An introduction to learning theory

王立威

北京大学

1. An introduction to generalization

2. Probability background

3. VC theory

4. An introduction to PAC-Bayes and Stability Theory

Outline

1. An introduction to generalization

2. Probability background

3. VC theory

4. An introduction to PAC-Bayes and Stability Theory

What is learning theory about

Study the fundamental of machine learning, like

- How to formalize machine learning problems
  we may use different framework, e.g. statistical, online, distributed
- What does it mean for a problem to be "learnable"
- How many data do we need to learn to a given accuracy
- How to build sound learning algorithms based on theory

The core of learning theory – generalization

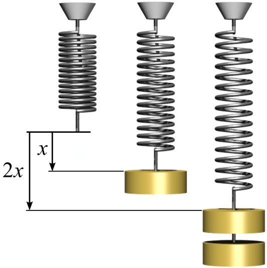
Aim: Collected data $\xRightarrow{learning}$ Model $\xRightarrow{predicting}$ Unknown data

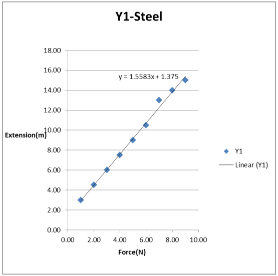
Generalization: Model should fit unknown data well, not training data!

Let us see several examples to make this idea clear.

An example

Hook’s Law: Suppose you are Hook now, and have some experimental data, what will you conclude?



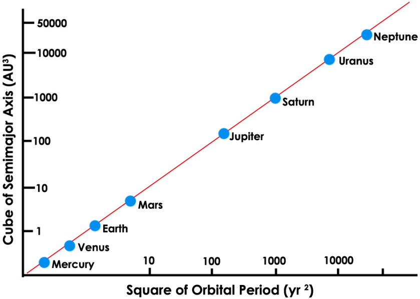


Why do we use linear function rather than complex polynomial function?

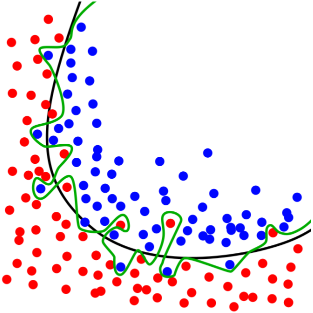
Simple and fit well for unknown data!

Some other examples

Kepler's Third Law



Learning a classifier



Ockham's Razor

William of Occam (circa 1287-1347)

"*entities should not be multiplied unncecessarily*" (in Latin)

Usually interpreted as "Among competing hypotheses, the one with the fewest assumptions should be selected"

In this lecture, we will see how to formalize the idea of generalization rigorously.

Before that, We will introduce some tools which will be used later.

Outline

1. An introduction to generalization

2. Probability background

3. VC theory

4. An introduction to PAC-Bayes and Stability Theory

Probability inequality

Two basic inequalities:

1. **Markov's inequality:** Random variable $X \geqslant 0$, then for any $a > 0$,there is

$$\mathsf{Pr}(X \geqslant a) \leqslant \frac{\mathbf{E}[X]}{a}$$

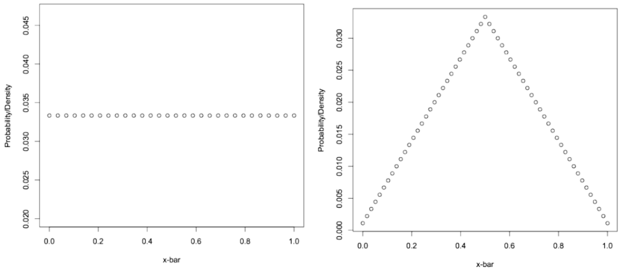
2. **Chebyshev's inequality:** Random variable X, if $\mathbf{E}[X]$ is bounded, then for any $a > 0$,there is

$$\mathsf{Pr}(X - \mathbf{E}[X] \geqslant a) \leqslant \frac{\mathsf{Var}[X]}{a^2}$$

Problem 1: Prove above two inequalities.

Law of large numbers

An intuitive example: If X and Y are uniform in $[0, 1]$, then $Z = \dfrac{X+Y}{2}$ is triangular:



A kind of **concentration** property!

Law of large numbers

Suppose $X_1, X_2, \cdots$ are i.i.d. random variables
$\mu = \mathbf{E}[X_i] < \infty, \sigma^2 = \mathrm{Var}[X_i] < \infty$
Let $\bar{X}_n = \frac{1}{n} \sum_{i=1}^{n} X_i$

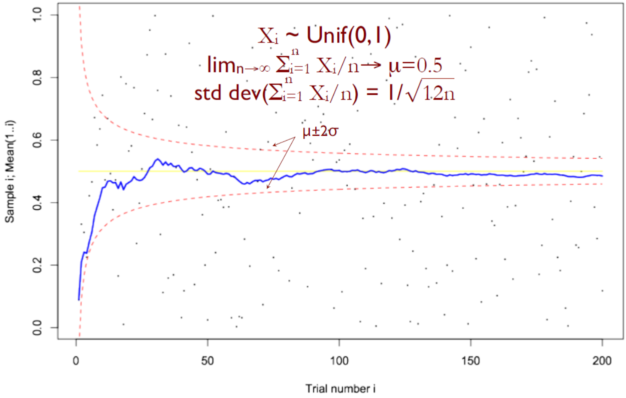
The weak law of large numbers (i.i.d. case)

For any $\epsilon > 0$, as $n \to \infty$

$$\Pr\left(|\bar{X}_n - \mu| > \epsilon\right) \longrightarrow 0$$

A simulation result

Consider toss a fair coin, consider the sample mean:



How fast does sample mean converge to the ture mean given error?

From Chebyshev's inequality, we know it converges as order $O(\frac{1}{n})$.

Can we obtain better result?

Central limit theorem

Suppose $X_1, X_2, \cdots$ are i.i.d. random variables
$\mu = \mathbf{E}[X_i] < \infty, \sigma^2 = \mathrm{Var}[X_i] < \infty$
Let $\bar{X}_n = \dfrac{1}{n}\displaystyle\sum_{i=1}^{n} X_i$

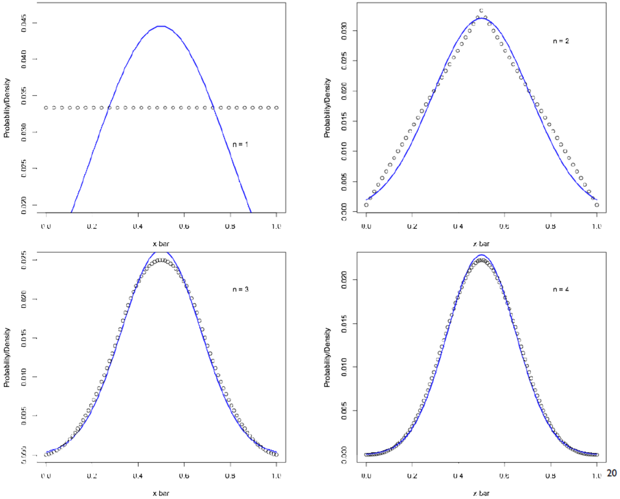
Central limit theorem

As $n \to \infty$, there is

$$\bar{X}_n \to N\left(\mu, \frac{\sigma^2}{n}\right)$$

A simulation result

If X_i are i.i.d. uniform in $[0, 1]$, then the distribution of $\bar{X}_n$:



Concentration inequality

Concentration inequality: provide bounds on how a random variable deviates from some value (e.g. expectation)
Chernoff Bound: Suppose $X_1, X_2, \ldots, X_n$ are i.i.d random variables and $p = \mathsf{Pr}(X_i = 1)$, then for any samll $\epsilon > 0$, there is

$$\mathsf{Pr}[\bar{X} - \mathbf{E}[X] \geqslant \epsilon] \leqslant \exp(-nD(p+\epsilon||p)) \leqslant \exp(-2n\epsilon^2)$$
$$\mathsf{Pr}[\bar{X} - \mathbf{E}[X] \leqslant -\epsilon] \leqslant \exp(-nD(p-\epsilon||p)) \leqslant \exp(-2n\epsilon^2)$$

where $\bar{X} = \frac{1}{n}\sum X_i, D(a||b) = a \ln \frac{a}{b} + (1-a)\ln\frac{1-a}{1-b}$

Compared with Chebyshev's inequality, the convergence rate now improved to $\boldsymbol{O(e^{-n})}$ from $\boldsymbol{O\left(\frac{1}{n}\right)}$

Concentration inequality

Two generalizations of Chernoff Bound:

1. **Chernoff bound for i.i.d. r.v.s in [0,1]:** If $X_1, X_2, \ldots, X_n$ are i.i.d r.v.s and $X_i \in [0,1], p = \mathbf{E}(X_i)$, then for any samll $\epsilon > 0$, there is

$$\text{Pr}[|\bar{X} - p| \geqslant \epsilon] \leqslant 2e^{-2n\epsilon^2}$$

2. **Chernoff bound for only independent r.v.s in [0,1]:** If $X_1, X_2, \ldots, X_n$ are independent r.v.s and $X_i \in [0,1]$, $p_i = \mathbf{E}(X_i), p = \frac{1}{n}\sum p_i$, then for any samll $\epsilon > 0$, there is

$$\text{Pr}[|\bar{X} - p| \geqslant \epsilon] \leqslant 2e^{-2n\epsilon^2}$$

Concentration inequality

Hoeffding's inequality

Let $X_1, X_2, \ldots, X_n$ are i.i.d r.v.s and $X_i \in [a_i, b_i]$, let $S_n = \sum X_i$, then for any $\epsilon > 0$, there is

$$\Pr[S_n - \mathbf{E}[S_n] \geqslant \epsilon] \leqslant e^{-\frac{2\epsilon^2}{\sum(b_i - a_i)^2}}$$

$$\Pr[S_n - \mathbf{E}[S_n] \leqslant -\epsilon] \leqslant e^{-\frac{2\epsilon^2}{\sum(b_i - a_i)^2}}$$

Proof sketch:

$$\begin{aligned}
\Pr[S_n - \mathbf{E}[S_n] \geqslant \epsilon] &\leqslant e^{-t\epsilon}\mathbf{E}[e^{t(S_n - \mathbf{E}[S_n])}] \\
&= \prod e^{-t\epsilon}\mathbf{E}[e^{t(X_i - \mathbf{E}[X_i])}] \\
&\leqslant \prod e^{-t\epsilon} e^{t^2(b_i - a_i)^2/8} \quad (\because \text{Hoeffding's lemma}) \\
&= e^{-t\epsilon} e^{t^2 \sum(b_i - a_i)^2/8} \\
&\leqslant e^{-\frac{2\epsilon^2}{\sum(b_i - a_i)^2}}
\end{aligned}$$

Outline

1. An introduction to generalization

2. Probability background

3. VC theory

4. An introduction to PAC-Bayes and Stability Theory

Basic notation – Data

- Input space/ Feature space: $\mathcal{X}$
  (e.g. various measures of an object, bag-of-words, vector of grey-scale values...)

  > Feature extraction itself is an important area in ML, is also an art, but we won't cover in this lecture.

- Output space/ Label space: $\mathcal{Y}$
  (e.g. $\{\pm 1\}, [K], \mathbb{R}-$valued output...)
- Assume an underlying unknown concept $c : \mathcal{X} \mapsto \mathcal{Y}$, which is the ture labeling function.
  A concept class is a set of concepts we want to learn, denoted by $\mathcal{C}$

Basic notation – Model

- **How is data generated?** data $x \in \mathcal{X}$ obeys some fixed but unknown distribution D
- **What do we observe?** a data set $S = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, each x_i is i.i.d according to D, and $y_i = h^*(x_i)$
- **How to measure performance or success?**
  Loss function: $\ell : \mathcal{Y} \times \mathcal{Y} \mapsto \mathbb{R}$
  (e.g. 0-1 loss $\ell(y', y) = 1\{y' \neq y\}$, sq-loss $\ell(y', y) = (y' - y)^2$...)
  y': predicting label, y: ture label.
- **Where do we place our prior assumption or model assumptions?**
  Model class/ Hpothesis class $\mathcal{H} \subset \mathcal{Y}^{\mathcal{X}}$
  (e.g. $\mathcal{H} = \{x \mapsto \text{sign}(f^T x)\}, \mathcal{H} = \{x \mapsto f^T x : \|f\|_2 \leqslant 1\}$)

In what next, we will fouce on the binary classification case

Generalization and empirical error

Note $\mathcal{Y} = \{0, 1\}$ now.

Generalization error

Given an $h \in \mathcal{H}$, a ture labeling function $c \in \mathcal{C}$ (Note $\mathcal{C}$ may be equal to $\mathcal{H}$ or not), an underlying distribution D, the generalization error or risk of h is defined by

$$R(h) = \mathsf{Pr}_{x \sim D}[h(x) \neq c(x)] = \mathbf{E}[1\{h(x) \neq c(x)\}]$$

Empirical error

Given an $h \in \mathcal{H}$, a ture labeling function $c \in \mathcal{C}$, a data set $S = \{x_1, \ldots, x_n\}$, the empirical error of h is defined by

$$\hat{R}_S(h) = \frac{1}{n} \sum_{i=1}^{n} 1\{h(x_i) \neq c(x_i)\}$$

When fix h, there is $\mathbf{E}[\hat{R}_S(h)] = R(h)$.

Probably Approximately Correct (PAC) learning framework

Our goal: make generalization error as small as possible!

PAC-learning

A concept class $\mathcal{C}$ is said to be PAC-learnable if there exists an algorithm $\mathcal{A}$ and a polynomial fucntion $poly(\cdot, \cdot)$, s.t. for any $\epsilon, \delta > 0$, any D on $\mathcal{X}$, any $c \in \mathcal{C}$, the following holds for any sample size $n \geqslant poly(\frac{1}{\epsilon}, \frac{1}{\delta})$:

$$\mathrm{Pr}_{S \sim D^n}[R(h_S) \leqslant \epsilon] \geqslant 1 - \delta$$

where $h_S = \mathcal{A}(S)$.

Further, if $\mathcal{A}$ runs in $poly(\frac{1}{\epsilon}, \frac{1}{\delta})$, then C is said to be efficiently PAC-learnable.

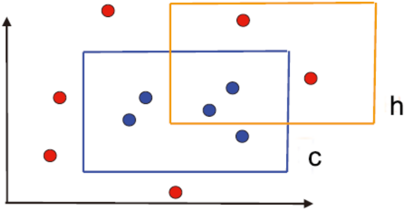
Several key points about PAC framework

1. It is a distribution free model, i.e. no particular assumption about D;
2. Training and test samples are drawn according to the same D (otherwise transfer learning);
3. It deals with the question of learnability for $\mathcal{C}$ not a particular concept.

A specific example

Learning axis-aligned rectangles:

- Data: points in the plane, $\mathcal{X} = \mathbb{R}^2$
- Concept class $\mathcal{C}$: all the axis-aligned rectangles lying in $\mathbb{R}^2$
- Points in the ture concept labeled with 1
- Assume $\mathcal{H} = \mathcal{C}$



For the figure, c represents the ture concept, h represents a hypothesis.

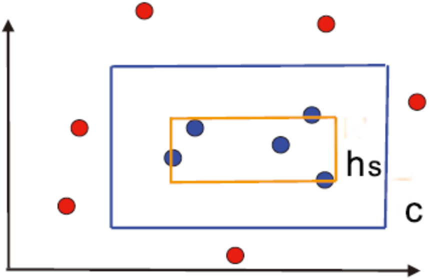
Where is the error region? fasle negatives and false positives.

Now, we will show this concept class is PAC-learnable.
How to? Give one PAC-learning algorithm!

A specific example

Our algorithm: For a data set S, return the tightest axis-aligned rectange h_S containing points labeled with 1.

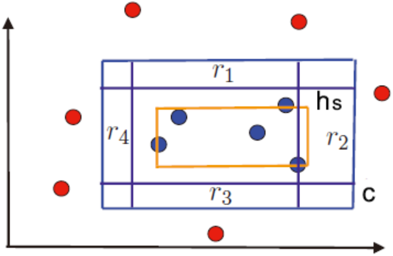
- h_S do not produce any false positives;
- Error region of h_S is included in c and out of h_S
- W.L.G., assume $\Pr(c) > \epsilon$ (here $\Pr(c)$ means the probability of area in c), otherwise $R(h_S) \leqslant \Pr(c) \leqslant \epsilon$



A specific example

Now, as $\Pr(c) > \epsilon$, define four rectangular regions r_1, r_2, r_3, r_4 along the sides of c, s.t. for each $i \in [4]$, there is $\Pr(r_i) > \epsilon/4$. Then

$$
\begin{aligned}
& \Pr_{S\sim D^n}[R(h_S) > \epsilon] \\
\leqslant & \Pr_{S\sim D^n}[\cup_{i=1}^4 \{h_S \cap r_i = \varnothing\}] \\
\leqslant & \sum_{i=1}^4 \Pr_{S\sim D^n}[\{h_S \cap r_i = \varnothing\}] \\
\leqslant & 4(1-\epsilon/4)^n \quad (\because \Pr(r_i) > \epsilon/4) \\
\leqslant & 4\exp(-n\epsilon/4) \quad (\because 1 - x \leqslant e^{-x})
\end{aligned}
$$



Thus, in order to have $\Pr_{S\sim D^n}[R(h_S) > \epsilon] \leqslant \delta$, we can impose

$$4\exp(-n\epsilon/4) \leqslant \delta \iff n \geqslant \frac{4}{\epsilon}\log\frac{4}{\delta}$$

The idea of uniform generalization

We want to learn $\operatorname{argmin}_{h\in\mathcal{H}} R(h)$, but trouble is we do not know underlying distribution D, thus no explicit form for $R(h)$.

A straightforward indicator of $R(h)$ is $\hat{R}_S(h)$, and according to Law of Large Number, we know $\hat{R}_S(h)$ is a good approximation of $R(h)$ when n is large enough.

So is it a good choice to choose h as $\operatorname{argmin}_{h\in\mathcal{H}} \hat{R}(h)$?

However, **this is not a good choice**. As we can easily choose h from $\mathcal{H}$, s.t. $\hat{R}_S(h) = 0$, but furture performance can be very poor...

The idea of uniform generalization

Where is it wrong with above idea?

Well, mainly two reasons:

1. we choose h from $\mathcal{H}$, which may be **too complex**, thus has a bad influence over future unknown examples;
2. the chosen h is **data dependent**, i.e. it just suits training examples well, has no guarantee for future performance.

Therefore, we want to bound the difference between them, i.e.

$$\mathsf{Pr}(R(h) \leqslant \hat{R}_S(h) + \textbf{?}) \geqslant 1 - \delta$$

Note, probability is over the training sampls.

The idea of uniform generalization

We may have guessed, the **?** term in above inequality may be dependent on the measurement of the complexity of $\mathcal{H}$. Now, we will make it clear.

Note, if h is fixed, we simply have $\Pr(R(h) \geqslant \hat{R}_S(h) + \epsilon) \leqslant e^{-2n\epsilon^2}$

But h is dependent of S

So we want a generalization bound uniformly for all $h \in \mathcal{H}$!

We can guess it is easier when $|\mathcal{H}|$ is finite (as we have union bound), and much harder when it is infinite.

Generalization bound – Consistent and Finite hypothesis

Note consistent means $\mathcal{C} \subset \mathcal{H}$, inconsistent means $\mathcal{C} \neq \mathcal{H}$

Learning bound - finite $\mathcal{H}$, consistent case

Suppose $\mathcal{H}$ is finite. $\mathcal{A}$ is an algorithm that for any $c \in \mathcal{H}$ and i.i.d sample S returns a consistent hypothesis $h_S : \hat{R}(h_S) = 0$. Then for any $\epsilon > 0$,

$$\Pr(R(h_S) \geqslant \epsilon) \leqslant |\mathcal{H}|(1-\epsilon)^n$$

Proof sketch:

$$\begin{aligned}\Pr(R(h_S) \geqslant \epsilon) &\leqslant \Pr\left(\exists h \in \mathcal{H}, \hat{R}_S(h) = 0 \cap R(h) > \epsilon\right) \\ &\leqslant \sum_{h \in \mathcal{H}} \Pr(\hat{R}_S(h) = 0 \cap R(h) > \epsilon) \\ &\leqslant \sum_{h \in \mathcal{H}} \Pr(\hat{R}_S(h) = 0 | R(h) > \epsilon) \\ &\leqslant |\mathcal{H}|(1-\epsilon)^n\end{aligned}$$

which also means $\Pr\left(R(h_S) \leqslant \frac{1}{n}\left(\log|\mathcal{H}| + \log\frac{1}{\delta}\right)\right) \geqslant 1-\delta$

Generalization bound – Inconsistent and Finite hypothesis

Learning bound - finite $\mathcal{H}$, inconsistent case

Suppose $\mathcal{H}$ is a finite hypothesis set. Then for any $\epsilon > 0$, there is

$$\Pr(\forall h \in \mathcal{H}, |R(h) - \hat{R}_S(h)| \leqslant \epsilon) \geqslant 1 - 2|\mathcal{H}|e^{-2n\epsilon^2}$$

Proof sketch:

$$\begin{aligned}
&\Pr\left(\exists h \in \mathcal{H}, |R(h) - \hat{R}_S(h)| > \epsilon\right) \\
=&\Pr\left((|R(h_1) - \hat{R}_S(h_1)| > \epsilon) \cup \cdots \cup (|R(h_{|\mathcal{H}|}) - \hat{R}_S(h_{|\mathcal{H}|})| > \epsilon)\right) \\
\leqslant&\sum_{h \in \mathcal{H}} \Pr(|R(h) - \hat{R}_S(h)| > \epsilon) \\
\leqslant&2|\mathcal{H}|e^{-2n\epsilon^2}
\end{aligned}$$

$$\iff \Pr\left(\forall h \in \mathcal{H}, |R(h) - \hat{R}_S(h)| \leqslant \sqrt{\frac{\log|\mathcal{H}| + \log\frac{2}{\delta}}{2n}}\right) \geqslant 1 - \delta$$

Note accuracy decreases from $O(1/n)$ to $O(1/\sqrt{n})$ in inconsistent case.

Generalization bound – Infinite hypothesis

What if $|\mathcal{H}|$ is infinite? As union bound is no more use..
The critical part is to measure the complexity of $\mathcal{H}$!

Suppose giving training samples $S = \{x_1, \ldots, x_n\}$, then we consider

$$\mathcal{H}_{x_1,\ldots,x_n} = \{h(x_1), \ldots, h(x_n) : h \in \mathcal{H}\}$$

Since h only takes $\{0, 1\}$, $\mathcal{H}_{x_1,\ldots,x_n}$ will be finite, no matter how big $\mathcal{H}$ is.

Growth Function

Growth function is defined as $N^{\mathcal{H}}(n) := \max_{x_1,\ldots,x_n} |\mathcal{H}_{x_1,\ldots,x_n}|$

It is easy to see $N^{\mathcal{H}}(n) \leqslant 2^n$.

VC-dimension

Though growth function function is a measurement for the complexity of $\mathcal{H}$, but it is still a function related to n, we want an index only related to $\mathcal{H}$.

Note, if $N^{\mathcal{H}}(n) = 2^n$, means there is a dataset of size n, s.t. $\mathcal{H}$ can generate any classification on these points. (say $\mathcal{H}$ shatters the set)

Thus, there we can define:

VC-dimension

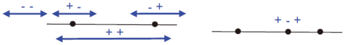
The VC-dimension of $\mathcal{H}$ is defined as follows:

$$\text{VC-dim}(\mathcal{H}) = \max\{n : N^{\mathcal{H}}(n) = 2^n\}$$

Some examples about VC-dimesnion

Example 1

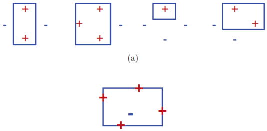
$\mathcal{H}$={intervals on the real line}, VC-dim($\mathcal{H}$) = 2



(a) (b)

Example 2

$\mathcal{H}$={axis-aligned rectangles on $\mathbb{R}^2$}, VC-dim($\mathcal{H}$) = 4



(a)

(b)

Problem 2: Prove the VC-dimension of the set of hyperplaces in $\mathbb{R}^d$ is $d + 1$.

Relation between VC-dimension and the growth function

Theorem, Sauer's lemma

Assume d is the VC-dimension of $\mathcal{H}$,then for all n, there is

$$N^{\mathcal{H}}(n) \leqslant \sum_{i=0}^{d} \binom{n}{i} \leqslant \left(\frac{en}{d}\right)^{d}$$

Proof sketch:

The lefthand inequality: once $n \leqslant d$, it obviously holds;
If $n > d$, for an intuitive view, assume $(h(x_1), \ldots, h(x_{d+1})) \neq 0^{d+1}$, which means there are at most d 0s in $(h(x_1), \ldots, h(x_n))$, thus

$$N^{\mathcal{H}}(n) \leqslant \sum_{i=0}^{d} \binom{n}{i}.$$

Proof of Sauer's lemma

Now we give the proof for a concrete example, and it is easy to obtain for general case.
Assume $n = 4, d = 2$, and the avoid set for $\mathcal{H}$ is

$$\left\{\begin{matrix} 001*, \\ 01*0, \\ 1*01, \\ *101 \end{matrix}\right\}.$$

And we can prove that

$$\left|\left\{\begin{matrix} 001*, \\ 01*0, \\ 1*01, \\ *101 \end{matrix}\right\}\right| \geq \left|\left\{\begin{matrix} 001*, \\ 01*0, \\ 0*01, \\ *101 \end{matrix}\right\}\right| \geq \left|\left\{\begin{matrix} 001*, \\ 00*0, \\ 0*01, \\ *001 \end{matrix}\right\}\right| \geq \left|\left\{\begin{matrix} 000*, \\ 00*0, \\ 0*01, \\ *001 \end{matrix}\right\}\right| \geq \left|\left\{\begin{matrix} 000*, \\ 00*0, \\ 0*00, \\ 0000 \end{matrix}\right\}\right|.$$

The above inequalities follow from the fact: if we set one column the same number (say 0) for non-star entries, the new set will not be bigger.

Proof of Sauer's lemma

The righthand inequality:

$$
\begin{aligned}
\sum_{i=0}^{d} \binom{n}{i} &\leqslant \sum_{i=0}^{d} \binom{n}{i} \left(\frac{n}{d}\right)^{d-i} \\
&\leqslant \sum_{i=0}^{n} \binom{n}{i} \left(\frac{n}{d}\right)^{d-i} \\
&= \left(\frac{n}{d}\right)^{d} \sum_{i=0}^{n} \binom{n}{i} \left(\frac{d}{n}\right)^{i} \\
&= \left(\frac{n}{d}\right)^{d} \left(1 + \frac{d}{n}\right)^{n} \\
&\leqslant \left(\frac{en}{d}\right)^{d} \qquad (\because (1 - x) \leqslant e^{-x})
\end{aligned}
$$

VC-Generalization bound

Through VC-dimension and growth function, we can obtain generalization bound for infinite case:

VC-Generalization bound

Over random draw of $S = \{x_1, \ldots, x_n\}$, with probability at least $1 - \delta$, there is

$$\forall h \in \mathcal{H}, R(h) \leqslant \hat{R}_S(h) + \mathcal{O}\left(\sqrt{\frac{d \log\left(\frac{n}{d}\right) + \log \frac{1}{\delta}}{n}}\right)$$

Thus for several $\mathcal{H}$, if their empirical error is close, then the simpler about $\mathcal{H}$, the better about generalization

A corollary of VC-generalization bound

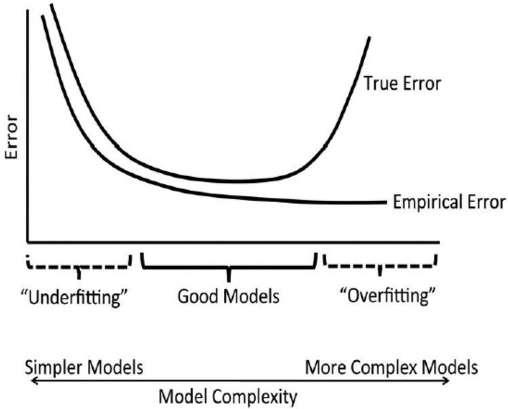
Generalization ability of ERM algorithm

Let $\mathcal{X}$ be the instance space, $\mathcal{Y} = \{0,1\}$. For any learning problem (i.e. $\forall D$ over $\mathcal{X}$), the Empirical Risk Minimization (ERM) algorithm learns a classifier $\hat{h}_S = \mathrm{argmin}_{h\in\mathcal{H}} \hat{R}_S(h)$, and let $h^* = \mathrm{argmin}_{h\in\mathcal{H}} R(h)$. Then with probability at least $1-\delta$ over the random draw of a training set S of n examples, there is

$$R(\hat{h}_S) \leqslant R(h^*) + \mathcal{O}\left(\sqrt{\frac{d\log\left(\frac{n}{d}\right) + \log\frac{1}{\delta}}{n}}\right)$$

Problem 3: Prove the above corollary.

An overview of generalization bound



We can see, small empirical error does not represent small generalization error, we also need consider the model complexity!

The idea of regularization

Based on above theory, how to design algorihtms?

1. ERM (Empirical Risk Minimization) algorithm:

$$h_S^{ERM} = \mathrm{argmin}_{h \in \mathcal{H}} \hat{R}_S(h)$$

   Regardless of complexity term, poor performance; computationally intractable

2. SRM (Structual Risk Minimization) algorithm: $\mathcal{H}_0 \subset \cdots \subset \mathcal{H}_k \subset \cdots$

$$h_S^{SRM} = \mathrm{argmin}_{h \in \mathcal{H}_k} \hat{R}_S(h) + \text{complexity}(\mathcal{H}_k, n)$$

   Computationally expensive, since it requires multiple ERM problems

3. Regularization methods:

$$h_S^{REG} = \mathrm{argmin}_{h \in \mathcal{H}} \hat{R}_S(h) + \lambda R(h)$$

   where $R(h)$ represents the complexity term, and $\lambda \geqslant 0$ is a regularization parameter.

Common Regularization Methods

L2-Regularization:

$$h_S^{L2} = \mathrm{argmin}_{h \in \mathcal{H}} \hat{R}_S(h) + \lambda \|h\|_2^2$$

Example: SVM

$$\min_{\mathbf{w},b} \frac{1}{n} \sum_{i=1}^{n} [1 - y_i(\mathbf{w} \cdot \mathbf{x_i} + b)]_+ + \frac{\lambda}{2} \|\mathbf{w}\|_2^2$$

L1-Regularization:

$$h_S^{L1} = \mathrm{argmin}_{h \in \mathcal{H}} \hat{R}_S(h) + \lambda \|h\|_1$$

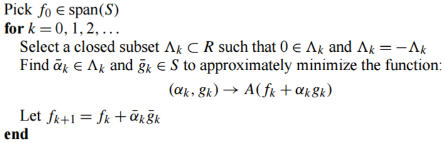
Example: LASSO

$$\min_{\mathbf{w},b} \frac{1}{n} \sum_{i=1}^{n} (\mathbf{w} \cdot \mathbf{x_i} + b - y_i)^2 + \frac{\lambda}{2} \|\mathbf{w}\|_1$$

Early Stopping in Boosting

Boosting: A family of algorithms whcih combine **weak learners** to produce a **strong learner**

General boosting framework



```
Pick $f_0 \in \text{span}(S)$
for $k = 0, 1, 2, \ldots$
    Select a closed subset $\Lambda_k \subset R$ such that $0 \in \Lambda_k$ and $\Lambda_k = -\Lambda_k$
    Find $\bar{\alpha}_k \in \Lambda_k$ and $\bar{g}_k \in S$ to approximately minimize the function:
        $(\alpha_k, g_k) \to A(f_k + \alpha_k g_k)$
    Let $f_{k+1} = f_k + \bar{\alpha}_k \bar{g}_k$
end
```

where S is the set of basic learners (or say hypothesis set $\mathcal{H}$), and A is the loss.

Concrete boosting algorithm: AdaBoost, Gradient Boosting etc...

When $\mathcal{H}$ is fixed, $\sum |\bar{\alpha}_k|$ can be seen as the complexity of the final model.

$$\text{Early stopping} \iff \text{regularization} \sum |\bar{\alpha}_k| \leqslant B \text{ in some sense}$$

Outline

1. An introduction to generalization

2. Probability background

3. VC theory

4. An introduction to PAC-Bayes and Stability Theory

Frequentist & Bayesian

Suppose: D, collected data; θ, unknown parameters of the model
Frequentist: Regard θ as the fixed constant

$$\text{Maximum Likelihood:} \max_{\theta} P(D|\theta)$$

Bayesian: Regard θ as a random variable

$$\text{Maximum A Posterior:} \max_{\theta} P(\theta|D)$$

From Bayesian's view, after observing data, what we know is a distribution Q over all classifiers $h \in \Omega$

With Q, there are two types of classifiers:

- Stochastic classifier h_Q, with generalization error $R(h_Q) = \mathbf{E}_Q[R(h)]$
- Voting classifier g_Q, with generalization error $R(g_Q) = R(\mathbf{E}_Q h)$

One can prove: $R(g_Q) \leqslant 2R(h_Q)$

PAC-Bayes Theorem

Similar with VC-generalization bound, there is a Bayesian version.

PAC-Bayes Theorem

For any prior P, with probability $1-\delta$, for all distributions Q over the hypothesis space, there is

$$R(h_Q) \leqslant \hat{R}(h_Q) + \sqrt{\frac{D(Q||P) + \log 3/\delta}{n}}$$

where $\hat{R}(h_Q)$ is the empirical error of stochastic classifier, and $D(Q||P) := \mathbf{E}_{h\sim Q} \log \frac{Q(h)}{P(h)}$ is the relative entropy between Q, P

Note in PAC-Bayes theorem, it does not contain any complexity measurement of the hypothesis space, like VC-dimension.

Algorithmic Stability

Motivation:

1. All of previous generalization bounds ignore concrete **learning algorithm**
2. Using algorithmic dependent analysis could lead to better guarantees

Firstly, we need define some general properties w.r.t. learning algorithms

Definition: ϵ-uniformly stable

A randomized algorithm A is ϵ-uniformly stable if for all data sets $S, S' \in Z^n$, such that S and S' differ in at most one example, we have

$$\forall z, \quad \mathbf{E}_A[\ell(A(S); z) - f(\ell(S'); z)] \leqslant \epsilon$$

where $\ell(\cdot; z)$ is the loss function w.r.t. data z

The infimum over all satisfied ϵ is denoted as $\epsilon_{stab}(A, n)$.

Main Theorem of Algorithmic Stability Theory

Theorem: Generalization in expectation

Suppose A is ϵ-uniformly stable, then

$$\left|\mathbf{E}_{S,A}[R[A(S)] - \hat{R}_S[A(S)]]\right| \leqslant \epsilon$$

A concrete example: stability of SGD for convex losses (Hardt et al. ICML2016)

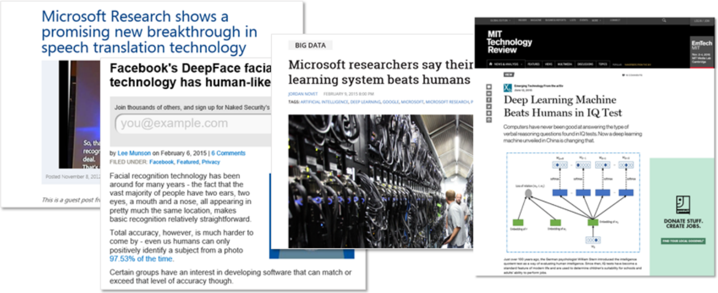
Assume $f(\cdot; z) \in [0, 1]$ is β-smooth, convex and L-Lipschitz for every z. Suppose we run SGD with step size $\alpha_t \leqslant 2/\beta$ for T steps. Then, SGD satisfies uniform stability with

$$\epsilon_{stab} \leqslant \frac{2L^2}{n} \sum_t \alpha_t$$

If $f(\cdot, z)$ is further γ-strongly convex, then $\epsilon_{stab} \leqslant \dfrac{2L^2}{\gamma n}$

Mystery of Deep Learning

Deep learning has achieved huge success in real world.



Microsoft Research shows a promising new breakthrough in speech translation technology

Posted November 8, 2012

This is a guest post fr

Facebook's DeepFace facia technology has human-like

Join thousands of others, and sign up for Naked Security's

you@example.com

by Lee Munson on February 6, 2015 | 6 Comments

FILED UNDER: Facebook, Featured, Privacy

Facial recognition technology has been around for many years - the fact that the vast majority of people have two ears, two eyes, a mouth and a nose, all appearing in pretty much the same location, makes basic recognition relatively straightforward.

Total accuracy, however, is much harder to come by - even us humans can only positively identify a subject from a photo 97.53% of the time.

Certain groups have an interest in developing software that can match or exceed that level of accuracy though.

BIG DATA

Microsoft researchers say their learning system beats humans

JORDAN NOVET FEBRUARY 9, 2015 8:00 PM

TAGS: ARTIFICIAL INTELLIGENCE, DEEP LEARNING, GOOGLE, MICROSOFT, MICROSOFT RESEARCH, P

MIT Technology Review

Deep Learning Machine Beats Humans in IQ Test

Computers have never been good at answering the type of verbal reasoning questions found in IQ tests. Now a deep learning machine unveiled in China is changing that.

Just over 100 years ago, the German psychologist William Stern introduced the intelligence quotient test as a way of evaluating human intelligence. Since then, IQ tests have become a standard feature of modern life and are used to determine children's suitability for schools and adults' ability to perform jobs.

In 2015, Lecun: "What's wrong with deep learning"

ICLR 2017 Best Paper:

Understanding deep learning requires rethinking generalization

Mystery of Deep Learning

Many interesting distinctions about DNN:

- number of parameters $\gg$ number of data
- Easy to fit data, even for random label or random feature

Traditional learning theory fails to explain:

For deep neural nets, $|E|$: number of parameters, L: number of layers

- **VC-dimension** (Shalev-Schwartz 2014): $O(|E|\log|E|)$
- Other complexity measurements, like covering number, has exponential dependence on L

All these measurements are far beyond the number of data points!

For stability theory, best result for non-convex but β smooth loss function:

$$\epsilon_{stab} \leqslant O\left(\frac{T^{1-1/(\beta c+1)}}{n}\right) \quad \text{(for SGM)}$$

which maybe linear dependent on training iterations!

Stochastic Gradient Langevin Dynamics

Suppose $F_S(w)$ is the empirical loss function,
$\mathbf{E}[g_k] = \nabla F_S(w_k), \xi_k \sim \mathcal{N}(0, I_d)$

SGLD

$$W_{k+1} = W_k - \eta g_k + \sqrt{\frac{2\eta}{\beta}}\xi_k$$

One can connect it with following stochastic differential equation

$$dW(t) = -\nabla F_z(W(t))dt + \sqrt{\frac{2}{\beta}}dB(t)$$

Consider an equivalent process with such evolution, w.r.t. density function

$$\frac{\partial \pi}{\partial t} = \nabla \cdot \left(\frac{1}{\beta}\nabla \pi + \pi g\right)$$

Our Results

For SGLD algorithm, from view of the PAC-Bayes theory,

PAC-Bayes Theory for SGLD w.r.t Non-convex Loss

For ERM with regularization term $\frac{\lambda}{2}\|w\|^2$. Let w_N be the result of SGLD at N-th round. Under mild conditions, with high probability, we have

$$R(w_N) - \hat{R}_S(w_N) \leqslant O\left(\sqrt{\frac{\beta}{n}\sum_{k=1}^{N}\eta_k e^{-\frac{\lambda}{2}(T_N - T_k)}\mathbf{E}\|g_k\|^2}\right)$$

where $T_k = \sum_{i=1}^{k}\eta_i$

Our Results

For SGLD algorithm, from view of the stability theory,

Stability Theory for SGLD w.r.t Non-convex Loss

For ERM problem, assume each $f_i(\cdot)$ is L-Lipschitz. Let w_N be the result of SGLD at N-th round. Under mild conditions, we have

$$\mathbf{E}[R(w_N) - \hat{R}_S(w_N)] \leqslant O\left(\frac{1}{n}\left(k_0 + L\sqrt{\beta \sum_{k=k_0+1}^{N} \eta_k}\right)\right)$$

where the expectation is taken w.r.t. random draw of training data, and $k_0 := \min\{k : \eta_k \beta L^2 < 1\}$

References

1. Foundations of Machine Learning;
2. Understanding deep learning requires rethinking generalization (ICLR'17)
3. The Expressive Power of Neural Networks: A View from the Width (NIPS'17)
4. Dropout Training, Data-dependent Regularization, and Generalization Bounds (ICML'18)
5. To What Extent Do Different Neural Networks Learn the Same Representation: A Neuron Activation Subspace Match Approach (NIPS'18)
6. Generalization bounds of SGLD for non-convex learning: two theoretical viewpoints (COLT'18)
7. Quadratic Upper Bound for Recursive Teaching Dimension of Finite VC Classes (COLT'17)