CONSISTENT ROBUST LOGARITHMIC TIME PREDICTION

JOHN LANGFORD (ABOUT JOINT WORK WITH SEVERAL COAUTHORS)

Yahoo! Research

(jl@{yahoo-inc.com, hunch.net})

Let's take as our goal prediction of 1 of k choices given input features x from a (possibly large) input space. It's clear from the problem statement that any algorithm requires $\Omega(\log k)$ as that is the answer complexity. A natural question is: Can we construct good learning algorithms with a matching $O(\log k)$ runtime? Many standard approaches for addressing this problem such as multiclass support vector machines, multiclass perceptron, one-against-all, all pairs, or error correcting output codes are $O(k)$ or worse, exhibiting an exponential gap between what might be possible and standard practice.

One plausible approach previously considered is prediction via a tree. The idea here is that you construct a binary tree over the set of choices, then for each node in the tree learn a predictor of whether the correct label is to the left or right using all examples with a label in the set of choices beneath the node. A surprising drawback of this approach is that it is *inconsistent* as proved in a theorem[5].

The above observations suggest the central question can only be answered pessimistically, but this turns out to not be the case. There are consistent algorithms for $O(\log k)$ training and testing, for which we have built a nearly complete understanding that I will outline next.

(1) The filtration trick used in the Filter Tree [5] allows consistent multiclass prediction in a binary tree. The basic algorithm can be generalized to cost-sensitive classification problems at the (necessary) cost of $O(k)$ training time per example, while keeping $O(\log k)$ prediction time. The filter tree sacrifices some robustness compared to the best-possible approaches ignoring computational complexity since an error is induced if any of $\log k$ predictions are incorrect. However, filter trees can be understood as the first instance of a larger family of algorithms, error correcting tournaments [5], with the property that any *constant* fraction of the binary classifications can be incorrect.

(2) Another form of prediction problem is in the partial-information setting, where you learn the reward of just one choice from a set of choices. In this setting, using the filtration trick and an additional offseting trick in the Offset Tree [6] provides a consistent method. The offset tree turns out to be as robust as possible given the (very limited) information in this setting—no other approach has a better reduction analysis.

(3) If we want to know the probability of 1 of k events, it turns out that the simple binary tree approach *is* consistent, and we can prove that the squared loss of the final estimate is bounded by $(\log_2 k)^2$ times the average squared loss of predictions at the nodes.

While all of the discussion above is about provable properties of learning algorithms, we have in fact tested these algorithms on a variety of different problems, and in every case found that they have performance similar to and often better than the more commonly used exponentially slower approaches. In one extreme test of the probability case, Andriy Mnih [7] (independently) constructed a tree-based algorithm which competes successfully with n-gram models in language learning.

References

[1] R. O. Duda and P. E. Hart. Pattern Classification and Scene Analysis, Wiley, 1973.

[2] Y. Lee, Y. Lin, and G. Wahba, Multicategory Support Vector Machines, Theory, and Applications to the Classification of Microarray Data and Satellite Radiance Data. University of Wisconsin TR 1064, September 2002.

[3] T. Hastie and R. Tibshirani, Classification by Pairwise Coupling, Annals of Statistics, 26(2): 451—471, 1998.

[4] T. Dietterich and G. Bakiri. Solving multiclass learning problems via error-correcting output codes, Journal of Artificial Intelligence Research, 2:263—286, 1995.

[5] A. Beygelzimer, J. Langford, and P. Ravikumar, Error Correcting Tournaments, http://arxiv.org/abs/0902.3176 .

[6] Alina Beygelzimer and John Langford, The Offset Tree for Learning with Partial Labels, http://arxiv.org/abs/0812.4044 .

[7] Andriy Mnih & Geoffrey Hinton, A Scalable Hierarchical Distributed Language Model, NIPS 2008.

0.0.1. *Topic: Learning Theory.*

0.0.2. *Preference: Oral.*