THE UNIVERSITY OF SYDNEY

Advanced Machine Learning (COMP 5328)

Hypothesis Complexity and Generalisation

Tongliang Liu

THE UNIVERSITY OF SYDNEY

Review

Best classifier

- The best classifier can be mathematically defined as:

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

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

Surrogate loss functions

- 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: $\ell(X, Y, h) = \exp(-Yh(X))$

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):

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

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

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

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

$$f(x) = f(a) + f'(a)(x - a) + h_1(x)(x - a)$$
$$= f(a) + f'(a)(x - a) + o(x - a).$$

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

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

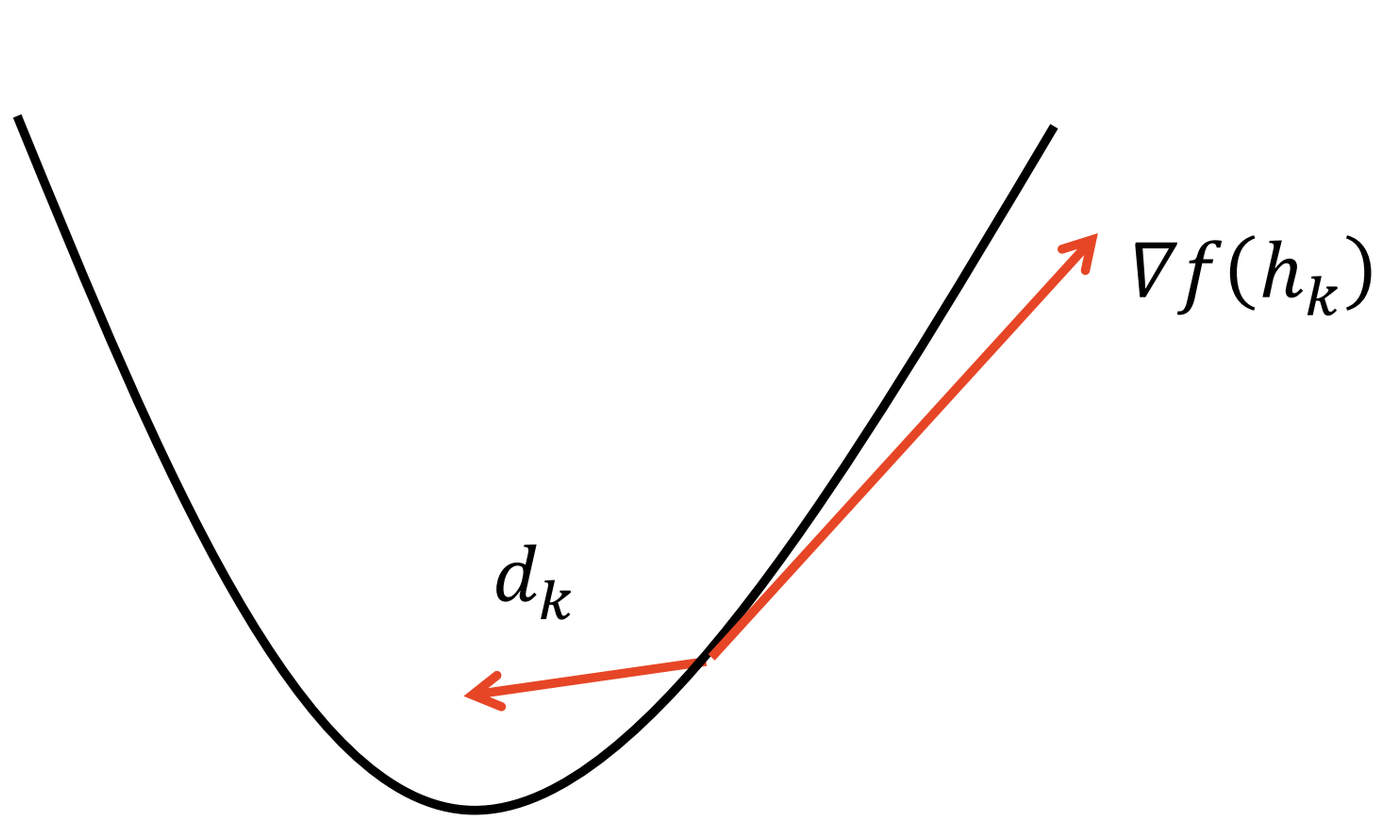
An iterative updating method

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

Two problems:

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

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



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

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

Gradient descent method

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

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

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.

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

Hypothesis class

What kind of hypothesis class should we choose?

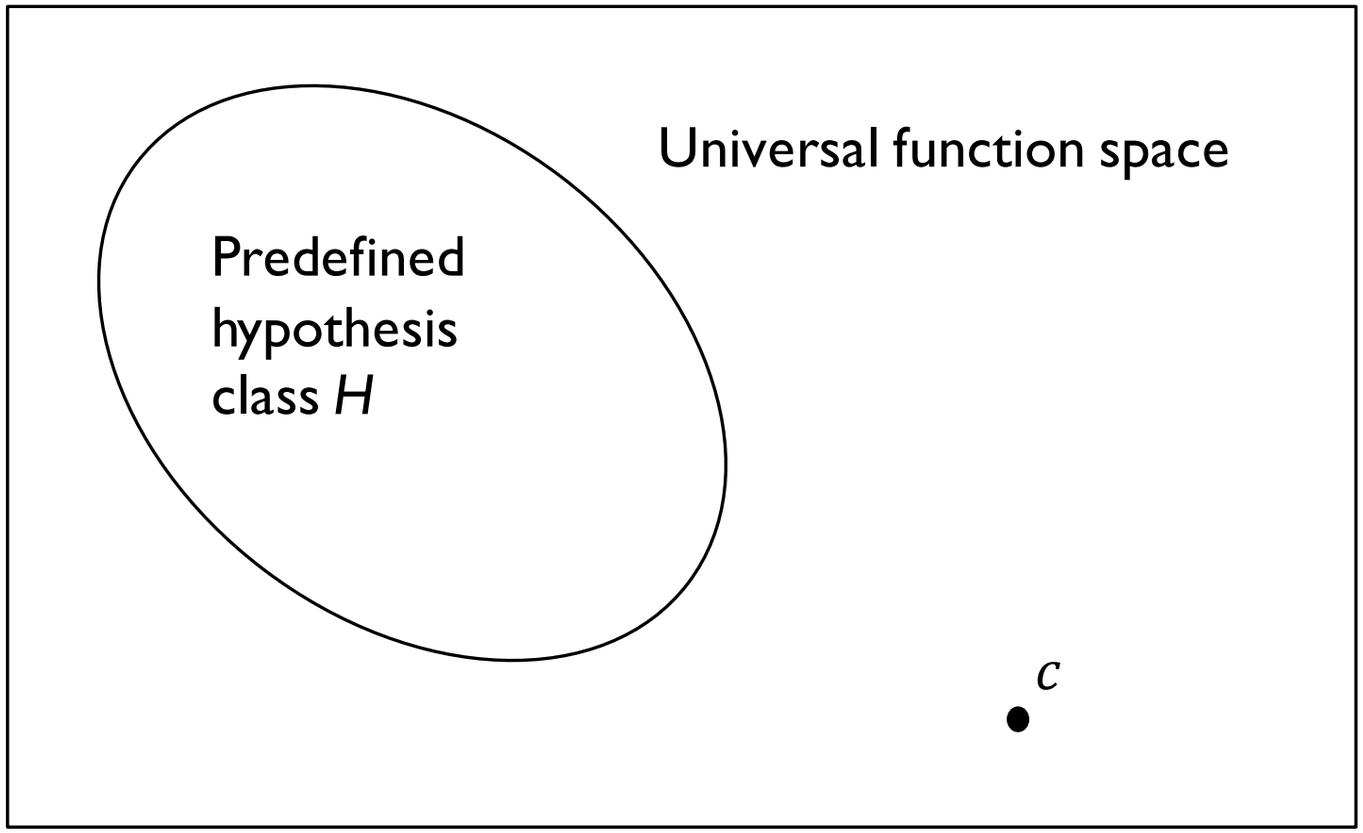
Polynomial functions vs linear functions

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

- 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):

$$c = \arg\min_h R(h).$$

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

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

- The hypothesis we can learn from data:

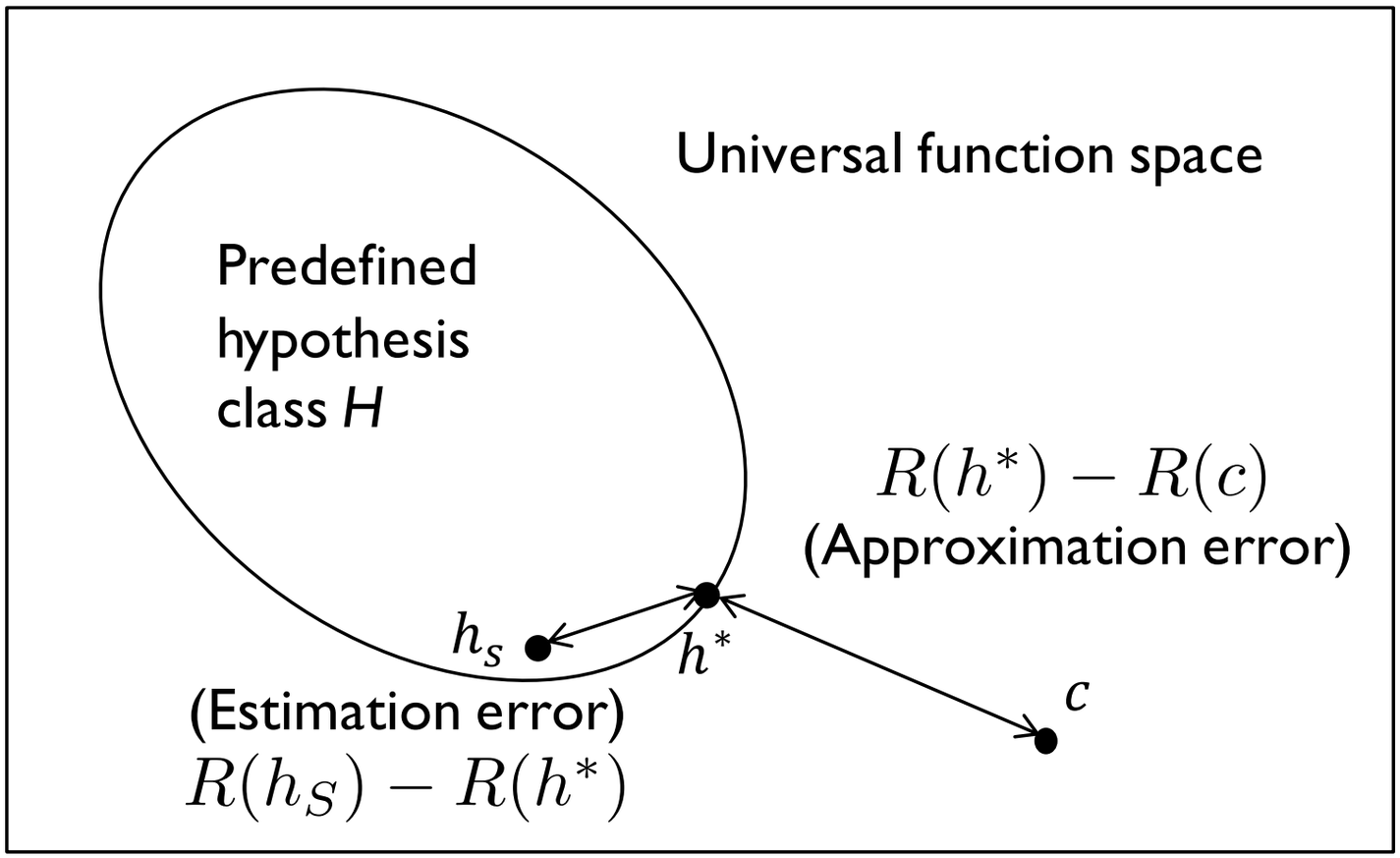
$$h_S = \arg\min_{h \in H} R_S(h) = \arg\min_{h \in H} \frac{1}{n} \sum_{i=1}^{n} \ell(X_i, Y_i, h).$$

Notation

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

Notation

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



- 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:

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 $\mathcal{A}$:

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

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

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

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

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

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

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

Question: which one is smaller?

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

THE UNIVERSITY OF SYDNEY

We start the proof

PAC learning checking

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

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

PAC learning checking

Very important inequality:

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

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

Generalisation error

PAC learning checking

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

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

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

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

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

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

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 $p\{A\} \leq p\{B\}$.

Hypothesis complexity

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

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

Union bound

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

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

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

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

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

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

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

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

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

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

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

VC dimension

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

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

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

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

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

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

The maximum group that have the same predictions.

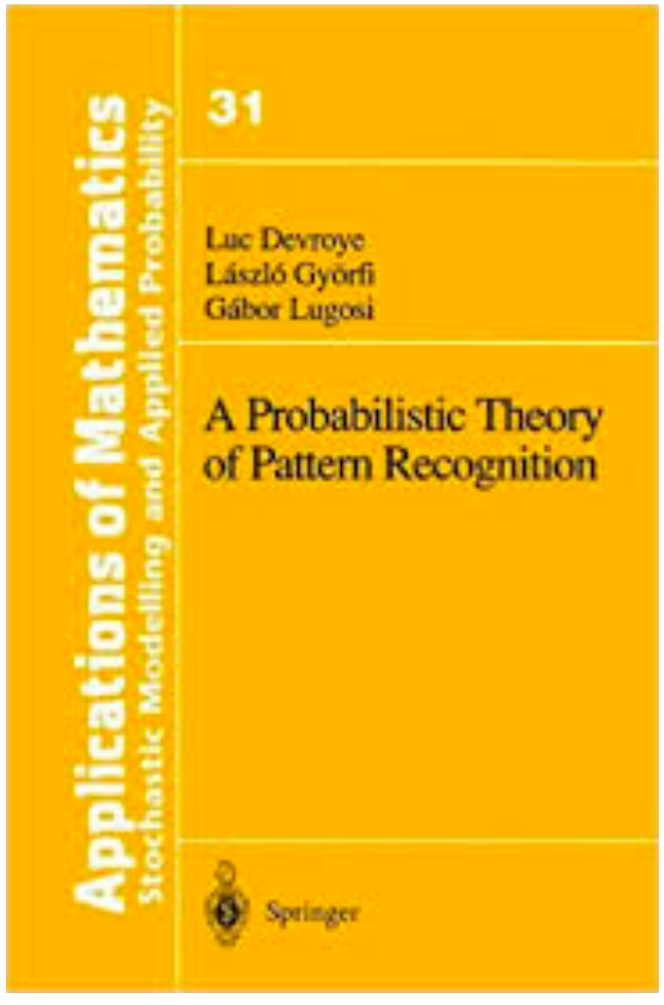
VC dimension

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

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 $\Pi_H(n) = 2^n$.

VC dimension

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:

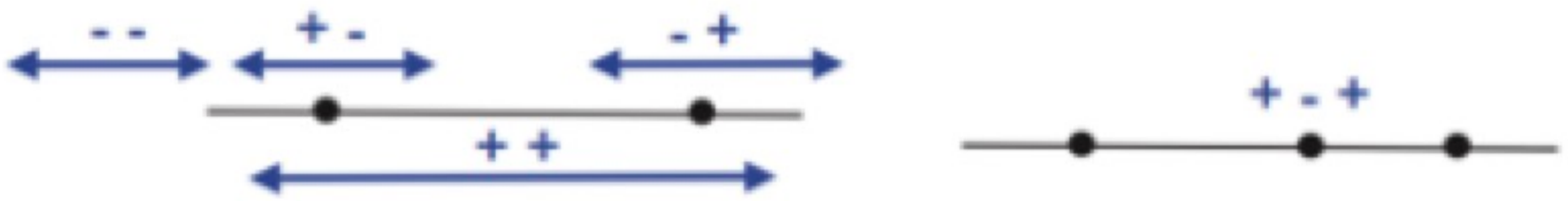
$$\text{VC dimension}(H) = \max_n \{n : \Pi_H(n) = 2^n\}.$$

VC dimension

Examples of the growth function:
Interval function class

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

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



The VC dimension of the iInterval function class is 2.

VC dimension

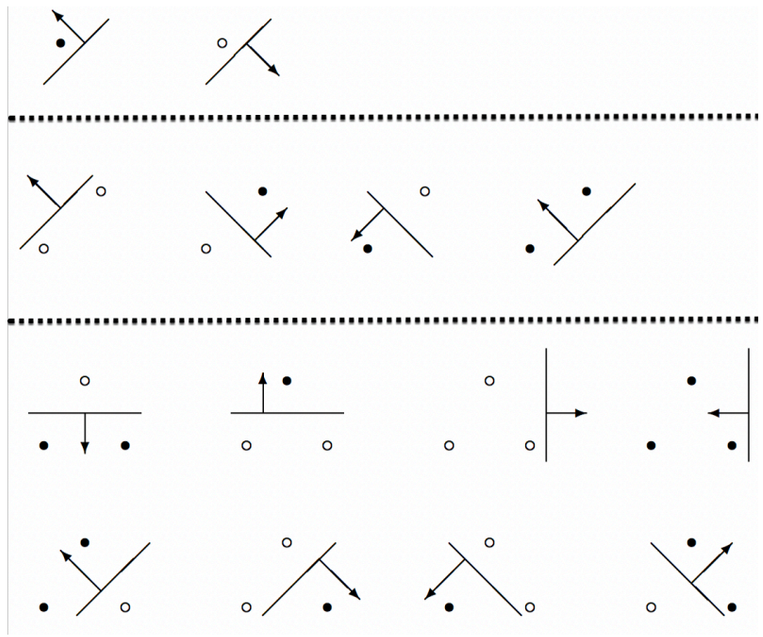
Examples of the growth function:
Linear classifiers in $\mathbb{R}^2$

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

$n = 1$

$n = 2$

$n = 3$



$n = 4$

The following two results cannot be achieved:

VC dimension

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

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

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

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

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

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

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!