JMLR: Workshop and Conference Proceedings vol 35:1–3, 2014

Open Problem: Efficient Online Sparse Regression

Satyen Kale SATYEN@YAHOO-INC.COM
Yahoo! Labs New York

Abstract

In practical scenarios, it is often necessary to be able to make predictions with very limited access to the features of any example. We provide one natural formulation as an online sparse regression problem with squared loss, and ask whether it is possible to achieve sublinear regret with efficient algorithms (i.e. polynomial running time in the natural parameters of the problem).

1. Introduction

In various real-world scenarios, features for examples are constructed by running some computationally expensive algorithms. With resource constraints, it is essential to be able to make predictions with only a limited number of features computed per example.

We provide one natural formulation of this prediction with limited feature access problem as an online sparse regression problem. Over a number of prediction rounds, the learner is required to make real-valued predictions in $[-1, 1]$ for a stream of examples online generated adversarially. Each example has d real-valued features, and has ℓ_2 norm bounded 1. For every example, the learner is restricted to observe only k features (of its choice) out of the d features for the example. Then the true label of the example is observed, and the learner suffers the squared loss. The goal is to minimize regret, i.e. total loss of the learner compared to the loss of the best k-sparse linear predictor of ℓ_2 norm at most 1.

Formally, for $t = 1, 2, \ldots, T$, the learner

1. selects a subset $S_t \subseteq \{1, 2, \ldots, d\}$ of size at most k,
2. observes $x_t(S_t)$, i.e. the values of the features of x_t restricted to the subset S_t,
3. makes a prediction $\hat{y}_t \in [-1, 1]$,
4. observes true label $y_t \in [-1, 1]$, and suffers loss $(\hat{y}_t - y_t)^2$.

Define regret as:

$$\text{Regret} := \sum_t (\hat{y}_t - y_t)^2 - \min_{w:\ \|w\|_0 \le k,\ \|w\| \le 1} \sum_t (w \cdot x_t - y_t)^2.$$

In case $\hat{y}_t$ is chosen using randomization, we consider expected regret instead. We are interested in the following 2 goals:

1. (Sublinear Regret) Make predictions $\hat{y}_t$ so that regret grows like $\text{poly}(d, k) \cdot o(T)$.
2. (Efficiency) Make these predictions efficiently, i.e. in $\text{poly}(d, k, T)$ time per iteration.

2. Open problem

Give an algorithm to achieve the goals of sublinear regret and efficiency simultaneously. Or prove a hardness result showing this is intractable.

The offline problem, finding the best k-sparse linear predictor, is known to be NP-hard by the results of Welch (1982); Natarajan (1995). Given this, for the upper bound it is acceptable to use any reasonable relaxation of the problem as long as the basic requirement that the predictions $\hat{y}_t$ be made using only a limited number of features of any example is satisfied. For example, one may allow the algorithm to use poly(k) features, while the comparator is restricted to use only k-sparse linear predictors.

We note that the question of resolving the computational complexity for a more general problem was also posed by Zolghadr et al. (2013).

3. Related Work and Known Results

A related setting is attribute-efficient learning (Cesa-Bianchi et al., 2011; Hazan and Koren, 2012). This is a batch learning problem where the examples are generated i.i.d., and the goal is to simply output a linear regressor using only a limited number of features per example with bounded excess risk compared to the optimal linear regressor, when given *full access* to the features at test time. While the aforementioned papers give efficient, near-optimal algorithms for this problem, these algorithms do not work in the online sparse regression setting we are interested in because here we are required to make predictions only using a limited number of features.

The simplest algorithm one might try[1] is running a bandit algorithm where the actions correspong to selecting one $O(d^k)$ subsets of coordinates of size k and running an online regression algorithm (such as the Online Newton Step algorithm of Hazan et al. (2007)) over these coordinates for B steps, initialized from scratch each time. Applying standard bandit algorithms, since there are $O(d^k)$ actions, T/B effective time steps, and a scaling of B for each time step to account for the B rounds of the online regression algorithm, the regret will become $O(\sqrt{d^k BT})$ with respect to the best periodically restarted regression algorithm. The loss of the latter relative to the best weight vector with a fixed set of k non-zero coordinates is $(T/B)O(k\log(B))$. Optimizing $B = O(T^{1/3}d^{-k/3})$ gives an overall regret of $O(k^2 d^{k/3} T^{2/3}\log(T/d))$. This has exponential dependence on k both in running time and the regret.

Zolghadr et al. (2013) consider a very closely related setting where features and labels may be obtained by the learner at some cost (which may be different for different features), and this cost is factored into the loss of the learner. In the special case of their setting corresponding to the problem considered here, they given an algorithm, LQDExp3, which relies on discretizing all k-sparse weight vectors and running an exponential weights experts algorithm on the resulting set with stochastic loss estimators, obtaining an $O(\sqrt{dT})$ regret bound. However the running time of their algorithm is prohibitive: $O((dT)^{O(k)})$ time per iteration.[2] This algorithm is based on the observation that the loss for a given weight vector

1. This was also suggested by an anonymous referee.
2. The paper of Zolghadr et al. (2013) gives a per-iteration running time bound of $O(T^{O(d)})$; the improved running time bound follows from the observation made by an anonymous referee that only k-sparse predictors need to be discretized.

w at time t can be rewritten as

$$w^\top x_t x_t^\top w - 2w^\top (y_t x_t) + y_t^2.$$

So an unbiased estimator for the loss can be constructed by sampling coordinates and using importance weighting to construct unbiased estimators for the two quantities

$$x_t x_t^\top \quad \text{and} \quad y_t x_t \tag{1}$$

In collaboration with Dean Foster, we can give an algorithm that has better running time, $O(d^k)$ per iteration, but worse[3] regret, $O(\text{poly}(d)\sqrt{T})$. Also, this algorithm requires access to $k' \geq k+2$ features per example, whereas the comparator is allowed access to only k features.

The idea is the following. Treat each subset S of $[d]$ of size at most k as an expert, and run the standard exponential weights algorithm on these experts. Each expert S internally runs stochastic gradient descent on weight vectors that are non-zero only on the coordinates in S. In each round, the learner samples an expert (i.e. a subset S_t) from the distribution generated by the exponential weights algorithm, obtains the features indexed by S_t, and uses the internal weight vector of the expert S_t to make its prediction $\hat{y}_t$. Next, using the extra $k' - k$ features that are available to the learner, it constructs stochastic gradients for all experts by sampling a uniformly random subset of coordinates of size $k' - k$ and using importance weighting to construct unbiased estimators for the quantities in (1). Overall, using the known regret bounds for the exponential weights algorithm and stochastic gradient descent, we obtain the stated regret bound. The running time is dominated by the exponential weights algorithm which needs to update weights for all the experts in each round, and is hence $O(d^k)$ per round.

References

Nicolò Cesa-Bianchi, Shai Shalev-Shwartz, and Ohad Shamir. Efficient learning with partially observed attributes. *Journal of Machine Learning Research*, 12:2857–2878, 2011.

Elad Hazan and Tomer Koren. Linear regression with limited observation. In *ICML*, 2012.

Elad Hazan, Amit Agarwal, and Satyen Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2-3):169–192, 2007.

B. K. Natarajan. Sparse approximate solutions to linear systems. *SIAM J. Computing*, 25 (2):227–234, 1995.

William J. Welch. Algorithmic complexity: three NP-hard problems in computational statistics. *Journal of Statistical Computation and Simulation*, 15(1):17–25, 1982.

Navid Zolghadr, Gábor Bartók, Russell Greiner, András György, and Csaba Szepesvári. Online learning with costly features and labels. In *NIPS*, pages 1241–1249, 2013.

3. The dependence on d is a small polynomial which we did not attempt to optimize.