DDA5001 Machine Learning

Training versus Testing (Part III)

Xiao Li

School of Data Science
The Chinese University of Hong Kong, Shenzhen

Recap: Generalization for Finite Hypothesis Space

Theorem: Generalization for finite hypothesis space

Let $\mathcal{H}$ be a finite hypothesis space, i.e., $|\mathcal{H}| < \infty$. For any $\delta > 0$, the following generalization bound holds with probability at least $1 - \delta$

$$\boxed{\forall f \in \mathcal{H} \qquad \mathrm{Er}_{\mathrm{out}}(f) \leq \mathrm{Er}_{\mathrm{in}}(f) + \sqrt{\frac{\log\left(\frac{2|\mathcal{H}|}{\delta}\right)}{2n}}} \tag{1}$$

- More samples (larger n) lead to better generalization.
- The generalization error increase when $|\mathcal{H}|$ grows, but only logarithmically.
- However, it is only for finite hypothesis case, i.e., $|\mathcal{H}| < +\infty$. This is impractical.

Recap: Dichotomy, Growth Function, and VC Dimension

Dichotomies of $\mathcal{H}$

Given $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$. The dichotomies generated by $\mathcal{H}$ on these points are defined by

$$\mathcal{H}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) = \{(f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_n)) : f \in \mathcal{H}\}.$$

Growth function

The growth function for the hypothesis set $\mathcal{H}$ is defined as:

$$\mathcal{G}_{\mathcal{H}}(n) = \max_{\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\} \subseteq \mathcal{X}} \left| \mathcal{H}(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) \right|$$

VC dimension

The VC dimension of a hypothesis space $\mathcal{H}$, denoted by $d_{\mathsf{VC}}(\mathcal{H})$ or simply d_{VC}, is the largest n so that it can be shattered by $\mathcal{H}$, i.e.,

$$d_{\mathsf{VC}}(\mathcal{H}) := \max\{n : \mathcal{G}_{\mathcal{H}}(n) = 2^n\}.$$

If $\mathcal{G}_{\mathcal{H}}(n) = 2^n$ for all n, then $d_{\mathsf{VC}}(\mathcal{H}) = \infty$.

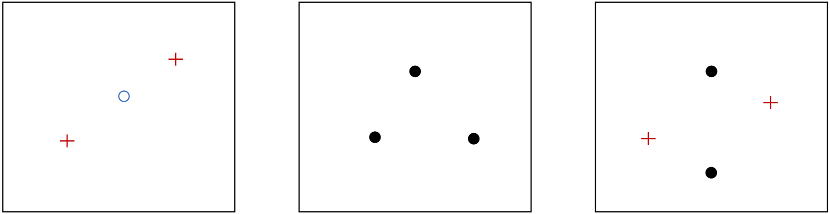
- VC dimension measures the complexity of $\mathcal{H}$, even when $|\mathcal{H}| = \infty$.

VC Dimension-induced Generation

Example: Perceptron in Two-Dimension

Suppose $\mathcal{X}$ is $\mathbb{R}^2$ and $\mathcal{H}$ is the two-dimensional perceptron.

What is $\mathcal{G}_{\mathcal{H}}(3)$ and $\mathcal{G}_{\mathcal{H}}(4)$?



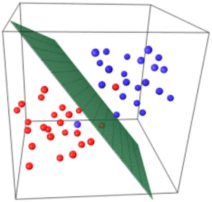
$\mathcal{G}_{\mathcal{H}}(3) = 8$ and $\mathcal{G}_{\mathcal{H}}(4) = 14$.

- One can indeed show that there are no 4 points that the two-dimensional perceptron can shatter.
- Therefore, $d_{\mathsf{VC}} = 3$ for a 2-dimensional perceptron.

Example: VC Dimension of General Linear Classifier

For a linear classifier, we can derive its VC dimension in a general sense.

This can be generalized to the following general result:



Theorem

For d-dimensional (binary) linear classifier, we have

$$\boxed{d_{\mathsf{VC}} = d + 1.}$$

- Proof is put in the supplementary material.

Property of VC Dimension of Linear Classifier

- d_{VC} is exactly the number of parameters of a d-dimensional binary linear classifier (think about perceptron), i.e., $\theta_0, \theta_1, \ldots, \theta_d$.
- d_{VC} measures the effective number of parameters, and hence the complexity of $\mathcal{H}$.
- The more parameter a model has, the more complex $\mathcal{H}$ is. This is reflected by a large d_{VC}.
- In some other models, the effective parameters may be less obvious.

VC Dimension Generalization Result

VC Dimension Generalization Result

After introducing all the related notions, we can now introduce the VC dimension generalization result.

VC generalization bound

For any $\delta > 0$, with probability at least $1 - \delta$, we have the following generalization bound:

$$\forall f \in \mathcal{H} \qquad \mathrm{Er}_{\mathrm{out}}(f) \leq \mathrm{Er}_{\mathrm{in}}(f) + \sqrt{\frac{8}{n} \log\left(\frac{4\mathcal{G}_{\mathcal{H}}(2n)}{\delta}\right)}$$

Upon invoking the upper bound on growth function using VC dimension, we have

$$\boxed{\forall f \in \mathcal{H} \qquad \mathrm{Er}_{\mathrm{out}}(f) \leq \mathrm{Er}_{\mathrm{in}}(f) + \sqrt{\frac{8}{n} \log\left(\frac{4((2n)^{d_{\mathrm{VC}}} + 1)}{\delta}\right)}}$$

- See the supplementary material for a proof sketch.

VC Generalization versus Previous Ones

- The VC generalization bound has the form

$$\boxed{\forall f \in \mathcal{H} \qquad \mathrm{Er}_{\mathrm{out}}(f) \leq \mathrm{Er}_{\mathrm{in}}(f) + \mathcal{O}\left(\sqrt{\frac{d_{\mathsf{VC}}}{n}}\right)}$$

  where $\mathcal{O}$ is used to hide a $\sqrt{\log n/\delta}$ term and some constants.
- Comparing the VC generalization bound to the finite $\mathcal{H}$ bound, it is easy to see that we not only replace $|\mathcal{H}|$ with $\mathcal{G}_{\mathcal{H}}$, but also change some constants. This is due to some technical issues. Fortunately, the overall idea is still maintained, that is, we use a much more reasonable effective number ($\mathcal{G}_{\mathcal{H}}$ or d_{VC}) to measure the complexity of $\mathcal{H}$ rather than using $|\mathcal{H}|$.
- Larger n means that $\mathrm{Er}_{\mathrm{in}}$ will generalize better to $\mathrm{Er}_{\mathrm{out}}$. When $n \to \infty$, we have $\mathrm{Er}_{\mathrm{in}} = \mathrm{Er}_{\mathrm{out}}$, which is consistent with our observation from the law of large numbers.

Is VC Generalization Bound Meaningful / Useful?

- The VC analysis is a universal result since it applies to all hypothesis space, learning algorithm, input space, probability distributions, binary targets (It can be extended to other target functions as well).
- Due to such a generality, the bound is indeed (quite) loose.

Though it is quite loose, it gives us useful guidance when conducting machine learning.

- It formally establishes the feasibility of learning for infinite $\mathcal{H}$. For $\mathcal{H}$ with finite d_{VC}, once we have enough training samples, learning is likely to be feasible.
- It tends to be equally loose for different models, enabling us to compare different models by comparing their d_{VC}. In real applications, model with smaller d_{VC} tend to generalize better than that with larger d_{VC}.
- It gives us some rules of thumb, e.g., about the number of training samples: $n \geq 10 \times d_{\mathsf{VC}}$.

$\rightsquigarrow$ We list several applications / guidance of the VC bound in practice.

Sample Complexity

Sample Complexity

Sample complexity: The sample complexity denotes how many training examples n are needed to achieve a certain generalization performance.

Suppose we want the result to hold with probability at least $1-\delta$, and generalization error (error between $\mathrm{Er}_{\mathrm{in}}$ and $\mathrm{Er}_{\mathrm{out}}$) to be less than some small number ε, we have

$$n \geq \frac{8}{\varepsilon^2} \log\left(\frac{4((2n)^{d_{\mathsf{VC}}}+1)}{\delta}\right).$$

Concisely, we need $n \geq \mathcal{O}\left(\frac{d_{\mathsf{VC}} \log(1/\delta) \log n}{\varepsilon^2}\right)$.

Example: Estimating the Sample Complexity

Example:
Suppose that we have a learning model with $d_{\mathsf{VC}} = 3$ and would like the generalization error to be at most 0.1 with confidence 90% (so $\varepsilon = 0.1, \delta = 0.1$). How big a data set do we need?

$$n \geq \frac{8}{0.1^2} \log\left(\frac{4((2n)^3 + 1)}{0.1}\right).$$

Solving the above inequality gives $n \approx 22000$. $\square$

- This obtained sample complexity is much bigger than the previously said rule of thumb $n \geq 10 \times d_{\mathsf{VC}}$, due to the fact that VC bound is quite loose.
- Nonetheless, the practical guidance is illustrated. With larger d_{VC}, we need more samples. This is consistent with practice.

Penalty for Model Complexity and Learning Curve

Example: Estimating the $\mathrm{Er}_{\mathrm{out}}$

In practice, we are often given $\mathcal{S}$. Hence, n is fixed. The question is what performance we can expect given n?

Example:
Suppose that $n = 10,000$ and we have a 90% confidence requirement ($\delta = 0.1$). What is the out-of-sample error can we guarantee with this confidence, given that $d_{\mathsf{VC}} = 3$?

By the generalization bound, we have

$$\begin{aligned}\mathrm{Er}_{\mathrm{out}}(f) &\leq \mathrm{Er}_{\mathrm{in}}(f) + \sqrt{\frac{8}{10000}\log\left(\frac{4((20000)^3+1)}{0.1}\right)} \\ &\approx \mathrm{Er}_{\mathrm{in}}(f) + 0.16. \end{aligned}$$

□

The Fundamental Trade-off

$$\boxed{\forall f \in \mathcal{H} \qquad \mathrm{Er}_{\mathrm{out}}(f) \leq \mathrm{Er}_{\mathrm{in}}(f) + \mathcal{O}\left(\sqrt{\frac{d_{\mathsf{VC}}}{n}}\right)}$$

To make $\mathrm{Er}_{\mathrm{out}}$ small:

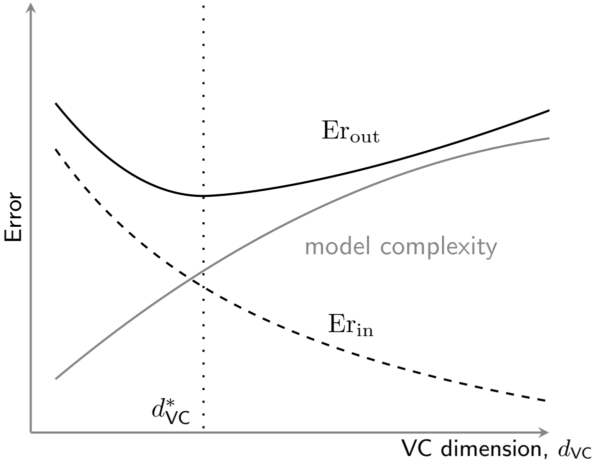
On the training side, we need:

more complex hypothesis $\mathcal{H}$ (larger d_{VC})

On the generalization side, we need:

less complex hypothesis $\mathcal{H}$ (smaller d_{VC})

Learning Curve from VC Analysis



- The optimal model is the one that minimizes the combinations of $\mathrm{Er}_{\mathrm{in}}$ and generalization error.
- Occam's Razor principle: The simplest workable model is the best.

VC Generalization Result for Regression

VC Generalization Bound for Linear Regression

- So far our VC generalization bound is established for binary classification case where $y = \{-1, +1\}$.
- By adopting certain generalized notion like pseudo-dimension, we can apply the similar VC analysis to linear regression model, i.e., $y = \boldsymbol{\theta}^\top \boldsymbol{x}$ where y is real-valued (continuous). Such a generalization result then applies to linear regression.
- Similar to binary classification case, a d-dimensional linear regression model has pseudo-dimension equal to $d + 1$.

VC generalization bound for linear regression

$$\boxed{\forall f \in \mathcal{H} \qquad \mathrm{Er}_{\mathrm{out}}(f) \leq \mathrm{Er}_{\mathrm{in}}(f) + \mathcal{O}\left(\sqrt{\frac{d_p}{n}}\right)}$$

where d_p is the pseudo-dimension.

- See the book "Foundations of Machine Learning" Chapter 11.2 for details.

$\rightsquigarrow$ Next lectures: Logistic regression and gradient-based optimization algorithm.