ML Lec 1.

- Introduction

1. Machine Learning ($\in$ artificial intelligence)

⇒ a field of study that gives the computer the ability to learn without explicitly programmed

⇒ study of methodologies and algorithms that learn from data.

☆ parametric : given parameters, prediction is independent on observed data

☆ non-parametric : given parameters, prediction is dependent on observed data

① Supervised learning

⇒ training the model by providing a desired output to each training instance

- Regression : fitting a model to predict continuous real values

(ex. linear regression, non-linear regression)

- Classification : identifying which of a set of categories a new instance belongs

(ex. logistic regression, support vector machines, random forest)

- Learning theory
  - bias/variance trade-off
  - PAC learning
  - Rademacher Complexity
  - VC-dimension

- Ensemble method : combining multiple hypothesis into one.
  - Bagging or Boosting

② Unsupervised learning

⇒ training a model without any labels

(but still needs to be designed to obtain desired outcomes)

- Clustering

  : automatic grouping so that similar instances belong to the same clusters

  (ex. k-means clustering, EM clustering)

- Dimensionallity Reduction

  : reduce the dimension of data to avoid dimensionallity

  (ex. PCA, t-SNE, UMAP, Manifold learning)

③ Reinforcement Learning

⇒ Learn by interacting with environment

2. Bayesian machine learning

⇒ uncertainty occurs by follows

: inherent uncertainty, insufficient observation, data ambiguity

⇒ use Bayes' rule to infer parameter θ from data D

$$\underset{\text{posterior}}{P(\theta \mid D)} = \frac{\overset{\text{likelihood}}{P(D \mid \theta)} \cdot \overset{\text{prior}}{P(\theta)}}{\underset{\text{evidence}}{P(D)}} = \frac{P(D \mid \theta) \cdot P(\theta)}{\int P(D \mid \theta) \cdot P(\theta)\, d\theta}$$

① Sampling

⇒ techniques to directly sample from the posterior for inference

② Variational inference

⇒ approximating the intractable posterior with simpler, tractable distribution

③ Graphical model

⇒ probabilistic models that use graphs to express the conditional dependencies

④ Gaussian Process

⇒ a bayesian non-parametric approach to learn a distributions of functions

3. Neural Network

⇒ systems of interconnected neurons which learns parameters that non-linearly converts some inputs to an output

① Multilayer Perceptron (MLP)

: Most basic type of neural network

② Convolutional Neural Network (CNN)

: a deep neural network whose connection resembles that of visual cortex

(Local connectivity, parameter sharing, pooling)

③ Recurrent Neural Network (RNN)

: networks whose connections are from a directed cycle.

- Empirical risk and generalization

1. True risk : expected risk on any given pair of new data

(also referred to as generalization error)

$\Rightarrow R(h) = \mathbb{E}\left[L(h(x), y)\right] = \int L(h(x), y)\, dP(x,y)$

$\Rightarrow h^* = \underset{h \in H}{\operatorname{argmin}}\, R(h)$ → goal of machine learning

☆ $R(h)$ cannot be computed since $P(x,y)$ is an unknown distribution

2. Empirical risk : approximation of true risk

$\Rightarrow R_{emp}(h) = \frac{1}{m} \sum_{i=1}^{m} L(h(x_i), y_i)$

$\Rightarrow \hat{h} = \underset{h \in H}{\operatorname{argmin}}\, R_{emp}(h)$ → ill posed problem

"Not" satisfying
① have a solution
② have a unique solution
③ have a solution that depends continuously on the parameter

☆ This does not guarantee obtaining a model that works well on test data

→ high generalization error due to "overfitting"

☆ Measure for a machine learning algorithm performance

① making training error small

② making the gap between training error and test error small

(These can be handled by controlling model capacity through hypothesis space)

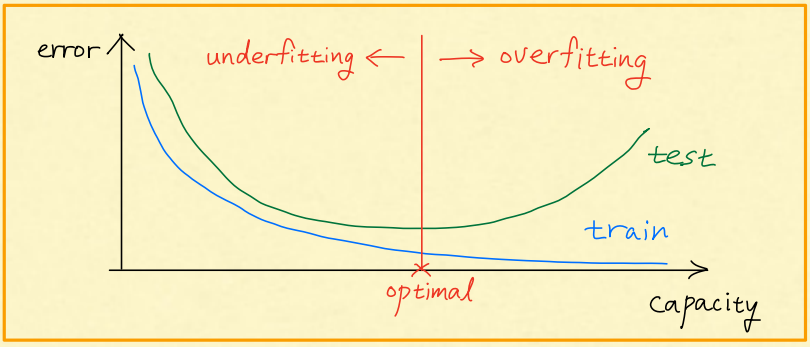
⇓

the set of functions the learning algorithm is allowed to select as solution

3. Effect of model capacity

⇒ simpler functions are more likely to generalize

⇒ still need to choose a sufficiently complex function to achieve lower training error



☆ No free lunch theorem

: averaged over all possible data-generating distributions, every classification algorithm has the same error rate when classifying unobserved points.

→ No machine learning algorithm is universally better than others.

→ With reasonable assumptions on data distribution, a certain algorithm can perform well

∴ The goal of ML is ① understanding distributions relevant to the real world and ② inventing algorithm that perform well on them.

ML Lec 2.

- Linear regression

1. Simple linear regression

$$\left[\begin{array}{l}\text{Suppose } y = \beta_0 + \beta_1 x + \varepsilon \\ \text{Let } \hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x\end{array}\right) \longrightarrow RSS = \sum_i^n (y_i - \hat{y})^2 = \sum_{i=1}^n (y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i)^2$$

☆ Determining β_0 and β_1

① $\frac{\partial RSS}{\partial \hat{\beta}_0} \Rightarrow \sum_{i=1}^n -2(y_i - \beta_0 - \beta_1 x_i) = 0$

$$\Rightarrow \boxed{\beta_0 = \sum_{i=1}^n (y_i - \beta_1 x_i) = \bar{y} - \beta_1 \bar{x}}$$

$$\Rightarrow RSS = \sum_{i=1}^n (y_i - \bar{y} - \beta_1 (x_i - \bar{x}))^2$$

② $\frac{\partial RSS}{\partial \hat{\beta}_1} \Rightarrow \sum_{i=1}^n -2\left(y_i - \bar{y} - \beta_1 (x_i - \bar{x})\right) \cdot (x_i - \bar{x}) = 0$

$$\Rightarrow \boxed{\beta_1 = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2} = \frac{Cov(x, y)}{Var(x)}}$$

2. Multivariate linear regression

$$\left[\begin{array}{l}\text{Suppose } y = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p + \varepsilon \iff y = X\beta + \varepsilon \\ \text{Let } \hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \cdots + \hat{\beta}_p x_p \iff \hat{y} = X\hat{\beta}\end{array}\right.$$

☆ Determining β

$$\Rightarrow J(\hat{\beta}) = \|y - \hat{y}\|_2^2 = \|y - X\hat{\beta}\|_2^2$$

$$\Rightarrow \nabla_{\hat{\beta}} J(\hat{\beta}) \Rightarrow -2X^T(y - X\beta) = 0$$

$$\Rightarrow X^T X\beta = X^T y \longrightarrow \text{Normal equation}$$

$\longrightarrow$ Moore-Penrose Pseudoinverse of X

: $\beta = (X^T X)^{-1} X^T y \longrightarrow$ computationally intensive

$\longrightarrow$ Singular Value Decomposition of $X^T X$

: $\beta = V\Sigma^{-1} U^T X^T y \longrightarrow$ computationally less intensive

(U, V : orthogonal matrix, Σ : diagonal matrix of singular values)

U, V ⇓ rotation Σ ⇓ scaling

3. Bayesian linear regression

Suppose $y = X\beta + \varepsilon$

Let $\hat{y} = X\hat{\beta} + \hat{\varepsilon}$ where $\hat{\varepsilon}_i \sim N(0, \sigma^2)$ are i.i.d. random variables

$$\Rightarrow L(\beta) = p(y \mid X;\beta) = \prod_{i=1}^{n} p(y_i \mid x_i;\beta) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y_i - x_i\beta)^2}{2\sigma^2}\right)$$

$$\Rightarrow \ell(\beta) = \sum_{i=1}^{n} \log \frac{1}{\sqrt{2\pi}\sigma} - \frac{(y_i - x_i\beta)^2}{2\sigma^2}$$

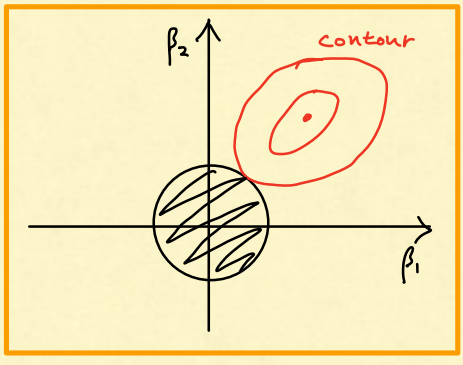
$$= n \cdot \log \frac{1}{\sqrt{2\pi}\sigma} - \frac{1}{2\sigma^2} \boxed{\sum_{i=1}^{n} (y_i - x_i\beta)^2} \rightarrow \text{least squares loss}$$

$\therefore$ maximizing log-likelihood $\Leftrightarrow$ minimizing $\|y - X\beta\|_2^2$

4. Ridge regression

: linear regression with L2-regularization

(The value of β shrinks so that the variance decreases)



☆ Determining β

$\Rightarrow$ X is standardized and y is centered

$$\Rightarrow J(\hat{\beta}) = \|y - X\hat{\beta}\|_2^2 + \lambda \|\hat{\beta}\|_2^2$$

$$\Rightarrow \nabla_{\hat{\beta}} J(\hat{\beta}) \Rightarrow -2X^T(y - X\beta) + 2\lambda\beta = 0$$

$$\Rightarrow \beta = (X^TX + \lambda I)^{-1} X^T y$$

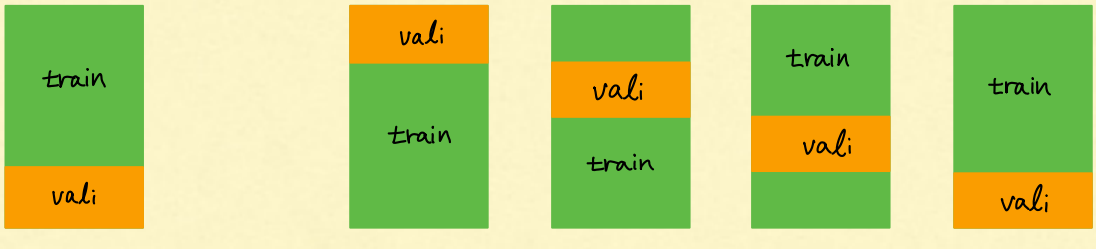
① Even if X^TX is singular, λI makes problem non-singular

② λ is a hyperparameter that can be choosed by cross-validation.

* Cross-Validation

: partition the training set into complementary subsets

(use one for the training and the other for validation)



Hold-out

K-fold cross-validation

ML Lec 3.

- Logistic regression

1. Nearest Neighbor Classification

: assign label of the nearest training data to each test data ⟶ too sensitive to outliers

2. k-Nearest Neighbor classification

: find k neighbors and use voting to decide the label of test instance

Strength
- non-linear decision boundary
- robust to noisy data
- effective in large data

Weakness
- need to determine k
- curse of dimensionality
- computation & memory cost

3. Logistic regression

: predict $p(y=1|x)$ by transforming linear regression by sigmoid function

(derivative of sigmoid function $(f(x)) = f(x)\cdot(1-f(x))$)

$$\Rightarrow L(\theta) = p(y|X;\theta) = \prod_{i=1}^{n} p(y_i|x_i;\theta) = \prod_{i=1}^{n} f_\theta(x_i)^{y_i}\,(1-f_\theta(x_i))^{1-y_i}$$

$$\Rightarrow \ell(\theta) = \sum_{i=1}^{n} y_i \log f_\theta(x_i) + (1-y_i)\log(1-f_\theta(x_i))$$

$$= -\boxed{\left(-\sum_{i=1}^{n} y_i \log f_\theta(x_i) + (1-y_i)\log(1-f_\theta(x_i))\right)}$$ ⟶ No analytic solution. (use descent method)

∴ maximizing log-likelihood ⟺ minimizing cross-entropy

☆ $\ell(\theta)$ is concave

pf) Let $g_\theta(x_i) = \log f_\theta(x_i)$, $h_\theta(x_i) = \log(1-f_\theta(x_i))$

$$\Rightarrow \nabla g_\theta(x_i) = \frac{1}{f_\theta(x_i)} f_\theta(x_i)\cdot(1-f_\theta(x_i))\cdot x_i = (1-f_\theta(x_i))\cdot x_i$$

$$\Rightarrow \nabla h_\theta(x_i) = \frac{1}{1-f_\theta(x_i)} \cdot f_\theta(x_i)\,(1-f_\theta(x_i))\,(-x_i) = -f_\theta(x_i)\cdot x_i$$

Then $\ell(\theta) = \sum_{i=1}^{n} y_i g_\theta(x_i) + (1-y_i) h_\theta(x_i)$

$$\Rightarrow \nabla\ell(\theta) = \sum_{i=1}^{n} y_i(1-f_\theta(x_i))\cdot x_i - (1-y_i) f_\theta(x_i)\cdot x_i = \sum_{i=1}^{n}(y_i - f_\theta(x_i))\cdot x_i$$

$$\Rightarrow \nabla^2\ell(\theta) = \sum_{i=1}^{n} -f_\theta(x_i)\cdot(1-f_\theta(x_i))\cdot x_i x_i^T$$

$$= X^T \mathrm{diag}\underbrace{(-f_\theta(x_i)(1-f_\theta(x_i))}_{(\le 0)}\cdot X$$ ⟶ negative semi-definite

4. Regularized logistic regression

$$\Rightarrow J(\theta) = -\sum_{i=1}^{n} y_i \log f_\theta(x_i) + (1-y_i) \log (1-f_\theta(x_i)) + \lambda \|\theta\|_2^2$$

5. Multinomial logistic regression

$$\Rightarrow f_\theta(x) = \begin{bmatrix} P(y=1|x;\theta) \\ P(y=2|x;\theta) \\ \vdots \\ P(y=K|x;\theta) \end{bmatrix} = \frac{1}{\sum_{j=1}^{K} \exp(\theta_j^T x)} \begin{bmatrix} \exp(\theta_1^T x) \\ \exp(\theta_2^T x) \\ \vdots \\ \exp(\theta_K^T x) \end{bmatrix}$$

$$\Rightarrow J(\theta) = -\sum_{i=1}^{n} \sum_{k}^{K} 1\{y_i = k\} \log \frac{\exp(\theta_k^T x)}{\sum_{j=1}^{K} \exp(\theta_j^T x)}$$

6. Descent method

: 1st order Taylor series approximation

$$\Rightarrow J(\theta) \approx J(\theta_0) + (\theta - \theta_0)^T \nabla J(\theta_0)$$

① determine a descent direction

⇒ (batch) gradient descent

→ $\theta := \theta - t \sum_{i=1}^{n} \nabla J_i(\theta)$

⇒ stochastic gradient descent ⟶ converge faster / escape local optimum

→ approximate true gradient using a single random example

→ $\theta := \theta - t \cdot \nabla J_i(\theta)$

⇒ mini-batch (stochastic) gradient descent

→ approximate true gradient using several random examples

→ $\theta := \theta - t \cdot \sum_{i=1}^{m} \nabla J_i(\theta)$

(large batch gives a more accurate estimate)

(small batch can result in a regularization)

↕ trade-off

② determine a step size

⇒ exact line search

⟶ $t = \underset{t>0}{\operatorname{argmin}}\ J(\theta + t\Delta\theta)$

⇒ backtracking line search

⟶ set $t = 1$ at start

⟶ set $t = \beta t$ until $J(\theta + t\Delta\theta) < J(\theta) + \alpha \cdot t \nabla J(\theta)^T \Delta\theta$

7. Newton's method

: 2nd order Taylor series approximation

⇒ $J(\theta) \approx J(\theta_0) + (\theta - \theta_0)^T \nabla J(\theta_0) + \frac{1}{2}(\theta - \theta_0)^T \nabla^2 J(\theta_0)(\theta - \theta_0)$

⇒ $\nabla J(\theta) \Rightarrow \nabla J(\theta_0) + \nabla^2 J(\theta_0) \cdot (\theta^* - \theta_0) = 0$

⇒ $\theta := \theta - \nabla^2 J(\theta)^{-1} \cdot \nabla J(\theta)$

ML Lec 4.

- Support vector machine

Pros
- available public packages
- flexible kernel framework
- work well in small examples

Cons
- No direct multi-class SVM
- not scalable kernel framework

*. Linear classifier

: $p-1$ dimensional separating hyperplane that divides p-dimensional data.

*. Linearly separable

: data points are separable by a linear classifier

*. Max-margin classifier

: linear classifier that results in confident prediction for all training indices

⇓ maximal margin around separating hyperplane

* Lagrangian dual

Primal $\min_{w} f(w)$

$$\text{s.t. } g_i(w) \le 0 \quad i=1,\cdots,k$$

$$h_j(w) = 0 \quad j=1,\cdots,l$$

Dual $\max_{\alpha,\beta} L(w,\alpha,\beta)$

$$\text{s.t. } \alpha_i \ge 0 \quad i=1,\cdots,k$$

$$\Rightarrow L(w,\alpha,\beta) = f(w) + \sum_i \alpha_i g_i(w) + \sum_j \beta_j h_j(x)$$

⇒ optimal value of dual is $f(w)$ if primal constraints are satisfied

$$\Rightarrow \min_w \max_{\alpha,\beta} L(w,\alpha,\beta) \text{ s.t. } \alpha \ge 0 \ge \max_{\alpha,\beta} \boxed{\min_w L(w,\alpha,\beta)} \text{ s.t. } \alpha \ge 0$$

⇓ concave

1. Linear support vector machine (separable)

$$\Rightarrow \max_{w,b} \ \frac{2}{\|w\|}$$

$$\text{s.t.} \quad y_i(w^T x_i + b) \geq 1 \ , \quad 1 \leq i \leq n$$

$$\Rightarrow \min_{w,b} \ \frac{1}{2}\|w\|^2$$

$$\text{s.t.} \quad y_i(w^T x_i + b) \geq 1 \ , \quad 1 \leq i \leq n$$

$$\Rightarrow \max_{\alpha} \min_{w,b} \ L(w,b,\alpha) = \frac{1}{2}\|w\|^2 - \sum_i \alpha_i \left(y_i(w^T x_i + b) - 1 \right)$$

$$\text{s.t.} \quad \alpha_i \geq 0 \ , \quad 1 \leq i \leq n$$

$$\rightarrow \nabla_w L(w,b,\alpha) \Rightarrow w - \sum_i \alpha_i y_i x_i = 0 \rightarrow w = \sum_i \alpha_i y_i x_i$$

$$\rightarrow \frac{\partial}{\partial b} L(w,b,\alpha) \Rightarrow -\sum_i \alpha_i y_i = 0 \rightarrow \sum_i \alpha_i y_i = 0$$

$$\Rightarrow \max_{\alpha} \ \frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j y_i y_j x_i^T x_i - \sum_i \sum_j \alpha_i \alpha_j y_i y_j x_i^T x_j + b \cancel{\sum_i \alpha_i y_i} + \sum_i \alpha_i$$

$$\text{s.t.} \quad \alpha_i \geq 0 \ , \quad 1 \leq i \leq n$$

$$\sum_i \alpha_i y_i = 0$$

$$\Rightarrow \max_{\alpha} \ \sum_i \alpha_i - \frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j y_i y_j x_i^T x_j$$

$$\text{s.t.} \quad \alpha_i \geq 0 \ , \quad 1 \leq i \leq n$$

$$\sum_i \alpha_i y_i = 0$$

$\Rightarrow$ quadratic programmin

☆ $\alpha_i = 0$ for all x_i except for the support vectors $\rightarrow y_i(w^T x_i + b) - 1 = 0$

☆ depends on inner product of feature vectors

Step 1. find optimal α^*

Step 2. find optimal $w^* = \sum_i \alpha_i^* y_i x_i$

Step 3. find optimal $b^* = -\frac{\max_{i:y_i=-1} w^{*T} x_i + \min_{i:y_i=1} w^{*T} x_i}{2}$

2. Soft-margin linear support vector machine (non separable)

$$\Rightarrow \min_{w,b,\xi} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n} L(w,x_i,y_i) = \frac{1}{2}\|w\|^2 + C\underbrace{\max(0, 1-y_i(w^Tx_i+b))}_{\text{hinge loss}}$$

$$\Rightarrow \min_{w,b,\xi} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i$$

$$\text{s.t. } y_i(w^Tx_i+b) \geq 1-\xi_i, \quad 1\leq i\leq n$$

$$\xi_i \geq 0, \quad 1\leq i\leq n$$

$$\Rightarrow \max_{\alpha,\beta}\min_{w,b,\xi} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\alpha_i(y_i(w^Tx_i+b)-1+\xi_i) - \sum_{i=1}^{n}\beta_i\xi_i$$

$$\text{s.t. } \alpha_i \geq 0, \quad 1\leq i\leq n$$

$$\beta_i \geq 0, \quad 1\leq i\leq n$$

$$\rightarrow \nabla_w L(\alpha,\beta,w,b,\xi) \Rightarrow w - \sum_{i=1}^{n}\alpha_i y_i x_i = 0 \rightarrow w = \sum_{i=1}^{n}\alpha_i y_i x_i$$

$$\rightarrow \nabla_b L(\alpha,\beta,w,b,\xi) \Rightarrow -\sum_{i=1}^{n}\alpha_i y_i = 0 \rightarrow \sum_{i=1}^{n}\alpha_i y_i = 0$$

$$\rightarrow \nabla_\xi L(\alpha,\beta,w,b,\xi) \Rightarrow C-\alpha_i-\beta_i = 0 \rightarrow \beta_i = C-\alpha_i$$

$$\Rightarrow \max_{\alpha,\beta} \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_i\sum_j \alpha_i\alpha_j y_i y_j x_i^T x_j$$

$$\text{s.t. } 0\leq\alpha_i\leq C, \quad 1\leq i\leq n$$

$$\sum \alpha_i y_i = 0$$

☆ The original input space can always map to some high-dimensional space where the training data is linearly separable

(lifting transformation : $\phi(x)$)

* kernel trick

: define a kernel, $K(x_i,x_j) = \phi(x_i)\cdot\phi(x_j)$ where $\phi(x)$ need not to be known

* Mercer's condition → condition for valid kernel

: $\forall g(x)$ s.t. $\underbrace{\int_{-\infty}^{+\infty}|g(x)|^2dx < \infty}_{\text{square integrable}}$, $\iint K(x_i,x_j)g(x_i)g(x_j)\,dxdy \geq 0$

★ Kernel matrix is positive semi-definite

pf) $$\iint k(x_i, x_j) g(x_i) g(x_j)\, dx_i dx_j = \iint \phi(x_i)^T \phi(x_j)\, g(x_i) g(x_j)\, dx_i dx_j$$
$$= \iiint \phi_k(x_i)\, \phi_k(x_j)\, dk\; g(x_i) g(x_j)\, dx_i dx_j$$
$$= \iiint \left(\phi_k(x_i) \cdot g(x_i)\right)\left(\phi_k(x_j)\, g(x_j)\right) dx_i dx_j dk$$
$$= \int \left(\int \phi_k(x_i)\, g(x_i)\, dx_i\right)^2 dk \geq 0$$

* Mercer's theorem

: if Mercer's condition is satisfied, it is necessary and sufficient that for any $\{x_1, \cdots x_m\}$ the corresponding kernel is symmetric and positive semi definite.

(symmetric and positive semi-definite matrix is a valid kernel)

3. Nonlinear support vector machine

⇒ replace inner product to kernel

⇒ $$\max_{\alpha} \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j\, \phi(x_i)^T \phi(x_j)$$

s.t. $\alpha_i \geq 0$, $1 \leq i \leq n$

$$\sum_i \alpha_i y_i = 0$$

* Multi-class classification

⟶ one to one classifier : A-B, A-C, B-C ⟶ voting

⟶ one to all classifier : $A-A^c$, $B-B^c$, $C-C^c$

ML Lec 5.

- Bias - Variance trade-off

1. Bias ($L(y^m, f(x))$) ⟶ underfitting

: discrepancy between the averaged function estimator and the true function

: the loss of the main prediction w.r.t the true label of x

⇒ source
- inability to represent certain decision boundaries
- incorrect assumptions on data
- too global model

2. Variance ($\mathbb{E}[L(h(x), y^m)]$) ⟶ overfitting

: discrepancy between models trained on different training sets

: the expected loss of $h(x)$ relative to the main prediction

⇒ source
- sharp decision making model
- randomization in model
- too local model

3. Noise ($\mathbb{E}[L(y, f(x))]$)

: irreducible error that describes how y varies from $f(x)$

: the expected loss of the noisy observed value y relative to the true label of x

☆ Bias - Variance decomposition (squared loss)

(Main prediction (y^m) : $\overline{h(x)}$)

$$\text{Bias}[h(x)] = \overline{h(x)} - f(x)$$

$$\text{Var}[h(x)] = \mathbb{E}\left[(h(x) - \overline{h(x)})^2\right]$$

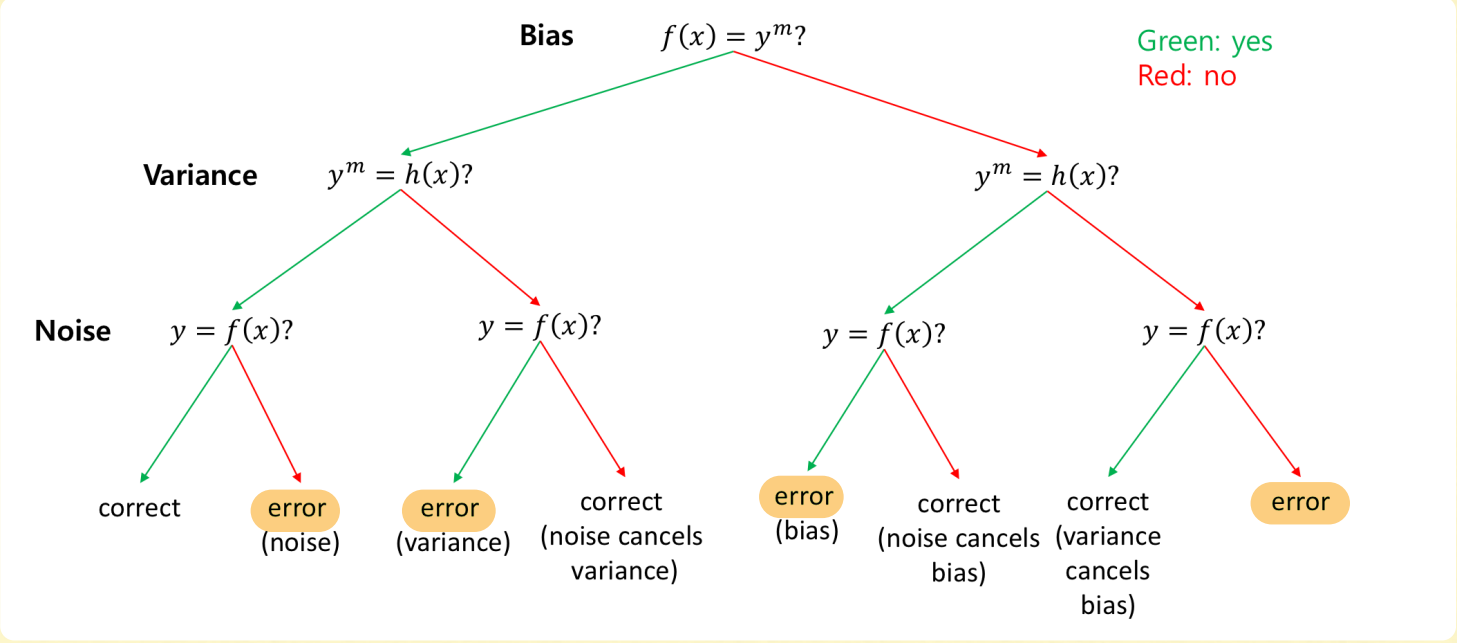
$$\text{Noise}[h(x)] = \mathbb{E}\left[(y - f(x))^2\right] = \mathbb{E}[\varepsilon^2] = \sigma^2 \longrightarrow \text{not dependent on } h(x)$$

$$\begin{aligned}
\mathbb{E}\left[(y - h(x))^2\right] &= \mathbb{E}\left[y^2 - 2yh(x) + h(x)^2\right] \\
&= \mathbb{E}[y^2] - 2\mathbb{E}[y]\cdot\overline{h(x)} + \mathbb{E}[h(x)^2] \\
&= \mathbb{E}[y^2] - 2f(x)\cdot\overline{h(x)} + \mathbb{E}\left[(h(x) - \overline{h(x)})^2\right] + \overline{h(x)}^2 \\
&= \mathbb{E}[(y - f(x))^2] + f(x)^2 - 2f(x)\cdot\overline{h(x)} + \mathbb{E}\left[(h(x) - \overline{h(x)})^2\right] + \overline{h(x)}^2 \\
&= \left(\overline{h(x)} - f(x)\right)^2 + \mathbb{E}\left[(h(x) - \overline{h(x)})^2\right] + \mathbb{E}\left[(y - f(x))^2\right] \\
&= \underbrace{\text{Bias}[h(x)]^2 + \text{Var}[h(x)]}_{\text{reducible}} + \underbrace{\text{Noise}[h(x)]}_{\text{irreducible}}
\end{aligned}$$

☆ Bias - Variance decomposition (0-1 loss)

(Main prediction (y^m): most common vote of $h(x)$)

- $\text{Bias}[h(x)] = \mathbb{I}(f(x) \neq y^m)$ ⟶ either 0 (unbiased) or 1 (biased)
- $\text{Var}[h(x)] = \Pr(h(x) \neq y^m) = \sigma$
- $\text{Noise}[h(x)] = \Pr(y \neq f(x)) = \tau$



(even number of green lines occurs error.)

① Biased ⇒ $L(h(x), y) = (1-\sigma)\tau + \sigma(1-\tau) = \sigma + \tau - 2\sigma\tau$

② Unbaised ⇒ $L(h(x), y) = (1-\sigma)(1-\tau) + \sigma\tau = 1 - (\sigma + \tau - 2\sigma\tau)$

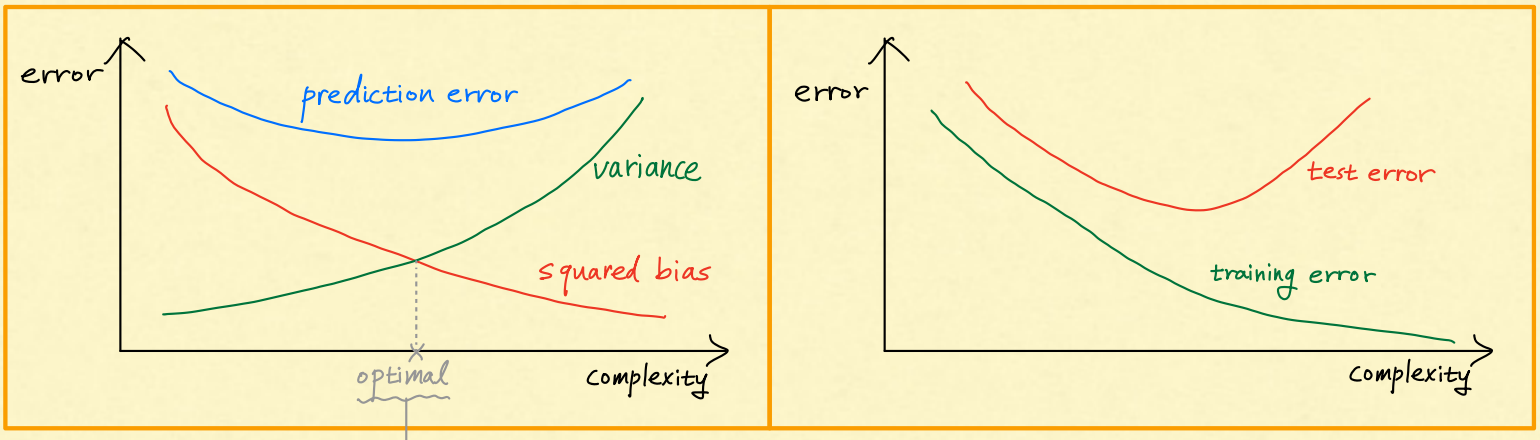
$$\therefore L(h(x), y) = \begin{cases} \text{Bias} + (\text{Variance} + \text{Noise} - 2 \cdot \text{Variance} \cdot \text{Noise}) & \text{if unbiased} \\ \text{Bias} - (\text{Variance} + \text{Noise} - 2 \cdot \text{Variance} \cdot \text{Noise}) & \text{if biased} \end{cases}$$

⟶ if unbiased, variance and noise increases the loss

⟶ if biased, variance and noise reduces the loss

(⇒ 0-1 loss has higher tolerance for variance than squared loss)

[increased loss due to variance in unbiased examples
  decreased loss due to variance in biased examples) offset



* Bagging

: train k models on different samples and average or vote their prediction.

⇒ results in model with smaller variance

* Regularization

: add penalty term that can be used to balance bias and variance

⇒ results in higher bias and lower variance

ML Lec 6.

- PAC learning

  ⇒ define the class of learnable concepts w.r.t. the number of sample points, sample complexity, and the time, space complexity of learning algorithm.

  - no assumption is made about the distribution D from which examples are drawn
  - the training and test examples are drawn from the same distribution D.
  - the concept set is known but the target concept is unknown

  * concept ($c : X \to Y$)

    : mapping from the input space to labels

  * concept set (C)

    : the set of concept that we wish to learn

  * Generalization error ($R(h)$)

    ⇒ $\Pr_{x \sim D}[h(x) \neq c(x)] = \mathbb{E}_{x \sim D}[I(h(x) \neq c(x))]$ → expectation over distribution

  * Empirical error ($\hat{R}(h)$)

    ⇒ $\frac{1}{m}\sum_{i=1}^{m} I(h(x_i) \neq c(x_i))$ → average over samples

  * Generalization bound ($\approx$ sample complexity bound)

    : with probability at least $1-\delta$, $R(h)$ is upper-bounded by some quantity that depends on the sample size m and δ.

1. PAC-learnable($\hat{C}$) ↦ PAC learning algorithm for $\hat{C}$

: there exists an algorithm A and a polynomial function poly such that

for any $\varepsilon > 0$ and $\delta > 0$, for all distributions D on X and for any target concept $c \in C$,

the following holds for any sample size $m \geq \text{poly}\left(\frac{1}{\varepsilon}, \frac{1}{\delta}, n, \text{size}(c)\right)$

$\longrightarrow \Pr_{S \sim D^m}[R(h_S) \leq \varepsilon] \geq 1 - \delta$ ($\Leftrightarrow$ algorithm is approximately correct with high probability)

(efficiently PAC learnable if A further runs in $\text{poly}\left(\frac{1}{\varepsilon}, \frac{1}{\delta}, n, \text{size}(c)\right)$)

(ε : error, δ : failure probability)

(n : $\dim(X)$, $\text{size}(c)$: $\log|\hat{C}|$) → omitted during analysis

(m : sample complexity)

Ex) Axis-aligned rectangles

$\Rightarrow X = \mathbb{R}^2$, C is the set of all axis-aligned rectangles lying on $\mathbb{R}^2$

$\Rightarrow C$ is PAC-learnable

(A is the tightest axis-alinged rectangles containing the true label points)

(errors always occur as False-Negative)

pf) Let $Pr(R) > \varepsilon$ and $Pr(r_i) \geq \frac{\varepsilon}{4}$

① if $R(R') > \varepsilon$, then R' must not overlap with at least one r_i

② if all m examples are not drawn from r_i, then the error happens on r_i.

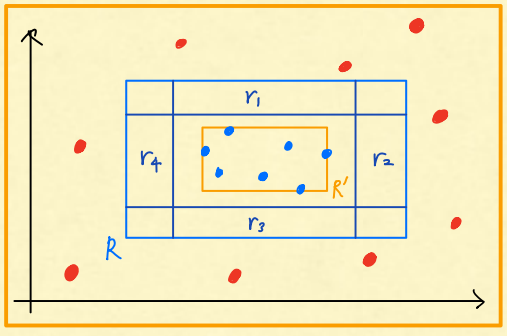
③ $n = 2$, $size(c) = 4$ ⟶ constant.

$$\Pr_{S\sim D^m}[R(R') > \varepsilon] \leq \Pr_{S\sim D^m}\left[\bigcup_{i=1}^{4}\{R' \cap r_i = \phi\}\right]$$

$$\leq \sum_{i=1}^{4} \Pr_{S\sim D^m}[R' \cap r_i = \phi]$$

$$\leq 4 \cdot \left(1 - \frac{\varepsilon}{4}\right)^m$$

$$\leq 4 \cdot \exp\left(-m \cdot \frac{\varepsilon}{4}\right) \leq \delta$$



$$\longrightarrow m \geq \frac{4}{\varepsilon}\log\frac{4}{\delta}$$

$$\longrightarrow \varepsilon \geq \frac{4}{m}\log\frac{4}{\delta}$$

$$\Rightarrow \Pr_{S\sim D^m}[R(R') > \varepsilon] \leq \delta \text{ if } m \geq \frac{4}{\varepsilon}\log\frac{4}{\delta}$$

PAC-learnable ↑

$$\therefore \begin{cases} \Pr_{S\sim D^m}[R(R') \leq \varepsilon] \geq 1-\delta \text{ if } m \geq \frac{4}{\varepsilon}\log\frac{4}{\delta} = \text{poly}\left(\frac{1}{\varepsilon}, \frac{1}{\delta}\right) \\ \Pr_{S\sim D^m}[R(R') \leq \frac{4}{m}\log\frac{4}{\delta}] \geq 1-\delta \longrightarrow \text{generalization bound} \end{cases}$$

Problem : The hypothesis returned by the algorithm is consistent

$\Rightarrow$ it admits no error on the training sample S

$\Rightarrow \hat{R}(h_S) = 0$

ML Lec 7.

- Learning bound

  * consistent (c) : error on training sample S is not admitted
  * conjunction ($\wedge$) : true only if all true
  * disjunction ($\vee$) : false only if all false

1. Learnable ($\mathcal{C}$)

   : there is an algorithm A that finds a concept $c \in \mathcal{C}$ that is consistent with the examples given any set of labels $(x_1, y_1), \cdots, (x_m, y_m)$

2. **Theorem 1** Finite H, Consistent case

   For any $\varepsilon, \delta > 0$, $\Pr_{S \sim D^m}[R(h_S) \le \varepsilon] \ge 1-\delta$, if $m \ge \frac{1}{\varepsilon}\left(\log|H| + \log\frac{1}{\delta}\right)$

   $\Rightarrow$ Generalization error bound increases with $|H|$ and decreases with m, δ

pf) For the worst case analysis, the proof will bound the probability that some $h \in H$ would be consistent and have error more than ε.

Let $H_\varepsilon = \{h \in H : R(h) > \varepsilon\}$

$$\begin{aligned}
\Pr_{S\sim D^m}[R(h) > \varepsilon] &\le \Pr_{S\sim D^m}[\exists h \in H_\varepsilon : \hat{R}(h) = 0] \\
&= \Pr_{S\sim D^m}\left[\bigvee_{i=1}^{|H_\varepsilon|} \{\hat{R}(h_i) = 0\}\right] \\
&\le \sum_{h\in H_\varepsilon} \Pr_{S\sim D^m}[\hat{R}(h) = 0] \\
&\le \sum_{h\in H_\varepsilon} (1-\varepsilon)^m \\
&\le |H|\cdot(1-\varepsilon)^m \le |H|\exp(-m\varepsilon) \le \delta
\end{aligned}$$

$\longrightarrow m \ge \frac{1}{\varepsilon}\left(\log|H| + \log\frac{1}{\delta}\right)$

$\longrightarrow \varepsilon \ge \frac{1}{m}\left(\log|H| + \log\frac{1}{\delta}\right)$

$\Rightarrow \Pr_{S\sim D^m}[R(h) > \varepsilon] \le \delta$ if $m \ge \frac{1}{\varepsilon}\left(\log|H| + \log\frac{1}{\delta}\right)$

PAC-learnable ↑

$\therefore$ $\Pr_{S\sim D^m}[R(h) \le \varepsilon] \ge 1-\delta$ if $m \ge \frac{1}{\varepsilon}\left(\log|H| + \log\frac{1}{\delta}\right) = \text{poly}\left(\frac{1}{\varepsilon}, \frac{1}{\delta}\right)$

$\Pr_{S\sim D^m}\left[R(h) \le \frac{1}{m}\left(\log|H| + \log\frac{1}{\delta}\right)\right] \ge 1-\delta \longrightarrow O\left(\frac{1}{m}\log|H|\right)$

For consistent algorithm, a larger hypothesis set is needed

⇒ but this increases the upper bound

(Ex) Conjunction of (at most n) boolean literals

⇒ boolean literal is either a variable or its negation

⇒ a negative example is non-informative

(we cannot tell which bits are incorrect)

⇒ Learning algorithm (A)

: for each positive example $(b_1, \cdots, b_n)$, rule out incompatible literals

⟶ if $b_i = 0$, rule out x_i and if $b_i = 1$, rule out $\hat{x}_i$

: conjunction of all literals not ruled out is a consistent hypothesis

⇒ since each literal can be positive / negative / excluded, $|H| = 3^n$

⟶ $m \geq \frac{1}{\varepsilon}\left((\log 3)n + \log\frac{1}{\delta}\right) = \text{poly}\left(\frac{1}{\varepsilon}, \frac{1}{\delta}, n\right)$

⇒ training cost per example is $O(n)$ ↓ PAC-learnable

(effectively PAC-learnable)

(Ex) Universal concept class

⇒ $X = \{0,1\}^n$, the boolean vectors with n component

⟶ $|X| = 2^n$

⇒ U_n, the concept class formed by all subsets of X

⟶ $|U_n| = 2^{|X|}$

⇒ To guarantee a consistent hypothesis, H must include U_n

⟶ $|H| \geq |U_n| = 2^{2^n}$ not PAC-learnable

⟶ $m \geq \frac{1}{\varepsilon}\left((\log 2)\cdot 2^n + \log\frac{1}{\delta}\right) = \text{poly}\left(\frac{1}{\varepsilon}, \frac{1}{\delta}, 2^n\right)$

(Ex) k-term DNF formula

⇒ Disjunctive Normal Form (DNF)

: a formula written as the disjunction of several terms, each term being a conjunction of Boolean literals.

⇒ k-term DNF

: DNF formula defined by the disjunction of k-terms, each term being a conjunction of at most n Boolean literals.

⇒ To guarantee a consistent hypothesis, H must include C_{nk}

⟶ $|H| \geq |C_{nk}| = 3^{nk}$ PAC-learnable

⟶ $m \geq \frac{1}{\varepsilon}\left((\log 3)\cdot nk + \log\frac{1}{\delta}\right) = \text{poly}\left(\frac{1}{\varepsilon}, \frac{1}{\delta}, n\right)$

⇒ training cost per example is $O(nk)$ ⟶ NP-hard

(not effectively PAC-learnable)

Ex) k-CNF formula

⇒ Conjunctive Normal Form (CNF)

: a formula written as the conjunction of several terms, each term being a disjunction of Boolean literals.

⇒ k-CNF

: CNF formula defined by the conjunction of arbitrary length $i \in N$, each term being a disjunction of at most k Boolean literals.

→ we can reduce the learning of k-CNF into the learning of conjunction of Boolean literals by introducing $(2n)^k$ new variables

$(a_i(x_1) \vee \cdots \vee a_i(x_n) \rightarrow Y_{a_i(x_1)\cdots a_i(x_n)})$

→ it may change the distribution D, but PAC learnability stays the same

(PAC-learnable, efficiently PAC learnable)

☆ k-term DNF can be rewritten as a k-CNF formula

⇒ $\bigvee_{i=1}^{k} a_i(x_1) \wedge \cdots \wedge a_i(x_n) = \bigwedge_{i_1,\cdots,i_k=1}^{n} a_1(x_1) \vee \cdots \vee a_k(x_{i_k})$ (∵ associativity)

ex) $(a \wedge b) \vee (c \wedge d) = (a \vee c) \wedge (a \vee d) \wedge (b \vee c) \wedge (b \vee d)$

⇒ the number of new variables needed is $O(n^k)$

⇒ k-term DNF is exponentially more compact than k-CNF

→ k-term DNF : not efficiently PAC-learnable

→ k-CNF : efficiently PAC-learnable

∴ Cost of representation of a concept & choice of the hypothesis test = the key concept of PAC-learning

3. **Theorem 2** Hoeffding's inequality (for Bernoulli R.V.)

For independent Bernoulli random variable $X_1, \cdots, X_m$, and for any $\varepsilon > 0$,

$\Pr\left[\left|\frac{1}{m}\sum_{i=1}^{m} X_i - \mathbb{E}[X]\right| \geq \varepsilon\right] \leq 2\exp(-2m\varepsilon^2)$ → Hoeffding's bound.

⇒ provide an upper bound on the probability that the mean of a sample of independent random variables deviate from its expected value

⇒ the probability decays exponentially with m and ε^2

⇒ it does not use any properties of distribution, but calculated only upon m, ε^2

Corollary 1

Fix $\varepsilon > 0$ and let S denote an i.i.d. sample of size m. Then the following holds

$\Pr_{S\sim D^m}\left[\hat{R}(h) - R(h) > \varepsilon\right] \leq \exp(-2m\varepsilon^2)$

$\Pr_{S\sim D^m}\left[\hat{R}(h) - R(h) < -\varepsilon\right] \leq \exp(-2m\varepsilon^2)$

→ $\Pr_{S\sim D^m}\left[\left|\hat{R}(h) - R(h)\right| > \varepsilon\right] \leq 2\exp(-2m\varepsilon^2) \leq \delta$

→ $m \geq \frac{1}{2\varepsilon^2}\log\frac{2}{\delta}$

→ $\varepsilon \geq \sqrt{\frac{1}{2m}\log\frac{2}{\delta}}$

⇒ $\Pr_{S\sim D^m}\left[\left|\hat{R}(h) - R(h)\right| > \varepsilon\right] \leq \delta$ if $m \geq \frac{1}{2\varepsilon^2}\log\frac{2}{\delta}$

$\therefore \begin{cases} \Pr_{S\sim D^m}\left[\left|\hat{R}(h) - R(h)\right| \leq \varepsilon\right] \geq 1-\delta \text{ if } m \geq \frac{1}{2\varepsilon^2}\log\frac{2}{\delta} \\ \Pr_{S\sim D^m}\left[\left|\hat{R}(h) - R(h)\right| \leq \sqrt{\frac{1}{2m}\log\frac{2}{\delta}}\right] \geq 1-\delta \end{cases}$

Corollary 2

For any $\delta > 0$, $\Pr_{S\sim D^m}\left[\left|\hat{R}(h) - R(h)\right| \leq \sqrt{\frac{1}{2m}\log\frac{2}{\delta}}\right] \geq 1-\delta$, $\forall h \in H$

4. **Theorem 3** Finite H, Inconsistent case

For any $\delta > 0$, $\Pr\left[R(h) \le \hat{R}(h) + \sqrt{\frac{1}{2m}\left(\log|H| + \log\frac{2}{\delta}\right)}\right] \ge 1-\delta$

$\Rightarrow$ larger m guarantees better generalization, but need quadratically larger m to attain the same guarantee as in the consistent case

$\Rightarrow$ large $|H|$ increases the upper bound but decreases empirical error (= trade-off)

pf) Let $H = \{h_1, h_2, \cdots, h_{|H|}\}$ and $H_\varepsilon = \{h \in H : |\hat{R}(h) - R(h)| > \varepsilon\}$

$$\begin{aligned}
\Pr\left[R(h) - \hat{R}(h) > \varepsilon\right] &\le \Pr\left[\exists h \in H : |\hat{R}(h) - R(h)| > \varepsilon\right] \\
&= \Pr\left[\bigvee_{i=1}^{|H|} |\hat{R}(h_i) - R(h_i)| > \varepsilon\right] \\
&\le \sum_{h \in H} \Pr\left[|\hat{R}(h) - R(h)| > \varepsilon\right] \\
&\le |H| \cdot 2\exp(-2m\varepsilon^2) \le \delta
\end{aligned}$$

$$\longrightarrow m \ge \frac{1}{2\varepsilon^2}\left(\log|H| + \log\frac{2}{\delta}\right)$$

$$\longrightarrow \varepsilon \ge \sqrt{\frac{1}{2m}\left(\log|H| + \log\frac{2}{\delta}\right)}$$

$\Rightarrow \Pr\left[R(h) - \hat{R}(h) > \varepsilon\right] \le \delta$ if $m \ge \frac{1}{2\varepsilon^2}\left(\log|H| + \log\frac{2}{\delta}\right)$

$\therefore$ $\Pr\left[R(h) \le \hat{R}(h) + \varepsilon\right] \ge 1-\delta$ if $m \ge \frac{1}{2\varepsilon^2}\left(\log|H| + \log\frac{2}{\delta}\right)$

$\Pr\left[R(h) \le \hat{R}(h) + \sqrt{\frac{1}{2m}\left(\log|H| + \log\frac{2}{\delta}\right)}\right] \ge 1-\delta \longrightarrow O\left(\sqrt{\frac{1}{m}\log|H|}\right)$

5. Occam's Razor

: Plurality should not be posited without necessity.

$\Rightarrow$ the simplest explanation is the best

$\Rightarrow$ to minimize true error, choose the most parsimonious explanation (smallest $|H|$)

- Generalities

  *. Deterministic scenario.

  : the label of a point can be uniquely determined by some measurable function

  ⇒ there exists a target function with no generalization error ($R^* = 0$)

  *. Stochastic scenario

  : the label is a probabilistic function of the input.

  ⇒ there exists a minimal non-zero error for any hypothesis ($R^* \neq 0$)

  * Agnostic PAC-learnable (C)

  : there exists an algorithm A and a polynomial function poly such that

  for any $\varepsilon > 0$ and $\delta > 0$, for all distributions D on $X \times Y$,

  the following holds for any sample size $m \geq poly\left(\frac{1}{\varepsilon}, \frac{1}{\delta}, n, size(c)\right)$

  → $\Pr_{S \sim D^m}\left[R(h_S) - \min_{h \in H} R(h) \leq \varepsilon\right] \geq 1 - \delta$ → based on estimation error

  (efficiently agnostic PAC learnable if A further runs in $poly\left(\frac{1}{\varepsilon}, \frac{1}{\delta}, n, size(c)\right)$)

  * Bayes error (R^*)

  : the infimum of the errors achieved by measurable functions $h : X \to Y$

  ⇒ $R^* = \inf_h R(h)$

  * Bayes hypothesis (≈ Bayes classifier) (h_{Bayes})

  : a hypothesis that achieves Bayes error

  ⇒ $\forall x \in X,\ h_{Bayes}(x) = \underset{y \in \{0,1\}}{\operatorname{argmax}} \Pr[y|x]$

  ⇒ $R(h_{Bayes}) = R^*$

  * Noise $= \mathbb{E}[Noise(x)] = R^*$

  ⇒ $Noise(x) = \min\{\Pr(0|x), \Pr(1|x)\}$

  ⇒ characteristic of the learning task indicative of its level of difficulty

  (if noise = 0.5, it's challenging)

☆ $R(h) - R^* = R(h) - R(h^*) + R(h^*) - R^*$

⇒ Estimation error $= R(h) - R(h^*)$

: the quality of hypothesis h w.r.t. the best-in-class hypothesis

⇒ h^* is a hypothesis in H with minimal error ($\approx$ best-in-class hypothesis)

⇒ $$\begin{aligned} R(h_s^{ERM}) - R(h^*) &= R(h_s^{ERM}) - \hat{R}(h_s^{ERM}) + \hat{R}(h_s^{ERM}) - R(h^*) \\ &\le R(h_s^{ERM}) - \hat{R}(h_s^{ERM}) + \hat{R}(h^*) - R(h^*) \\ &\le 2 \cdot \sup_{h \in H} |R(h) - \hat{R}(h)| \\ &\le 2\sqrt{\frac{1}{2m}\left(\log |H| + \log \frac{2}{\delta}\right)} \quad \text{with probability at least } 1-\delta \end{aligned}$$

⇒ Approximation error $= R(h^*) - R^*$

: how well the Bayes error can be approximated using H ($\approx$ H's richness)

⇒ unaccessable as the underlying distribution is unknown

ML Lec 8.

- Rademacher complexity

  ☆ Finding h that minimizes empirical risk maximizes the correlation and vice-versa

  pf) $\hat{R}(h) = \frac{1}{m}\sum_{i=1}^{m} I(h(x_i) \neq c(x_i)) = \frac{1}{m}\sum_{i=1}^{m} \frac{1 - y_i h(x_i)}{2} = \frac{1}{2} - \frac{1}{2m}\sum_{i=1}^{m} y_i h(x_i)$

  $\Rightarrow \arg\min_h \hat{R}(h) = \arg\max_h \sum_{i=1}^{m} y_i h(x_i)$

  * Rademacher random variable $(\sigma_i) = \begin{cases} +1 & \text{with probability } \frac{1}{2} \\ -1 & \text{with probability } \frac{1}{2} \end{cases}$

  1. Empirical Rademacher complexity $(\hat{\mathcal{R}}_S(\mathcal{G}))$ $(S = (z_1, \cdots, z_m))$

     : the degree a hypothesis set correlate with a random noise on average over S

     (captures the richness and expressiveness of a family of functions.)

     ① $g \in \mathcal{G} : Z \to \{-1, 1\}$

     $\Rightarrow \hat{\mathcal{R}}_S(\mathcal{G}) = \mathbb{E}_\sigma\left[\max_{g \in \mathcal{G}} \frac{1}{m}\sum_{i=1}^{m} \sigma_i g(z_i)\right]$

     $\longrightarrow$ if $|\mathcal{G}| = 1$, $\mathbb{E}_\sigma\left[\max_{g \in \mathcal{G}} \frac{1}{m}\sum_{i=1}^{m} \sigma_i g(z_i)\right] = 0$

     $\longrightarrow$ if $|\mathcal{G}| = 2^m$, $\mathbb{E}_\sigma\left[\max_{g \in \mathcal{G}} \frac{1}{m}\sum_{i=1}^{m} \sigma_i g(z_i)\right] = 1$

     $\Rightarrow$ range $= [0, 1]$

     ② $g \in \mathcal{G} : Z \to [a, b]$

     $\Rightarrow \hat{\mathcal{R}}_S(\mathcal{G}) = \mathbb{E}_\sigma\left[\sup_{g \in \mathcal{G}} \frac{1}{m}\sum_{i=1}^{m} \sigma_i g(z_i)\right]$

  2. Rademacher complexity $(\mathcal{R}_m(\mathcal{G}))$

     : the expectation of empirical Rademacher complexity over all samples of size m

     $\Rightarrow \mathcal{R}_m(\mathcal{G}) = \mathbb{E}_{S \sim D^m}\left[\hat{\mathcal{R}}_S(\mathcal{G})\right]$

3. McDiarmid's inequality ⟶ generalization of Hoeffding's inequality

: For independent random variable $X_1, \cdots, X_m$, assume that there exists $c_1, \cdots, c_m > 0$ s.t.

$|f(x_1, \cdots, x_i, \cdots, x_m) - f(x_1, \cdots, x_i', \cdots, x_m)| \le c_i$ ⟶ changing argument has limited effect

Let $f(S)$ denotes $f(X_1, \cdots, X_m)$. Then for any $\varepsilon > 0$, the followings hold.

$$Pr[f(S) - \mathbb{E}[f(S)] \ge \varepsilon] \le \exp\left(-\frac{2\varepsilon^2}{\sum_{i=1}^m c_i^2}\right) \le \frac{\delta}{2}$$

$$Pr[f(S) - \mathbb{E}[f(S)] \le -\varepsilon] \le \exp\left(-\frac{2\varepsilon^2}{\sum_{i=1}^m c_i^2}\right) \le \frac{\delta}{2}$$

$$\Rightarrow Pr\left[f(S) \le \mathbb{E}[f(S)] + \sqrt{\frac{1}{2}\sum_{i=1}^m c_i^2 \log\frac{2}{\delta}}\right] \ge 1 - \frac{\delta}{2}$$

$$\Rightarrow Pr\left[\mathbb{E}[f(S)] \le f(S) + \sqrt{\frac{1}{2}\sum_{i=1}^m c_i^2 \log\frac{2}{\delta}}\right] \ge 1 - \frac{\delta}{2}$$

4. **Theorem**

Let $g \in G : Z \to [0,1]$

For any $\delta > 0$, $\forall g \in G$, the followings hold with probability at least $1 - \frac{\delta}{2}$

$$\mathbb{E}[g(z)] \le \frac{1}{m}\sum_{i=1}^m g(z_i) + 2\mathcal{R}_m(G) + \sqrt{\frac{1}{2m}\log\frac{1}{\delta}}$$

$$\mathbb{E}[g(z)] \le \frac{1}{m}\sum_{i=1}^m g(z_i) + 2\hat{\mathcal{R}}_S(G) + 3\sqrt{\frac{1}{2m}\log\frac{2}{\delta}}$$

⇒ Rademacher complexity plays a role of the size of the hypothesis test.

pf) Let $\hat{\mathbb{E}}_S[g] = \frac{1}{m}\sum_{i=1}^m g(z_i)$, $\phi(S) = \sup_{g \in G} \mathbb{E}[g] - \hat{\mathbb{E}}_S[g]$

Let S, S' be two samples differing by exactly one point. ($z_m \in S$, $z_m' \in S'$)

$$\phi(S') - \phi(S) \le \sup_{g \in G} \hat{\mathbb{E}}_S[g] - \hat{\mathbb{E}}_{S'}[g] = \sup_{g \in G} \frac{1}{m}(g(z_m) - g(z_m')) \le \frac{1}{m}$$

$$\Rightarrow |\phi(S') - \phi(S)| \le \frac{1}{m}$$

$$\Rightarrow Pr\left[\phi(S) \le \mathbb{E}_S[\phi(S)] + \sqrt{\frac{1}{2m}\log\frac{2}{\delta}}\right] \ge 1 - \frac{\delta}{2}$$ (∵ McDiarmid's inequality)

(continued)

$$
\begin{aligned}
\mathbb{E}[\phi(S)] &= \mathbb{E}_S\Big[\sup_{g\in G} \mathbb{E}[g] - \hat{\mathbb{E}}_S[g]\Big] \\
&= \mathbb{E}_S\Big[\sup_{g\in G} \mathbb{E}_{S'}[\hat{\mathbb{E}}_{S'}[g]] - \hat{\mathbb{E}}_S[g]\Big] \\
&= \mathop{\mathbb{E}}_{S,S'}\Big[\sup_{g\in G} \hat{\mathbb{E}}_{S'}[g] - \hat{\mathbb{E}}_S[g]\Big] \\
&= \mathop{\mathbb{E}}_{S,S'}\Big[\sup_{g\in G} \frac{1}{m}\sum_{i=1}^{m} (g(z_i') - g(z_i))\Big] \\
&= \mathop{\mathbb{E}}_{\sigma,S,S'}\Big[\sup_{g\in G} \frac{1}{m}\sum_{i=1}^{m} \sigma_i (g(z_i') - g(z_i))\Big] \\
&\le \mathop{\mathbb{E}}_{\sigma,S'}\Big[\sup_{g\in G} \frac{1}{m}\sum_{i=1}^{m} \sigma_i\, g(z_i')\Big] + \mathop{\mathbb{E}}_{\sigma,S}\Big[\sup_{g\in G} \frac{1}{m}\sum_{i=1}^{m} (-\sigma_i) g(z_i)\Big] \\
&= 2\mathop{\mathbb{E}}_{\sigma,S}\Big[\sup_{g\in G} \frac{1}{m}\sum_{i=1}^{m} \sigma_i\, g(z_i)\Big] = 2\mathcal{R}_m(G)
\end{aligned}
$$

① $\Rightarrow \Pr\Big[\phi(S) \le 2\mathcal{R}_m(G) + \sqrt{\frac{1}{2m}\log\frac{2}{\delta}}\Big] \ge 1 - \frac{\delta}{2}$

$\Rightarrow \Pr\Big[\sup_{g\in G} \mathbb{E}[g] - \hat{\mathbb{E}}_S[g] \le 2\mathcal{R}_m(G) + \sqrt{\frac{1}{2m}\log\frac{2}{\delta}}\Big] \ge 1 - \frac{\delta}{2}$

$\Rightarrow \Pr\Big[\mathbb{E}[g] \le \frac{1}{m}\sum_{i=1}^{m} g(z_i) + 2\mathcal{R}_m(G) + \sqrt{\frac{1}{2m}\log\frac{2}{\delta}}\Big] \ge 1 - \frac{\delta}{2}$

$\hat{\mathcal{R}}_{S'}[G] - \hat{\mathcal{R}}_S[G] \le \mathop{\mathbb{E}}_{\sigma}\Big[\sup_{g\in G} \frac{1}{m}\sum_{i=1}^{m} \sigma_i (g(z_i') - g(z_i))\Big] \le \frac{1}{m}$

② $\Rightarrow \Pr\Big[\mathcal{R}_m[G] \le \hat{\mathcal{R}}_S[G] + \sqrt{\frac{1}{2m}\log\frac{2}{\delta}}\Big] \ge 1 - \frac{\delta}{2}$ ($\because$ McDiarmid's inequality)

①+2×② $\Rightarrow \Pr\Big[\phi(S) \le 2\hat{\mathcal{R}}_S[G] + 3\cdot\sqrt{\frac{1}{2m}\log\frac{2}{\delta}}\Big] \ge 1 - \delta$

$\Rightarrow \Pr\Big[\sup_{g\in G} \mathbb{E}[g] - \hat{\mathbb{E}}_S[g] \le 2\hat{\mathcal{R}}_S(G) + 3\sqrt{\frac{1}{2m}\log\frac{2}{\delta}}\Big] \ge 1 - \delta$

$\Rightarrow \Pr\Big[\mathbb{E}[g] \le \frac{1}{m}\sum_{i=1}^{m} g(z_i) + 2\hat{\mathcal{R}}_S(G) + 3\sqrt{\frac{1}{2m}\log\frac{2}{\delta}}\Big] \ge 1 - \delta$

5. **Lemma**

Let H be a family of functions taking values in $\{-1, 1\}$

Let G be a family of loss functions associated with H for the 0-1 loss

Let $S_X = (x_1, \cdots, x_m)$ be projection of $S = ((x_1, y_1), \cdots, (x_m, y_m))$ over X.

$\hat{\mathcal{R}}_S(G) = \frac{1}{2}\hat{\mathcal{R}}_{S_X}(H)$ $\xrightarrow{\text{expectation}}$ $\mathcal{R}_m(G) = \frac{1}{2}\mathcal{R}_m(H)$

pf) $\hat{\mathcal{R}}_S(G) = \mathop{\mathbb{E}}_{\sigma}\left[\sup_{h\in H} \frac{1}{m}\sum_{i=1}^{m} \sigma_i I(h(x_i) \neq y_i)\right]$

$= \mathop{\mathbb{E}}_{\sigma}\left[\sup_{h\in H} \frac{1}{m}\sum_{i=1}^{m} \sigma_i \frac{1 - y_i h(x_i)}{2}\right]$

$= \frac{1}{2}\mathop{\mathbb{E}}_{\sigma}\left[\sup_{h\in H} \frac{1}{m}\sum_{i=1}^{m} -\sigma_i y_i h(x_i)\right]$

$= \frac{1}{2}\mathop{\mathbb{E}}_{\sigma}\left[\sup_{h\in H} \frac{1}{m}\sum_{i=1}^{m} \sigma_i h(x_i)\right] = \frac{1}{2}\hat{\mathcal{R}}_{S_X}(H)$

6. **Theorem** Rademacher complexity bound. (For binary classification)

For any $\delta > 0$, $\forall h \in H$, the followings hold with probability at least $1 - \delta$

$R(h) \leq \hat{R}(h) + \mathcal{R}_m(H) + \sqrt{\frac{1}{2m}\log\frac{1}{\delta}}$

$R(h) \leq \hat{R}(h) + \hat{\mathcal{R}}_S(H) + 3\sqrt{\frac{1}{2m}\log\frac{2}{\delta}}$ $\longrightarrow$ data-dependent. $\hat{\mathcal{R}}_S(H)$

$\hat{\mathcal{R}}_S(H)$ $\longrightarrow$ $\mathop{\mathbb{E}}_{\sigma}\left[\sup_{h\in H} \frac{1}{m}\sum_{i=1}^{m} (-\sigma_i) h(x_i)\right] = \mathop{\mathbb{E}}_{\sigma}\left[\inf_{h\in H} \frac{1}{m}\sum_{i=1}^{m} \sigma_i h(x_i)\right]$

Empirical Risk Minimization

$\Rightarrow$ Computing $\hat{\mathcal{R}}_S(H)$ is computationally hard for some hypothesis sets

7. **Theorem** Massart's Lemma

Let $A \subseteq \mathbb{R}^m$ be a finite set with $r = \max_{x \in A} \|x\|_2$

$$\mathbb{E}_\sigma\left[\frac{1}{m} \sup_{x \in A} \sum_{i=1}^m \sigma_i x_i\right] \le \frac{r}{m}\sqrt{2 \log |A|}$$

pf) $\exp\left(t \mathbb{E}_\sigma\left[\sup_{x \in A} \sum_{i=1}^m \sigma_i x_i\right]\right) \le \mathbb{E}_\sigma\left[\exp\left(t \sup_{x \in A} \sum_{i=1}^m \sigma_i x_i\right)\right]$

$$= \mathbb{E}_\sigma\left[\sup_{x \in A} \exp\left(t \sum_{i=1}^m \sigma_i x_i\right)\right]$$

$$\le \sum_{x \in A} \mathbb{E}_\sigma\left[\exp\left(t \sum_{i=1}^m \sigma_i x_i\right)\right]$$

$$\le \sum_{x \in A} \prod_{i=1}^m \mathbb{E}_\sigma\left[\exp\left(t \sigma_i x_i\right)\right]$$

$$\le \sum_{x \in A} \prod_{i=1}^m \exp\left(\frac{t^2 (2x_i)^2}{8}\right) \quad (\because \text{Hoeffding's Lemma})$$

$$= \sum_{x \in A} \exp\left(\frac{t^2}{2} \sum_{i=1}^m x_i^2\right)$$

$$\le \sum_{x \in A} \exp\left(\frac{t^2 r^2}{2}\right) = |A| \exp\left(\frac{t^2 r^2}{2}\right)$$

$$\Rightarrow \mathbb{E}_\sigma\left[\sup_{x \in A} \sum_{i=1}^m \sigma_i x_i\right] \le \frac{\log |A|}{t} + \frac{t r^2}{2} \le r \cdot \sqrt{2 \log |A|} \quad \left(t = \frac{\sqrt{2 \log |A|}}{r}\right)$$

$$\Rightarrow \mathbb{E}_\sigma\left[\frac{1}{m} \sup_{x \in A} \sum_{i=1}^m \sigma_i x_i\right] \le \frac{r}{m}\sqrt{2 \log |A|}$$

8. Growth function ($\Pi_H : \mathbb{N} \to \mathbb{N}$)

: maximum number of distinct ways (dichotomies) in which m points can be classified using hypothesis in H

(hypothesis $h : \mathcal{X} \to \{-1, 1\}$, dichotomy $h : \{x_1, \cdots, x_m\} \to \{-1, 1\}$)

$\Rightarrow \forall m \in \mathbb{N}, \quad \Pi_H(m) = \max_{\{x_1, \cdots x_m\} \subseteq \mathcal{X}} \left| \{ (h(x_1), \cdots, h(x_m)) : h \in H \} \right| \leq 2^m$

ex. H : 2D Linear Classifier $\longrightarrow \Pi_H(3) = 8$, $\Pi_H(4) = 14$

$\Rightarrow$ Let $\mathcal{G}$ be a family of functions taking values in $\{-1, 1\}$

$$\mathcal{R}_m(\mathcal{G}) \leq \sqrt{\frac{2}{m} \log \Pi_{\mathcal{G}}(m)}$$

(Rademarcher complexity is bounded by the growth function)

$\left(R(g) \leq \hat{R}(g) + \sqrt{\frac{2}{m} \log \Pi_{\mathcal{G}}(m)} + \sqrt{\frac{1}{2m} \log \frac{1}{\delta}} \longrightarrow \text{generalization bound}\right)$

pf) Fix a sample $S = (x_1, \cdots, x_m)$

Let $\mathcal{G}_{|S}$ be the set of vectors $(g(x_1), \cdots, g(x_m))^T$ where $g \in \mathcal{G}$

$$\Rightarrow \max_{x \in S} \|x\|_2 \leq \sqrt{m}$$

$$\Rightarrow \mathcal{R}_m(\mathcal{G}) \leq \mathbb{E}_S \left[\mathbb{E}_\sigma \left[\sup_{u \in \mathcal{G}_{|S}} \frac{1}{m} \sum_{i=1}^{m} \sigma_i u_i \right] \right]$$

$$\leq \mathbb{E}_S \left[\frac{\sqrt{m}}{m} \sqrt{2 \log |\mathcal{G}_{|S}|} \right] \quad (\because \text{Massart's Lemma})$$

$$\leq \sqrt{\frac{2}{m} \log \Pi_{\mathcal{G}}(m)}$$

ML Lec 9.

- VC dimension

  * shattered (H)

    : the hypothesis set H realizes all possible dichotomy of a sample S

    $\Rightarrow \Pi_H(m) = 2^m$

  1. VC dimension (VC dim H)

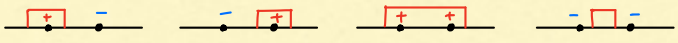
     : the size of the largest set that can be fully shattered by H

     $\Rightarrow$ VC dim $(H) = \max\{m : \Pi_H(m) = 2^m\} = d$

     $\Rightarrow$ it does not imply that all sets of size d or less are fully shattered

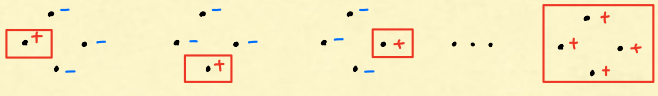
     $\Rightarrow$ it only needs a single placement of points that can be shattered by H

Ex. VC dim (Intervals on the real line) = 2

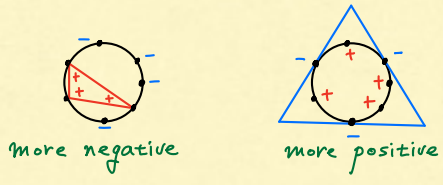


Ex. VC dim (Linear hyperplanes in $\mathbb{R}^N$) = $N+1$

Ex. VC dim (Axis-aligned rectangles) = 4



Ex. VC dim (Convex d-gons) = $2d+1$



Ex. VC dim (Convex polygons) = ∞

2. **Theorem** Sauer's Lemma

Let H be a hypothesis set with $VCdim(H) = d$.

For any $m \in \mathbb{N}$, $\Pi_H(m) \le \sum_{i=0}^{d} \binom{m}{i}$

(growth function is bounded by the VC dimension)

pf) Mathematical Induction

① Base step

$\Rightarrow$ for any d and $m=0$, $\Pi_H(0) \le \sum_{i=0}^{d} \binom{0}{i} = 1$

$\Rightarrow$ for any m and $d=0$, $\Pi_H(m) \le \sum_{i=0}^{0} \binom{m}{i} = 1$

② Inductive step

Fix $S = \{x_1, x_2, \cdots, x_m\}$ with $\Pi_H(m)$ and $S' = \{x_1, \cdots, x_{m-1}\}$

Let $G = H_{|S}$ be the set of concepts H includes given S.

Let $G_1 = H_{|S'}$ be the set of concepts H includes given S'

Let $G_2 = \{g' \subseteq S' : (g' \in G) \wedge (g' \cup \{x_m\} \in G)\}$

$\rightarrow$ $g' \in G$: without adding x_m, it is a concept of G

$\rightarrow$ $g' \cup \{x_m\} \in G$: adding x_m, it is also a concept of G

$VCdim(G_1) \le VCdim(G) \le d$

$\rightarrow |G_1| \le \Pi_{G_1}(m-1) \le \sum_{i=0}^{d} \binom{m-1}{i}$

$VCdim(G_2) \le VCdim(G) - 1 \le d-1$

$\rightarrow |G_2| \le \Pi_{G_2}(m-1) \le \sum_{i=0}^{d-1} \binom{m-1}{i}$

$$|G| = |G_1| + |G_2| \le \sum_{i=0}^{d} \binom{m-1}{i} + \sum_{i=0}^{d-1} \binom{m-1}{i}$$

$$= \binom{m}{0} + \sum_{i=1}^{d} \binom{m-1}{i} + \sum_{i=1}^{d} \binom{m-1}{i-1} = \sum_{i=0}^{d} \binom{m}{i}$$

3. **Corollary**

For all $m \geq d$, $\Pi_H(m) \leq \left(\frac{em}{d}\right)^d = O(m^d)$

$\Rightarrow$ $d \leq \infty$ when $\Pi_H(m) = O(m^d)$

$\Rightarrow$ $d = \infty$ when $\Pi_H(m) = 2^m$

$\left(R(g) \leq \hat{R}(g) + \sqrt{2 \cdot \frac{d}{m} \log \frac{em}{d}} + \sqrt{\frac{1}{2m} \log \frac{1}{\delta}} \rightarrow \text{generalization bound}\right)$

pf) $\Pi_H(m) \leq \sum_{i=0}^{d} \binom{m}{i}$ ($\because$ Saner's Lemma)

$\leq \sum_{i=0}^{d} \binom{m}{i} \left(\frac{m}{d}\right)^{d-i}$

$\leq \sum_{i=0}^{m} \binom{m}{i} \left(\frac{m}{d}\right)^{d-i}$ ($\because m \geq d$)

$= \left(\frac{m}{d}\right)^d \sum_{i=0}^{m} \binom{m}{i} \left(\frac{d}{m}\right)^i$

$= \left(\frac{m}{d}\right)^d \cdot \left(1 + \frac{d}{m}\right)^m$ ($\because$ binomial formula)

$\leq \left(\frac{m}{d}\right)^d \cdot e^d = \left(\frac{em}{d}\right)^d$

ML Lec 10.

- Ensemble methods

1. Ensemble : combining multiple hypotheses into one.

⇒ improves accuracy and robustness over a single model

⇒ the target function may not be implemented with a single classifier

, but might be approximated well by model averaging

① Combine by consensus

ex. Bagging, random forest, model averaging of probabilities

Pros
- no need for label
- improve generalization

Cons
- No feedback from label
- need assumption for good of consensus

② Combine by learning

ex. Boosting, rule ensemble, Bayesian model averaging

Pros
- feedback from label
- may improve generalization

Cons
- need to keep label for train
- may overfit to label
- do not work without label

2. Bagging → reduce variance

: given m data samples and a class of learning models, train k different models on different samples and average (or vote) their predictions

⇒ repeatedtly random sample with replacement from the training data (Bootstrap)

⇒ weight on each classifier is the same

☆ Bagging results in a model with smaller variance

pf) Let $x \sim N(\mu, \sigma^2)$ where $\mathbb{E}[x_i] = \mu$, $\mathrm{Var}[x_i] = \sigma^2$

$\Rightarrow \mathbb{E}[(x_1 + x_2 + \cdots + x_k)/k] = \mu$

$\Rightarrow \mathrm{Var}[(x_1 + x_2 + \cdots + x_k)/k] = \frac{1}{k}\sigma^2$

3. Boosting → reduce bias & variance

: combine a set of weak learners to obtain a strong learner.

⇒ weak learners usually have low variance and high bias.

⇒ the number of the boosting round decides whether overfit or underfit

($R(h) \le \hat{R}_S(h) + O\left(\sqrt{\frac{1}{m} d_t}\right)$ where d is the complexity of weak classifier.)

4. AdaBoost (Adaptive Boosting)

① Initialize weights on examples $D_0(i) = \frac{1}{m}$, $1 \le i \le m$

Repeat

② $h_t = \underset{h_t \in H}{\arg\min}\ \varepsilon_t = \underset{h_t \in H}{\arg\min} \sum_{i=1}^{m} D_t(i)\, I(y_i \neq h_t(x_i))$

③ $\alpha_t = \frac{1}{2} \log\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)$ (≈ accuracy)

④ $D_{t+1}(i) = \frac{1}{Z_t} D_t(i) \exp(-\alpha_t y_i h(x_i))$ where Z_t is a normalizing constant

⑤ $H(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$ → final hypothesis

⇒ h_t achieves the lowest classification error

⇒ misclassified examples are up-weighted

$$\begin{cases} \exp(-\alpha_t y_i h(x_i)) < 1 & y_i = h(x_i) \\ \exp(-\alpha_t y_i h(x_i)) > 1 & y_i \neq h(x_i) \end{cases}$$

⇒ weight of each learner is proportional to its accuracy.

Pros
- fast, versatile
- simple and easy to program
- No parameter except T to tune
- No prior knowledge needed
- Provably effective

Cons
- may overfit by complex learners
- may overfit by weak learners
- vulnerable to uniform noise

☆ $\alpha_t = \frac{1}{2}\log\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)$

pf) $D_{T+1}(i) = \frac{1}{m}\cdot\prod_{t=1}^{T}\frac{\exp(-y_i\alpha_t h_t(x_i))}{Z_t} = \frac{\exp\left(\sum_{t=1}^{T}-y_i\alpha_t h_t(x_i)\right)}{m\prod_{t=1}^{T}Z_t}$

$= \frac{\exp\left(-y_i\sum_{t=1}^{T}\alpha_t h_t(x_i)\right)}{m\prod_{t=1}^{T}Z_t} = \frac{\exp(-y_i f(x_i))}{m\prod_{t=1}^{T}Z_t}$ where $f(x) = \sum_t \alpha_t h_t(x)$

$\frac{1}{m}\sum_{i=1}^{m} I[H(x_i)\neq y_i] \le \frac{1}{m}\sum_{i=1}^{m}\exp(-y_i f(x_i))$

$= \sum_{i=1}^{m}\prod_{t=1}^{T} Z_t D_{T+1}(i) = \prod_{t=1}^{T} Z_t$

⇒ minimizing Z_t will minimize the generalization error bound.

→ find α_t that minimizes Z_t

$Z_t = \sum_{i=1}^{m} D_t(i)\exp(-\alpha_t y_i h_t(x_i))$

$= \sum_{i:y_i\neq h_t(x_i)} D_t(i)\cdot\exp(\alpha_t) + \sum_{i:y_i=h_t(x_i)} D_t(i)\cdot\exp(-\alpha_t)$

$= \sum_{i=1}^{m} D_t(i)\cdot I(y_i\neq h_t(x_i))\cdot\exp(\alpha_t) + \left(1-\sum_{i=1}^{m} D_t(i)\cdot I(y_i\neq h_t(x_i))\right)\exp(-\alpha_t)$

$= \varepsilon_t\exp(\alpha_t) + (1-\varepsilon_t)\exp(-\alpha_t)$

⇒ $\frac{\partial Z_t}{\partial \alpha_t} = \varepsilon_t\exp(\alpha_t) - (1-\varepsilon_t)\exp(-\alpha_t) \Rightarrow 0$

⇒ $\alpha_t = \frac{1}{2}\log\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)$