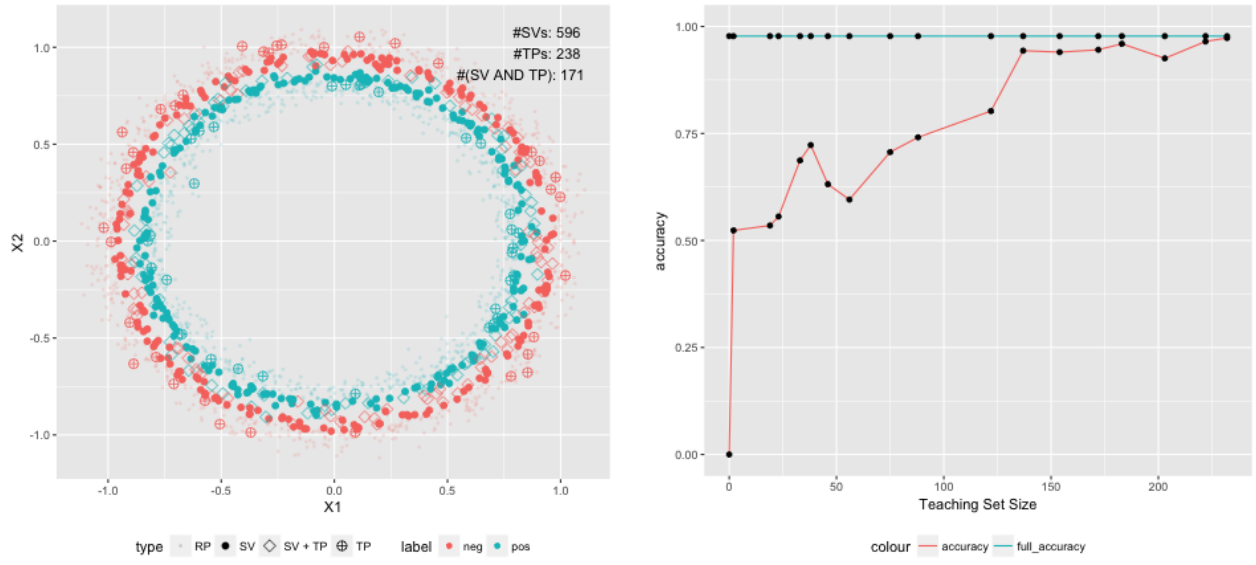
(a) (b)

Figure 9. Circular dataset (separable), Linear kernel



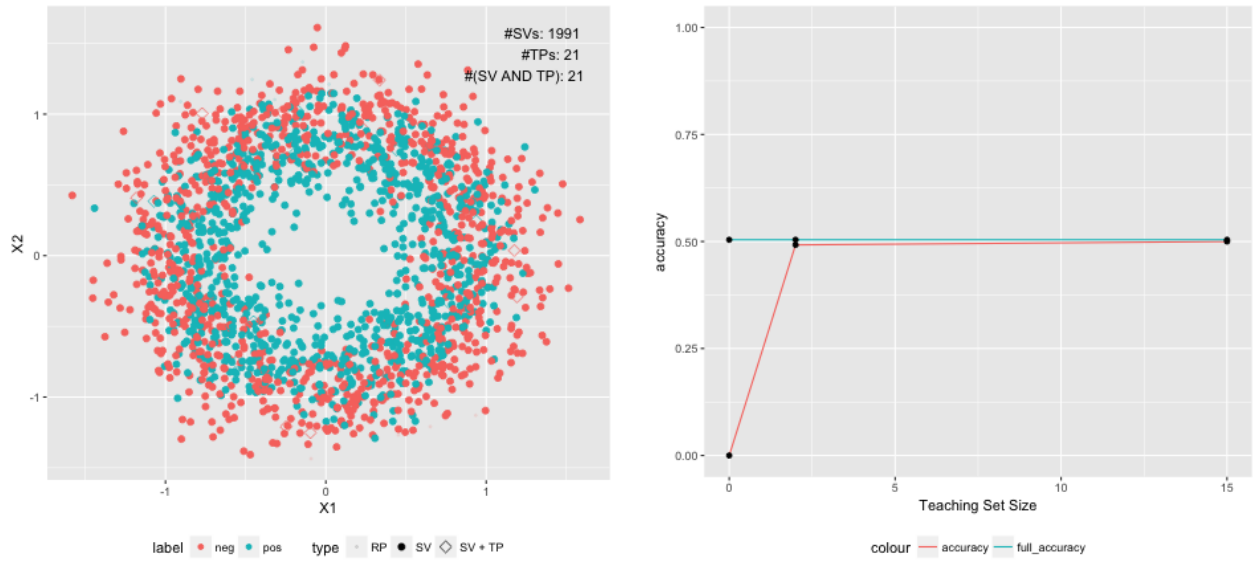
(a) (b)

Figure 10. Circular dataset (separable), Quadratic kernel



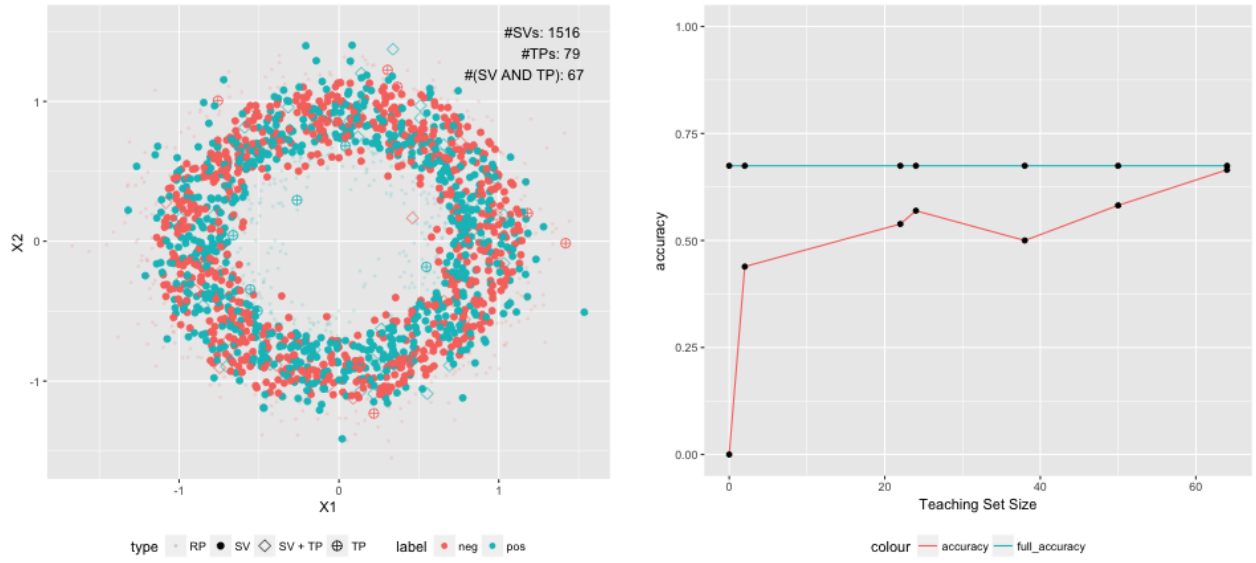
(a) (b)

Figure 11. Circular dataset (separable), RBF kernel



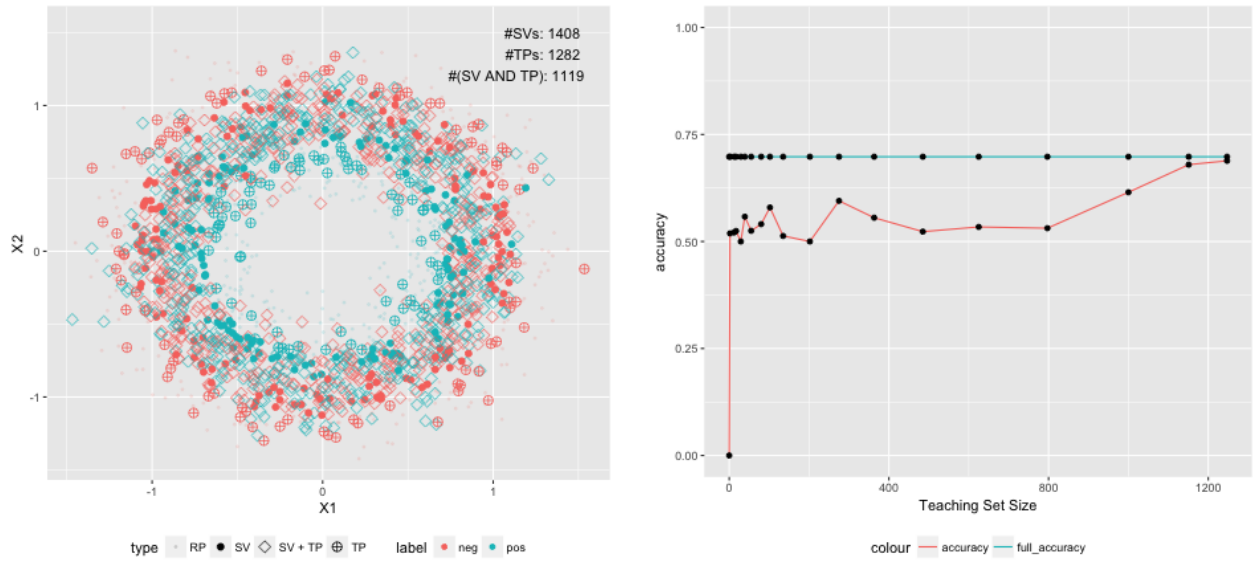
(a) (b)

Figure 12. Circular dataset (non-separable), Linear kernel



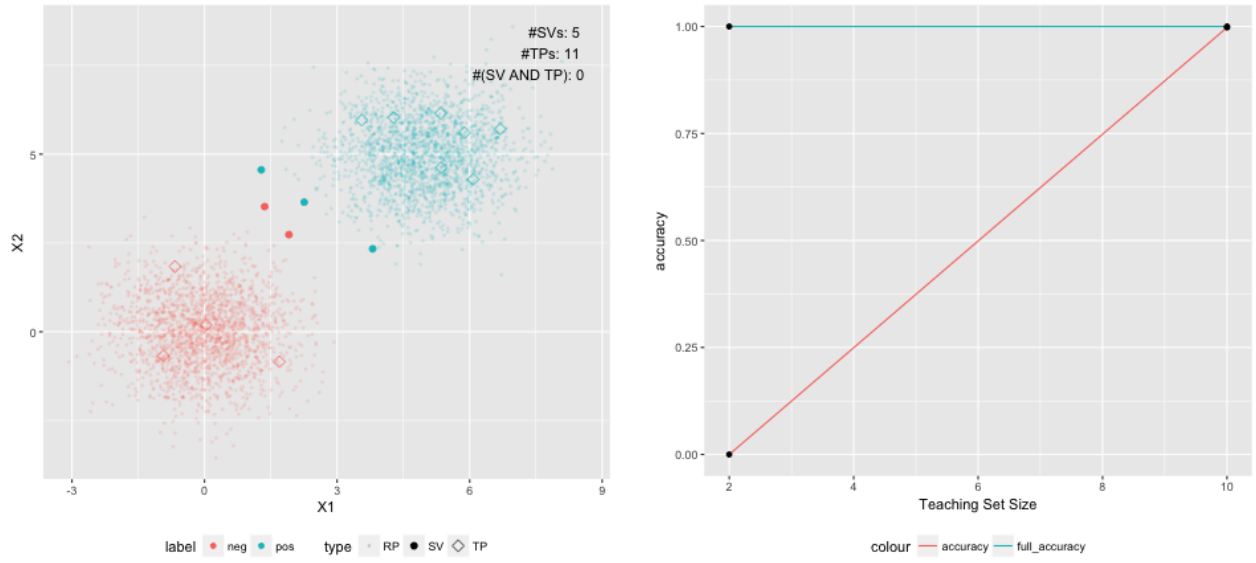
(a) (b)

Figure 13. Circular dataset (non-separable), Quadratic kernel



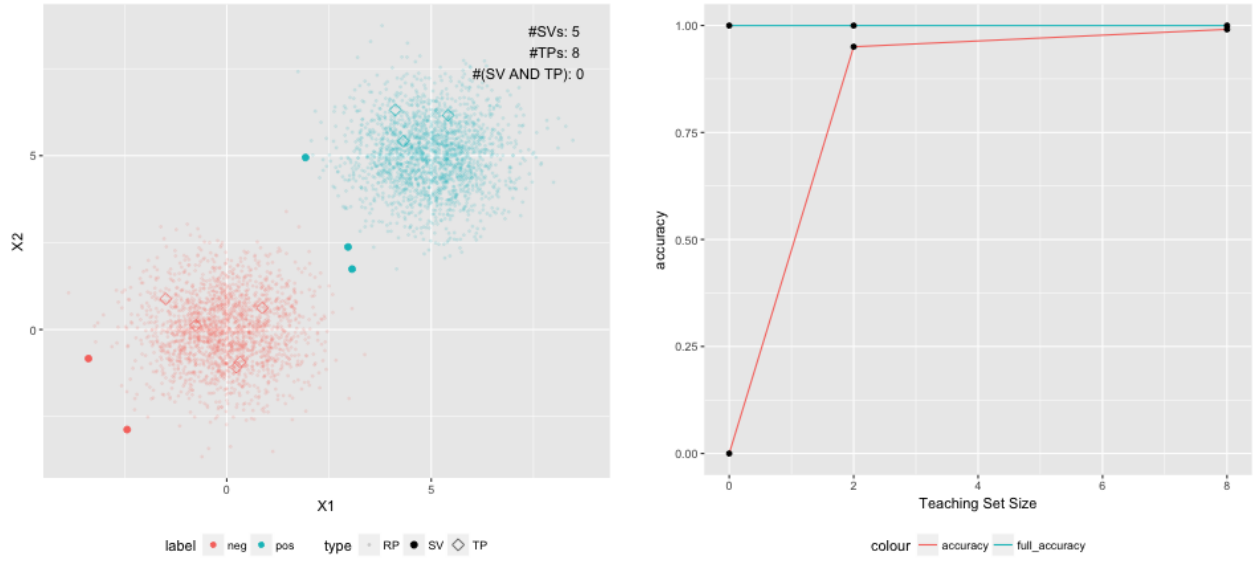
(a) (b)

Figure 14. Circular dataset (non-separable), RBF kernel



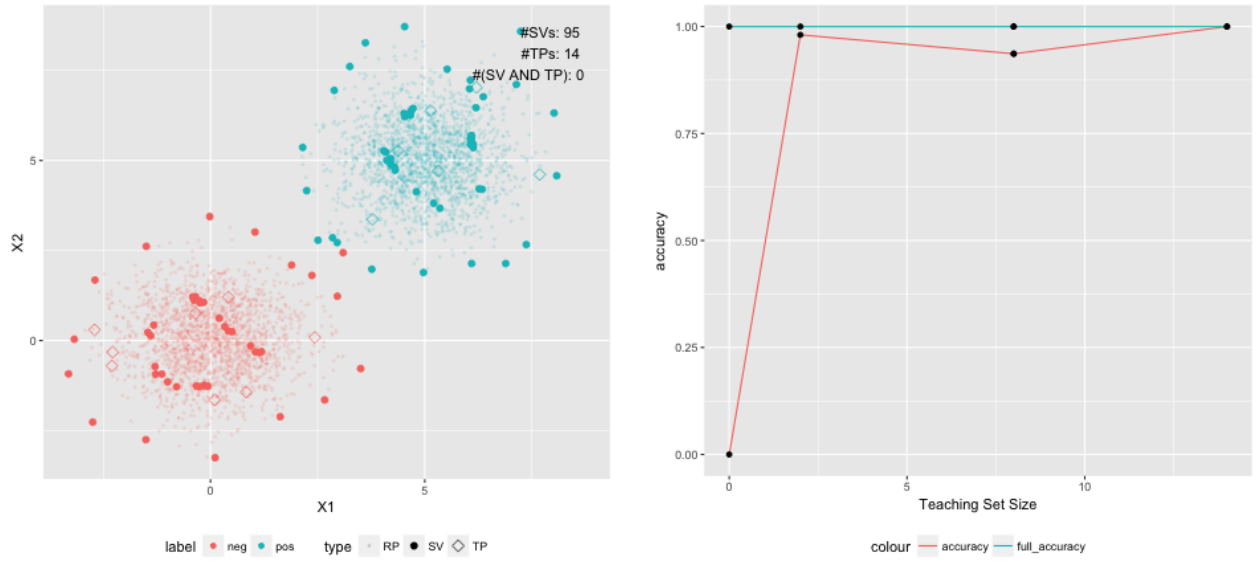
(a) (b)

Figure 15. Mixtures of Gaussians dataset (separable), Linear kernel



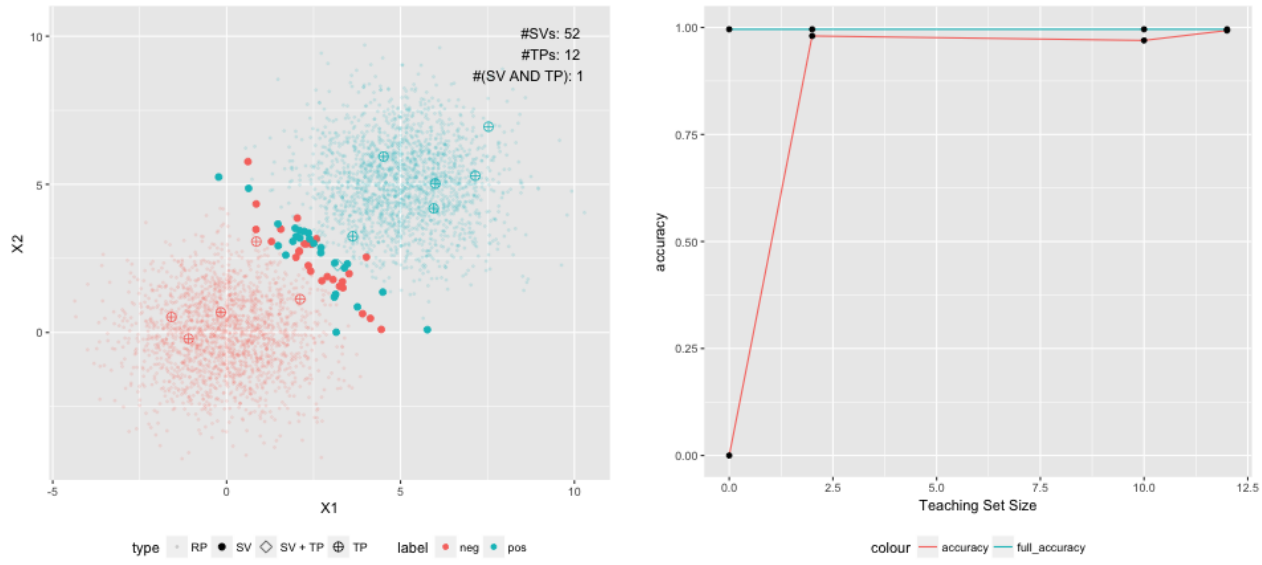
(a) (b)

Figure 16. Mixtures of Gaussians dataset (separable), Quadratic kernel



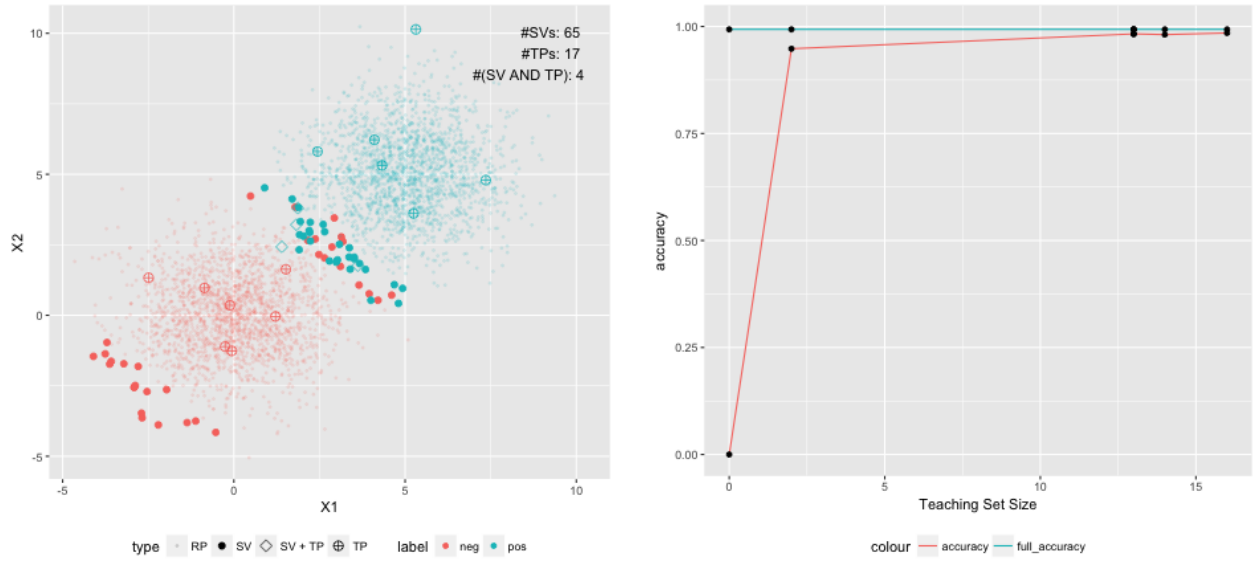
(a) (b)

Figure 17. Mixtures of Gaussians dataset (separable), RBF kernel



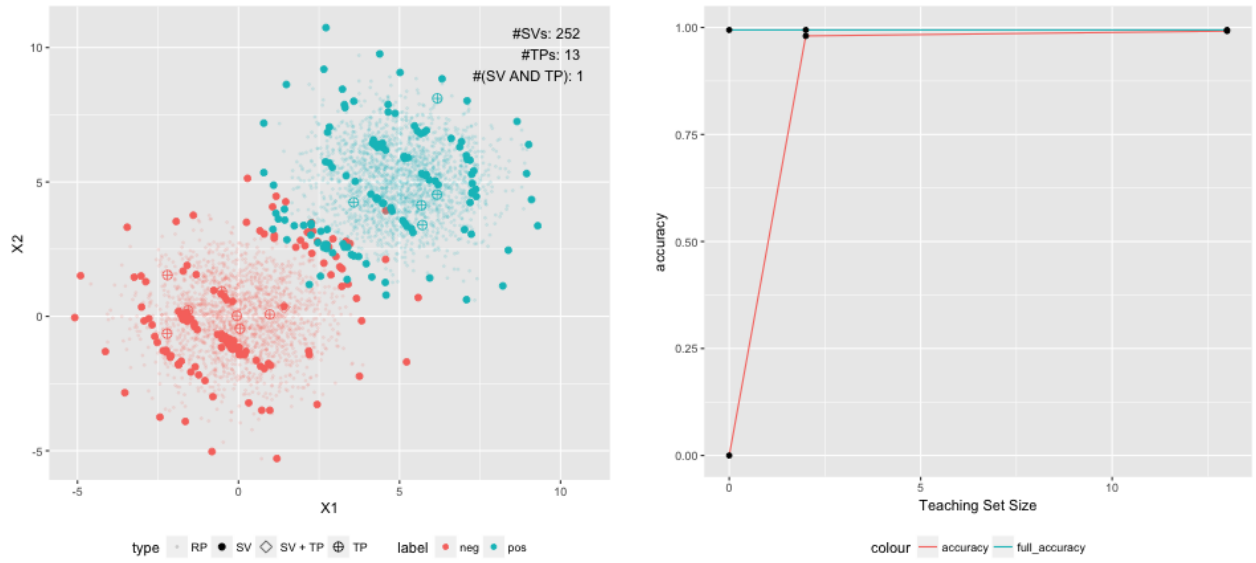
(a) (b)

Figure 18. Mixtures of Gaussians dataset (non-separable), Linear kernel



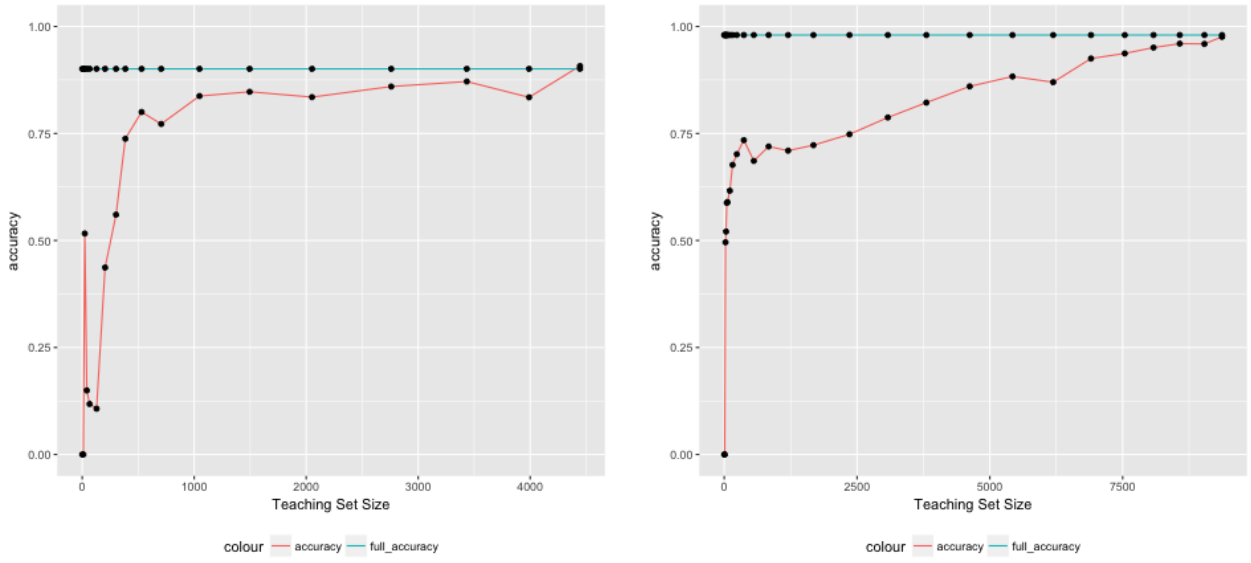
(a) (b)

Figure 19. Mixtures of Gaussians dataset (non-separable), Quadratic kernel



(a) (b)

Figure 20. Mixtures of Gaussians dataset (non-separable), RBF kernel



(a) (b)

Figure 21. (a) MNIST data set, quadratic kernel SVM (b) Fashion MNIST data set, convolutional neural network

# SVs	32,320
# TPs	4,445
#TPs AND SVs	4,357

Table 1. Number of SVs, TPs, and points that are both SVs and TPs on MNIST.