Neural Network Learning: Theoretical Foundations
Chapter 22 and 23

Martin Anthony and Peter L. Bartlett

Speaker : Young-geun Kim

November 29, 2017

Contents

outline

Part 4 : Algorithmics

Part 4 : Algorithmics

- In the previous parts, we have discussed the sample complexity of learning. VC-dimension and fat-shattering dimension were important concepts to quantify it.
- In this part, we consider the time complexity of learning. To get a practical value, an algorithm should be possible to produce a good output ‘quickly‘.

Part 4 : Algorithmics

Part 4 focuses on

- what is efficient learning (Chap. 22)
- the relation between efficient learning and optimization problem (Chap. 23)
- the time complexity of the Boolean Perceptron (Chap. 24)
- the hardness of the consistency problem with neural networks (Chap. 25)
- constructive learning algorithms iteratively adding basis functions to a convex combination such as *Construct* and *Adaboost* (Chap. 26)

outline

22.1. Introduction

In this chapter, we see

- the definition of efficient learning.
- the role of VC-dimension on efficient learning for binary function class.
- the role of fat-shattering dimension on efficient learning for real function class.
- the efficient learnability of binary classes in the restricted model.

22.2. Graded Function Classes

- The increasing speed of learning time w.r.t. the number of inputs, n, should be considered, but a learning algorithm is defined on fixed n.
- Graded function classes $\bigcup_{n=1}^{\infty} F_n$, an union of function class F_n for input size n, formalize the notion of 'scaling' w.r.t. the number of inputs.
- For instance, let $Z_n = X_n \times \{0, 1\}$ and $H = \bigcup_{n=1}^{\infty} H_n$ be a graded binary function class. Then a learning algorithm for H is a mapping

$$L : \bigcup_{n=1}^{\infty} \bigcup_{m=1}^{\infty} Z_n^m \to \bigcup_{n=1}^{\infty} H_n$$

such that if $z \in Z_n^m$, then $L(z) \in H_n$, and for each n, L is a learning algorithm for H_n.

22.3. Efficient Learning

Definition 22.1 Let $F = \bigcup_{n=1}^{\infty} F_n$ be a graded class of functions and suppose that L is a learning algorithm for F. We say that L is efficient if:

- the worst-case running time $R_L(m, n)$ of L on samples $z \in Z_n^m$ is polynomial in m and n
- the sample complexity $m_L(n, \epsilon, \delta)$ of L on F_n is polynomial in n, $1/\epsilon$ and $\ln(1/\delta)$.

22.3. Efficient Learning

- We separate the running time of the algorithm and the sample complexity, which is standard or based on standard definitions: other definitions are possible, but they are all, in a sense, equivalent (Haussler et al., 1991).
- Roughly speaking, if the sample size is doubled, an efficient learning algorithm should give approximately squared confidence for fixed accuracy, and approximately halved accuracy for fixed confidence.

Theorem 22.2 Let $H = \bigcup_{n=1}^{\infty} H_n$ be a graded binary function class.

- If VCdim(H_n) is polynomial in n, then any SEM algorithm for H is a learning algorithm with sample complexity $m_L(n, \epsilon, \delta)$ polynomial in n, $1/\epsilon$ and $\ln(1/\delta)$.
- If there is an efficient learning algorithm for H, then VCdim(H_n) is polynomial in n.

Proof

- By thm 4.2, for any SEM algorithm L for H,

$$m_L(n, \epsilon, \delta) \leq \frac{64}{\epsilon^2}\left(2VCdim(H_n)\ln\left(\frac{12}{\epsilon}\right) + \ln\left(\frac{4}{\delta}\right)\right).$$

- By thm 5.2, for any learning algorithm L for H with $0 < \epsilon, \delta < 1/64$,

$$m_L(n, \epsilon, \delta) \geq \frac{1}{320\epsilon^2} VCdim(H_n).$$

Theorem 22.3 Let $F = \bigcup_{n=1}^{\infty} F_n$ be a graded real function class.

- If the fat-shattering dimension $fat_{F_n}(\alpha)$ is polynomial in n and $1/\alpha$, and L is the learning algorithm based on any approximate SEM algorithm $\mathcal{A}$ (as in Theorem 19.1), then L has sample complexity $m_L(n, \epsilon, \delta)$ polynomial in n, $1/\epsilon$ and $\ln(1/\delta)$.
- If there is an efficient learning algorithm for F, then $fat_{F_n}(\alpha)$ is polynomial in n and $1/\alpha$.

Proof

- In thm 19.1, $L(z) = \mathcal{A}(z, \epsilon_0/6)$ where $\epsilon_0 = \frac{16}{\sqrt{m}}$ satisfies

$$m_L(n, \epsilon, \delta) \le \frac{256}{\epsilon^2}\left(18 fat_{F_n}(\epsilon/256) \ln^2\left(\frac{128}{\epsilon}\right) + \ln\left(\frac{16}{\delta}\right)\right).$$

- By thm 19.5, for any learning algorithm L for F_n with $B \ge 2$, $0 < \epsilon < 1$, $0 < \delta < 1/100$ and $0 < \alpha < 1/4$,

$$m_L(\epsilon, \delta, B) \ge \frac{fat_{F_n}(\epsilon/\alpha) - 1}{16\alpha}.$$

Definition 22.4 An efficient approximate-SEM algorithm for the graded real function class $F = \bigcup_{n=1}^{\infty} F_n$ is an algorithm that taked as input $z \in Z_n^m$ and $\epsilon \in (0,1)$ and, in time polynomial in m, n and $1/\epsilon$, produces an output hypothesis $f \in F_n$ such that

$$\hat{er}_z(f) < \inf_{g \in F_n} \hat{er}_z(g) + \epsilon.$$

An efficient SEM algorithm for the graded binary function class $H = \bigcup_{n=1}^{\infty} H_n$ is an algorithm that takes as input $z \in Z_n^m$ and, in time polynomial in m and n, returns $h \in H_n$ such that

$$\hat{er}_z(h) = \min_{g \in H_n} \hat{er}_z(g).$$

Theorem 22.5

- Suppose that $H = \bigcup_{n=1}^{\infty} H_n$ is a graded binary function class and that $VCdim(H_n)$ is polynomial in n. Then, any efficient SEM algorithm for H is an efficient learning algorithm for H.
- Suppose that $F = \bigcup_{n=1}^{\infty} F_n$ is a graded real function class and that $fat_{F_n}(\alpha)$ is polynomial in n and $1/\alpha$. Then any learning algorithm for F based on an efficient approximate-SEM algorithm is efficient.

Proof

- By thm 22.2, any SEM algorithm for H is a learning algorithm with $m_L(n, \epsilon, \delta)$ polynomial in n, $1/\epsilon$ and $\ln(1/\delta)$.
- By thm 22.3, $m_L(n, \epsilon, \delta)$ is polynomial in n, $1/\epsilon$ and $\ln(1/\delta)$. For any efficient approximate-SEM algorithm $\mathcal{A}$, the learning algorithm for F based on $\mathcal{A}$ computes $\mathcal{A}(\mathbf{z}, \epsilon_0)$ in the time polynomial in m, n and $1/\epsilon_o$. Here, $\epsilon_0 = 16/\sqrt{m}$.

Definition 22.6 An algorithm L is an efficient consistent-hypothesis-finder for the graded binary class $H = \bigcup_{n=1}^{\infty} H_n$ if, given any training sample z of length m for a target function in H_n, L halts in time polynomial in m and n and returns $h = L(z) \in H_n$ such that $\hat{er}_z(h) = 0$.

Theorem 22.7 Suppose that $H = \bigcup_{n=1}^{\infty} H_n$ is a binary graded function class and that $VCdim(H_n)$ is polynomial in n. Then any algorithm that is an efficient consistent-hypothesis-finder for H is an efficient learning algorithm for H.

outline

23.1. Introduction

In this chapter, we see

- the definition of randomized algorithms.
- the relation between efficient learning and efficient randomized SEM algorithm.
- the relation between efficient learning and optimization problem of finding a hypothesis with small sample error.

Definition 2.1 Suppose that H is a class of functions that map from a set X to $\{0,1\}$. A learning algorithm L for H is a function

$$L : \bigcup_{m=1}^{\infty} Z^m \to H$$

from the set of all training samples to H, with the following property:

- given any $\epsilon, \delta \in (0,1)$,

there is an integer $m_0(\epsilon, \delta)$ such that if $m \geq m_0(\epsilon, \delta)$ then,

- for any probability distribution P on $Z = X \times \{0,1\}$,

if z is a training sample of length m, drawn randomly according to the product probability distribution P^m, then for $m \geq m_0(\epsilon, \delta)$,

$$P^m\{er_P(L(z)) < \inf_{g \in H} er_P(g) + \epsilon\} \geq 1 - \delta.$$

We say that H is learnable if there is a learning algorithm for H.

23.2. Randomized Algorithms

Definition 23.1 A randomized learning algorithm for the graded class $F = \bigcup_{n=1}^{\infty} F_n$ is a mapping

$$L : \{0,1\}^* \times \bigcup_{n=1}^{\infty} \bigcup_{m=1}^{\infty} Z_n^m \to \bigcup_{n=1}^{\infty} F_n$$

such that if $z \in Z_n^m$, then $L(b, z) \in F_n$, and:

- given any $\epsilon, \delta \in (0, 1)$ and positive integer n,

there is an integer $m_0(n, \epsilon, \delta)$ such that if $m \geq m_0(n, \epsilon, \delta)$ then

- for any probability distribution P on Z_n,

if z is a training sample of length m, drawn randomly according to the product probability distribution P^m, and b is a sequence of independent, uniformly chosen bits, then for $m \geq m_0(n, \epsilon, \delta)$,

$$EP^m\{er_P(L(b, z)) < \inf_{g \in F_n} er_P(g) + \epsilon\} \geq 1 - \delta.$$

We say that F is learnable if there is a learning algorithm for F.

23.2. Randomized Algorithms

Definition 23.2 A randomized algorithm $\mathcal{A}$ is an efficient randomized SEM algorithm for the graded binary function class $H = \bigcup_{n=1}^{\infty} H_n$ if given any $z \in Z_n^m$, $\mathcal{A}$ halts in time polynomial in n and m and outputs $h \in H_n$ which, with probability at leat $1/2$, satisfies

$$\hat{er}_z(h) = \min_{g \in H_n} \hat{er}_z(g).$$

A randomized algorithm $\mathcal{A}$ is an efficient randomized approximate-SEM algorithm for the graded real function class $F = \bigcup_{n=1}^{\infty} F_n$ if the following holds: given any $z \in Z_n^m$, and any $\epsilon \in (0,1)$, $\mathcal{A}$ halts in time polynomial in n, m and $1/\epsilon$ and outputs $f \in F_n$ which, with probability at least $1/2$, satisfies

$$\hat{er}_z(f) < \inf_{g \in F_n} \hat{er}_z(g) + \epsilon.$$

- Suppose we run a randomized approximate-SEM algorithm k times on a fixed input, keeping the output hypothesis $f^{(k)}$ with minimal sample error among all the k hypotheses returned.
- The probability that $f^{(k)}$ has error that is not within ϵ of the optimal is at most $(1/2)^k$. This enables us to handle the confidence of randomized approximate-SEM by manipulating k .

Theorem 23.3

- Suppose that $H = \bigcup_{n=1}^{\infty} H_n$ is a graded binary function class and that $VCdim(H_n)$ is polynomial in n. If there is an efficient randomized SEM algorithm $\mathcal{A}$ for H, then there is an efficient learning algorithm for H that uses $\mathcal{A}$ as a subroutine.

Proof

- By results from previous parts, w.p. at least $1 - 4\prod_H(2m)\exp(-\epsilon^2 m/8)$,

$$er_P(h) < opt_P(H_n) + 2\epsilon$$

for all h achieving minimum sample error.
A randomized SEM algorithm with k iterations for z gives $h^{(k)}$ satisfying w.p. at least $1 - 1/2^k$,

$$\hat{er}_z(h^{(k)}) = \min_{g \in H_n} \hat{er}_z(g).$$

Now, choose $k = \max(m_0(n, \epsilon, \delta), C\log(1/\delta))$ where C is sufficiently large constant and

$$m_0(n, \epsilon, \delta) = \frac{64}{\epsilon^2}\left(VCdim(H_n)\ln(128/\epsilon^2) + \ln(8/\delta)\right).$$

Theorem 23.3

- Suppose that $F = \bigcup_{n=1}^{\infty} F_n$ is a graded real function class with $fat_{F_n}(\alpha)$ polynomial in n and $1/\alpha$. If there is an efficient randomized approximate SEM algorithm $\mathcal{A}$ for H, then there is an efficient learning algorithm for F that uses $\mathcal{A}$ as a subroutine.

23.3. Learning as Randomized Optimization

- It is possible to construct efficient learning algorithm using efficient approximate-SEM or SEM algorithm.
- The converse also true. i.e., the existence of efficient learning algorithm implies the existence of efficient randomized SEM or approximate-SEM algorithm.

Theorem 23.4

- If there is an efficient learning algorithm for the graded binary class $H = \bigcup_{n=1}^{\infty} H_n$, then there is an efficient randomized SEM algorithm.
- If there is an efficient learning algorithm for the graded real class $F = \bigcup_{n=1}^{\infty} F_n$, then there is an efficient randomized approximate-SEM algorithm.

Proof

It is sufficient to show that the second statement.
Resample (uniformly)z to z^* of length $m^* = m_L(n, \epsilon, 1/2)$ and get output for z^*.
This randomized approximate-SEM algorithm gives f^* satisfying w.p. at least 1/2,

$$\hat{er}_P(f^*) < opt_P(F) + \epsilon/2.$$

Since $er_P(f^*) = \hat{er}_P(f)$ and $opt_P(F) = \inf_{g \inf_n} er_P(g) = \inf_{g \inf_n} \hat{er}_z(g)$, it is an efficient randomized approximate-SEM algorithm.

Theorem 23.5 Suppose that $F = \bigcup_{n=1}^{\infty} F_n$ is a graded function class. Then F is efficiently learnable if and only if $fat_{F_n}(\alpha)$ is polynomial in n and $1/\alpha$ and there is an efficient randomized approximate-SEM algorithm for F.

Theorem 23.6 Suppose that $H = \bigcup_{n=1}^{\infty} H_n$ is a graded binary function class. Then H is efficiently learnable if and only if $VCdim(H_n)$ is polynomial in n and there is an efficient randomized SEM algorithm for F.

23.5. The Hardness of Learning

- We have seen that H can be efficiently learned only if there is an efficient randomized SEM algorithm for H.
- Checking the existence of such efficient randomized SEM algorithm may difficult.
- It is enough to confirm that a certain decision problem associated with H is NP-hard.

23.5. The Hardness of Learning

H-FIT
Instance: $z \in (R^n \times \{0,1\})^m$ and an integer k between 1 and m.
Question : Is there $h \in H_n$ such that $\hat{er}_z(h) \leq k/m$?
H-CONSISTENCY
Instance: $z \in (R^n \times \{0,1\})^m$.
Question : Is there $h \in H_n$ such that $\hat{er}_z(h) = 0$?

Theorem 23.7 Let $H = \bigcup_{n=1}^{\infty} H_n$ be a graded binary function class. If there is an efficient learning algorithm for H then there is a polynomial time randomized algorithm for H-FIT; in other words, H-FIT is in RP.

Proof

Let $\mathcal{A}$ be an efficient randomized SEM algorithm for H. Calculating $\mathcal{A}(z)$ and answering whether its sample error is less or equal than k/m is a polynomial-time randomized algorithm.

This gives 'no' if true answer is 'no', and 'yes' w.p. at least $1/2$ if true answer is 'yes'. This is the definition of solving decision problem with randomized algorithm in polynomial time.

23.5. The Hardness of Learning

Theorem 23.8 Suppose RP $\neq$ NP and that H is a graded class of binary functions. If H-FIT is NP-hard then there is no efficient learning algorithm for H.

Corollary 23.9 Suppose RP $\neq$ NP and that H is a graded class of binary functions. If H-CONSISTENCY is NP-hard then there is no efficient learning algorithm for H.

23.6.Remarks

Theorem 23.10 Suppose that $H = \bigcup_{n=1}^{\infty} H_n$ is a graded binary function class. Then H is efficiently learnable in the restricted model if and only if $VCdim(H_n)$ is polynomial in n and there is an efficient randomized consistent-hypothesis-finder for H.