Algorithms, Geometry, and Learning Reading Group Notes: Agnostically Learning Decision Trees

Steve Mussmann

November 29, 2016

Abstract

This reading group is on the paper "Agnostically Learning Decision Trees" by Gopalan, Kalai, and Klivans. This paper provides a query algorithm for agnostically learning decision trees with respect to the uniform distribution on inputs. Given black-box access to an arbitrary binary function f on the n-dimensional hypercube, the algorithm finds a function that agrees with f on almost as many inputs as the best size-t decision tree, in polynomial time. This is the first polynomial-time algorithm for learning decision trees in a harsh noise model. The core of the learning algorithm is a procedure to implicitly solve a convex optimization problem over the L_1 ball in 2^n dimensions using an approximate gradient projection method. The problem and basic concepts will be presented, the main result will be stated, and the proof will be shown.

1 Motivation

Decision trees are a model widely used in practice. Typically, q decision tree is learned from a top-down approach, splitting nodes based on a greedy heuristic. Later, pruning possibly takes place.

From a theoretical perspective, there exist algorithms with guarantees for learning decision trees without noise (i.e. the data distribution is modeled by a decision tree). Here, we examine the noisy problem where the data distribution is arbitrary and we try to achieve as much error as the best decision tree. This is known as the agnostic learning paradigm.

2 Main Result

Let there be a function $g : \{-1, 1\}^n \to \{-1, 1\}$ that is computed by a decision tree.

This model is PAC-learnable for data from a size-t decision tree, known from earlier work.

This paper shows it is agnostically learnable. In particular,

Define

$$\text{opt}_{\mathcal{C}} = \min_{c \in C} \text{err}(c) \tag{1}$$

Theorem 1 *Let $\mathcal{C}$ be the class of decision trees with at most t leaves. For any $t, \epsilon > 0$ and black-box access to the data distribution. There exists a polynomial time algorithm that outputs a hypothesis h such that $err(h) \leq opt_{\mathcal{C}} + \epsilon$.*

3 Basic Concepts

3.1 Agnostic Learning

Agnostic Learning can be defined for arbitrary distributions using random samples from $\mathcal{D}$. However, here, we solve the strictly easier problem of membership queries (active learning) with evaluation on the uniform distribution.

We define $\text{err}(c) = \Pr_{(x,y)\sim\mathcal{D}}[c(x) \neq y]$. Then the optimal error rate is $opt = min_{c\in\mathcal{C}}\text{err}(c)$

Precisely, we say a concept class $\mathcal{C}$ is agnostically learnable with queries under the uniform distribution if there is an algorithm which when given a query oracle for $\mathcal{D}$ and parameters ϵ, δ as inputs, returns a hypothesis h such that $\Pr_{(x,y)\sim\mathcal{D}}[h(x) \neq y] \leq opt + \epsilon$ with probability $1 - \delta$.

Note that we do not require $h \in \mathcal{C}$.

3.2 Boolean Functions

A boolean function is a function

$$P : \{-1, 1\}^n \to \mathbb{R} \tag{2}$$

For the rest of this talk, we'll assume the only distribution is the uniform distribution over boolean tuples.

With this distribution, we can define the dot product as

$$P \cdot Q = \mathbb{E}[P(x)Q(x)] \tag{3}$$

and thus,

$$||P||_2 = \sqrt{\mathbb{E}[|P(x)|^2]} \tag{4}$$

3.3 Polynomial Representation

Any boolean function can be written as a polynomial. Define the mononomials as $\mathcal{X}_S(x) = \Pi_{i\in S}x_i$

Let $\hat{P}(S)$ be the coefficients for $S \subset [n]$. Thus, $P(x) = \sum_{S\subset[n]} \hat{P}(S)\mathcal{X}_S(x)$.

We can define norms for the coefficients with respect to the uniform distribution. Then,

$$L_2(P) = \sqrt{\sum_S |\hat{f}(P)|^2} \tag{5}$$

$$L_1(P) = \sum_S |\hat{P}(S)| \tag{6}$$

$$L_\infty(P) = \max |\hat{P}(S)| \tag{7}$$

We say P is d-degree if $\hat{P}(S) = 0$ for $|S| > d$.

We say P is k-sparse if $L_1(P) \leq k$.

Finally, the support is $supp(P) = \{S : \hat{P}(S) \neq 0\}$.

By orthogonality,

$$P \cdot Q = \sum_S \hat{P}(S)\hat{Q}(S) \tag{8}$$

4 Useful Tools

4.1 Depth restricted decision trees as sparse polynomials

If a boolean function P can be computed by a decision tree with t leaves, then $L_1(f) \leq t$.

4.2 KM

The paper which gives the noiseless learning algorithm has some more general results using sampling arguments and membership queries.

In particular, given oracle access to P and a parameter θ, the algorithm runtime is $\text{poly}(n, \theta^{-1}, L_2(P))$ and returns $Q = KM(P, \theta)$. With the following properties.

Lemma 1 *For $Q = KM(P, \theta)$, $|supp(Q)| = O(L_2(P)^2\theta^{-2})$ and $L_\infty(P - Q) \leq \theta$.*

Lemma 2 *If P is t-sparse ($L_1(P) \leq t$), then for $Q = KM(P, \frac{\epsilon^2}{2t})$, $||P - Q||_2 \leq \epsilon$.*

Put another way, if P is t-sparse, then, $||P - Q||_2 \leq \sqrt{2\theta t}$.

5 Idealized Projected Subgradient Descent

Define $\nabla_f P(x) = \text{sgn}(P(x) - f(x))$.

Lemma 3 *For polynomials P and Q,*

$$\nabla_f P \cdot (P - Q) \geq err_f(P) - err_f(Q)$$

Thus, $\nabla_f P(x)$ is a subgradient.

Define K_t as the t-sparse polynomials. This is a convex set and we define the projection onto K_t as proj_K.

Input: T, η
$P_0 = 0$
for $k = 1$ **to** T **do**
  $P'_k = P_{k-1} - \eta \nabla_f P_{k-1}$
  $P_k = \text{proj}_K(P'_k)$
end for
Return best P_k

This algorithm yields a good solution, but requires $O(2^n)$ computational time.

6 Efficient Projected Subgradient Descent

Input: T, η, θ
$P_0 = 0$
for $k = 1$ **to** T **do**
  $P'_k = P_{k-1} - \eta KM(\nabla_f P_{k-1}, \theta)$
  $P_k = KM(\text{proj}_K(P'_k), \theta)$
end for
Return best P_k

This algorithm is polynomial time yet still achieves the optimal accuracy for a decision tree of size t.

7 Analysis

7.1 L_1 Ball Projection

Formally, the projection operator is defined as

$$\text{proj}_K(P) = \text{argmin}_{L_1(Q) \leq t} ||P - Q||_2 \tag{9}$$

Define $\text{shrink}(P, l)$ as the function Q such that

$$\hat{Q}(S) = \begin{cases} \hat{P}(S) - l & \hat{P}(S) \geq l \\ \hat{P}(S) + l & \hat{P}(S) \leq -l \\ 0 & |\hat{P}(S)| \leq l \end{cases} \tag{10}$$

Lemma 4 *For any P, $proj_K(P) = shrink(P, l)$ for the smallest $l \geq 0$ so that $shrink(P, l) \in K_t$.*

7.2 L_∞ approximations

Lemma 5 *Let P, P' be such that $L_\infty(P - P') \leq \epsilon$. Then $L_\infty(proj_K(P) - proj_K(P')) \leq 2\epsilon$*

Lemma 6 *Let P, P' be such that $L_\infty(P - P') \leq \epsilon$. Then $||proj_K(P) - proj_K(P')||_2 \leq \sqrt{4\epsilon t}$*

7.3 Analysis of subgradient descent

Define P_k and P'_k as in the efficient algorithm. Define $Q'_k = P_{k-1} - \eta \nabla_f P_{k-1}$ and $Q_k = \text{proj}_K(Q'_k)$.

We wish to bound $||P_k - Q_k||_2$.

Lemma 7 *The polynomials P_k and Q_k satisfy $||P_k - Q_k|| \leq 4\sqrt{\theta t}$.*

Let $P^* \in K$ be the polynomial that minimizes err_f.

Lemma 8

$$||P_k - P_*||_2^2 - ||P_{k+1} - P_*||_2^2 \geq 2\eta(err_f(P_k) - err_f(P_*)) - 2\eta^2$$

Theorem 2 *For any $T \geq 0$, if $\eta \leq \frac{t}{\sqrt{T}}$ and $\theta \leq \frac{\eta^2}{C^2 t^3}$, then for some $k \leq T$,*

$$err_f(P_k) \leq err_f(P_*) + 2\eta$$

The runtime is poly(n, t, T).