Linear Algebra

l-p norm

- $\|x\|_p = \left(\sum_{i=1}^{n} |x_i|^p\right)^{1/p}$

e.g.
$\|x\|_1 = (|x_1| + |x_2| + |x_3| + \cdots + |x_n|)^1$
$\|x\|_2 = (|x_1|^2 + |x_2|^2 + |x_3|^2 + \cdots + |x_n|^2)^{1/2}$
⋮
$\|x\|_\infty = \max\{|x_1|, \ldots, |x_n|\}$

$\lim_{p\to\infty} |x_1|^p + |x_2|^p + |x_3|^p + \cdots + |x_n|^p$
$\Rightarrow \lim_{p\to\infty} \max\{|x_1|^p + \cdots + |x_n|^p\}$
let $x_k = \max\{|x_1| + \cdots + |x_n|\}$
$\Rightarrow \lim_{p\to\infty} |x_k|^p$ (1)
$\therefore \lim_{p\to\infty} (|x_k|^p)^{1/p}$
$\Rightarrow \lim_{p\to\infty} |x_k| = |x_k|$ (2)
from (1), (2)
$\therefore \|x\|_\infty = \max\{|x_1| + \cdots + |x_n|\}$

Frobenius Norm

- $\|A\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} |a_{ij}|^2}$

e.g. $A = \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix}$
$\Rightarrow \|A\|_F = \sqrt{5^2+6^2+7^2+8^2} = \sqrt{174}$

- $\|A\|_F = \sqrt{tr(AA^T)}$ (should be AA^H)

e.g.
$AA^T = \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix}\begin{bmatrix} 5 & 7 \\ 6 & 8 \end{bmatrix} = \begin{bmatrix} 61 & 83 \\ 83 & 113 \end{bmatrix}$
$tr(AA^T) = 174$

Trace

- $tr(A) = \sum_{i=1}^{n} A_{ii}$

e.g.
$tr\left(\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}\right) = 15$
$tr\left(\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}\right) = 6$

- $tr(A^T) = tr(A)$
$tr(A \pm B) = tr(A) \pm tr(B)$
$tr(ABC) = tr(CAB) = tr(BCA)$

Matrix

(there is actually no orthonormal matrix)

- Orthogonal · orthonormal
$AA^T = I$ (column norm = 1)
e.g. $\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$
$\begin{bmatrix} c\theta & -s\theta \\ s\theta & c\theta \end{bmatrix}\begin{bmatrix} c\theta & s\theta \\ -s\theta & c\theta \end{bmatrix}$
$= \begin{bmatrix} c^2\theta + s^2\theta & c\theta s\theta - s\theta c\theta \\ s\theta c\theta - c\theta s\theta & s^2\theta + c^2\theta \end{bmatrix}$
$= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

- Eigendecomposition
  - $Av = \lambda v$ (right eigen; eigenvalue, eigenvector (usually unit))
  - $v^T A = \lambda v^T$ (left eigen)
  - n independent eigenvectors
    $\{v^{(1)}, \ldots, v^{(n)}\}$
    $\{\lambda_1, \ldots, \lambda_n\}$
  - $V = [v^{(1)}, \ldots, v^{(n)}]$
    $\lambda = [\lambda_1, \lambda_2 \cdots \lambda_n]^T$
  - $A = V\,diag(\lambda)\,V^{-1}$
  - if A is symmetric then
    $\exists\ A = Q\Lambda Q^T$
    ↑ scaling $v^{(i)}$ by λ_i

Notes:
△ $f(x) = x^T A x$ s.t. $\|x\|_2 = 1$
if $x \in \{v^{(1)}, \ldots, v^{(n)}\}$
then $f(x) = \lambda_i$
△ $\lambda \succ 0$, $A \in PD$
$\lambda \succeq 0$, $A \in PSD$ $x^T A x \geq 0$
$\lambda \prec 0$, $A \in ND$
$\lambda \preceq 0$, $A \in NSD$

- Singular Value Decomposition
  - All matrix ∃ SVD
  - $A = UDV^T$ (A: m×n, U: m×m, D: m×n, V: n×n)
    $U, V \in$ orthogonal (U: left singular vectors, V: right singular vectors)
    $D \in$ Diag (singular values here)

- Moore-Penrose Pseudoinverse
  - $Ax = b$
  - $x^* = \arg\min_x \|Ax - b\|_2$
  - $\nabla \|Ax - b\|^2 = 2A^TAx - 2A^Tb$

let ∇ = 0
$A^TAx = A^Tb$
$x = (A^TA)^{-1}A^Tb$
$x = A^+ b$
$A^+ = (A^TA)^{-1}A^T$

Matrix Derivatives

- $f \in R$, $x \in R^n$
$\frac{\partial f}{\partial x} = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{bmatrix}$

- $f \in R^m$, $x \in R$
$\frac{\partial f}{\partial x} = \begin{bmatrix} \frac{\partial f_1}{\partial x} \\ \frac{\partial f_2}{\partial x} \\ \vdots \\ \frac{\partial f_m}{\partial x} \end{bmatrix}$

- $f \in R^m$, $x \in R^n$
$J = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \frac{\partial f_m}{\partial x_2} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}$

- $f \in R$, $x \in R^{p\times q}$
$\frac{\partial f}{\partial X} = \begin{bmatrix} \frac{\partial f}{\partial x_{11}} & \frac{\partial f}{\partial x_{12}} & \cdots & \frac{\partial f}{\partial x_{1q}} \\ \frac{\partial f}{\partial x_{21}} & \frac{\partial f}{\partial x_{22}} & \cdots & \frac{\partial f}{\partial x_{2q}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f}{\partial x_{p1}} & \frac{\partial f}{\partial x_{p2}} & \cdots & \frac{\partial f}{\partial x_{pq}} \end{bmatrix}$

- $f = \|Xw - y\|^2$ $\quad Xw - y \in R^m$
$\frac{\partial f}{\partial w} = \frac{\partial}{\partial w}((Xw-y)^T(Xw-y))$
$= \frac{\partial}{\partial w}(w^TX^TXw - 2y^TXw + y^Ty)$
$= 2X^TXw - 2X^Ty$
$= 2X^T(Xw - y)$

$\arg\min_x \|Xw - y\|^2$
$\Rightarrow$ let $2X^T(Xw - y) = 0$
$\Rightarrow X^TXw = X^Ty$
$w = (X^TX)^{-1}X^Ty$
$(w = X^+ y)$

via SVD
- minimize $\|Xw - y\|$
- $X = U\Sigma V^T$
$U^TU = I$
$V^TV = I$
$\Sigma = diag[\sigma_1, \sigma_2, \ldots \sigma_n, 0, \ldots 0]$
- $w = X^+ y$
$= V\Sigma^+U^T$ $\quad \Sigma^+ = diag[1/\sigma_1, 1/\sigma_2, \ldots 1/\sigma_n, 0 \ldots]$
- sol^n
$\|Xw - y\|^2 = \|U\Sigma V^Tw - y\|^2$
minimize$_w$ $\|U\Sigma V^Tw - y\|^2$
⇕
minimize$_w$ $\|U^T(U\Sigma V^Tw - y)\|^2$
⇕
minimize$_w$ $\|\Sigma V^Tw - U^Ty\|^2$
⇕ z
minimize$_z$ $\|\Sigma z - U^Ty\|^2$ (diagonal)
$\Rightarrow \Sigma z - U^Ty = 0$
$\Rightarrow \Sigma z = U^Ty$
$\Rightarrow \Sigma^+\Sigma z = \Sigma^+U^Ty$
$\Rightarrow z = \Sigma^+U^Ty$
$\Rightarrow z = V^Tw = \Sigma^+U^Ty$
$\Rightarrow VV^Tw = V\Sigma^+U^Ty$
$\Rightarrow w = V\Sigma^+U^Ty$ *

Linear Models

Logistic Regression

- Binary classification
- $P(y=1|x) = \theta(w^Tx)$
$\theta(s) = \frac{e^s}{1+e^s} = \frac{1}{1+e^{-s}}$
- $P(y=-1|x) = \theta(-w^Tx)$
$1 - \frac{1}{1+e^{-s}} = \frac{1+e^{-s}-1}{1+e^{-s}} = \frac{e^{-s}}{1+e^{-s}} = \frac{1}{1+e^{s}}$
$= \theta(-s)$
- sum:
$P(y|x) = \theta(ys) = \theta(yw^Tx)$
- maximum likelihood
$\prod_{n=1}^{N} P(y_n|x_n)$
$= \prod_{n=1}^{N} \theta(y_n w^T x_n)$

maximize$_w$ $\prod_{n=1}^{N} \theta(y_n w^T x_n)$
⇕
maximize$_w$ $\log\left(\prod_{n=1}^{N} \theta(y_n w^T x_n)\right)$
⇕
minimize$_w$ $-\sum_{n=1}^{N} \log(\theta(y_n w^T x_n))$ $\quad \left(\frac{1}{1+e^{-s}}\right)^{-1}$
⇕
minimize$_w$ $\sum_{n=1}^{N} \log(1 + e^{-y_n w^T x_n})$

Linear Regression

- minimize$_w$ $\|X^Tw - y\|_2$
- sol^n: from previous!

Lagrange duality

- primal
min$_w$ $f(w)$
s.t. $g_i(w) \leq 0$ $\quad i = 1, \ldots, k$
$h_i(w) = 0$ $\quad i = 1, \ldots, l$
- generalized Lagrangian
- $\mathcal{L}(w, \alpha, \beta)$
$= f(w) + \sum_{i=1}^{k} \alpha_i g_i(w) + \sum_{i=1}^{l} \beta_i h_i(w)$
- $\theta_P(w) = \max_{\alpha,\beta:\alpha_i\geq 0} \mathcal{L}(w, \alpha, \beta)$

$\therefore \theta_P(w) = \begin{cases} f(w) & w \in \text{constraints} \\ \infty & \text{o.w.} \end{cases}$

↳ $\max_{\alpha,\beta,\alpha\geq 0} f(w) + \sum_{i=1}^{k}\alpha_i g_i(w) + \sum_{i=1}^{l}\beta_i h_i(w)$
- if $g_i > 0$, then ∞
- if $h_i \neq 0$, then ∞
∴ g_i, h_i constraints should be obeyed
- to get max, we want $\sum_{i=1}^{k}\alpha_i g_i(w)$ as big as possible
∴ $\sum_{i=1}^{k}\alpha_i g_i(w) = 0$
$\sum_{i=1}^{l}\beta_i h_i(w) = 0$
∴ $\max_{\alpha\beta\alpha\geq 0} \mathcal{L}(w, \alpha, \beta) = \begin{cases} f(w) \\ \infty \end{cases}$

- $\theta_D(w) = \min_{\alpha,\beta:\alpha\geq 0} \mathcal{L}(w, \alpha, \beta)$

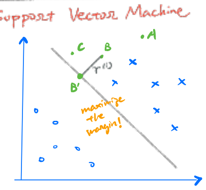
- Primal Problem (it is equivalent to original problem)
$\min_w \theta_P(w) = \min_w \max_{\alpha,\beta:\alpha_i\geq 0} \mathcal{L}(w, \alpha, \beta)$
- Dual Problem
$\max_{\alpha,\beta:\alpha_i\geq 0} \theta_D(w) = \max_{\alpha,\beta:\alpha_i\geq 0} \min_w \mathcal{L}(w, \alpha, \beta)$

$d^* = \max_{\alpha,\beta:\alpha_i\geq 0} \min_w \mathcal{L}(w, \alpha, \beta)$
$\leq \min_w \max_{\alpha,\beta:\alpha_i\geq 0} \mathcal{L}(w, \alpha, \beta) = p^*$

KKT condition (when will $p^* = d^* = \mathcal{L}(w^*, \alpha^*, \beta^*)$ i.e., strong duality; Slater's condition; zero duality gap)
- $\nabla \mathcal{L}(w^*, \alpha^*, \beta) = 0$
- $h_i(w^*) = 0$
- $\alpha_i^* g_i(w) = 0$ dual complementarity (complementary slackness)
- $g_i(w^*) \leq 0$
- $\alpha^* \geq 0$

now back to SVM!

Support Vector Machine



- Classifier
$h_{w,b}(x) = g(w^Tx + b)$
$g(s) \begin{cases} 1 & , s \geq 0 \\ -1 & , s < 0 \end{cases}$

- functional margin
- $\hat{\gamma}^{(i)} = y^{(i)}(w^Tx + b)$ (+ positive number, − negative number)
if $\hat{\gamma}^{(i)} > 0 \Rightarrow$ correct prediction
- $\hat{\gamma} = \min_{i=1,\ldots,m} \hat{\gamma}^{(i)}$
define the functional margin as the min. of all margins, we then try to maximize it.

- Geometric margin
@ B, we have $x^{(i)}$
∴ $B' = x^{(i)} - \gamma^{(i)}\frac{w}{\|w\|}$
∵ $w B' + b = 0$
∴ $w^T(x^{(i)} - \gamma^{(i)}\frac{w}{\|w\|}) + b = 0$
∴ $\gamma^{(i)} = \frac{w^Tx^{(i)} + b}{\|w\|}$
$= \left(\frac{w}{\|w\|}\right)^T x^{(i)} + \frac{b}{\|w\|}$ (this is for (+))
∴ $\gamma^{(i)} = y^{(i)}\left[\left(\frac{w}{\|w\|}\right)^T x^{(i)} + \frac{b}{\|w\|}\right]$
$\gamma = \min_{i=1,\ldots,m} \gamma^{(i)}$ (each training example could have at least γ margin)
⇓
$\max_{\gamma,w,b}$ γ
s.t. $y^{(i)}(w^Tx^{(i)} + b) \geq \gamma$
$\|w\| = 1$
this ensures geometric = functional

(from Lagrangian we can express w in terms of α, x; since must only SV $\alpha \neq 0$; we can write $w^Tx + b$ with (α, x, y) of support vectors!)

- rewrite
$\max_{\hat{\gamma},w,b} \frac{\hat{\gamma}}{\|w\|}$
s.t. $y^{(i)}(w^Tx^{(i)} + b) \geq \hat{\gamma}$, $i = 1, \ldots, m$
⟺ $\max_{w,b} \frac{1}{\|w\|}$
s.t. $y^{(i)}(w^Tx^{(i)} + b) \geq 1$, $i = 1, \ldots, m$
⟺ $\min_{w,b} \frac{1}{2}\|w\|^2$
s.t. $y^{(i)}(w^Tx^{(i)} + b) \geq 1$, $i = 1, \ldots, m$
↳ Lagrangian
$\mathcal{L}(w,b,\alpha) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{m}\alpha_i[y^{(i)}(w^Tx^{(i)} + b) - 1]$ — (0)
we try to find dual problem
$\Rightarrow \nabla_w \mathcal{L}(w,b,\alpha) = w - \sum_{i=1}^{m}\alpha_i y^{(i)}x^{(i)} = 0$
∴ $w = \sum_{i=1}^{m}\alpha_i y^{(i)}x^{(i)}$ — (1)
$\nabla_b \mathcal{L}(w,b,\alpha) = \sum_{i=1}^{m}\alpha_i y^{(i)} = 0$ — (2)
from (0), (1)
$\Rightarrow \mathcal{L}(w,b,\alpha) = \frac{1}{2}\sum_{i=1}^{m} y^{(i)2}\alpha_i^2 x^{(i)T}x^{(i)}$
$- \sum_{i=1}^{m} y^{(i)2}\alpha_i^2 x^{(i)T}x^{(i)}$
$- b\sum_{i=1}^{m}\alpha_i y^{(i)}$ ---- (2)
$+ \sum_{i=1}^{m}\alpha_i$
$\Rightarrow \mathcal{L}(w,b,\alpha) = -\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} y^{(i)2}\alpha_i^2 x^{(i)T}x^{(j)}$
$+ \sum_{i=1}^{m}\alpha_i$

Dual Problem
$\max_\alpha -\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n} y^{(i)2}\alpha_i^2 x^{(i)T}x^{(j)}$
s.t. $\alpha_i \geq 0$ $\quad i = 1, \ldots, m$
$\sum_{i=1}^{m}\alpha_i y^{(i)} = 0$

△ - solve α, then use (1) get w
- w/ w, b is
$b^* = -\frac{\max_{i:y^{(i)}=-1} w^{*T}x^{(i)} + \min_{i:y^{(i)}=1} w^{*T}x^{(i)}}{2}$ (support vector for (−); support vector for (+))

△ from (1)
$w = \sum_{i=1}^{m}\alpha_i y^{(i)}x^{(i)}$ get w in terms of α
@ inference (deployment)
$wx + b$
$= \left(\sum_{i=1}^{m}\alpha_i y^{(i)}x^{(i)}\right)^T x + b$
$\sum_{i=1}^{m}\alpha_i y^{(i)}\langle x^{(i)}, x\rangle + b$
↳ $\alpha_i > 0$ only @ support vector
↳ $wx + b$ is actually just $\sum \langle$support vector$_{(i)}$, $x_{input}\rangle$

This is why it is called Support Vector Machine! SVM

SVD calculation

$Av = \lambda v$
$(A - \lambda I)v = 0$

- recall eigendecomposition
- given $A = \begin{bmatrix} 26 & 18 \\ 18 & 74 \end{bmatrix}$
- $\det(A - \lambda I) = \det\begin{bmatrix} 26-\lambda & 18 \\ 18 & 74-\lambda \end{bmatrix}$
$\Rightarrow \lambda^2 - 100\lambda + 1600 = (\lambda - 80)(\lambda - 20)$

△ $\lambda = 20$
$A - \lambda I = \begin{bmatrix} 6 & 18 \\ 18 & 54 \end{bmatrix}$ $\quad v_1 = \frac{1}{\sqrt{10}}\begin{bmatrix} -3 \\ 1 \end{bmatrix}$
△ $\lambda = 80$
$A - \lambda I = \begin{bmatrix} -54 & 18 \\ 18 & -6 \end{bmatrix}$ $\quad v_2 = \frac{1}{\sqrt{10}}\begin{bmatrix} 1 \\ 3 \end{bmatrix}$

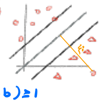
∴ $A = Q\Lambda Q^T$
$= \begin{bmatrix} -3/\sqrt{10} & 1/\sqrt{10} \\ 1/\sqrt{10} & 3/\sqrt{10} \end{bmatrix}\begin{bmatrix} 20 & 0 \\ 0 & 80 \end{bmatrix}\begin{bmatrix} -3/\sqrt{10} & 1/\sqrt{10} \\ 1/\sqrt{10} & 3/\sqrt{10} \end{bmatrix}$

$\arg\max_{x:\|x\|=1} \|Ax\|$
⇕
$\arg\max_{x:\|x\|=1} x^TA^TAx$ (symmetric)
$A^TAx = \lambda x$
$x^TA^TAx = \lambda x^Tx$
$x^TA^TAx = \lambda$
∴ x^TA^TAx happens @ max λ
$A = U\Sigma V^T$
orthogonal · A · still orthogonal
$AV = U\Sigma$

- SVD
we use C^TC's eigenvectors to determine V; Σ is then $\sqrt{\Lambda}$

$C = \begin{bmatrix} 5 & 5 \\ -1 & 7 \end{bmatrix}$
$C = U\Sigma V^T$
$C^TC = V\Sigma^TU^TU\Sigma V^T$
$= V\Sigma^T\Sigma V^T$ — ①
$CV = U\Sigma V^TV$
$= U\Sigma$ — ②
$C^TC = \begin{bmatrix} 26 & 18 \\ 18 & 74 \end{bmatrix}$
∴ $V = \begin{bmatrix} -3/\sqrt{10} & 1/\sqrt{10} \\ 1/\sqrt{10} & 3/\sqrt{10} \end{bmatrix}$
$\Sigma = \begin{bmatrix} \sqrt{20} & 0 \\ 0 & \sqrt{80} \end{bmatrix}$
∵ $CV = U\Sigma$
$\Rightarrow \begin{bmatrix} -\sqrt{10} & 2\sqrt{10} \\ \sqrt{10} & 2\sqrt{10} \end{bmatrix} = U\begin{bmatrix} \sqrt{20} & 0 \\ 0 & \sqrt{80} \end{bmatrix}$
∴ $U = \begin{bmatrix} -1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix}$

SVM + regularization (for outlier)



- recall SVM

$$\min_{w,b} \frac{1}{2}\|w\|^2 \quad s.t.\ y^{(i)}(w^Tx^{(i)}+b)\ge 1$$

- add L1 regularization

$$\min_{w,b} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m}\xi_i \quad s.t.\ y^{(i)}(w^Tx^{(i)}+b)\ge 1-\xi_i,\ \xi_i\ge 0$$

$$\iff \min_{w} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m}\xi_i \quad s.t.\ y^{(i)}(w^Tx^{(i)})\ge 1-\xi_i,\ \xi_i\ge 0$$

(we can always let w^Tx+b become $[b\ w_1\ w_2 \dots w_n]\begin{bmatrix}1\\x_1\\x_2\\\vdots\\x_n\end{bmatrix}$)

$$\iff \min_{w} C\sum_{i=1}^{n}\max(1-y_iw^Tx_i, 0) + \frac{1}{2}w^Tw$$

(hinge loss; L2-reg.)

use subgradient: $-y_ix_i$, 0, 0

Non-linear SVM

- some data ≠ linear separable
- perform ϕ non-linear transformation to a space that is linear separable
- $z = \phi(x)$ (↑ input features, ↑ input attributes) — we can use different kernel: Gaussian, Quadratic.
- Kernel Trick (this allow us not to explicitly write down ϕ, but apply kernel directly).
  - $\langle a, b\rangle$ replaced by
  - $K(a, b)$ for non-linear transformation

Multi Class Classification

- 1 v. All
  - build t-binary classifier
    - (+), $\in t^{th}$ class
    - (−), $\notin t^{th}$ class
  - decision value for t^{th} class
- 1 v. 1
  - $L(L-1)$ binary classifier
  - s-t
    - (+), $\in s^{th}$ class
    - (−), $\in t^{th}$ class
- multi class loss function
  - softmax, i-th data

$$-\log\left(\frac{e^{w_{y_i}^Tx_i}}{\sum_j e^{w_j^Tx_i}}\right)$$

  - optimization problem

$$\min_{w_1 \dots w_L} \sum_{i=1}^{n} -\log\left(\frac{e^{w_{y_i}^Tx_i}}{\sum_j e^{w_j^Tx_i}}\right) + \lambda\|w\|_2$$

Optimization Convexity

△ $f(\theta x_1+(1-\theta)x_2) \le \theta f(x_1)+(1-\theta)f(x_2)$
△ 1st order: $f(x) \ge f(x_0)+\nabla f(x_0)^T(x-x_0)$
△ 2nd order: $\nabla^2 f(x)\succeq 0$, $z^T\nabla^2 f(x) z\ge 0$

- convex function

FACT $f: \mathbb{R}^n \to \mathbb{R}$

$f(\theta x+(1-\theta)y)\le \theta f(x)+(1-\theta)f(y)$, $\theta \in [0,1]$

FACT 1st order condition

$f(y)\ge f(x)+\nabla f(x)^T(y-x)$

FACT 2nd order condition

$\nabla^2 f(x)\succeq 0$

$z^T\nabla^2 f(x) z \ge 0 \quad \forall z \in \mathbb{R}^n$

eg.

- $f(w) = g(x^Tw+y)$

$g(z)$ is convex, how about $f(w)$?

proof:

△ f is convex i.f.f.

$f(\theta w_1+(1-\theta)w_2)\le \theta f(w_1)+(1-\theta)f(w_2)$

$\Rightarrow g(x^T[\theta w_1+(1-\theta)w_2]+y)$
$\le \theta g(x^Tw_1+y)+(1-\theta)g(x^Tw_2+y)$

$\Rightarrow g(\theta x^Tw_1+(1-\theta)x^Tw_2+\theta y+(1-\theta)y)$
$\le \theta g(x^Tw_1+y)+(1-\theta)g(x^Tw_2+y)$

$\Rightarrow g(\theta(x^Tw_1+y)+(1-\theta)(x^Tw_2+y))$
$\le \theta g(x^Tw_1+y)+(1-\theta)g(x^Tw_2+y)$

$z_1 = x^Tw_1+y$, $z_2=x^Tw_2+y$

$\Rightarrow g(\theta z_1+(1-\theta)z_2)\le \theta g(z_1)+(1-\theta)g(z_2)$

as $g(z)$ is convex, $f(w)$ is convex

Lipschitz Continuity Smoothness

L is a constant

- L-Lipschitz continuous: $\|f(x_1)-f(x_2)\|_2 \le L\|x_1-x_2\|_2$
- L-Lipschitz smooth: $\|\nabla f(x_1)-\nabla f(x_2)\|_2 \le L\|x_1-x_2\|_2$

FACT $\nabla^2 f(x) \preceq LI$ if f is L-smooth

proof

△ mean value theorem, $\exists c\in[a,b]$

$$f'(c) = \frac{f(b)-f(a)}{b-a}$$

$\therefore \exists z \in [a,b]$ s.t. $\nabla^2 f(z)\cdot(x-y) = \nabla f(x)-\nabla f(y)$

△ L-smooth property: $\|\nabla f(x)-\nabla f(y)\|\le L\|x-y\|$

△ $\therefore \|\nabla^2 f(z)(x-y)\|$
$= \|\nabla f(x)-\nabla f(y)\|$
$\le L\|x-y\|$ (L-smooth)

$\therefore \|\nabla^2 f(z)(x-y)\|\le L\|x-y\|$

from Cauchy-Schwarz inequality (upper bound)

$\|\nabla^2 f(z)(x-y)\|\le \|\nabla^2 f(z)\|\|x-y\|$ (this one might)

$\therefore \|\nabla^2 f(z)\|\ \|x-y\| \le L\|x-y\|$

$\therefore \|\nabla^2 f(z)\|\le L$

$\nabla^2 f(z)$ is symmetric

$\|\nabla^2 f(z)\|\le L$

Cauchy-Schwarz inequality: $\|ab\|\le\|a\|\|b\|$

$\Rightarrow \nabla^2 f(z)\preceq LI$

FACT $f(y)\le f(x)+\nabla f(x)^T(y-x)+\frac{1}{2}L\|y-x\|^2$

△ $z(t) = x+t(y-x)$, $g(t) = f(z(t))$ $\therefore g(0)=f(x)$, $g(1)=f(y)$

$\therefore g'(t) = \nabla f(z(t))^T(y-x)$, $g'(0)=\nabla f(x)^T(y-x)$

△ fundamental theorem

$g(1)-g(0) = \int_0^1 g'(t)dt$

$\therefore g(1)-g(0)-g'(0) = \int_0^1 g'(t)-g'(0)\,dt \le \int_0^1 \|g'(t)-g'(0)\|dt$

△ $\|g'(t)-g(0)\|$
$= \|\nabla f(z(t))^T(y-x)-\nabla f(x)^T(y-x)\|$
$= \|[\nabla f(z(t))-\nabla f(x)]^T(y-x)\|$
$\le L\|z(t)-x\|\cdot\|y-x\|$
$= L\|x+ty-tx-x\|\cdot\|y-x\|$
$= tL\|y-x\|^2$

△ $\int_0^1\|g'(t)-g'(0)\|dt \le \int_0^1 tL\|y-x\|^2dt = L\|y-x\|^2[\frac{1}{2}t^2]_0^1 = \frac{1}{2}L\|y-x\|^2$

△ $\therefore g(1)-g(0)-g'(0)$
$= f(y)-f(x)-\nabla f(x)^T(y-x)$
$\le \frac{1}{2}L\|y-x\|^2$

△ $\therefore f(y)\le f(x)+\nabla f(x)^T(y-x)+\frac{1}{2}\|y-x\|^2$

FACT Gradient Descent converges if $\alpha < \frac{1}{L}$ ($\alpha<\frac{1}{L}$, $\frac{1}{2\alpha}>\frac{L}{2}$)

proof

$g(d) = f(w^t)+\nabla f(w^t)^Td+\frac{1}{2\alpha}\|d\|^2$
$> f(w^t)+\nabla f(w^t)^Td+\frac{L}{2}\|d\|^2$
$\ge f(w^t+d)$

$\therefore f(w^t+d^*) < g(d^*)\le g(0) = f(w^t)$

$\therefore$ let $\alpha<\frac{1}{L}$ during training

learning rate scheduling:
- @ start: lr ↑ speed up the process
- @ end: lr ↓ avoid overshoot, divergence
- @ start: we could sometimes increase lr, to explore different dimension.

Gradient Descent

- $w \leftarrow w - \alpha\nabla f(w)$
- stop: $\|\nabla f(w)\| < \epsilon$

Momentum Gradient Descent

- use previous gradient

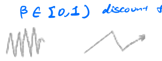
$v_t = (1-\beta)\nabla f(w_t) + \beta(1-\beta)\nabla f(w_{t-1}) + \beta^2(1-\beta)\nabla f(w_{t-2})+\dots$ (similar to IIR)

@ update:

$v_t = \beta v_{t-1} + (1-\beta)\nabla f(w_t)$

$w_{t+1} \leftarrow w_t - \alpha v_t$

$\beta \in [0,1)$ discount factors



Adagrad

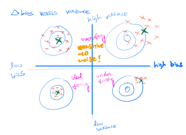
- $$\begin{bmatrix}w_{t+1}^1\\w_{t+1}^2\\\vdots\\w_{t+1}^n\end{bmatrix} \leftarrow \begin{bmatrix}w_t^1\\w_t^2\\\vdots\\w_t^n\end{bmatrix} - \begin{bmatrix}\frac{\alpha}{(\epsilon+G_t^{(1,1)})^{0.5}} & & 0\\ & \frac{\alpha}{(\epsilon+G_t^{(2,2)})^{0.5}} & \\ 0 & & \ddots \\ & & & \frac{\alpha}{(\epsilon+G_t^{(n,n)})^{0.5}}\end{bmatrix}\begin{bmatrix}g_t^{(1)}\\g_t^{(2)}\\\vdots\\g_t^{(n)}\end{bmatrix}$$

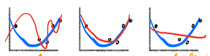
- $g_t \approx \frac{1}{b}\sum_{i=1}^{B}\nabla f_i(w)$ • $G_t = \sum_{\tau=1}^{t} g_\tau g_\tau^T$

Regularization

- L1-norm
- L2-norm

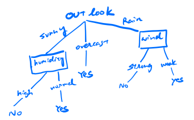
to prevent over-/under-fitting





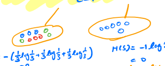
Tree-based Models

- Play Baseball?



- Q: given a data, how do we do splitting? I.G. as big as possible
- entropy:

$$H(S) = -\sum_{c=1}^{C} p(c)\log p(c)$$ (classes)



$H(S) = -(\frac{1}{3}\log\frac{1}{3}+\frac{1}{3}\log\frac{1}{3}+\frac{1}{3}\log\frac{1}{3}) = 1.58$ (2 as base) bad split

$H(S) = -1\log 1 = 0$ good split

we want $H(S)$↓!

- averaged entropy of a split

$S \to S_1, S_2$

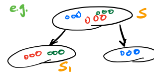
$$\frac{|S_1|}{|S|}H(S_1) + \frac{|S_2|}{|S|}H(S_2)$$

information gain:

$$H(S) - \left(\frac{|S_1|}{|S|}H(S_1) + \frac{|S_2|}{|S|}H(S_2)\right)$$

(H from S; H from S_1; H from S_2)

e.g.



$H(S) = -(\frac{1}{3}\log_2(\frac{1}{3})+\frac{1}{3}\log_2(\frac{1}{3})+\frac{1}{3}\log_2(\frac{1}{3})) = 1.58$

$H(S_1) = -(\frac{1}{2}\log_2(\frac{1}{2})+\frac{1}{2}\log_2(\frac{1}{2})) = 1$

$H(S_2) = -(1\log_2(1)) = 0$

I.G.: $H(S)-(\frac{|S_1|}{|S|}H(S_1)+\frac{|S_2|}{|S|}H(S_2)) = 1.58 - \frac{6}{9}\cdot 1 + \frac{3}{9}\cdot 0 = 0.91$

eg.

$H(S) = 1.58$

$H(S_1) = -(\frac{2}{5}\log\frac{2}{5}+\frac{2}{5}\log\frac{2}{5}+\frac{1}{5}\log\frac{1}{5}) = 1.52$

$H(S_2) = -(\frac{1}{2}\log\frac{1}{2}+\frac{1}{4}\log\frac{1}{4}+\frac{1}{4}\log\frac{1}{4}) = 1.5$

I.G.: $H(S)-(\frac{|S_1|}{|S|}H(S_1)+\frac{|S_2|}{|S|}H(S_2)) = 1.58-(\frac{5}{9}1.52+\frac{4}{9}1.5) = 1.58-1.51 = 0.07$

- Solve
  - find all features/threshold
  - compute information gain after each splitting
  - choose best (maximal I.G.)

Stochastic Gradient Descent

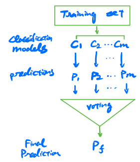
- recall $\nabla f(w) = \frac{1}{N}\sum_{n=1}^{N}\nabla f_n(w)$
- SGD: $\nabla f(w) \approx \frac{1}{B}\sum_{n=1}^{B}\nabla f_n(w)$, $|B|$ = batch size

Adam

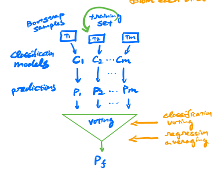
- Adagrad + Momentum
  - $g_t \leftarrow \nabla f_i(w_t)$
  - $m_t \leftarrow \beta_1 m_{t-1} + (1-\beta_1)g_t$
  - $v_t \leftarrow \beta_2 v_{t-1} + (1-\beta_2)g_t^2$
  - $\hat{m}_t \leftarrow m_t/(1-\beta_1^t)$
  - $\hat{v}_t \leftarrow v_t/(1-\beta_2^t)$
  - $w_t \leftarrow w_{t-1} - \alpha\cdot\frac{\hat{m}_t}{\sqrt{\hat{v}_t+\epsilon}}$ g_t

Ensemble Methods

- Stronger Together — ensembles of weak learner!!!



- Bagging (Bootstrap aggregation) ↑ independent from each other



e.g. Random Forest

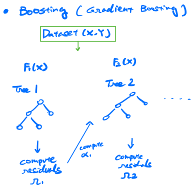
can converge on this dataset; perform better on other dataset

low bias + high variance (start) → train → reduced variance (end)

might perform poorly on other dataset

- subsample dataset/features
- avoid over-fitting
- improve stability & accuracy

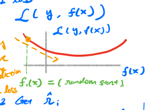
- Boosting (Gradient Boosting)



$x \to F(x)$

$F(x) = \sum_{i=1}^{m} f_i(x)$

1. loss $\mathcal{L}(y, f(x))$ — bias ↓ variance ↑



$f_1(x)$ = (random start)

2. Get $\hat{r}_i$

$$\hat{r}_{mi} = \frac{d\mathcal{L}(y_i, \hat{y}_i)}{d\hat{y}_i}\Bigg|_{\hat{y}_i = F_m(x_i)}$$

$\forall$ data points $i = 1, \dots, N$

3. fit new weak learner

$[\hat{r}_m \sim x] \Rightarrow f_{m+1}(x)$

4. line search

$$\gamma_m = \arg\min_{\gamma}\left[\sum_{i=1}^{N}\mathcal{L}(y_i, F_m(x_i)+\gamma f_{m+1}(x_i))\right]$$

5. update

$F_{m+1}(x) = F_m(x) + \gamma_m f_{m+1}(x)$

6. return (2) or stop

why it works?

$\Rightarrow F_{m+1}(x) = F_m(x)+\gamma_m f_{m+1}(x)$
$\approx F_m(x)+\gamma_m\hat{r}_m$
$= F_m(x)+\gamma_m\frac{d\mathcal{L}(y, F_m(x))}{dF_m(x)}$

⇒ Hey! It is Gradient Descent!

Neural Network

Forward/Backward Propagation

- Forward Prop.
- Backward Prop. (get gradient)
  - find gradient $\frac{\partial f}{\partial x}\ \frac{\partial f}{\partial y}\ \frac{\partial f}{\partial z}$
  - establish computing graph

e.g. $f(x,y,z) = (x+y)z$

find $\frac{\partial f}{\partial x}\ \frac{\partial f}{\partial y}\ \frac{\partial f}{\partial z}$ @ $x=-2$, $y=5$, $z=-4$

△ $q = g(x,y)$
$\Rightarrow 3 = g(-2,5)$
$\Rightarrow \frac{\partial q}{\partial x} = 1$
$\frac{\partial q}{\partial y} = 1$

△ $f = f(q,z)$
$\Rightarrow -12 = f(3,-4)$
$\Rightarrow \frac{\partial f}{\partial z} = q = 3$
$\frac{\partial f}{\partial q} = z = -4$

△ $\frac{\partial f}{\partial x} = \frac{\partial q}{\partial x}\frac{\partial f}{\partial q} = 1\cdot -4 = -4$
$\frac{\partial f}{\partial y} = \frac{\partial q}{\partial y}\frac{\partial f}{\partial q} = 1\cdot -4 = -4$
$\frac{\partial f}{\partial z} = 3$

Activation function

- sigmoid $\sigma(x) = \frac{1}{1+e^{-x}}$
- tanh $\tanh(x)$
- ReLU $\max(0, x)$
- Leaky ReLU $\max(0.1x, x)$
- Maxout $\max(w_1^Tx+b_1, w_2^Tx+b_2)$
- ELU $\begin{cases}x & x\ge 0\\ \alpha(e^x-1) & x<0\end{cases}$

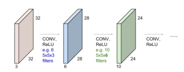
CNN kernel

e.g. 1

△ 32×32 (input) * 6 × 5×5×1 (convolution), 0 padding, 1 stride

output? 6×28×28

e.g. 2



input 32×32×3 * 6 conv 5×5×3

⇒ output? 28×28×6

CNN Pooling

- to down sampling

6×28×28 * max pool stride 2

output? 6×14×14

CNN Padding

- padding 2

Overfitting

- Dropout: turn off neurons w/ prob. $1-\alpha$
- Data augmentation
- SGD, Adagrad, Adam
- L1, L2-norm Regularization
- Learning Rate decaying

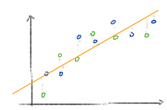
Vanishing/Exploding

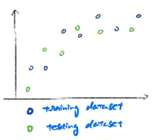
- Add Residual (vanishing) — ResNet

- use proper activation to prevent saturation (ReLU)
- batch normalization (retain it around zero)
- proper initialization
- learning rate scheduling.

Bias v. Variance !!!

- Bias: unable to capture real relation
- Variance: difference in fits between datasets





- Bias ↑ for linear regression
- Var. ↓ for linear regression between dataset

- Bias ↓ for polynomial regression
- Var. ↑ for polynomial regression between dataset

ML target: Bias ↓ Var. ↓

method: bagging/boosting/regularization

Theory of Generalization

FACT

△ with distribution in
training data $\mathcal{D}$
testing data
⇒ low training error
$\not\Rightarrow$
low testing error

Def

△ Training error :

- $E_{tr}(h) = \frac{1}{N}\sum_{n=1}^{N} e(h(x_n), f(x_n))$

where $x_1, \dots x_N$ sampled from $\mathcal{D}$

- h is determined by $x_1 \dots x_N$

△ Testing error :

- $E_{te}(h) = \frac{1}{N}\sum_{n=1}^{N} e(h(x_n), f(x_n))$

where $x_1, \dots x_n$ sampled from $\mathcal{D}$

- h is independent from $x_1 \dots x_N$

△ Generalization error $E(h)$

- G. error = Test error (expected performance on $\mathcal{D}$)
- $E(h) = E_{x\sim\mathcal{D}}[e(h(x), f(x))] = E_{te}(h)$

△ summary

if $E(h) = 0$

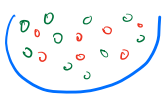
then $E(h) \approx E_{tr}(h)$ → How ? (but we hope this happens, so that what is seen (trained) could be applied to unseen data. $E(h)$ ensures w/ sufficient $E(h)$ ensures)

and

$E_{tr}(h) \approx 0$ → Training

Q: How do we make sure $E(h) \approx E_{tr}(h)$ ← thru this

FACT Hoeffding's inequality



△ $P[\text{pick red ball}] = \mu$
$P[\text{pick green ball}] = 1-\mu$
→ we DO NOT know μ

△ by pick ball's independently
we get fraction of ν

△ $\nu \to \mu$?
perhaps

△ Hoeffding's inequality
拼存的 真实的

$$P[|\nu-\mu| > \epsilon] \le 2e^{-2\epsilon^2 N}$$

annote : ν & μ 的差距，比 ϵ 大的 P，很小
多小？ → 比 $2e^{-2\epsilon^2 N}$ 还小

△ statement $\mu = \nu$ is
probably approximately correct
(PAC !)

$\nu = \mu$ probably approximately correct

FACT

△ $P[|\nu-\mu| > \epsilon] \le 2e^{-2\epsilon^2 N}$

- valid for N
- $\epsilon > 0$
- independent from μ (real probability)

$P[|\nu-\mu| > \epsilon] \le 2e^{-2\epsilon^2 N}$

↓

△ in learning :

- given a function h
- we randomly draw $x_1, \dots, x_n$] independent
- generalization error

$E(h) = E_{x\sim\mathcal{D}}[h(x) \neq f(x)] \Leftrightarrow \mu$ unknown

sample data error

$E_{tr}(h) = \frac{1}{N}\sum_{n=1}^{N}[h(x_n) \neq y_n] \Leftrightarrow \nu$ known

$P[|\nu-\mu| \ge \epsilon] \le 2e^{-2\epsilon^2 N}$

$P[|\nu-\mu| > \epsilon] \le 2e^{-2\epsilon^2 N}$

FACT

△ for each h, h is a hypothesis

$$P[|E_{tr}(h) - E(h)| > \epsilon] \le 2e^{-2\epsilon^2 N}$$

△ for all h, $\mathcal{H}$ is a hypothesis set

$P[|E_{tr}(h_1) - E(h_1)| > \epsilon]$,
$P[|E_{tr}(h_2) - E(h_2)| > \epsilon]$,
⋮
$P[|E_{tr}(h_{|\mathcal{H}|}) - E(h_{|\mathcal{H}|})| > \epsilon]$

$\le P[\sup_{h\in\mathcal{H}} |E_{tr}(h) - E(h)| > \epsilon]$

$\le \sum_{m=1}^{|\mathcal{H}|} P[|E_{tr}(h_m) - E(h_m)|] \le 2|\mathcal{H}|e^{-2\epsilon^2 N}$

from $P(\bigcup_{i=1}^{\infty} A_i) \le \sum_{i=1}^{\infty} P(A_i)$

$P[|\nu-\mu| > \epsilon] \le 2e^{-2\epsilon^2 N}$
⇓
$P[|E_{tr}(h) - E(h)| > \epsilon] \le 2e^{-2\epsilon^2 N}$
⇓
$P[|E_{tr}(h) - E(h)| > \epsilon]$
$\le P[\sup_{h\in\mathcal{H}} |E_{tr}(h) - E(h)| > \epsilon]$
$\le 2|\mathcal{H}|e^{-2\epsilon^2 N}$

△ summary

$$P[|E_{tr}(h) - E(h)| > \epsilon] \le P[\sup_{h\in\mathcal{H}} |E_{tr}(h) - E(h)| > \epsilon] \le 2|\mathcal{H}|e^{-2\epsilon^2 N}$$

NOTE More on Hoeffding's inequality

	D_1	D_2	...	D_N	
h_1	Bad			Bad	P_D [BAD D for h_1] ≤ ...
h_2		Bad			P_D [BAD D for h_2] ≤ ...
⋮					...
h_M	Bad			Bad	P_D [BAD D for h_M] ≤ ...

inferred hypothesis (假設現在我設定的模型 i.e., 不是學出來的) → 對於到我手上的資料 $D_1, D_2 \dots D_N$, inferred 的 h on D 可能導致 "Bad" 亦即 $E_{in}(h) \neq E_{out}(h)$ → 之所以要 Hoeffding's ineq. 就是為了量化 P[Bad] 機率多高?

→ 答案是很低: bounded by $2e^{-2\epsilon^2 N}$ → 不過剛剛是 1 個 h 啊... 我 algo 都是在 $\mathcal{H}$ 找 h... upper bound 是啥?

∴ P_D[BAD D]

$= P_D$[BAD D for h_1 or BAD D for h_2 ... or BAD D for h_M]

$\leq P_D$[BAD D for h_1] + P_D[BAD D for h_2] + ... + P_D[BAD D for h_M] (union bound)

$\leq 2M e^{-2\epsilon^2 N} = 2|\mathcal{H}| e^{-2\epsilon^2 N}$

- finite-bin version of Hoeffding
- & hope... $E_{in}(g) = E_{out}(g)$ is PAC.

→ $\mathcal{A}$ will pick h_m w/ min. $E_{in}(h_m)$ as g

△ summary: statistical learning flow

unknown target function $f: \mathcal{X} \to \mathcal{Y}$ (e.g. ideal credit approval function) ↓

unknown P on $\mathcal{X}$ → $x_1, x_2 \dots x_N$ (to training examples); → x (to final hypothesis)

training examples D: (x_1, y_1), (x_2, y_2), ⋮ (x_N, y_N) (e.g. historical bank record & approval results) → learning algorithm $\mathcal{A}$ → final hypothesis $g \approx f$ (learned formula to be used)

hypothesis set $\mathcal{H}$ (set of candidate formula) → learning algorithm $\mathcal{A}$

& hope $E_{out}(g) \approx E_{in}(g) \approx 0$ (test / train)

- for batch & supervised learning, $g \approx f \Leftrightarrow E_{out}(g) \approx 0$ achieved through $E_{out}(g) \approx E_{in}(g)$ & $E_{in}(g) \approx 0$
  ① can we make sure $E_{out}(g) \approx E_{in}(g)$ G. Error
  ② can we make $E_{in}(g)$ small enough Training

FACT $|\mathcal{H}| = \infty$

△ Question: How do we deal with it?

- small $|\mathcal{H}|$
  - 當 hypothesis set 小
  - P[BAD] 發生小
  - 但小 $|\mathcal{H}|$, 我訓練 找不到好 h 來讓 $E_{in}(h)$ or $E_{in}(h)$ 小

  $P[BAD] \leq 2|\mathcal{H}| e^{-2\epsilon^2 N}$
  small! great!
  but $|\mathcal{H}|$ too little
  $E_{in}(g)$ ↑

- large $|\mathcal{H}|$ (我 $|\mathcal{H}|$ 大 訓練找得到! $E_{in}(h)$ / $E_{out}(h)$ 小 但 P[BAD] 大!)

  $E_{in}(g) \to 0$
  small error! great!
  but $|\mathcal{H}|$ too large
  P[BAD] ↑ → $|\mathcal{H}|$ is ∞ no doubt, but can control the number!

註: 我們如何找 finite $|\mathcal{H}|$

FACT establish a finite quantity replace $|\mathcal{H}|$

let $|\mathcal{H}|$ replaced by $m_{\mathcal{H}}$
s.t.

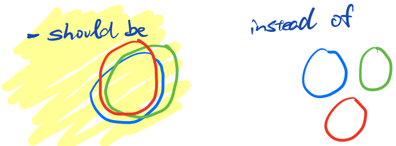
$$P[|E_{in}(g) - E_{out}(g)| > \epsilon] \leq 2\, m_{\mathcal{H}}\, e^{-2\epsilon^2 N}$$

(P[BAD])

FACT $|\mathcal{H}|$ is over-estimated for BAD events

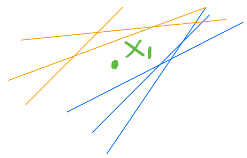
- BAD events B_m: $|E_{in}(h_m) - E_{out}(h_m)| > \epsilon$
- over-lapping for similar hypothesis $h_1 \approx h_2$
- as ① $E_{out}(h_1) \approx E_{out}(h_2)$
  ② for most D $E_{in}(h_1) \approx E_{out}(h_2)$

- should be / instead of

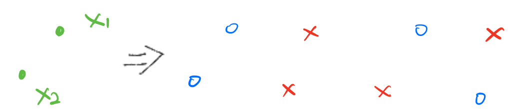


- so: can we group similar kinds?

e.g. $\mathcal{H}$ in $\mathbb{R}^2$ $|\mathcal{H}| \to \infty$



→ $|\mathcal{H}| = 2$



$|\mathcal{H}| = 4$

⋮

N=3 $|\mathcal{H}| = 8$ but if on same line, different
N=4 $|\mathcal{H}| = 14$ but if on same line, differ

FACT observation: effective $|\mathcal{H}| \leq 2^N$
perhaps can replace $|\mathcal{H}|$ by effective $|\mathcal{H}|$?
need more rigorous proof

FACT Dichotomies: mini-hypotheses (the split into two exclusive subsets of set)

△ limited hypothesis: $\mathcal{H}(x_1, x_2, \dots x_N)$

△ $|\mathcal{H}(x_1, x_2, \dots, x_N)|$: depend on inputs $(x_1, x_2 \dots, x_N)$

△ growth function: remove dependence by taking max of all possible $(x_1, x_2, \dots, x_N)$

$$m_{\mathcal{H}}(N) = \max_{x_1, x_2 \dots x_N \in \mathcal{X}} |H(x_1, x_2, \dots x_N)|$$

△ finite, upper-bounded by 2^N

Q: How to calculate growth function

FACT shattered

△ if $m_{\mathcal{H}}(N) = 2^N \iff$ exists N inputs that can be shattered

△ eg. convex set

FACT summary of 4 growth function

- positive rays $N+1$
- positive intervals $C_2^{N+1}+1$
- convex sets 2^N
- 2D perceptrons $< 2^N$

polynomial good !

exponential bad !

FACT Break point ⇝ k 個時，無法被 shattered

△ if no k inputs can be shattered by $\mathcal{H}$, call k a break point for $\mathcal{H}$

△ $m_{\mathcal{H}}(k) < 2^k$ (in binary case)

△ $k+1, k+2, k+3 \ldots$ are all break points

△ strictly minimum break point

eg linear case break point $k=4$

note: 4 個無法被 shattered

FACT conjecture:

△ no break point : $m_H(N) = 2^N$

△ break point k : $m_{\mathcal{H}}(N) = O(N^{k-1})$

proof ?

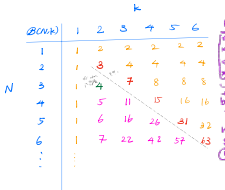
FACT $m_{\mathcal{H}}(N) \le$ maximum possible $m_{\mathcal{H}}(N)$ given k $\le$ poly (N)

FACT Bounding function 如果我有 break point k upper bound 在哪？

- 從上面結論：
  - 在 break point $= k$ ⇝ 露出一條曙光！
  - 任 k 個度，不能被 shattered (不管 2^k 組合) exists no 2^k dichotomies !
- Now, for bounding function:

  max $m_{\mathcal{H}}(N)$ @ break point $= k$

- $B(N,k)$



YELLOW

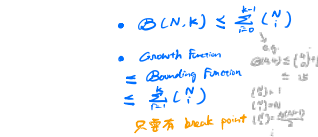
△ $k=1$, max. dichotomies $= 1$

△ $k>N$, max. dichotomies $= 2^N$

△ $k=N$, max. dichotomies $= 2^N - 1$

△ $k \le N$? PINK

- $B(N,k) \le \sum_{i=0}^{k-1} \binom{N}{i}$
- Growth Function $=$ Bounding Function $\le \sum_{i=0}^{k-1} \binom{N}{i}$

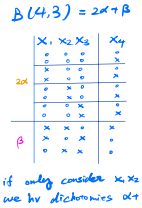


又當有 break point 有界定！

可被 polynomial 上限住！

PINK :

△ $B(4,3) = 2\alpha + \beta$



△ if only consider $x_1 x_2 x_3$ we hv dichotomies $\alpha + \beta$

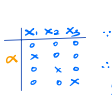
∵ $B(4,3) \to$ break pt $= 3$

∴ $B(3,3) \to$ break pt $= 3$ i.e., cannot be shattered i.e, we cannot hv 2^3

∴ $\alpha + \beta$ cannot shatter any 3 pts

∴ $\alpha + \beta \le B(3,3)$

△



∵ x_3 provide dichotomies "o" & "x" within $x_1 x_2$ we cannot shatter any 2 pts

∴ $\alpha \le B(3,2)$

△ $\therefore 2\alpha + \beta \le B(3,2) + B(3,3)$

$B(4,3) \le B(3,2) + B(3,3)$

$\Rightarrow B(N,k) \le B(N-1,k-1) + B(N-1,k)$

Hoeffding's Inequality Revisit

- recall

$$P[\exists h \in \mathcal{H} \text{ s.t. } |E_{in}(h) - E_{out}(h)| > \epsilon] \le 2\, m_{\mathcal{H}}(N) \exp(-2\epsilon^2 N)$$

- Actually

$$P[\exists h \in \mathcal{H} \text{ s.t. } E_{in}(h) - E_{out}(h)| > \epsilon] \le 2 \cdot 2 m_{\mathcal{H}}(2N) \cdot \exp\left(-2\tfrac{1}{16}\epsilon^2 N\right)$$

- Sketch of proof

1 - $P[\exists h \in \mathcal{H} \text{ s.t. } |E_{in}(h) - E_{out}(h)| > \epsilon]$ (E_{out} is actually ∞)

- replace $E_{out}(h)$ w/ $E_{in'}(h)$

$$\Rightarrow \tfrac{1}{2} P[|E_{in}(h) - E_{out}(h)| > \epsilon] \le P[|E_{in}(h) - E_{in'}(h)| > \tfrac{\epsilon}{2}]$$

($E_{in'}(h)$: ghost data)

2 - E_{in} w/ D $E_{in'}$ w/ D'

- $\mathcal{H} = \{\mathcal{H}(x_1, \ldots, x_N, x_1', \ldots, x_N')\}$
- ∵ union bound $m_{\mathcal{H}}(2N)$

$$\Rightarrow BAD \le 2P[|E_{in}(h) - E_{in'}(h)| > \tfrac{\epsilon}{2}] \le 2\, m_H(2N)\, P[\text{fixed } h \;|E_{in}(h) - E_{in'}(h)| > \tfrac{\epsilon}{2}]$$

3 - $|E_{in} - E_{in'}| > \frac{\epsilon}{2} \iff |E_{in} - \frac{E_{in} + E_{in'}}{2}| > \frac{\epsilon}{4}$

$$\Rightarrow BAD \le 2\, m_{\mathcal{H}}(2N)\, P[\text{fixed } h \;|E_{in}(h) - E_{in'}(h)| > \tfrac{\epsilon}{2}] \le 2\, m_{\mathcal{H}}(2N) \cdot 2\exp\left(-2\left(\tfrac{\epsilon}{4}\right)^2 N\right)$$

Vapnik-Chervonenkis (VC) bound :

$$P[\exists h \in \mathcal{H} \text{ s.t. } |E_{in}(h) - E_{out}(h)| > \epsilon] \le 4\, m_{\mathcal{H}}(2N)\, e^{-\frac{1}{8}\epsilon^2 N}$$

V.C. Dimension

- recall
  - $E_{out} \approx E_{in}$ possible if $\exists\, k$ if N large enough
  - $m_{\mathcal{H}}(N)$ (when $\exists$ break pt k) $m_{\mathcal{H}}(N) \le B(N,k) = \sum_{i=0}^{k-1} \binom{N}{i}$
  - when $N \ge 2, k \ge 2$ $m_{\mathcal{H}}(N) \le B(N,k) = \sum_{i=0}^{k-1} \binom{N}{i} = N^{k-1}$
- V.C. Dimension
  - max non-break point
  - break point -1

⇒ largest N for which $m_{\mathcal{H}}(N) = 2^N$ ⇒ $d_{VC} = \min k - 1$ if $\exists\, k$, $d_{VC} < \infty$

△ $N \le d_{VC} \Rightarrow \mathcal{H}$ can shatter some N inputs

△ $N > d_{VC} \Rightarrow k$ also cannot shatter N points

Perceptron Learning Algorithm

1D $d=1$, $d_{VC}=2$

2D $d=2$, $d_{VC}=3$

3D $d=3$, $d_{VC}=$! 4?

- FACT $d_{VC} = d+1$

proof using 3D case

- target $d_{VC} \ge d+1$ $d_{VC} \le d+1$
- what is $d_{VC} \ge d+1$ ⇒ we can shatter some $d+1$ inputs

△ $d_{VC} \ge d+1$

let $X = \begin{bmatrix} -x_1^T- \\ -x_2^T- \\ \vdots \\ -x_{d+1}^T- \end{bmatrix} = \begin{bmatrix} 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \end{bmatrix}$

eg 2D

- $\exists\, X^{-1}$
- recall perceptron : $\text{sign}(Xw) = y$

let $Xw = y$

$\therefore w = X^{-1} y$

- any y correspond to w all dichotomies exist! shattered!

- what is $d_{VC} \le d+1$ ⇒ we cannot shatter any set of $d+2$ inputs
- 2D special case

$X = \begin{bmatrix} -x_1^T- \\ -x_2^T- \\ -x_3^T- \\ -x_4^T- \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}$

$\Rightarrow w^T x_4 = \underbrace{w^T x_2}_{\circ} + \underbrace{w^T x_3}_{\circ} - \underbrace{w^T x_1}_{\times}$

$\Rightarrow w^T x_4$ must be (-) cannot be (+) therefore : not shattered.

- if linear dependencies exist $d+2$ cannot be shattered

$X = \begin{bmatrix} -x_1^T- \\ -x_2^T- \\ \vdots \\ -x_{d+2}^T- \end{bmatrix} \Rightarrow$ non-zero a_i's

- $x_{d+2} = a_1 x_1 + a_2 x_2 + \cdots + a_{d+1} x_{d+1}$

$w^T x_{d+2} = a_1 \underbrace{w^T x_1}_{(+)} + a_2 \underbrace{w^T x_2}_{\circ} + \cdots + a_{d+1} \underbrace{w^T x_{d+1}}_{(-)} > 0$

therefore $w^T x_{d+2}$ must be (+) ! ($y_{d+2} = -1$)

cannot be shattered !