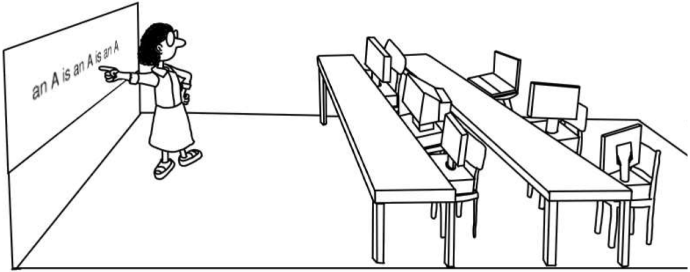
Machine Learning

Krystian Mikolajczyk & Deniz Gunduz

Department of Electrical and Electronic Engineering
Imperial College London



Machine Learning - Part 2.2 Summary

Department of Electrical and Electronic Engineering

Imperial College London

- Multiple hypothesis
- Growth function
- VC inequality

Example: Linear classification

- Features: $\mathbf{x} = (x_1, \ldots, x_d) \in \mathbb{R}^d$

  $\mathbf{x} = (x_0, x_1, \ldots, x_d) \in \mathbb{R}^{d+1}$ (with $x_0 = 1$).

- Labels: $y \in \{+1, -1\}$.

- Data points: $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$.

- Hypothesis class:

  $h_{\mathbf{w}}(\mathbf{x}) = \mathrm{sign}(\mathbf{w}^\top \mathbf{x})$, with $\mathbf{w} = (w_0, w_1, \ldots, w_d)$

Perceptron finds $g \in \mathcal{H}$ such that $g(\mathbf{x}_i) = y_i$ for all $i = 1, \ldots, n$

assuming it exists

$$g \in \operatorname*{argmin}_{h \in \mathcal{H}} \underbrace{\sum_{t=1}^{n} \mathbb{I}\left(h(\mathbf{x}_t) \neq y_t\right)}_{\widehat{R}_n(h)}$$

minimize $\widehat{R}_n(h_{\mathbf{w}})$ in $\mathbf{w}$

Empirical Risk Minimization

Is $|\mathcal{H}|$ finite? Does the theory apply?

Overlapping hypotheses - real case scenario

Hypotheses are overlapping!

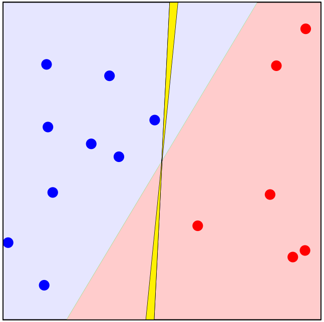
If $h_1 \approx h_2$ then:

$$\widehat{R}_n(h_1) \approx \widehat{R}_n(h_2)$$

$$R(h_1) \approx R(h_2)$$

Thus

$$|\widehat{R}_n(h_1) - R(h_1)| \approx |\widehat{R}_n(h_2) - R(h_2)|$$



$|\widehat{R}_n(h_1) - R(h_1)| > \varepsilon$ often implies $|\widehat{R}_n(h_2) - R(h_2)| > \varepsilon$

$h_1 \neq h_2$ if $\exists \mathbf{x}_i \in \mathcal{X} : h_1(\mathbf{x}_i) \neq h_2(\mathbf{x}_i) \rightarrow$ dichotomy

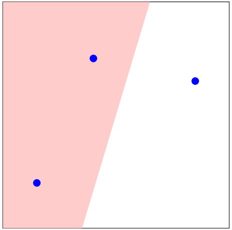
Growth Function - Perceptron

Use data samples $\mathbf{x}$ instead of entire input space $\mathcal{X}$.

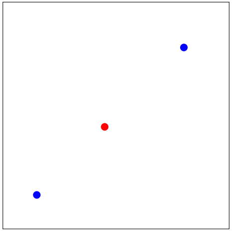
How many ways can we partition data points? Number of dichotomies?

number of hypotheses $|\mathcal{H}(\mathcal{X})| \gg |\mathcal{H}(\mathbf{x}_1, \ldots, \mathbf{x}_n)|$ number of dichotomies

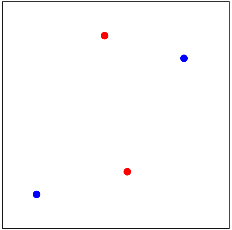
Growth function: $m_{\mathcal{H}}(n) = \max_{\mathbf{x}_1,\ldots,\mathbf{x}_n \in \mathcal{X}} |\mathcal{H}(\mathbf{x}_1, \ldots, \mathbf{x}_n)| \leqslant 2^n$ shattered



$n = 3$, in general position



$n = 3$, colinear



$n = 4$, in general position

$m_{\mathcal{H}}(3) = 8$

$m_{\mathcal{H}}(4) = 14 < 16$

Break point (capacity of $\mathcal{H}$, $\exists k : m_{\mathcal{H}(k)} \nexists (\mathbf{x}_1, \ldots, \mathbf{x}_k) \in \mathcal{X}$: shattered by $\mathcal{H}$)

Growth Function - Perceptron

Hoeffding's inequality

$$P\left(|\widehat{R}_n(g) - R(g)| > \varepsilon\right) \leqslant 2Me^{-2\varepsilon^2 n}$$

Replace $M = \infty$ with $m_{\mathcal{H}}(k) \rightarrow P\left(|\widehat{R}_n(g) - R(g)| > \varepsilon\right) \leqslant 2m_H(k)e^{-2\varepsilon^2 n}$

- no break point $\rightarrow m_{\mathcal{H}}(n) = 2^n$
- any break point $\exists k \rightarrow m_{\mathcal{H}}(k)$ is **polynomial** in n.

  $m_{\mathcal{H}}(k) = a_k n^k + a_{k-1} n^{k-1} +, \ldots, a_1 n + a_0$
- e^{-n} and large n data points will then reduce the whole probability

  Generalisation possible with probability assurance!
- no need to know k, only that it exists
- use $m_{\mathcal{H}}(2n)$, why? $|\widehat{R}_n(g) - R(g)| \approx |\widehat{R}_{n^{(1)}}(g) - \widehat{R}_{n^{(2)}}(g)|$

Generalisation Bounds

Extended Hoeffding's Inequality

$$P\left(\sup_{h\in\mathcal{H}} |\widehat{R}_n(h) - R(h)| > \varepsilon\right) \leqslant 2 \underbrace{m_{\mathcal{H}}(n)}_{incorrect: R(h) missing} e^{-2\varepsilon^2 n}$$

Vapnik-Chervonenkis Inequality

$$P\left(\sup_{h\in\mathcal{H}} |\widehat{R}_n(h) - R(h)| > \varepsilon\right) \leqslant 4 \underbrace{m_{\mathcal{H}}(2n)}_{\leqslant (2n+1)^{d_{VC}(\mathcal{H})}} e^{-\varepsilon^2 n/8}$$

where $d_{VC}(\mathcal{H}) = \max\{n : m_{\mathcal{H}}(n) = 2^n\}$ is the VC-dimension

The most important statement in theoretical machine learning

Vapnik-Chervonenkis Inequality

$$P\left(\sup_{h\in\mathcal{H}} |\widehat{R}_n(h) - R(h)| > \varepsilon\right) \leqslant 4 \underbrace{m_{\mathcal{H}}(2n)}_{\leqslant (2n+1)^{d_{VC}(\mathcal{H})}} e^{-\varepsilon^2 n/8}$$

where $d_{VC}(\mathcal{H}) = \max\{n : m_{\mathcal{H}}(n) = 2^n\}$ is the VC-dimension

- $m_{\mathcal{H}}(n+1) = 2^n$, $n+1$ is first break point, $n+2$ is also a break point
- max points that can be shattered $\approx$ "effective number of parameters"
- order of the polynomial that bounds $\mathcal{H}$: $m_{\mathcal{H}} \leqslant \sum_{i=0}^{d_{VC}} \binom{n}{i} \approx n^{d_{VC}}$
- independent of learning algorithm because $g \in \mathcal{H}$
- independent of input distribution p on $\mathcal{X}$ i.e. $\mathbf{x} \sim p$, only n matters
- independent of target distribution $P(y|\mathbf{x})$
- it concerns g and $\mathcal{H}$ and $(\mathbf{x}_i, \ldots, \mathbf{x}_n) \sim p$
- if d_{VC} is finite $\Rightarrow g \in \mathcal{H}$ will generalise with probability δ

Generalisation Bounds

Vapnik-Chervonenkis Inequality:

$$P\left(\sup_{h\in\mathcal{H}} |\widehat{R}_n(h) - R(h)| > \varepsilon\right) \leqslant 4 \quad \underbrace{m_{\mathcal{H}}(2n)}_{\leqslant (2n+1)^{d_{VC}(\mathcal{H})}} \quad e^{-\varepsilon^2 n/8}$$

where $d_{VC}(\mathcal{H}) = \max\{n : m_{\mathcal{H}}(n) = 2^n\}$ is the VC-dimension $\approx$ "effective number of parameters".

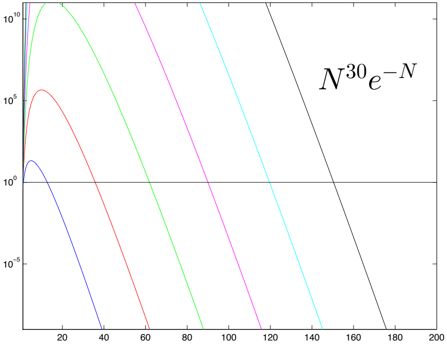
Example: for linear classification in d dimension,

$d_{VC} = d + 1,\ (w_0, \ldots, w_d)$

Figure: relation $n^{d_{VC}(\mathcal{H})} e^{-n}$.

Change $\mathcal{H}$ or n to make $P(.)$ small! N is proportional to d_{VC}

Rule of thumb: $n \geqslant 10 d_{VC}(\mathcal{H})$.

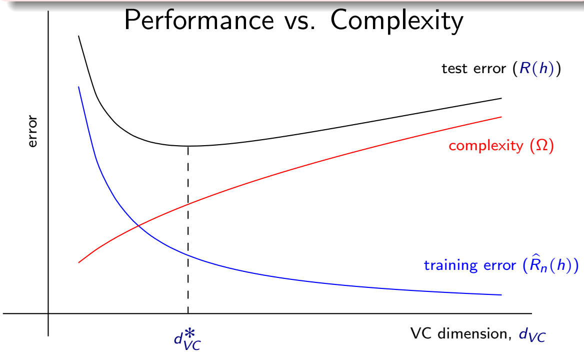


Generalisation Error

With probability at least $1-\delta$, for all $h \in \mathcal{H}$ simultaneously,

$$R(h) \leqslant \widehat{R}_n(h) + \underbrace{\sqrt{\frac{8d_{VC}(\mathcal{H})}{n}\log(2n+1) + \frac{8}{n}\log\frac{4}{\delta}}}_{\Omega(n,\mathcal{H},\delta)}$$

$$\boxed{R(h) \leqslant \widehat{R}_n(h) + \Omega(n,\mathcal{H},\delta)}$$



Example

In a binary classification problem, the test error is 27% on 100 random test samples. Give a confidence interval that contains the expected test error with 90% probability? How does the result change if the test error is obtained on 1000 samples?

Use Hoeffding, and then VC assuming VC dimension of the hypothesis class is 3.

Example

Let $X_i \in \{0, 1\}$ denote the error in classifying sample i, n the number of samples (100 or 1000), $\mu = \mathbb{E}[X_i]$ the expected test error, and $\nu = \frac{1}{n}\sum_{t=1}^{n} X_t = 0.27$ the empirical test error. By Hoeffding's inequality,

$$P(|\nu - \mu| > \varepsilon) \leqslant 2e^{-2\varepsilon^2 n}.$$

Therefore, $\mu \in [\nu - \varepsilon, \nu + \varepsilon]$ with probability at least $1 - \delta = 1 - 2e^{-2\varepsilon^2 n}$. Setting $1 - \delta = 0.9$ and solving for ε we get $\varepsilon = \sqrt{\log(2/\delta)/(2n)}$, which gives $\varepsilon \approx 0.12$ for $n = 100$ and $\varepsilon \approx 0.04$ for $n = 1000$. The required confidence intervals are $[0.15, 0.39]$ for $n = 100$ and $[0.23, 0.31]$ for $n = 1000$.

Example

Repeat with Vapnik-Chervonenkis Inequality:

$$P\left(|v-\mu|>\varepsilon\right) \leqslant 4(2n+1)^{d_{VC}} e^{-\varepsilon^2 n/8}$$

$\nu = 0.27$

$d_{VC} = 3$

$$\varepsilon = \sqrt{??}$$

Polynomial Bounds on growth function

Different bounds lead to different approximations of $\Omega(n, \mathcal{H}, \delta)$

Polynomial Bounds on growth function

- $m_{\mathcal{H}}(k)$ is **polynomial** in n, with break point $k+1$, $d_{VC} = k$
- $m_{\mathcal{H}}(n, k) = a_k n^k + a_{k-1} n^{k-1} +, \ldots, a_1 n + a_0$
  - inconvenient to use, so "nicer" bounds needed
- Order of the polynomial that bounds $\mathcal{H}$: $m_{\mathcal{H}}(n, d_{VC}) \leqslant \sum_{i=0}^{d_{VC}} \binom{n}{i} \approx n^{d_{VC}}$
- If $d_{VC}(n) < \infty$, then for all n:

$$m_{\mathcal{H}}(n) \leqslant n^{d_{VC}} + 1 \leqslant (n+1)^{d_{VC}}$$

and for all $n \geqslant d_{VC}$, an improved bound is:

$$m_{\mathcal{H}}(n) \leqslant \left(\frac{ne}{d_{VC}}\right)^{d_{VC}} \leqslant n^{d_{VC}} + 1$$

Example

Given VC inequality $P\left(\sup_{h\in\mathcal{H}}|\widehat{R}_n(h)-R(h)|>\varepsilon\right)\leqslant 4m_{\mathcal{H}}(2n)e^{-\varepsilon^2 n/8}$

show how to arrive to the following generalisation bounds $R(h)\leqslant\widehat{R}_n(h)+\Omega(n,\mathcal{H},\delta)$

(a)

$$\Omega(n,\mathcal{H},\delta)=\sqrt{\frac{8d_{VC}}{n}\log(2n+1)+\frac{8}{n}\log\frac{4}{\delta}}$$

(b)

$$\Omega(n,\mathcal{H},\delta)=\sqrt{\frac{8d_{VC}}{n}\log\frac{2n\mathrm{e}}{d_{VC}}+\frac{8}{n}\log\frac{4}{\delta}}$$

(c)

$$\Omega(n,\mathcal{H},\delta)=\sqrt{\frac{8d_{VC}}{n}\log 2n+\frac{8}{n}\log\frac{4}{\delta}}$$

Example

From the slide on polynomial bounds we can use the following bounds in 4 $\mathrm{m}_{\mathcal{H}}(2n)e^{-\varepsilon^2 n/8}$

- $m_{\mathcal{H}}(n) = n^{d_{VC}}$
- $m_{\mathcal{H}}(n) = n^{d_{VC}} + 1$
- $m_{\mathcal{H}}(n) = (n+1)^{d_{VC}}$
- $m_{\mathcal{H}}(n) = (\frac{ne}{d_{VC}})^{d_{VC}}$

Example

Show how to arrive to the following expression:

Let $g \in \operatorname{argmin}_{h\in\mathcal{H}} \widehat{R}_n(h)$ and $h^* \in \operatorname{argmin}_{h\in\mathcal{H}} R(h)$. Then with probability at least $1-\delta_1-\delta_2$,

$$R(g) \leqslant R(h^*) + \sqrt{\frac{8d_{VC}(\mathcal{H})}{n}\log(2n+1) + \frac{8}{n}\log\frac{4}{\delta_1}} + \sqrt{\frac{1}{2n}\log\frac{2}{\delta_2}}$$

Hint: Use $R(g), R(h^*), \widehat{R}_n(g), \widehat{R}_n(h^*)$,

Example

Proof:

$$R(g) - R(h^*) \leqslant \underbrace{R(g) - \widehat{R}_n(g)}_{\text{VC bound}} + \underbrace{\widehat{R}_n(g) - \widehat{R}_n(h^*)}_{\leqslant 0 \text{ by def. of } g} + \underbrace{\widehat{R}_n(h^*) - R(h^*)}_{\text{Hoeffding bound}}$$

Practical ML scenario

HMRC is considering using ML to identify suspicious tax return cases. Overall, it estimates about $\$4.4 \cdot 10^9$ in taxes is lost due to tax evasion each year. The average cost of investigating a taxpayer is approximately $\$10^4$. There are approximately 10 million taxpayers submitting their own tax returns, which are in structured form consisting of 100 fields, that can be converted into real value numbers. There are approx 4400 tax evasion cases every year and their records from the past 10 years are available. HMRC will find ML useful if it can guarantee that the test error will not differ from training by more than 20% with 99% certainty.

- Identify relevant ML components and formulate it as an ML problem.
- What is required to guarantee that the predictor meets HMRC criteria?

Practical ML scenario

$\mathbf{x}_i \in \mathbb{R}^{100}$ – there are $n_p = 44000$ positive data points and $100M$ in total

To learn, we should choose similar number of negative examples from $100M$, e.g. $n_n = 44000$, thus $n = 88000$

$f(\mathbf{x}_i) = y$, $y \in \{-1, 1\}$ – binary classification problem

$\widehat{R}_n(h) = \frac{1}{n} \sum_n \mathbb{I}\left(g(x_i) \neq y\right)$ – simple error function

H – hypothesis class with $d_{VC} \leqslant 8800$, eg. polynomial of degree k and linear classifier

ERM $g = \text{argmin}_{h \in H} \widehat{R}_n(h)$ – algorithm to find the best predictor g, so PLA

Better loss:

false negative – cost of not finding evasion: $= (4.4 \cdot 10^9)/(4.4 \cdot 10^3) \approx 10^6$

false positive – cost of investigating a tax return $= 10^4$

false negative leads to 100 times higher cost than false positive

therefore better loss: $\mathbb{I}\left(g(x_i) \neq y\right) = 100$ if $y = 1$ otherwise $\mathbb{I}\left(g(x_i) \neq y\right) = 1$ if $y = -1$

Practical ML scenario

error $\varepsilon = 0.2$, $n = 88000$, $P(\cdot) = 0.99$,

choose H with d_{VC} according to VC inequality

From VC inequality, the test error can be larger than the training error with probability 0.01

$$P\left(|\widehat{R}_n(h) - R(h)| > \varepsilon\right) \leqslant 4n^{d_{VC}} e^{-\varepsilon^2 n/8} = 4 \cdot 88000^{d_{VC}} e^{-0.2^2 88000/8} \approx 0.01$$

hence $d_{VC} \leqslant 39$.

No need to derive $d_{VC} = \ldots$, try a few numbers 5, 50, etc and you see if you need to reduce or increase.