Robust Efficient Conditional Probability Estimation

John Langford
Yahoo! Research
jl@yahoo-inc.com

1 The Problem

The problem is finding a general, robust, and efficient mechanism for estimating a conditional probability $P(y|x)$ where robustness and efficiency are measured using techniques from learning reductions.

In particular, suppose we have access to a binary regression oracle B which has two interfaces—one for specifying training information and one for testing. Training information is specified as $B(x', y')$ where x' is an unspecified feature vector and $y' \in [0, 1]$ is a bounded range scalar with no value returned. This operation is stateful, possibly altering the return value of the testing interface in arbitrary ways. Testing is done according to $B(x')$ with a value in $[0, 1]$ returned. The testing operation operation is stateless.

A learning reduction consists of two algorithms R and R^{-1}.

The algorithm R takes as input a single example (x, y) where x is a feature vector and $y \in \{1, ..., k\}$ is a discrete variable. R then specifies a training example (x', y') for the oracle B. R can then create another training example for B based on all available information. This process repeats some finite number of times before halting without returning information.

A basic observation is that for any oracle algorithm, a distribution $D(x, y)$ over multiclass examples and a reduction R induces a distribution over a sequence $(x', y')^*$ of oracle examples. We collapse this into a distribution $D'(x', y')$ over oracle examples by drawing uniformly from the sequence.

The algorithm R^{-1} takes as input a single example (x, y) and returns a value $v \in [0, 1]$ after using (only) the testing interface of B zero or more times.

We measure the power of an oracle and a reduction according to squared-loss regret according to:

$$\text{reg}(D, R^{-1}) = E_{(x,y)\sim D}[(R^{-1}(x, y) - D(y|x))^2]$$

and similarly letting $\mu_{x'} = E_{(x',y')\sim D'}[y']$.

$$\text{reg}(D', B) = E_{(x',y')\sim D'}(B(x') - \mu_{x'})^2$$

The open problem is to specify R and R^{-1}satisfying the following theorem:

Theorem 1 *For all multiclass distributions $D(x, y)$, for all binary oracles B: The computational complexity of R and R^{-1} are $O(\log k)$ and*

$$\textit{reg}(D, R^{-1}) \leq C\,\textit{reg}(D', B)$$

where C is a universal constant.

Alternatively, this open problem is satisfied by proving there exists no deterministic algorithms R, R^{-1} satisfying the above theorem statement.

2 Motivation

The problem of conditional probability estimation is endemic to machine learning applications. In fact, in some branches of machine learning, this is simply considered "the problem". Typically conditional probability estimation is done in situations where the conditional probability of only one bit is required, however there are a growing number of applications where a well-estimated conditional probability over a more complex object is required. For example, all known methods for solving contextual bandit algorithms over an arbitrary policy class require knowledge of or good estimation of $P(a \mid x)$ where a is an action.

There is a second intrinsic motivation which is matching the lower bound. No method faster than $O(\log k)$ can be imagined because the label y requires $\log_2 k$ bits to specify and hence read. Similarly it's easy to prove no learning reduction can provide a regret ratio with $C < 1$.

The motivation for using the learning reduction framework to specify this problem is a combination of generality and the empirical effectiveness in application of learning reductions. Any solution to this will be general because any oracle B can be plugged in, even ones which use many strange kinds of prior information, features, and active multitask hierachical (insert your favorite adjective here) structure.

3 Related Results

The state of the art is summarized by [1] which shows it's possible to have a learning reduction satisfying the above theorem with either:

1. C replaced by $\log_2^2 k$ (using a binary tree structure)
2. or the computational time increased to $O(k)$ (using an error correcting code structure).

Hence, answering this open problem in the negative shows that there is an inherent computation vs. robustness tradeoff.

There are two other closely related problems, where similar analysis can be done.

1. For multiclass classification, where the goal is predicting the most likely class, a result analogous to the open problem is provable using error correcting tournaments [2].
2. For multiclass classification in a partial label setting, no learning reduction can provide a constant regret guarantee [3].

4 Silly tricks that don't work

Because Learning reductions are not familiar to everyone, we note certain tricks which do not work here to prevent false leads and provide some intuition.

4.1 Ignore B's predictions and use your favorite learning algorithm instead.

This doesn't work, because the quantification is for all D. Any specified learning algorithm will have some D on which it has nonzero regret. On the other hand, because R calls the oracle at least once, there is a defined induced distribution D'. Since the theorem must hold for all D and B, it must hold for a D your specified learning algorithm fails on and for a B for which $\text{reg}(D', B) = 0$ implying the theorem is not satisfied.

4.2 Feed random examples into B and vacuously satisfy the theorem by making sure that the right hand side is larger than a constant.

This doesn't work because the theorem is stated in terms of squared loss regret rather than squared loss. In particular, if the oracle is given examples of the form (x', y') where $y' \in \{0, 1\}$ is drawn uniformly at random, any oracle specifying $B(x') = 0.5$ has zero regret.

4.3 Feed pseudorandom examples into B and vacuously satisfy the theorem by making sure that the right hand side is larger than a constant.

This doesn't work, because the quantification is "for all binary oracles B", and there exists one which, knowing the pseudorandom seed, can achieve zero loss (and hence zero regret).

4.4 Just use Boosting to drive the LHS to zero.

Boosting theorems require a stronger oracle—one which provides an edge over some constant baseline for each invocation. The oracle here is not limited in this fashion since it could completely err for a small fraction of invocations.

4.5 Take an existing structure, parameterize it, randomize over the parameterization, and then average over the random elements.

Employing this approach is not straightforward, because the average in D' is over an increased number of oracle examples. Hence, at a fixed expected (over oracle examples) regret, the number of examples allowed to have a large regret is increased.

References

[1] Alina Beygelzimer, John Langford, Yuri Lifshits, Gregory Sorkin, and Alex Strehl, Conditional Probability Tree Estimation Analysis and Algorithms, UAI 2009.

[2] Alina Beygelzimer, John Langford, and Pradeep Ravikumar, Error-Correcting Tournaments, ALT 2009.

[3] Alina Beygelzimer and John Langford, The Offset Tree for Learning with Partial Labels, KDD 2009.