JMLR: Workshop and Conference Proceedings vol 35:1–5, 2014

Open Problem: Online Local Learning

Paul Christiano PAULFCHRISTIANO@EECS.BERKELEY.EDU
UC Berkeley

1. Introduction

In many learning problems, we attempt to infer *global* structure in the interest of making *local* predictions. For example, we might try to infer the skills of the competitors in a tournament in order to predict who will win a match, or we might try to predict characteristics of users and films in order to predict which users will like which films. In even relatively simple settings of this type, it is typically NP-hard to find the latent data which best explain some observations. But do these complexity-theoretic obstructions actually prevent us from making good predictions? Because each prediction depends on only a small number of variables, it might be possible to make good predictions without actually finding a good global assignment. This may seem to be a purely technical distinction, but recent work has shown that several local prediction problems actually *are* easy even though the corresponding global inference problem is hard. The question we pose is: how general is this phenomenon?

2. Problem statement

Fix a universe of items $\mathcal{U}$ and a space of possible labels $\mathcal{L}$. Let $n = |\mathcal{U}|$ and $k = |\mathcal{L}|$, and let r be an arbitrary positive integer. Let $\mathcal{U}^r$ be the collection of r-element subsets of $\mathcal{U}$. A *local prediction problem* is an online learning problem where in each round $t = 0, 1, \ldots$:

- Nature presents a set $X^t = \left\{x_1^t, \ldots, x_r^t\right\} \in \mathcal{U}^r$.
- The learner chooses a labeling, $a^t : X^t \to \mathcal{L}$.
- Nature reveals a payoff function $p^t : \left(X^t \to \mathcal{L}\right) \to [-1, 1]$, and the learner receives the payoff $p^t\left(a^t\right)$.

As usual, we assume the choices of nature are adversarial, and allow the learner to submit a probability distribution over labelings (the objective is then to maximize their expected payoff). For any fixed global labeling $\ell : \mathcal{U} \to \mathcal{L}$ we consider the "benchmark" strategy which always selects $a^t = \ell|_{X^t}$, i.e. the restriction of ℓ to X^t. We aim to find an efficient strategy A which minimizes the expected *regret* against this class of benchmarks:

$$\text{Regret}\,(A) = \max_{\ell:\mathcal{U}\to\mathcal{L}} \sum_{t=1}^{T} p^t\left(\ell|_{X^t}\right) - \sum_{t=1}^{T} p^t\left(a^t\right)$$

where a^t is the decision of A in round t. This goal may be achievable despite the fact that *finding* the maximizing ℓ is generally NP-hard.

It is relatively easy to see that for any r the best possible regret bound is $\mathcal{O}\left(\sqrt{n \log(k) T}\right)$. This regret is achieved by an intractable algorithm which performs multiplicative updates over the space of all k^n possible maps $\ell : \mathcal{U} \to \mathcal{L}$. Efficient strategies with regret $\mathcal{O}\left(\sqrt{nT}\right)$ are known in the case $k = \mathcal{O}(1)$ and $r = 2$ [Christiano (2014)], and near-optimal strategies are known in a few other special cases such as the online gambling problem [Hazan et al. (2012)]. But for general online local learning the best known efficient algorithms have regret that depends polynomially on k rather than logarithmically.

An optimal algorithm for online local learning in the case $r = 2$ would yield an algorithm with regret $\mathcal{O}\left(\sqrt{n^{\lceil \frac{r}{2} \rceil} \log(k) T}\right)$ for general r. Obtaining substantially better regret would yield improved algorithms for planted constraint satisfaction, which may be impossible or require overcoming fundamental algorithmic obstructions [Feldman et al. (2013)].

This leads to our open problems:

- Is there an efficient algorithm that achieves optimal regret $\mathcal{O}\left(\sqrt{n \log(k) T}\right)$ for online local learning with $r = 2$?
- Is there an efficient algorithm that achieves regret $o\left(\sqrt{n^{\frac{r}{2}} T}\right)$ for online local learning for *any* r and k?

For each of these two questions, we offer a $500 cash reward for an affirmative answer.

3. Existing work and semidefinite programming

The online gambling problem[1] was proposed in Abernethy (2010) and Kleinberg et al. (2010) and is a special case of the general online local learning problem. The first non-trivial guarantee for this problem was obtained in Hazan et al. (2012), which achieved nearly optimal worst-case regret $\mathcal{O}\left(\sqrt{n \log(n)^3 T}\right)$. The same algorithm can be applied to general online local learning with $r = 2$ and $k = \mathcal{O}(1)$, to achieve regret $\mathcal{O}\left(\sqrt{n \log(n) T}\right)$. Christiano (2014) proposed the online local learning model, and used similar techniques to achieve optimal regret $\mathcal{O}\left(\sqrt{nT}\right)$ for $k = \mathcal{O}(1)$. All of these algorithms can be applied to the case of general r by assigning each tuple in $\mathcal{U}^{\lceil \frac{r}{2} \rceil}$ a label in $\mathcal{L}^{\lceil \frac{r}{2} \rceil}$, resulting in a regret of $\mathcal{O}\left(\sqrt{n^{\lceil \frac{r}{2} \rceil} T}\right)$ for $k = \mathcal{O}(1)$.

These algorithms are all based on follow the regularized leader over the space of positive semidefinite matrices. Semidefinite programming (SDP) relaxations have the desirable characteristics that they are tight enough that there aren't too many possible solutions, so that there is no information-theoretic obstruction to learning, but at the same time they admit efficient optimization. Similar characteristics make SDP relaxations suitable for approximation algorithms; as a result, SDP's play a central role in the best known algorithms for approximate constraint satisfaction. In some sense recent algorithms for online local learning can be seen as "cutting out the middle man," and using SDP solutions directly to make predictions rather than first trying to obtain a global labeling and then using that labeling to make predictions.

1. Defined and discussed in the appendix.

It is possible that the same SDP relaxations, together with standard approaches to online linear optimization, could be used to achieve optimal regret bounds for $r = 2$ and general $\mathcal{L}$. In the case of $r > 2$, however, we show in the appendix that online learning over known SDP hierarchies cannot achieve regret $o\left(\sqrt{n^{\frac{r}{2}}T}\right)$. Moreover, in the appendix we also show that an algorithm with regret $\mathcal{O}\left(\sqrt{cT}\right)$ for online local learning is able to solve planted constraint satisfaction problems with c clauses. Together with our current understanding of planted constraint satisfaction [Feldman et al. (2013)], this suggests that there might be algorithmic obstructions to obtaining regret $o\left(\sqrt{n^{\frac{r}{2}}T}\right)$.

References

Jacob Abernethy. Can we learn to gamble efficiently? In Adam Tauman Kalai and Mehryar Mohri, editors, *COLT*, pages 318–319. Omnipress, 2010. ISBN 978-0-9822529-2-5. URL http://colt2010.haifa.il.ibm.com/papers/COLT2010proceedings.pdf#page=326.

Paul Christiano. Online local learning via semidefinite programming. In *STOC: ACM Symposium on Theory of Computing (STOC)*, 2014.

Vitaly Feldman, Will Perkins, and Santosh Vempala. On the complexity of random satisfiability problems with planted solutions. *CoRR*, abs/1311.4821, 2013. URL http://arxiv.org/abs/1311.4821.

Elad Hazan, Satyen Kale, and Shai Shalev-Shwartz. Near-optimal algorithms for online matrix prediction. *CoRR*, abs/1204.0136, 2012.

Robert Kleinberg, Alexandru Niculescu-Mizil, and Yogeshwer Sharma. Regret bounds for sleeping experts and bandits. *Machine Learning*, 80(2-3):245–272, 2010. URL http://dx.doi.org/10.1007/s10994-010-5178-7.

Laurent. A comparison of the sherali-adams, lovasz-schrijver, and lasserre relaxations for 0-1 programming. *MOR: Mathematics of Operations Research*, 28, 2003.

Grant Schoenebeck. Linear level lasserre lower bounds for certain k-CSPs. In *FOCS*, pages 593–602. IEEE Computer Society, 2008. URL http://doi.ieeecomputersociety.org/10.1109/FOCS.2008.74.

4. Appendix

4.1. Online gambling

In the online gambling problem, in each round a learner is given a pair of teams x_i, x_j and asked to predict which team will win in an upcoming match; after making their prediction, they learn the result of the match. The goal of the learner is to predict as many matches correctly as possible. The learner's regret is measured against the space of strategies which choose a fixed ranking of the teams and always predict that the higher-ranked team will win.

This problem can be fit in the online local learning setting, by taking $\mathcal{U}$ to be the space of teams and $\mathcal{L} = \{1, 2, \ldots, |\mathcal{U}|\}$. The output of the online local learning algorithm can be used to make predictions about which team will win in the obvious way: if the local predictor outputs ℓ with $\ell(x_i) > \ell(x_j)$, we predict that team x_i will beat team x_j. If team x_i actually beats team x_j in round t, then the payoff $p^t(\ell)$ is defined to be 1 if $\ell(x_i) > \ell(x_j)$ and 0 otherwise.

For any ranking R of the teams, we can construct an $\ell : \mathcal{U} \to \mathcal{L}$ such that $x_i >_R x_j$ if and only if $\ell(x_i) > \ell(x_j)$. This shows that a solution to the local online learning problem with $k = n$ leads to a solution to the online gambling problem with the same regret.

4.2. Lower bounds for the Lasserre hierarchy

We will show that the Lasserre hierarchy of relaxations is not sufficient for a successful attack on the online local learning problem with $r > 2$. Because the Lasserre hierarchy is the strongest of the usual SDP or linear programming hierarchies [Laurent (2003)], this provides evidence that this problem is out of reach for existing techniques.

Fix an integer R corresponding to the number of rounds of the Lasserre hierarchy which we will use. We take $R = \mathcal{O}(1)$ because the complexity of an R-round Lasserre relaxation is n^R. Let $\mathcal{A}_R$ be the space of all "pseudodistributions" on $\mathcal{U} \to \mathcal{L}$, i.e. all maps which associate each $X^t \in \mathcal{U}^R$ with a distribution over the space of labelings $X^t \to \mathcal{L}$, and which are consistent with the constraints of R rounds of the Lasserre hierarchy.

Theorem 1 *In the local prediction setting, no strategy can achieve a regret of* $\mathcal{O}\left(\sqrt{n^{\frac{r}{2}-\varepsilon}T}\right)$ *against* $\mathcal{A}_R$ *for any constants* $\varepsilon > 0, R \in \mathbb{N}$.

Proof

Let $\mathcal{L} = \{+1, -1\}$, fix $0 < \delta < \varepsilon$, and fix $r \in \mathbb{N}$. Consider the following behavior for nature. In each round, a random $X^t \subset \mathcal{U}$ is chosen with $\left|X^t\right| = r$, a random sign $b \in \{+1, -1\}$ is chosen, and the payoff is defined as

$$p^t(\ell) = b * \Pi_{x \in X^t} \ell(x).$$

Since b has expectation 0, it is easy to see that every prediction strategy has an expected payoff of 0.

We will show that for $T = n^{\frac{r}{2}-\delta}$, the Lasserre relaxation to the constraint satisfaction problem

$$\forall t < T : 1 = b * \Pi_{x \in X^t} \ell(x)$$

is satisfiable with high probability. This implies that there is a strategy in $\mathcal{A}_R$ which achieves a payoff of 1 in every round, and consequently that the expected regret of any fixed prediction strategy is at least $\sqrt{n^{\frac{r}{2}-\delta}T}$ for this value of T.

We appeal directly to Lemma 13 and Theorem 11 of Schoenebeck (2008). Theorem 11 implies that for $T = n^{\frac{r}{2}-\delta}$ and for sufficiently large n, with overwhelming probability constant width resolution cannot derive a contradiction from the given constraints. Lemma 13 then implies that a constant number of rounds of the Lasserre hierarchy cannot derive a contradiction.

■

In fact, an identical argument applies for $R = n^\alpha$ rounds of the Lasserre hierarchy for an appropriate $\alpha > 0$ (which depends on ε but not on r). It is interesting to observe that the $\frac{1}{2}$ in the exponent of our lower bound, $n^{\frac{r}{2}}$, is due to a fact about resolution proofs unrelated to the positive semidefiniteness condition. By contrast, the $\frac{1}{2}$ in the exponent of our upper bound, $n^{\lceil \frac{r}{2} \rceil}$, is due to characteristics of our algorithm unrelated to resolution.

4.3. Planted constraint satisfaction

In a (noisy) planted r-XOR-SAT instance, the learner is given a number of r-variable XOR clauses over the variables $x_1, x_2, \ldots, x_n$. Their task is to distinguish two cases, for a fixed $p > \frac{1}{2}$:

1. The distribution of clauses is uniform.
2. There is a "planted" solution $x_1, x_2, \ldots, x_n$, and each clause is satisfied by that solution with probability p.

Such an instance can be translated into an online local learning problem in a straightforward way (as in Section 4.2): for each clause, the learner is told the variables that participate in that clause and guesses an assignment to them: the payoff for a satisfying assignment is 1 and the payoff for an unsatisfying assignment is -1.

It is easy to see that in case (1), no algorithm can achieve a positive expected payoff. In case (2), however, there is a fixed labeling (given by the planted solution) that achieves a payoff of $(2p-1)T$.

Now suppose that we have a learning algorithm with a regret of $\mathcal{O}\left(\sqrt{cT}\right)$. Such an algorithm has a per-round regret of $\mathcal{O}\left(\sqrt{\frac{c}{T}}\right)$. So for some $T = \mathcal{O}(c)$, the per-round regret is less than $(2p-1)$. But this implies that in case (2), our learning algorithm receives a positive expected payoff each round, and consequently can be used to distinguish case (1) from case (2).

The best known algorithms for planted constraint satisfaction require $n^{\frac{r}{2}}$ clauses in order to distinguish cases (1) and (2). Moreover, Feldman et al. (2013) shows that a natural class of "statistical" algorithms cannot distinguish these cases[2] after examining substantially fewer than $n^{\frac{r}{2}}$ clauses and exhibits a statistical algorithm using $\tilde{\mathcal{O}}\left(n^{\frac{r}{2}}\right)$ clauses. The algorithms produced by applying the reduction in this section to typical online learning algorithms are not statistical in this sense, and converting these algorithms into statistical algorithms may introduce a polynomial blowup in the number of clauses required. Nevertheless, the apparent difficulty of finding better algorithms for planted constraint satisfaction suggests that there may be fundamental algorithmic obstructions to attaining regret $o\left(\sqrt{n^{\frac{r}{2}}T}\right)$.

2. In fact they show that a statistical algorithm cannot output a matrix that is substantially correlated with the real clause distribution, but it is easy to see that a successful online learning algorithm must do just that.