

Learning From Data

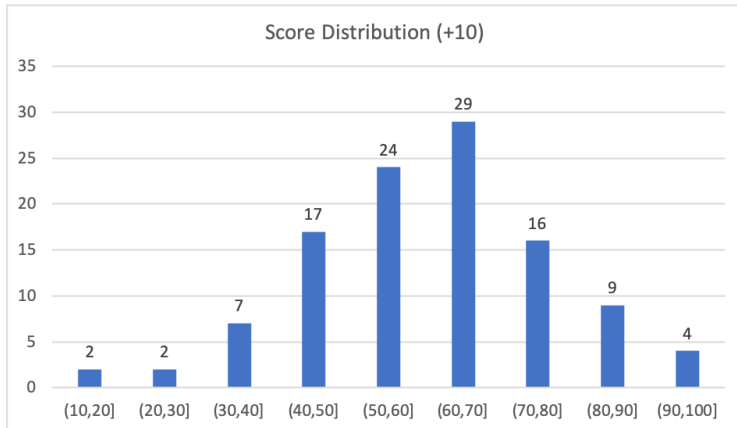
Lecture 7: Learning Theory

Yang Li yangli@sz.tsinghua.edu.cn

TBSI

November 18, 2022

Midterm Results



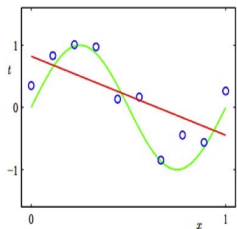
	max	mean	median
curved score	100	60	61

Review

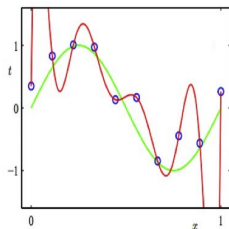
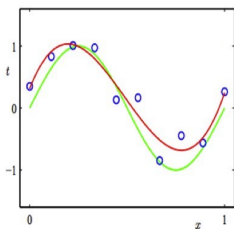
Overfit & Underfit

Underfit Both training error and testing error are large

Overfit Training error is small, testing error is large



underfit

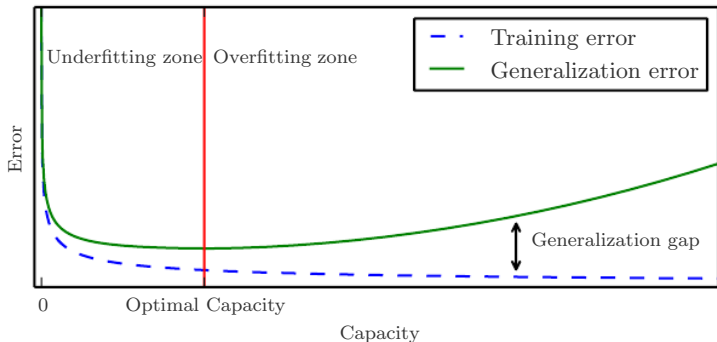


overfit

Model capacity: the ability to fit a wide variety of functions

Model Capacity

Changing a model's **capacity** controls whether it is more likely to overfit or underfit



How to formalize this idea?

Bias and Variance

Suppose data is generated by the following model:

$$y = h(x) + \epsilon$$

with $\mathbb{E}[\epsilon] = 0$, $\text{Var}(\epsilon) = \sigma^2$

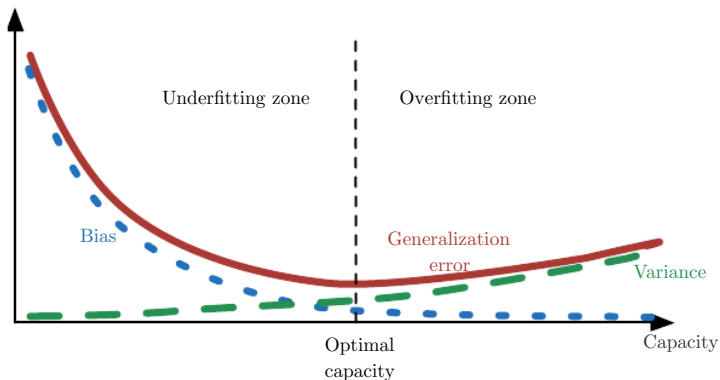
- ▶ $h(x)$: true hypothesis function, unknown
- ▶ $\hat{h}_D(x)$: estimated hypothesis function based on training data $D = \{(x^{(1)}, y^{(1)}), \dots, (x^{(m)}, y^{(m)})\}$ sampled from P_{XY}
- ▶ **Model bias:** $\text{Bias}(\hat{h}_D(x)) = \mathbb{E}_D[\hat{h}_D(x) - h(x)]$ *Expected estimation error of the model over all choices of training data D*
- ▶ **Model variance:** $\text{Var}(\hat{h}_D(x)) = \mathbb{E}_D[\hat{h}_D(x)^2] - \mathbb{E}_D[\hat{h}_D(x)]^2$ *Variance of the model over all choices of D*

Bias - Variance Tradeoff

If we measure generalization error by MSE

$$MSE = \mathbb{E}[(\hat{h}_D(x) - y)^2] = \text{Bias}(\hat{h}_D(x))^2 + \text{Var}(\hat{h}_D(x)) + \sigma^2,$$

- ▶ σ^2 represents irreducible error (*caused by noisy data*)
- ▶ in practice, increasing capacity tends to increase variance and decrease bias.



Exercise:

When the training error is much smaller than the testing error in a regression problem, what should be done? Select all that apply.

- ▶ A) Add more training data.
- ▶ B) Reduce model complexity.
- ▶ C) Add more features.
- ▶ D) Apply random transformation to the training data (*data augmentation*).
- ▶ Train multiple models on random subsets of the training data; Make prediction by averaging of the output of each model. (*bagging a.k.a. bootstrap aggregation*)

Today's Lecture

- ▶ How to measure model capacity?
- ▶ Can we find a theoretical guarantee for model generalization?

A brief introduction to learning theory

- ▶ Empirical risk estimation
- ▶ Generalization bound for finite and infinite hypothesis space

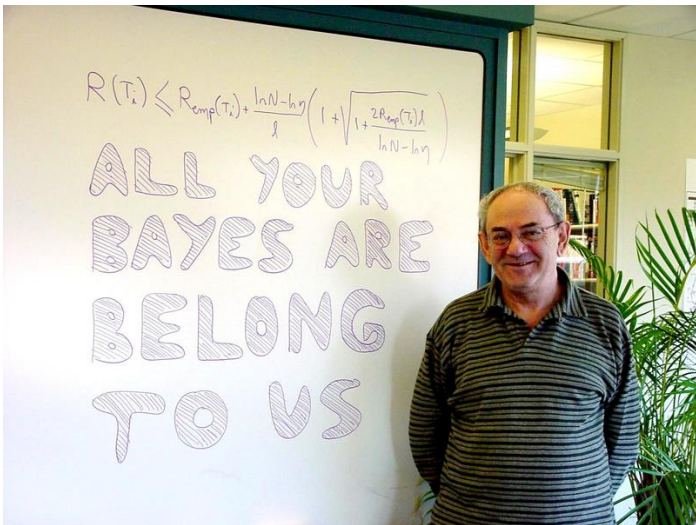
Final project information.

Introduction to Learning Theory

- ▶ Empirical risk estimation
- ▶ Learning bounds
 - ▶ Finite Hypothesis Class
 - ▶ Infinite Hypothesis Class

Learning theory

How to quantify generalization error?



Prof. Vladimir Vapnik in front of his famous theorem

Empirical risk

Simplified assumption: $y \in (0, 1)$

- ▶ Training set: $S = (x^{(i)}, y^{(i)}); i = 1, \dots, m$ with $(x^{(i)}, y^{(i)}) \sim \mathcal{D}$
- ▶ For hypothesis h , the **training error** or **empirical risk/error** in learning theory is defined as

$$\hat{\epsilon}(h) = \frac{1}{m} \sum_{i=1}^m 1\{h(x^{(i)}) \neq y^{(i)}\}$$

- ▶ The **generalization error** is

$$\epsilon(h) = P_{(x,y) \sim \mathcal{D}}(h(x) \neq y)$$

- ▶ **PAC assumption**: assume that training data and test data (for evaluating generalization error) were drawn from the same distribution \mathcal{D}

Hypothesis Class and ERM

Hypothesis class

The **hypothesis class** \mathcal{H} used by a learning algorithm is the set of all classifiers considered by it.

e.g. Linear classification considers $h_{\theta}(x) = 1\{\theta^T x \geq 0\}$

Empirical Risk Minimization (ERM): the "simplest" learning algorithm: pick the best hypothesis h from hypothesis class \mathcal{H}

$$\hat{h} = \operatorname{argmin}_{h \in \mathcal{H}} \hat{\epsilon}(h)$$

How to measure the generalization error of empirical risk minimization over \mathcal{H} ?

- ▶ Case of finite \mathcal{H}
- ▶ Case of infinite \mathcal{H}

Case of Finite \mathcal{H}

Goal: give guarantee on generalization error $\epsilon(h)$

- ▶ Show $\hat{\epsilon}(h)$ (training error) is a good estimate of $\epsilon(h)$
- ▶ Derive an upper bound on $\epsilon(h)$

For any $h_i \in \mathcal{H}$, the event of h_i miss-classification given sample $(x, y) \sim \mathcal{D}$:

$$Z = 1\{h_i(x) \neq y\}$$

$Z_j = 1\{h_i(x^{(j)}) \neq y^{(j)}\}$: event of h_i miss-classifying sample $x^{(j)}$

Training error of $h_i \in \mathcal{H}$ is:

$$\hat{\epsilon}(h_i) = \frac{1}{m} \sum_{j=1}^m 1\{h_i(x^{(j)}) \neq y^{(j)}\}$$

$$\hat{\epsilon}(h_i) = \frac{1}{m} \sum_{j=1}^m Z_j$$

Preliminaries

Here we make use of two famous inequalities:

Lemma 1 (Union Bound)

Let A_1, A_2, \dots, A_k be k different events, then

$$P(A_1 \cup \dots \cup A_k) \leq P(A_1) + \dots + P(A_k)$$

Probability of any one of k events happening is less the sums of their probabilities.

Preliminaries

Lemma 2 (Hoeffding Inequality, Chernoff bound)

Let Z_1, \dots, Z_m be m i.i.d. random variables drawn from a Bernoulli(ϕ) distribution. i.e. $P(Z_i = 1) = \phi$, $P(Z_i = 0) = 1 - \phi$. Let $\hat{\phi} = \frac{1}{m} \sum_{i=1}^m Z_i$ be the sample mean of RVs.

For any $\gamma > 0$,

$$P(|\phi - \hat{\phi}| > \gamma) \leq 2 \exp(-2\gamma^2 m)$$

The probability of $\hat{\phi}$ having large estimation error is small when m is large!

Case of Finite \mathcal{H}

Training error of $h_i \in \mathcal{H}$ is:

$$\hat{\epsilon}(h_i) = \frac{1}{m} \sum_{j=1}^m Z_j$$

where $Z_j \sim \text{Bernoulli}(\epsilon(h_i))$

By Hoeffding inequality,

$$P(|\epsilon(h_i) - \hat{\epsilon}(h_i)| > \gamma) \leq 2e^{-2\gamma^2 m}$$

By Union bound,

$$P(\forall h \in \mathcal{H}. |\epsilon(h) - \hat{\epsilon}(h)| \leq \gamma) \geq 1 - 2ke^{-2\gamma^2 m}$$

Uniform Convergence Results

Corollary 3

Given γ and $\delta > 0$, If

$$m \geq \frac{1}{2\gamma^2} \log \frac{2k}{\delta}$$

Then with probability at least $1 - \delta$, we have $|\epsilon(h) - \hat{\epsilon}(h)| \leq \gamma$ for all h .
 m is called the algorithm's **sample complexity**.

Remarks

- ▶ Lower bound on m tell us how many training examples we need to make generalization guarantee.
- ▶ # of training examples needed is logarithm in k

Uniform Convergence Results

Corollary 4

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

$$|\hat{\epsilon}(h) - \epsilon(h)| \leq \sqrt{\frac{1}{2m} \log \frac{2k}{\delta}}$$

What is the convergence result when we pick $\hat{h} = \operatorname{argmin}_{h \in \mathcal{H}} \hat{\epsilon}(h)$

Uniform Convergence Theorem for Finite \mathcal{H}

Using previous corollaries, we can bound $\epsilon(\hat{h})$:

Theorem 5 (Uniform convergence)

Let $|\mathcal{H}| = k$, and m, δ be fixed. With probability at least $1 - \delta$, we have

$$\epsilon(\hat{h}) \leq \left(\min_{h \in \mathcal{H}} \epsilon(h) \right) + 2\sqrt{\frac{1}{2m} \log \frac{2k}{\delta}}$$

Infinite hypothesis class: Challenges

Can we apply the same theorem to infinite \mathcal{H} ?

Example

- ▶ Suppose \mathcal{H} is parameterized by d real numbers. e.g. $\theta = [\theta_1, \theta_2, \dots, \theta_d] \in \mathbb{R}^d$ in linear regression with $d - 1$ unknowns.
- ▶ In a 64-bit floating point representation, size of hypothesis class: $|\mathcal{H}| = 2^{64d}$
- ▶ How many samples do we need to guarantee $\epsilon(\hat{h}) \leq \epsilon(h^*) + 2\gamma$ to hold with probability at least $1 - \delta$?

$$m \geq O\left(\frac{1}{\gamma^2} \log \frac{2^{64d}}{\delta}\right) = O\left(\frac{d}{\gamma^2} \log \frac{1}{\delta}\right) = O_{\gamma, \delta}(d)$$

To learn well, the number of samples has to be linear in d

Infinite hypothesis class: Challenges

Size of \mathcal{H} depends on the choice of parameterization

Example

$2n + 2$ parameters:

$$h_{u,v} = \mathbf{1}\{(u_0^2 - v_0^2) + (u_1^2 - v_1^2)x_1 + \dots + (u_n^2 - v_n^2)x_n \geq 0\}$$

is equivalent the hypothesis with $n + 1$ parameters:

$$h_\theta(x) = \mathbf{1}\{\theta_0 + \theta_1x_1 + \dots + \theta_nx_n \geq 0\}$$

We need a complexity measure of a hypothesis class invariant to parameterization choice

Infinite hypothesis class: Vapnik-Chervonenkis theory

A computational learning theory developed during 1960-1990 explaining the learning process from a statistical point of view.



Alexey Chervonenkis (1938-2014), Russian mathematician

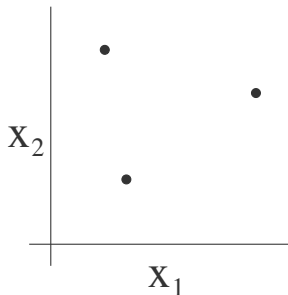


Vladimir Vapnik (Facebook AI Research, Vencore Labs)
Most known for his contribution in statistical learning theory

Shattering a point set

- ▶ Given d points $x^{(i)} \in \mathcal{X}$, $i = 1, \dots, d$, \mathcal{H} **shatters** S if \mathcal{H} can realize any labeling on S .

Example: $S = \{x^{(1)}, x^{(2)}, x^{(3)}\}$ where $x^{(i)} \in \mathbb{R}^2$.

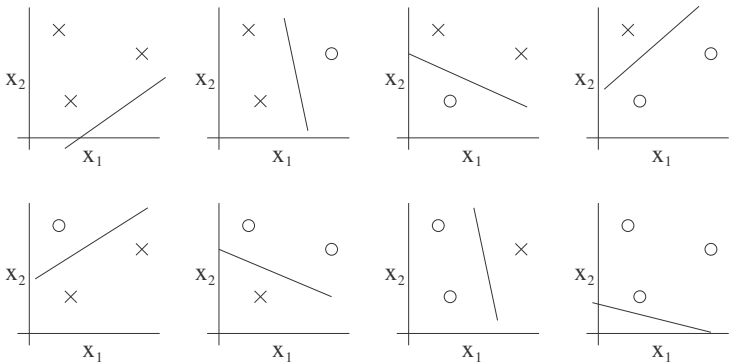


Suppose $y^{(i)} \in \{0, 1\}$, how many possible labelings does S have?

Shattering a point set

- Example: Let $\mathcal{H}_{LTF,2}$ be the linear threshold function in \mathbb{R}^2 (e.g. in the perceptron algorithm)

$$h(x) = \begin{cases} 1 & w_1x_1 + w_2x_2 \geq b \\ 0 & \text{otherwise} \end{cases}$$

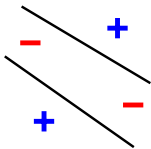


$\mathcal{H}_{LTF,2}$ shatters $S = \{x^{(1)}, x^{(2)}, x^{(3)}\}$

VC Dimension

The **Vapnik-Chervonenkis** dimension of \mathcal{H} , or $VC(\mathcal{H})$, is the cardinality of the largest set shattered by \mathcal{H} .

- ▶ Example: $VC(H_{LTF,2}) = 3$

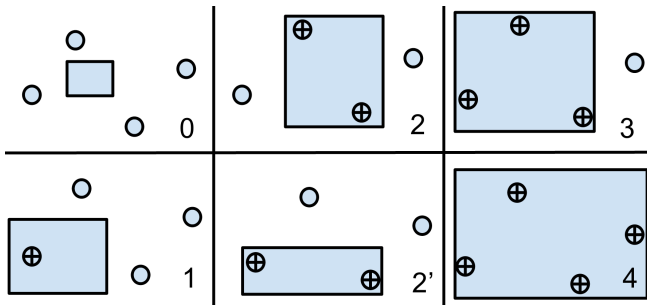


\mathcal{H}_{LTF} can not shatter 4 points: for any 4 points, label points on the diagonal as '+'. (See Radon's theorem)

- ▶ To show $VC(\mathcal{H}) \geq d$, it's sufficient to find **one** set of d points shattered by \mathcal{H}
- ▶ To show $VC(\mathcal{H}) < d$, need to prove \mathcal{H} doesn't shatter any set of d points

VC Dimension

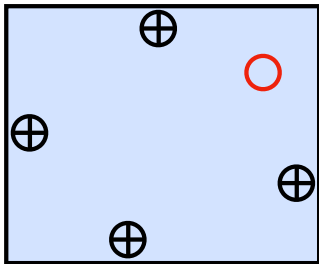
- Example: $VC(\text{AxisAlignedRectangles}) = 4$



Axis-aligned rectangles can shatter 4 points. $VC(\text{AxisAlignedRectangles}) \geq 4$

VC Dimension

- ▶ Example: $VC(\text{AxisAlignedRectangles}) = 4$



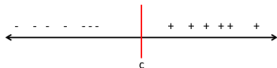
For any 5 points, label topmost, bottommost, leftmost and rightmost points as “+”.

$$VC(\text{AxisAlignedRectangles}) < 5$$

Discussion on VC Dimension

More VC results of common \mathcal{H} :

- ▶ $VC(\text{ConstantFunctions}) = 0$
- ▶ $VC(\text{PositiveHalf-Lines}) = 1, \mathcal{X} = \mathbb{R}$



- ▶ $VC(\text{Intervals}) = 2, \mathcal{X} = \mathbb{R}$
- ▶ $VC(\text{LTF in } \mathbb{R}^n) = n + 1, \mathcal{X} = \mathbb{R}^n \leftarrow \text{prove this at home!}$

Proposition 1

If \mathcal{H} is finite, VC dimension is related to the cardinality of \mathcal{H} :

$$VC(\mathcal{H}) \leq \log|\mathcal{H}|$$

Proof. Let $d = VC|\mathcal{H}|$. There must exist a shattered set of size d on which \mathcal{H} realizes all possible labelings. Every labeling must have a corresponding hypothesis, then $|\mathcal{H}| \geq 2^d$



Learning bound for infinite \mathcal{H}

Theorem 6

Given \mathcal{H} , let $d = VC(\mathcal{H})$.

- ▶ With probability at least $1 - \delta$, we have that for all h

$$|\epsilon(h) - \hat{\epsilon}(h)| \leq O\left(\sqrt{\frac{d}{m} \log \frac{m}{d} + \frac{1}{m} \log \frac{1}{\delta}}\right)$$

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

$$\epsilon(\hat{h}) \leq \epsilon(h^*) + O\left(\sqrt{\frac{d}{m} \log \frac{m}{d} + \frac{1}{m} \log \frac{1}{\delta}}\right)$$

Learning bound for infinite \mathcal{H}

Corollary 7

For $|\epsilon(h) - \hat{\epsilon}(h)| \leq \gamma$ to hold for all $h \in \mathcal{H}$ with probability at least $1 - \delta$, it suffices that $m = O_{\gamma, \delta}(d)$.

Remarks

- ▶ Sample complexity using \mathcal{H} is linear in $VC(\mathcal{H})$
- ▶ For “most”^a hypothesis classes, the VC dimension is linear in terms of parameters
- ▶ For algorithms minimizing training error, # training examples needed is roughly linear in number of parameters in \mathcal{H} .

^aNot always true for deep neural networks

VC Dimension of Deep Neural Networks

Theorem 8 (Cover, 1968; Baum and Haussler, 1989)

Let \mathcal{N} be an arbitrary feedforward neural net with w weights that consists of linear threshold activations, then $VC(\mathcal{N}) = O(w \log w)$.

Recent progress

- ▶ For feed-forward neural networks with piecewise-linear activation functions (e.g. ReLU), let w be the number of parameters and l be the number of layers, $VC(\mathcal{N}) = O(wl \log(w))$ [Bartlett et. al., 2017]
- ▶ *Among all networks with the same size (number of weights), more layers have larger VC dimension*, thus more training samples are needed to learn a deeper network

Bartlett and W. Maass (2003) Vapnik-Chervonenkis Dimension of Neural Nets

Bartlett et. al., (2017) Nearly-tight VC-dimension and pseudodimension bounds for piecewise linear neural networks.

Final Project Information

See <http://yangli-feasibility.com/home/classes/lfd2022fall/project.html>