THE UNIVERSITY OF SYDNEY

Advanced Machine Learning

(COMP 5328)

Hypothesis Complexity and Generalisation

Tongliang Liu

THE UNIVERSITY OF SYDNEY

Review

Best classifier

THE UNIVERSITY OF SYDNEY

- The best classifier can be mathematically defined as:

$$\arg\min_{h} \mathbb{E}[1_{\{Y \neq \text{sign}(h(X))\}}]$$

- The objective function is not convex or smooth, hard to optimise.

what is the difference between 0-1 loss & surrogate loss functions ?

Surrogate loss functions

THE UNIVERSITY OF SYDNEY

- Popular surrogate loss functions:
- Hinge loss: $\ell(X, Y, h) = \max\{0, 1 - Yh(X)\}$
- Logistic loss: $\ell(X, Y, h) = \log_2(1 + \exp(-Yh(X)))$
- Least squares loss: $\ell(X, Y, h) = (Y - h(X))^2 = (1 - Yh(X))^2$
- Exponential loss: $\ell(X, Y, h) = \exp(-Yh(X))$

Let $\phi(Yh(X)) = \ell(X, Y, h)$.

Surrogate loss functions

THE UNIVERSITY OF SYDNEY

- Not all surrogate loss functions are convex
- Cauchy loss:

$$\ell(X, Y, h) = \log_2 \left(1 + \left(\frac{1 - Yh(X)}{\sigma}\right)^2\right)$$

- Correntropy loss (Welsch loss):

$$\ell(X, Y, h) = \left(1 - \exp\left(-\left(\frac{1 - Yh(X)}{\sigma}\right)^2\right)\right)$$

Objective function

- When employing classification-calibrated surrogate loss function, the empirical estimator is unbiased:

$$h_n = \arg\min_h \frac{1}{n} \sum_{i=1}^{n} \ell(X_i, Y_i, h)$$

$$\mathbb{E}[1_{Y \neq \text{sign}(h_n(X))}] \xrightarrow{n \to \infty} \arg\min_h \mathbb{E}[1_{Y \neq \text{sign}(h(X))}]$$

h learned by surrogate function

h learned by 0-1 loss function

- If the surrogate loss function is convex and smooth, in the next, we show how to find the h!

Gradient descent method

- Unconstraint convex optimisation problem

$$\arg\min_{h\in H} \frac{1}{n}\sum_{i=1}^{n} \ell(X_i, Y_i, h) = \arg\min_{h\in H} f(h)$$

By Taylor's theorem, we have

update rule

$h_{k+1} = h_k + \eta d_k$

$$f(h_{k+1}) = f(h_k) + \eta \nabla f(h_k)^\top d_k + o(\eta)\,.$$

Design η and d_k so that

$$\nabla f(h_k)^\top d_k < 0 \quad \text{when} \quad \nabla f(h_k) \neq 0.$$

Taylor's Theorem

Let $k \geq 1$ be an integer and let the function $f : \mathbb{R} \to \mathbb{R}$ be k times differentiable at the point $a \in \mathbb{R}$. Then there exists a function $h_k : \mathbb{R} \to \mathbb{R}$ such that

$$f(x) = f(a) + f'(a)(x-a) + ... + \frac{f^{(k)}(a)}{k!}(x-a)^k + h_k(x)(x-a)^k$$

and $\lim_{x \to a} h_k(x) = 0.$

Set $k = 1$, when x is approaching a, we have

$$\begin{aligned} f(x) &= f(a) + f'(a)(x-a) + h_1(x)(x-a) \\ &= f(a) + f'(a)(x-a) + o(x-a). \end{aligned}$$

Taylor's Theorem

Set $k = 1$, we have

$$\begin{aligned} f(x) &= f(a) + f'(a)(x-a) + h_1(x)(x-a) \\ &= f(a) + f'(a)(x-a) + o(x-a). \end{aligned}$$

Note the update rule: $h_{k+1} = h_k + \eta d_k$. Let $x = h_{k+1}$ and $a = h_k$. We have

$$f(h_{k+1}) = f(h_k) + \nabla f(h_k)^\top \eta d_k + o(\eta d_k).$$

Note that η is a small value, we further have

$$f(h_{k+1}) = f(h_k) + \eta \nabla f(h_k)^\top d_k + o(\eta)$$

An iterative updating method

$$f(h_{k+1}) = f(h_k) + \eta \nabla f(h_k)^\top d_k + o(\eta).$$

Two problems:

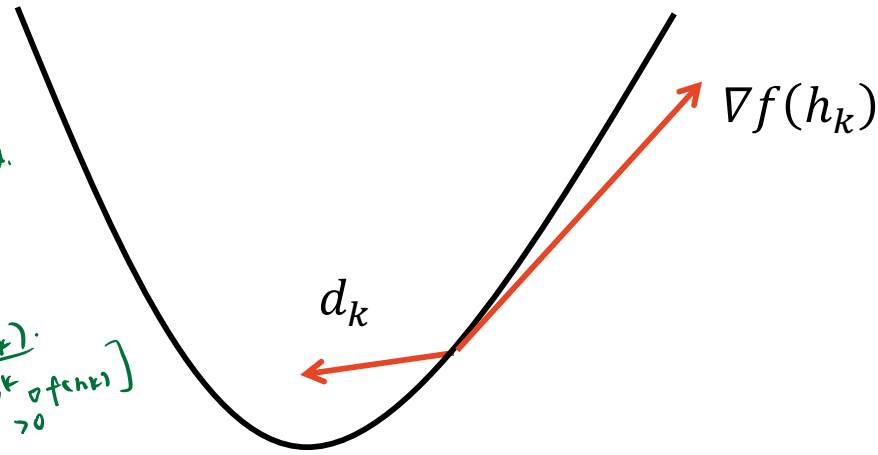
- How to design d_k?
- How to choose η?

positive definite matrix ($a^T D a > 0$ for $D \in \mathbb{R}^{d \times d}$, $a \in \mathbb{R}^d$)

$$d_k = -D^k \nabla f(h_k)$$

why?

$$\eta \nabla f(h_k)^T d_k = \eta \nabla f(h_k)^T \cdot (-D^k) \nabla f(h_k) = -\eta \left[\nabla f(h_k)^T D^k \nabla f(h_k) \right] < 0$$



Gradient convergence rate

How many iteration steps do we need to achieve the optimal solution ?

$$h_S = \arg\min_{h \in H} f(h) = \arg\min_{h \in H} \frac{1}{n} \sum_{i=1}^{n} \ell(X_i, Y_i, h)$$

Gradient convergence rate

If the objective function is strongly-convex, and has Lipschitz Gradient, we have a **linear convergence rate**, i.e., defined by

μ & positive

gradient is Lipschitz continuous

$$f(h_{k+1}) - f(h_S) \leq \left(1 - \frac{\mu}{L}\right)^k (f(h_1) - f(h_S)).$$

k: # iterations

upper bound

randomly initialization

when k is large, upper bound is small

Function f is L-Lipschitz continuous if

$$|f(x_1) - f(x_2)| \leq L\|x_1 - x_2\|, \forall x_1, x_2 \in \text{domain } f.$$

A function is μ-strongly convex:

$$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle + \frac{\mu}{2} \|x - y\|^2, \forall x, y,$$

$$\Leftrightarrow$$

$$\mu \boldsymbol{I} \preccurlyeq \nabla^2 f(x), \forall x$$

Gradient descent method

when k is large
largest $O(1/k)$
smallest $\prod_{i=1}^{k} \rho_k$

$O(1-\mu/L)^k$ v.s $\prod_{i=1}^{k} \rho_k$
base is constant
base is smaller

# of iteration ≠ # of time
time per iteration is different for each method

Algorithm	Assumption	Convergence rate
Gradient	Lipshitz Gradient, Convex	sub-linear convergence rate $O(1/k)$
Gradient	Lipshitz Gradient, Strongly-Convex	linear convergence rate $O(1-\mu/L)^k$
Newton	Lipshitz Gradient, Strongly-convex	super-linear convergence rate $\prod_{i=1}^{k} \rho_k$, $\rho_k \to 0$

$$f(h_{k+1}) - f(h_S) \leq \left(1 - \frac{\mu}{L}\right)^k (f(h_1) - f(h_S)).$$

Gradient descent method

$$f(h_{k+1}) = f(h_k) + \eta \nabla f(h_k)^\top d_k + o(\eta)$$

Set $d_k = -D^k \nabla f(h_k)$.

Newton's method sets

$$D^k = [\nabla^2 f(h_k)]^{-1}.$$

problem: time-consuming to calculate $[\nabla^2 f(h_k)]^{-1}$

Obtaining D^k my be difficult. There are many practical variants of Newton's method:

- Modify the Hessian to be positive-definite
- Only compute the Hessian every m iterations
- Only use the diagonals of the Hessian
- Quasi-Newton: Update a approximate of the Hessian (BFGS, L-BFGS)

How to deal with constrained optimisation problem?

Constrained optimisation

constrained → unconstrained

- Constrained convex optimisation problem

$$\begin{aligned} &\min_h f_0(h) \\ &\text{s.t. } f_i(h) \leq 0, i = 1, \ldots, k \\ &\quad\; g_i(h) = 0, i = 1, \ldots, l, \end{aligned}$$

where $f_0(h), f_1(h), \ldots, f_k(h)$ are convex functions, $g_i(h)$ are affine functions, i.e., $g_i(h) = a_i^\top h - b_i$.

THE UNIVERSITY OF SYDNEY

Predefined hypothesis class

Hypothesis class

Recall that a machine learning algorithm is a mapping to find a hypothesis to fit the data

$$\mathcal{A} : S \in (\mathcal{X} \times \mathcal{Y})^n \mapsto h_S \in H.$$

Here H is the predefined hypothesis class.

The mapping is an optimisation procedure that picks a hypothesis from the predefined hypothesis class to minimise or maximise the objective.

$$\arg\min_{h \in H} R_S(h).$$

empirical risk

$R_S(h) = \frac{1}{n} \sum_{i=1}^{n} l(x_i, y_i, h)$

risk training samples.

Hypothesis class

What kind of hypothesis class should we choose?

An example:

Linear SVM vs Kernel SVM

The predefined hypothesis class for Linear SVM is a set of linear functions, e.g., vectors in the Euclidean space.

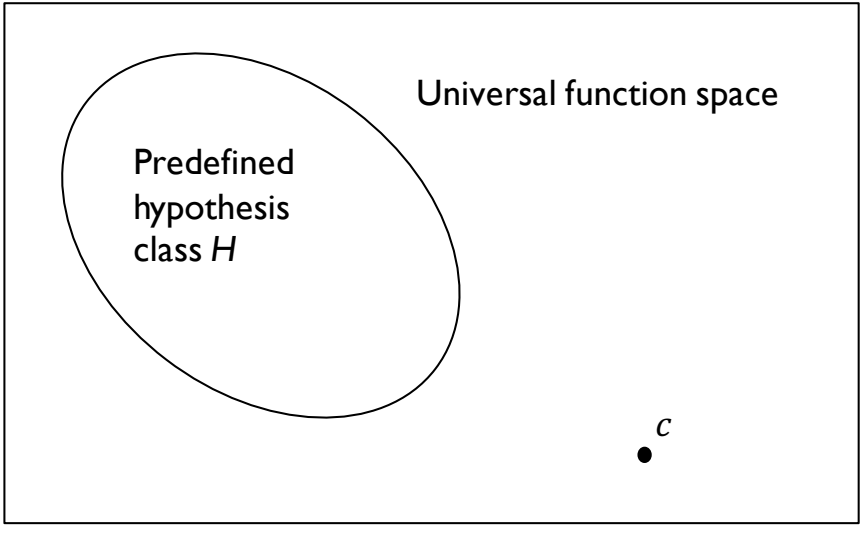
The predefined hypothesis class for Kernel SVM is a set of non-linear functions, e.g., functions in the Reproducing kernel Hilbert space (RKHS).

Which one is better?

Hypothesis class

Assume the target concept (or function) is *c*, which fits the data best, i.e., $c = \arg\min_h R(h)$.

Is the target concept (or function) *c* in the predefined hypothesis class *H*?



Notation: risks

measure of error

risk : loss on many training samples

loss : on one data point

- Empirical risk

$$R_S(h) = \frac{1}{n}\sum_{i=1}^{n} \ell(X_i, Y_i, h)$$

where $S = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$ is the training sample.

- Expected risk

$$R(h) = \mathbb{E}[R_S(h)] = \mathbb{E}[\ell(X, Y, h)]$$

Notation

- The best hypothesis in the universal function space (target concept):

$$c = \arg\min_{h} R(h).$$

h → any hypothesis

- The optimal (best) hypothesis in the predefined hypothesis class:

$$h^* = \arg\min_{h \in H} R(h).$$

since we can't have expected risk ⇒ from empirical risk

→ hypothesis must belong to predefined class

- The hypothesis we can learn from data:

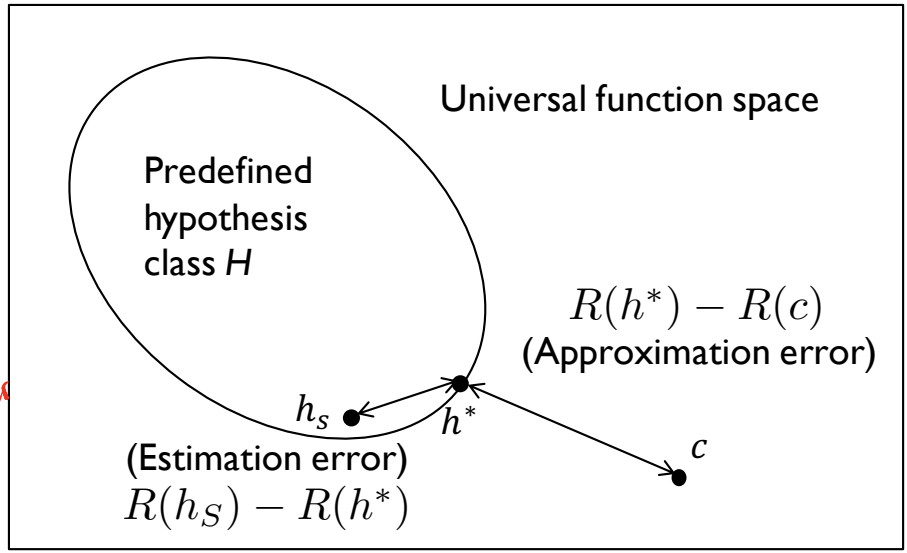
$$h_S = \arg\min_{h \in H} R_S(h) = \arg\min_{h \in H} \frac{1}{n} \sum_{i=1}^{n} \ell(X_i, Y_i, h).$$

Notation

What are the differences between c, h^*, and h_S?

Notation

What are the differences between c, h^*, and h_S?



Q: do we need a very large predefined hypothesis class

① difficult to learn approximation error maybe small but estimation error is large (need more samples to learn)

② computation cost

$R(h^*)$ is smaller

- Approximation error is caused by the difference between h^* and c
- Estimation error is cause by the difference between h_S and h^*

Hypothesis class

If the target *c* is within the predefined hypothesis class *H*, the approximation error will be zero.

It seems we should choose a large enough predefined hypothesis class to contain the target *c*. Does this help?

Large and complex hypothesis class would make it hard to learn.
The estimation error will become large!
To explain this, we need to introduce the PAC learning framework!

PAC learning framework

Probably approximately correct learning (PAC learning) is a framework for mathematical analysis of machine learning. It was proposed in 1984 by Leslie Valiant.

The PAC learning framework explains how many training examples are needed to learn the best hypothesis in the predefined class.

PAC learning framework

Definition:

event happen with high probability

A hypothesis class H is said to be PAC (probably approximately correct)-learnable if there exists an learning algorithm $\mathcal{A}$ and a polynomial function $poly(\cdot,\cdot)$ such that for any $\epsilon > 0$ and $\delta > 0$, for all distribution D on $X \times Y$, the following holds for any sample of size $n > poly(1/\delta, 1/\epsilon)$ and the hypothesis h_S learned by $\mathcal{A}$:

$$p\left\{R(h_S) - \min_{h\in H} R(h) \leq \epsilon\right\} \geq 1 - \delta.$$

$= \min_{h\in H} R(h^*)$.

h^*

δ small, probability large

ϵ small, $R(h_S)$ is close to $\min_{h\in H} R(h^*)$

$\Rightarrow$ we can learn h_S which is very close to h^* & event occurs with high probability

PAC learning framework

learned hypothesis approximately probably

$$p\left\{R(h_S) - \min_{h \in H} R(h) \leq \epsilon\right\} \geq 1 - \delta.$$

If the training sample size is large enough, e.g., $n > poly(1/\delta, 1/\epsilon)$ with a high probability, the learned hypothesis h_S can be an approximation of the best one in the predefined hypothesis class for any task.

PAC learning framework

A counterexample!

If a hypothesis class *H* is too complex, we may need exponentially many training examples, i.e., $n > exp(1/\delta, 1/\epsilon)$ to guarantee the following

$$p\left\{R(h_S) - \min_{h \in H} R(h) \leq \epsilon\right\} \geq 1 - \delta.$$

In this case, the hypothesis class *H* is not PAC-learnable.

THE UNIVERSITY OF
SYDNEY

PAC learning checking

PAC learning checking

To check if a given hypothesis class *H* is PAC learnable, we need to find a learning algorithm $\mathcal{A}$ and a polynomial function $poly(\cdot,\cdot)$ such that for any $\epsilon > 0$ and $\delta > 0$, for all distribution D on $X \times Y$, the following holds for any sample of size $n > poly(1/\delta, 1/\epsilon)$ and hypothesis h_S learned by $\mathcal{A}$:

$$p\left\{R(h_S) - \min_{h \in H} R(h) \leq \epsilon\right\} \geq 1 - \delta.$$

PAC learning checking

Recall notation

① assume loss function ℓ is convex

⇓

any optimiser get the same h_S.

$$R_S(h) = \frac{1}{n}\sum_{i=1}^{n} \ell(X_i, Y_i, h)$$

$$R(h) = \mathbb{E}[R_S(h)] = \mathbb{E}[\ell(X, Y, h)]$$

$$h^* = \arg\min_{h \in H} R(h).$$

$$h_S = \arg\min_{h \in H} R_S(h) = \arg\min_{h \in H} \frac{1}{n}\sum_{i=1}^{n} \ell(X_i, Y_i, h).$$

We use the Empirical Risk Minimisation (ERM) algorithm to verify if a hypothesis class is PAC learnable or not.

Question: which one is smaller? $R_S(h_S) \le R_S(h^*)$

$$R_S(h_S) \text{ or } R_S(h^*)?$$

h_S minimizer for R_S

h^* minimizer for R

THE UNIVERSITY OF
SYDNEY

We start the proof

PAC learning checking

① check non equality

We have $R_S(h_S) \leq R_S(h^*)$, then

$$
\begin{aligned}
R(h_S) - \min_{h \in H} R(h) &= R(h_S) - R(h^*) \qquad (= R(h^*)) \\
&= R(h_S) - R_S(h_S) + R_S(h_S) - R_S(h^*) + R_S(h^*) - R(h^*) \\
&\leq R(h_S) - R_S(h_S) + R_S(h^*) - R(h^*) \\
&\leq |R(h_S) - R_S(h_S)| + |R(h^*) - R_S(h^*)| \\
&\leq \sup_{h \in H} |R(h) - R_S(h)| + \sup_{h \in H} |R(h) - R_S(h)| \\
&= 2 \sup_{h \in H} |R(h) - R_S(h)|.
\end{aligned}
$$

PAC learning checking

Very important inequality:

$$R(h_S) - \min_{h \in H} R(h) \leq 2 \sup_{h \in H} |R(h) - R_S(h)|$$

$$R(h_S) - R_S(h_S) \leq \sup_{h \in H} |R(h) - R_S(h)|.$$

expected risk
⇒ represents test error

training error

generalisation error small
hope to be
⇒ test error small

Generalisation error

PAC learning checking

② check "high probability" part

We need to find a way to upper bound $R(h_S) - \min_{h \in H} R(h)$ or $\sup_{h \in H} |R(h) - R_S(h)|$ with a high probability.

$$R_S(h) = \frac{1}{n} \sum_{i=1}^{n} \ell(X_i, Y_i, h)$$

$$R(h) = \mathbb{E}[R_S(h)] = \mathbb{E}[\ell(X, Y, h)]$$

According to the law of large numbers, we know that $R_S(h)$ will converge to $R(h)$ when the sample size n is sufficiently large. This a an asymptotical property.

PAC learning checking

Non-asymptotical measurement between $R(h)$ and $R_S(h)$:

Concentration inequality, e.g.,
Chebyshev's inequality
Hoeffding's inequality
Bernstein's inequality
McDiarmid's inequality

Hoeffding's inequality

Let $X_1, \ldots, X_n$ be independent random variables, such that $X_i \in [a_i, b_i]$ with probability one. Let $S_n = \frac{1}{n}\sum_{i=1}^{n} X_i$. Then for any $\epsilon > 0$, we have

upper bound by small ϵ with high probability

$$p\left\{|S_n - \mathbb{E}[S_n]| \geq \epsilon\right\} \leq 2\exp\left(\frac{-2n^2\epsilon^2}{\sum_{i=1}^{n}(b_i - a_i)^2}\right).$$

Let $\delta = 2\exp\left(\frac{-2n^2\epsilon^2}{\sum_{i=1}^{n}(b_i - a_i)^2}\right)$. Then

n>>0, close to 0

$$p\left\{|S_n - \mathbb{E}[S_n]| \geq \epsilon\right\} \leq \delta.$$

$\Leftrightarrow$

$P\{|S_n - E[S_n]| < \epsilon\} \geq 1-\delta$

Hoeffding's inequality

Note that $\ell(X_1, Y_1, h), \ldots, \ell(X_n, Y_n, h)$ are independent random variables. Assume that $\ell(X, Y, h) \in [0, M]$ Then, for any $\epsilon > 0$, we have

$$p\left\{\left|\mathbb{E}[\ell(X,Y,h)] - \frac{1}{n}\sum_{i=1}^{n}\ell(X_i,Y_i,h)\right| \geq \epsilon\right\} \leq 2\exp\left(\frac{-2n\epsilon^2}{M^2}\right),$$

$\mathbb{E}[\ell(X,Y,h)]$: $\uparrow$ E[sn

$\frac{1}{n}\sum_{i=1}^{n}\ell(X_i,Y_i,h)$: $\uparrow$ Sn

or

$$p\left\{|R(h) - R_S(h)| \geq \epsilon\right\} \leq 2\exp\left(\frac{-2n\epsilon^2}{M^2}\right).$$

$$\sup_{h\in H}|R(h) - R_S(h)|$$

THE UNIVERSITY OF SYDNEY

Hypothesis (class)

Hypothesis complexity

- Union bound

  For any events $A_1, A_2, \ldots, A_n$, we have

$$p\left\{\cup_{i=1}^{n} A_i\right\} \leq \sum_{i=1}^{n} p\{A_i\}.$$

- If A implies B, then $p\{A\} \leq p\{B\}$.

  if A happens, B will always happen

Hypothesis complexity

If A implies B, then $p\{A\} \leq p\{B\}$

Union bound

$$p\{|R(h) - R_S(h)| \geq \epsilon\} \leq 2\exp\left(\frac{-2n\epsilon^2}{M^2}\right).$$

$$p\left\{\sup_{h\in H}|R(h) - R_S(h)| \geq \epsilon\right\}$$

at least one hypothesis makes it bigger than ϵ

$$\leq p\left\{\cup_{h\in H}|R(h) - R_S(h)| \geq \epsilon\right\}$$

at least one happen

$$\leq \sum_{h\in H} p\left\{|R(h) - R_S(h)| \geq \epsilon\right\}$$

$$\leq 2|H|\exp\left(\frac{-2n\epsilon^2}{M^2}\right).$$

Hypothesis complexity

$$p\left\{\sup_{h\in H}|R(h)-R_S(h)|\geq\epsilon\right\}\leq 2|H|\exp\left(\frac{-2n\epsilon^2}{M^2}\right).$$

Let $\delta = 2|H|\exp\left(\frac{-2n\epsilon^2}{M^2}\right)$. We have

$$\epsilon = M\sqrt{\frac{\log|H|+\log 2/\delta}{2n}}.$$

Hypothesis complexity

# of hypothesis

Thus, with probability at least $1-\delta$, we have

$$\sup_{h\in H}|R(h)-R_S(h)|\leq M\sqrt{\frac{\log|H|+\log 2/\delta}{2n}}.$$

Generalisation bound

Very important inequality:

$$R(h_S) - R_S(h_S) \leq \sup_{h \in H} |R(h) - R_S(h)|.$$

We have

$$R(h_s) \leq R_S(h_S) + \sup_{h \in H} |R(h) - R_S(h)|.$$

Generalisation error bound

PAC learning checking

If the hypothesis class is of finite hypotheses, it is PAC learnable. Because

$$\epsilon = M\sqrt{\frac{\log|H| + \log 2/\delta}{2n}}.$$

$$p\left\{R(h_S) - \min_{h\in H} R(h) \leq 2\sup_{h\in H}|R(h) - R_S(h)| \leq 2M\sqrt{\frac{\log|H| + \log(2/\delta))}{2n}}\right\} \geq 1-\delta.$$

Since $\delta = 2|H|\exp\left(\frac{-2n\epsilon^2}{M^2}\right)$. We have

$$n = \frac{M^2}{\epsilon^2}\log\left(\frac{2|H|}{\delta}\right).$$

$n > poly(1/\delta, 1/\epsilon)$

PAC learning checking

If the hypothesis class is of finite hypotheses, it is PAC learnable. Because

$$p\left\{R(h_S) - \min_{h\in H} R(h) \leq 2\sup_{h\in H}|R(h) - R_S(h)| \leq 2M\sqrt{\frac{\log|H| + \log(2/\delta))}{2n}}\right\} \geq 1-\delta.$$

$P\{R(h_S) - R(h^*) \leq 2\epsilon\} \geq 1-\delta$

We can find that if the hypothesis class H is large, to find a good hypothesis with a small prediction error, we need a large training sample size n, which means that we should choose small hypothesis space.

THE UNIVERSITY OF
SYDNEY

The proof ends

PAC learning checking

If the hypothesis class is of infinite many hypotheses, is it PAC learnable?

How to derive a generalisation bound?

THE UNIVERSITY OF SYDNEY

VC dimension

(we consider binary classification in this subsection)

VC dimension

If the predefined hypothesis class *H* has infinite many hypotheses, how can we upper bound

$$\sup_{h \in H} |R_S(h) - R(h)|?$$

Hint: we consider binary classifier s and group the hypothesis infinite hypotheses but finite groups.

$$H = \{(h_1^1, \ldots, h_{n_1}^1), (h_1^2, \ldots, h_{n_2}^2), \ldots, (h_1^G, \ldots, h_{n_G}^G)\}.$$

VC dimension

$$H = \{(h_1^1, \ldots, h_{n_1}^1), (h_1^2, \ldots, h_{n_2}^2), \ldots, (h_1^G, \ldots, h_{n_G}^G)\}.$$

for binary classification, the prediction will be the same, either +1 or -1

Although the predefined hypothesis class *H* has infinitely many hypotheses, we can group them into finite groups, where the hypotheses in each group having the same value of

$$h(X_1), h(X_2), \ldots, h(X_n)$$

Let $h^1, \ldots, h^G$ be the representatives of each group, we have a new set of representatives:

$$H' = \{h^1, \ldots, h^G\}.$$

VC dimension

$$H = \{(h_1^1, \ldots, h_{n_1}^1), (h_1^2, \ldots, h_{n_2}^2), \ldots, (h_1^G, \ldots, h_{n_G}^G)\}.$$

$$H' = \{h^1, \ldots, h^G\}.$$

假设：Hypothesis set H 是 Binary function 的集合　　$H=\{h_1, h_2, \cdots\}$.

Input：sample data set，size N　$(x_1, x_2, \cdots x_{N-1}, x_N)$.

output：for one hypothesis h (Dichotomy)　$(h(x_1), h(x_2), \cdots h(x_N))$，$h(x_i) \in \{-1, +1\}$

$H(x_1, x_2, \cdots x_N)$ 是 H 在 sample $(x_1, \cdots x_N)$ 上所有 dichotomy 的集合

Growth function：Hypothesis set 在任意 N 个点上所能产生的最多 dichotomy 的数量.

$$m_H(N) = \max_{x_1, \cdots x_N} |H(x_1, \cdots x_N)|$$

trivial bound　$m_H(N) \leq 2^N$

shatter：如果 H 能在 data set 上产生所有的 dichotomy，H 可以 shatter $x_1, \cdots, x_N$

break point：如果所有大小为 k 的 data set 都不能被 H shatter，则 k 为 H 一个 break point

VC dimension：对于一个 H，使得 $m_H(N) = 2^N$ 的最大 N，记为 $d_{vc}(H)$.

如果 growth function 恒等于 2^N，则 $d_{vc}(H)$ 为 ∞

VC dimension

How to find H' ?

Definition:

Growth function

The growth function $\Pi_H : \mathbb{N} \to \mathbb{N}$ for a hypothesis class H is defined by

given training samples n, the maximum number of different predictions

$$\forall n \in \mathbb{N},\ \Pi_H(n) = \max_{X_1,...,X_n} |\{h(X_1), ..., h(X_n) : h \in H\}|$$

The maximum group that have the same predictions.

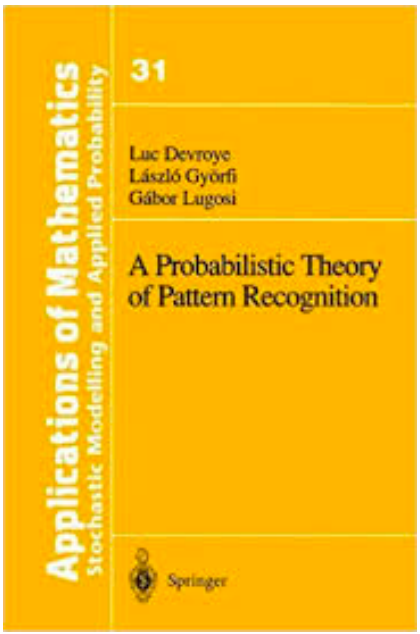
VC dimension

$$
\begin{aligned}
& p\left\{\sup_{h\in H}|R_S(h)-R(h)|\geq\epsilon\right\} \\
& \leq 2p\left\{\sup_{h\in H}|R_S(h)-R_{S'}(h)|\geq\epsilon/2\right\} \\
& \leq 4p\left\{\sup_{h\in H}\frac{1}{n}\left|\sum_{i=1}^{n}\sigma_i\ell(X_i,Y_i,h)\right|\geq\epsilon/4\right\} \\
& \leq 4\Pi_H(n)p\left\{\frac{1}{n}\left|\sum_{I=1}^{n}\sigma_i\ell(X_i,Y_i,h)\right|\geq\epsilon/4\right\} \\
& \leq 8\Pi_H(n)\exp(-n\epsilon^2/32M^2).
\end{aligned}
$$

VC dimension

Glivenko-Cantelli inequality: Proof

Chapter 12 of the following book



VC dimension

Definition:

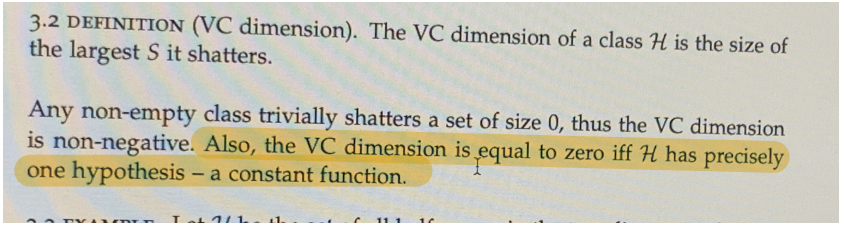
Shattering

The data points $\{X_1, \ldots, X_n\}$ is said to be shattered by a hypothesis class H when H realises all possible binary predictions. That is $\Pi_H(n) = 2^n$.

count # hypotheses which makes different prediction result

if $H = \{h\}$ growth function = 1 "only one hypothesis"

VC dimension


3.2 DEFINITION (VC dimension). The VC dimension of a class $\mathcal{H}$ is the size of the largest S it shatters.

Any non-empty class trivially shatters a set of size 0, thus the VC dimension is non-negative. Also, the VC dimension is equal to zero iff $\mathcal{H}$ has precisely one hypothesis – a constant function.

Definition:

VC dimension

The VC dimension of a hypothesis class H is the size of the largest set that can be fully shattered by H:

$$\text{VC dimension}(H) = \max_n \{n : \Pi_H(n) = 2^n\}.$$

VC dimension

Examples of the growth function:
Interval function class

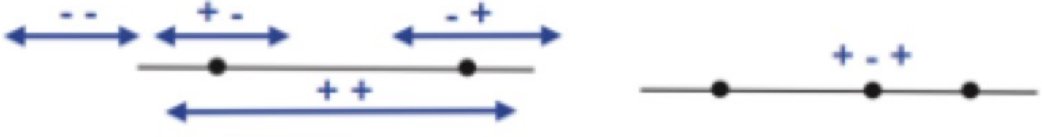
x within (a,b) infinitely hypotheses

$$H = \{x \mapsto 1_{\{x \in (a,b)\}} : a < b \in \mathbb{R}\}.$$

$$\Pi_H(1) = 2; \Pi_H(2) = 4; \Pi_H(3) = 7.$$

$= 2^1$ $= 2^2$ $< 2^3$

for 3 points, can never achieve "+ - +"



The VC dimension of the iInterval function class is 2.

VC dimension

Examples of the growth function:
Linear classifiers in $\mathbb{R}^2$

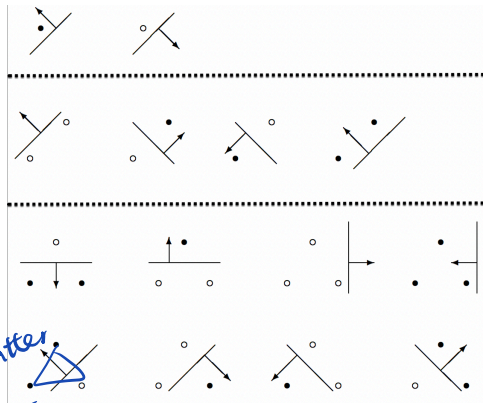
$$H = \{(x_1, x_2) \mapsto 1_{\{w_1 x_1 + w_2 x_2 + b \geq 0\}} : w_1, w_2, b \in \mathbb{R}\}.$$

in the space +1
out of the space -1

$n = 1$

$n = 2$

$n = 3$



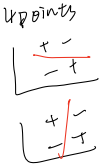
can shatter

can't shatter . . .

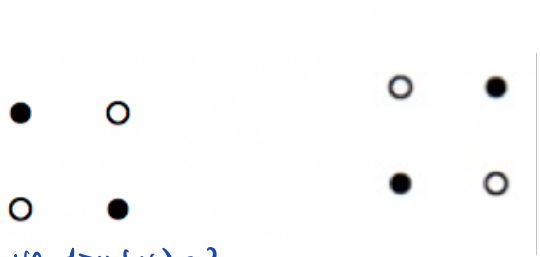
for n=3, there exist cases to be shattered, so VC dim(H) = 3

4 points

$n = 4$



The following two results cannot be achieved:



VC dimension

Examples of the growth function:
Linear classifiers in $\mathbb{R}^2$

$$H = \{(x_1, x_2) \mapsto 1_{\{w_1 x_1 + w_2 x_2 + b \geq 0\}} : w_1, w_2, b \in \mathbb{R}\}.$$

$$\Pi_H(1) = 2; \Pi_H(2) = 4; \Pi_H(3) = 8; \Pi_H(4) = 14.$$

$= 2^3$ $< 2^4$

The VC dimension of the linear function class is 3.

VC dimension

Let H be a hypothesis set with $\mathrm{VC\ dimension}(H) = d$ then for all $n \geq d$

$$\Pi_H(n) \leq \left(\frac{en}{d}\right)^d.$$

The proof is in Chapter 3 of the book "Foundations of ML"

VC dimension

include vc dimension

$$
\begin{aligned}
& p\left\{\sup_{h\in H}|R_S(h)-R(h)|\geq\epsilon\right\} \\
& \leq 2p\left\{\sup_{h\in H}|R_S(h)-R_{S'}(h)|\geq\epsilon/2\right\} \\
& \leq 4p\left\{\sup_{h\in H}\frac{1}{n}\left|\sum_{i=1}^{n}\sigma_i\ell(X_i,Y_i,h)\right|\geq\epsilon/4\right\} \\
& \leq 4\Pi_H(n)p\left\{\frac{1}{n}\left|\sum_{I=1}^{n}\sigma_i\ell(X_i,Y_i,h)\right|\geq\epsilon/4\right\} \\
& \leq 8\Pi_H(n)\exp(-n\epsilon^2/32M^2) \\
& \leq 8\left(\frac{en}{d}\right)^d\exp(-n\epsilon^2/32M^2).
\end{aligned}
$$

Let $8\left(\frac{en}{d}\right)^d\exp(-n\epsilon^2/32M^2)=\delta.$

We have

$$\epsilon=M\sqrt{\frac{32\left(d\log\frac{en}{d}+\log(8/\delta)\right)}{n}}.$$

With probability at least $1-\delta$, we have

$$\sup_{h\in H}|R(h)-R_S(h)|=M\sqrt{\frac{32\left(d\log\frac{en}{d}+\log(8/\delta)\right)}{n}}.$$

PAC learning checking

finite hypothesis → finite VC dimension

If the hypothesis class is of finite VC dimension, it is PAC learnable. Because

$$p\left\{R(h_S) - \min_{h\in H} R(h) \leq 2\sup_{h\in H}|R(h) - R_S(h)| \leq 2M\sqrt{\frac{32(d\log(en/d) + \log(8/\delta))}{n}}\right\} \geq 1-\delta.$$

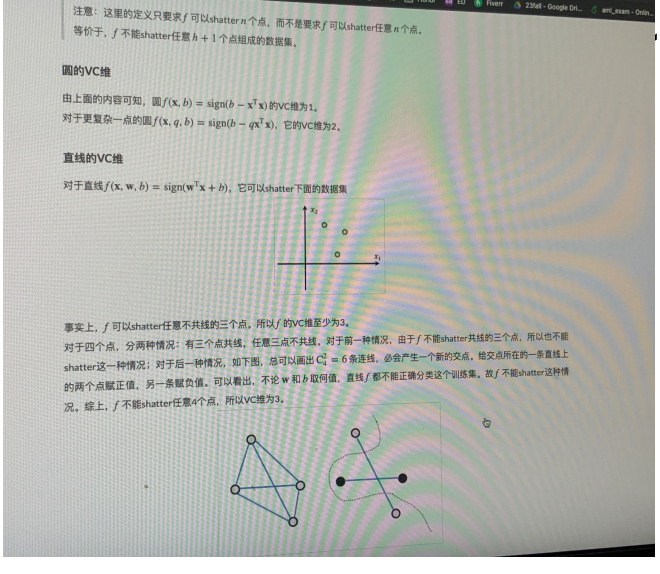
Since $\delta = 8\left(\frac{en}{d}\right)^d \exp(-n\epsilon^2/32M^2)$, we have

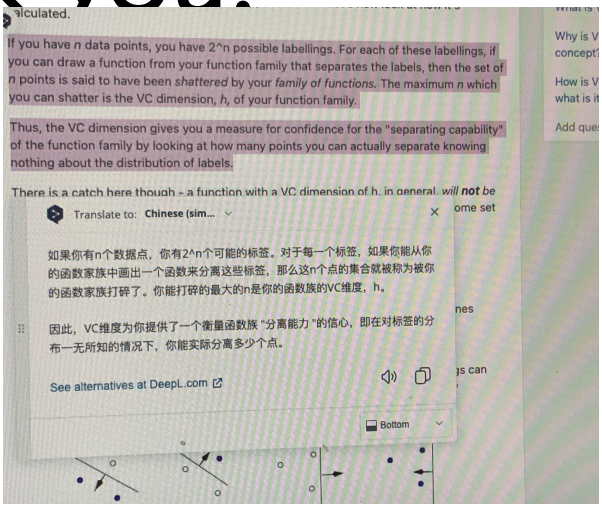
$$n = \frac{32M^2}{\epsilon^2}(d\log(en/d) + \log(8/\delta)).$$

$$n > poly(1/\delta, 1/\epsilon)$$

THE UNIVERSITY OF SYDNEY

Thank you!





Quiz Next Week

20 multiple choices

- Content: first three lectures and basic programming
- How to study
  - Lecture slides and tutorials
  - Examples and exercises
- Time: 7pm-8pm on Friday
- Venue: online

Assignment 1

- Assignment 1 will be released.
  - Group-based (2-3 students per group). Find you teammates by yourselves.