Linear Algebra

L-p norm

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

e.g.

$\|x\|_1 = (|x_1| + |x_2| + |x_3| + \cdots + |x_n|)$

$\|x\|_2 = (|x_1|^2 + |x_2|^2 + |x_3|^2 + \cdots + |x_n|^2)^{1/2}$

$\|x\|_\infty = \max\{|x_1|, \ldots, |x_n|\}$

$\lim_{p\to\infty} |x_1|^p + |x_2|^p + |x_3|^p + \cdots + |x_n|^p$

$\Rightarrow \lim_{p\to\infty} \max\{|x_1|^p + \cdots + |x_n|^p\}$

let $x_k = \max\{|x_1| + \cdots + |x_n|\}$ (1)

$\Rightarrow \lim_{p\to\infty} |x_k|^p$

$\therefore \lim_{p\to\infty} (|x_k|^p)^{1/p}$

$\Rightarrow \lim_{p\to\infty} |x_k| = |x_k|$ (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^T)

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.

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

$tr\left(\begin{bmatrix} 1 & 2 \\ 4 & 5 \end{bmatrix}\begin{bmatrix} 3 \\ 6 \end{bmatrix}\right) = 6$

- $tr(A^T) = tr(A)$

$tr(A \pm B) = tr(A) \pm tr(B)$

$tr(ABC) = tr(CAB) = tr(BCA)$

Matrix

- Orthogonal, orthonormal (these actually no orthonormal matrix.)

$AA^T = I$ (columns norm = 1)

e.g. $\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$

$\begin{bmatrix} c\theta & -s\theta \\ s\theta & c\theta \end{bmatrix}\begin{bmatrix} c\theta & s\theta \\ -s\theta & c\theta \end{bmatrix}$

$= \begin{bmatrix} c^2\theta + s^2\theta & cs\theta - sc\theta \\ sc\theta - cs\theta & s^2\theta + c^2\theta \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

- Eigendecomposition

$Av = \lambda v$ (right eigen; v: eigenvector (usually unit), λ: eigenvalue)

$v^T A = \lambda v^T$ (left eigen)

- n independent eigenvectors $\{v^{(1)}, \ldots, v^{(n)}\}$

$\{\lambda_1, \ldots, \lambda_n\}$

- $V = [v^{(1)}, \ldots, v^{(n)}]$

$\lambda = [\lambda_1, \lambda_2 \cdots \lambda_n]^T$

- $A = V\,\mathrm{diag}(\lambda)\,V^{-1}$

- if A is symmetric then

$\exists\ A = Q\Lambda Q^T$ (scaling $v^{(i)}$ by λ_i)

Notes:

- $f(x) = x^TAx$ s.t. $\|x\|_2 = 1$ if $x \in \{v^{(1)}, \ldots, v^{(n)}\}$ then $f(x) = \lambda_i$
- $\lambda \geq 0$, $A \in PD$
- $\lambda \geq 0$, $A \in PSD$, $x^TAx \geq 0$
- $\lambda \not\geq 0$, $A \in ND$
- $\lambda \ngeq 0$, $A \in NSD$

- Singular Value Decomposition

- All matrix $\exists$ SVD
- $A = \underset{m\times n}{U}\,\underset{m\times n}{D}\,\underset{n\times n}{V^T}$ ($U, V \in$ orthogonal (left/right singular vectors); $D \in Diag$ (singular values here))

- Moore-Penrose Pseudoinverse

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

let $\nabla = 0$: $A^TAx = A^Tb$, $x = (A^TA)^{-1}A^Tb$, $x = A^+b$, $A^+ = (A^TA)^{-1}A^T$

Matrix Derivatives

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

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

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

$J = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \frac{\partial f_m}{\partial x_2} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}$

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

- $f = \|Xw - y\|^2$ ($Wx - y \in R^m$)

$\frac{\partial f}{\partial w} = \frac{\partial}{\partial w}((Xw-y)^T(Xw-y))$

$= \frac{\partial}{\partial w}(w^TX^TXw - 2y^TXw + y^Ty)$

$= 2X^TXw - 2X^Ty$

$= 2X^T(Xw - y)$

$\underset{x}{\arg\min}\ \|Xw - y\|^2$

$\Rightarrow$ let $2X^T(Xw - y) = 0$

$\Rightarrow X^TXw = X^Ty$

$w = (X^TX)^{-1}X^Ty$

($w = X^\dagger y$)

---

via SVD

- $\min_w \|Xw - y\|$

- $X = U\Sigma V^T$

$U^TU = I$

$V^TV = I$

$\Sigma = \mathrm{diag}[\sigma_1, \sigma_2, \ldots \sigma_n, 0, \ldots]$

- $w = X^\dagger y$

$= V\Sigma^+U^T$ $\quad \Sigma^+ = \mathrm{diag}\left[\frac{1}{\sigma_1}, \frac{1}{\sigma_2} \cdots \frac{1}{\sigma_n}, 0, \ldots\right]$

- solve

$\|Xw - y\|^2 = \|U\Sigma V^Tw - y\|^2$

$\min_w \|U\Sigma V^Tw - y\|^2$

$\min_w \|U^T(U\Sigma V^Tw - y)\|^2$

$\min_w \|\Sigma V^Tw - U^Ty\|^2$ ($\gamma = V^Tw$)

$\min_\gamma \|\Sigma\gamma - U^Ty\|^2$ (Σ: diagonal)

$\Rightarrow \Sigma\gamma - U^Ty = 0$

$\Rightarrow \Sigma\gamma = U^Ty$

$\Rightarrow \Sigma^T\Sigma\gamma = \Sigma^TU^Ty$

$\Rightarrow \gamma = \Sigma^TU^Ty$

$\Rightarrow \gamma = V^Tw = \Sigma^+U^Ty$

$\Rightarrow VV^Tw = V\Sigma^+U^Ty$

$\Rightarrow w = V\Sigma^+U^Ty$ ✱

SVD calculation

- recall eigendecomposition

- given $A = \begin{bmatrix} 26 & 18 \\ 18 & 74 \end{bmatrix}$

- $\det(A - \lambda I) = \det\begin{bmatrix} 26-\lambda & 18 \\ 18 & 74-\lambda \end{bmatrix}$

$\Rightarrow \lambda^2 - 100\lambda + 1600 = (\lambda - 80)(\lambda - 20)$

▲ $\lambda = 20$

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

▲ $\lambda = 80$

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

$\therefore A = Q\Lambda Q^T$

$= \begin{bmatrix} -\frac{3}{\sqrt{10}} & \frac{1}{\sqrt{10}} \\ \frac{1}{\sqrt{10}} & \frac{3}{\sqrt{10}} \end{bmatrix}\begin{bmatrix} 20 & 0 \\ 0 & 80 \end{bmatrix}\begin{bmatrix} -\frac{3}{\sqrt{10}} & \frac{1}{\sqrt{10}} \\ \frac{1}{\sqrt{10}} & \frac{3}{\sqrt{10}} \end{bmatrix}$

A orthogonal, still orthogonal

$AV = U\Sigma$

$Av = \lambda v$

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

$\arg\max \|Ax\|$ s.t. $\|x\| = 1$

$\Leftrightarrow \arg\max_x \frac{x^TA^TAx}{x^Tx}$ (Rayleigh quotient)

$A^TAx = \lambda x$

$x^TA^TAx = \lambda x^Tx$

$x^TA^TAx = \lambda x^Ty$

$x^TA^Tx = \lambda x^Ty$

x^TAx happens @ max λ

- 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

$C^TC = \begin{bmatrix} 26 & 18 \\ 18 & 74 \end{bmatrix}$

$\therefore V = \begin{bmatrix} -\frac{3}{\sqrt{10}} & \frac{1}{\sqrt{10}} \\ \frac{1}{\sqrt{10}} & \frac{3}{\sqrt{10}} \end{bmatrix}$ $\quad \Sigma = \begin{bmatrix} \sqrt{20} & 0 \\ 0 & \sqrt{80} \end{bmatrix}$

$\because CV = U\Sigma$

$\Rightarrow \begin{bmatrix} -\frac{3}{\sqrt{10}} & \frac{2\sqrt{10}}{\sqrt{10}} \\ \frac{1}{\sqrt{10}} & \frac{2\sqrt{10}}{\sqrt{10}} \end{bmatrix} = U\begin{bmatrix} \sqrt{20} & 0 \\ 0 & \sqrt{80} \end{bmatrix}$

$\therefore U = \begin{bmatrix} \frac{1}{\sqrt{10}} & \frac{3}{\sqrt{10}} \\ -\frac{3}{\sqrt{10}} & \frac{1}{\sqrt{10}} \end{bmatrix}$

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

$\prod_{n=1}^{N} P(y_n|x_n)$

$= \prod_{n=1}^{N} \theta(y_n w^Tx_n)$

$\underset{w}{\text{maximize}} \prod_{n=1}^{N} \theta(y_n w^Tx_n)$

$\Updownarrow$

$\underset{w}{\text{maximize}} \log\left(\prod_{n=1}^{N} \theta(y_n w^Tx_n)\right)$

$\Updownarrow$

$\underset{w}{\text{minimize}} -\sum_{n=1}^{N} \log(\theta(y_n w^Tx_n))$

$\frac{1}{1+e^{-s}} \rightarrow \left(\frac{1}{1+e^{-s}}\right)^{-1}$

$\Updownarrow$

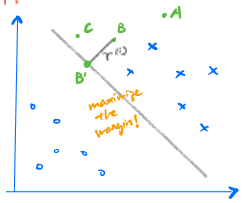
$\underset{w}{\text{minimize}} \sum_{n=1}^{N} \log(1 + e^{-y_n w^Tx_n})$

Linear Regression

- $\min_w \|X^Tw - y\|_2$

- solve: from previous!

Support Vector Machine



- Classifier

$h_{w,b}(x) = g(w^Tx + b)$

$g(z) \begin{cases} 1, & z \geq 0 \\ -1, & z < 0 \end{cases}$

- 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 hv $x^{(i)}$

$\therefore B' = x^{(i)} - \gamma^{(i)}\frac{w}{\|w\|}$

$\because w^TB' + b = 0$

$\therefore w^T\left(x^{(i)} - \gamma^{(i)}\frac{w}{\|w\|}\right) + b = 0$

$\therefore \gamma^{(i)} = \frac{w^Tx^{(i)} + b}{\|w\|}$

$= \left(\frac{w}{\|w\|}\right)^Tx^{(i)} + \frac{b}{\|w\|}$ (this is for (4))

$\therefore \gamma^{(i)} = y^{(i)}\left[\left(\frac{w}{\|w\|}\right)^Tx^{(i)} + \frac{b}{\|w\|}\right]$

$\gamma = \min_{i=1,\ldots,m} \gamma^{(i)}$

$\Downarrow$

$\max_{\gamma,w,b} \gamma$

s.t. $y^{(i)}(w^Tx^{(i)} + b) \geq \gamma$

$\|w\| = 1$ (this ensure geometric = functional)

- rewrite

$\max_{\hat{\gamma},w,b} \frac{\hat{\gamma}}{\|w\|}$

s.t. $y^{(i)}(w^Tx^{(i)} + b) \geq \hat{\gamma}$, $i=1,\ldots,m$

$\Leftrightarrow \max_{w,b} \frac{1}{\|w\|}$

s.t. $y^{(i)}(w^Tx^{(i)} + b) \geq 1$, $i=1,\ldots,m$

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

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

$\Rightarrow \nabla_w\mathcal{L}(w,b,\alpha) = w - \sum_{i=1}^{m}\alpha_i y^{(i)}x^{(i)} = 0$

$\therefore w = \sum_{i=1}^{m}\alpha_i y^{(i)}x^{(i)}$ — (1)

$\nabla_b\mathcal{L}(w,b,\alpha) = \sum_{i=1}^{m}\alpha_i y^{(i)} = 0$ — (2)

from (0), (1)

$\mathcal{L}(w,b,\alpha) = \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} y^{(i)}\alpha_i\alpha_j x^{(i)T}x^{(j)} - \sum_{i=1}^{m}\sum_{j=1}^{m} y^{(i)}\alpha_i\alpha_j x^{(i)T}x^{(j)} - b\sum_{i=1}^{m}\alpha_i y^{(i)} + \sum_{i=1}^{m}\alpha_i$ — (2)

$\Rightarrow \mathcal{L}(w,b,\alpha) = -\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} y^{(i)}y^{(j)}\alpha_i\alpha_j x^{(i)T}x^{(j)} + \sum_{i=1}^{m}\alpha_i$

Dual Problem

$\max_\alpha -\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} y^{(i)}y^{(j)}\alpha_i\alpha_j x^{(i)T}x^{(j)}$

s.t. $\alpha_i \geq 0$, $i=1,\ldots,m$

$\sum_{i=1}^{m}\alpha_i y^{(i)} = 0$

▲ - solve α, then use (1) get w

- w, b is $b^* = -\frac{\max_{i:y^{(i)}=-1} w^Tx^{(i)} + \min_{i:y^{(i)}=1} w^Tx^{(i)}}{2}$ (support vectors for (−) / support vectors for (+))

▲ from (1)

$w = \sum_{i=1}^{m}\alpha_i y^{(i)}x^{(i)}$ get w in terms of α

@ inference (deployment)

$wx + b$

$= \left(\sum_{i=1}^{m}\alpha_i y^{(i)}x^{(i)}\right)^Tx + b$

$= \sum_{i=1}^{m}\alpha_i y^{(i)}\langle x^{(i)}, x\rangle + b$

↳ $\alpha_i > 0$ only @ support vector

↳ w&b is actually just $\Sigma\langle$support vector$_{(i)}$, $x_{input}\rangle$

(Since we can express w in terms of $y^{(i)}$, $x^{(i)}$ and α_i (& since most $\alpha_i = 0$), we can write $w^Tx + b$ with $\langle x, x^{(i)}\rangle$ of support vectors!)

(Lagrangian) we can express w in terms of $y^{(i)}$, $x^{(i)}$ and α_i (& since most $\alpha_i = 0$), we can write $w^Tx + b$ with $\langle x, x^{(i)}\rangle$ of support vectors!

each training example could have at least γ margin

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

$\mathcal{L}(w,\alpha,\beta)$

$= f(w) + \sum_{i=1}^{k}\alpha_i g_i(w) + \sum_{i=1}^{l}\beta_i h_i(w)$

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

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

↳ $\max_{\alpha,\beta:\alpha\geq 0} f(w) + \sum_{i=1}^{k}\alpha_i g_i(w) + \sum_{i=1}^{l}\beta_i h_i(w)$

- if $g_i > 0$, then ∞
- if $h_i \neq 0$, then ∞
- $\therefore$ g_i, h_i constraints should be obeyed
- to get max, we want $\sum_{i=1}^{k}\alpha_i g_i(w)$ as possible as $\therefore \sum_{i=1}^{k}\alpha_i g_i(w) = 0$
- $\sum_{i=1}^{l}\beta_i h_i(w) = 0$

$\therefore \max_{\alpha,\beta:\alpha\geq 0} \mathcal{L}(w,\alpha,\beta) = \begin{cases} f(w) \\ \infty \end{cases}$

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

- Primal Problem (it is equivalent to original problem)

$\min_w \theta_P(w) = \min_w \max_{\alpha,\beta:\alpha\geq 0} \mathcal{L}(w,\alpha,\beta)$

- Dual Problem

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

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

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

KKT condition (when will $p^* = d^* = \mathcal{L}(w^*,\alpha^*,\beta^*)$ i.e., strong duality; Slater's condition; zero duality gap)

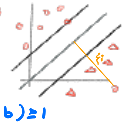
- $\nabla\mathcal{L}(w^*,\alpha^*,\beta) = 0$
- $h_i(w^*) = 0$
- $\alpha_i^* g_i(w^*) = 0$ dual complementarity (complementary slackness)
- $g_i(w^*) \leq 0$
- $\alpha^* \geq 0$

now back to SVM!

Support Vector Machine

This is why it is called Support Vector Machine! SVM

SVM + regularization (for outliers)



- recall SVM

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

s.t. $y^{(i)}(w^T x^{(i)} + b) \geq 1$

- add L1 regularization

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

s.t. $y^{(i)}(w^T x^{(i)} + b) \geq 1 - \xi_i$, $\xi_i \geq 0$

$\Leftrightarrow \min_w \frac{1}{2}\|w\|^2 + C\sum \xi_i$

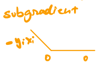
s.t. $y^{(i)}(w^T x^{(i)}) \geq 1 - \xi_i$, $\xi_i \geq 0$

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

$\Leftrightarrow \min_w C\sum_{i=1}^{m}\max(1 - y_i w^T x_i, 0) + \frac{1}{2}w^T w$

hinge loss — L2-reg.

use subgradient



Non-linear SVM

- some data ≠ linear separable
- perform ϕ nonlinear 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 compute ϕ, but apply kernel directly)
  - $\langle a, b \rangle$ replaced by
  - $K(a, b)$ for nonlinear transformation

Multi Class Classification

- 1 v. All
  - build t-binary classifier
    - (+), ∈ t^{th} class
    - (−), ∉ t^{th} class
  - decision value for t^{th} class
- 1 v. 1
  - $L(L-1)$ binary classifier
  - s-t
    - (+), ∈ s^{th} class
    - (−), ∈ t^{th} class
- multi class loss function
  - soft max
  - i-th data

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

- optimization problem

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

Optimization

Convexity

- convex function

$f(\theta x_1 + (1-\theta)x_2) \leq \theta f(x_1) + (1-\theta)f(x_2)$

1st order: $f(x) \geq f(x) + \nabla f(x)^T (x - x)$ ... 2nd order: $\nabla^2 f(x) \succeq 0$, $z^T \nabla^2 f(x) z \geq 0$

FACT

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

$f(\theta x + (1-\theta)y) \leq \theta f(x) + (1-\theta)f(y)$

$\theta \in [0, 1]$

FACT 1st order condition

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

FACT 2nd order condition

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

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

eg.

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

$g(z)$ is convex

how about $f(w)$?

proof:

- f is convex i.f.f.

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

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

$\leq \theta g(x^T w_1 + y) + (1-\theta)g(x^T w_2 + y)$

$\Rightarrow g(\theta x^T w_1 + (1-\theta)x^T w_2 + \theta y + (1-\theta)y)$

$\leq \theta g(x^T w_1 + y) + (1-\theta)g(x^T w_2 + y)$

$\Rightarrow g(\theta(x^T w_1 + y) + (1-\theta)(x^T w_2 + y))$

$\leq \theta g(x^T w_1 + y) + (1-\theta)g(x^T w_2 + y)$

$z_1 = x^T w_1 + y$, $z_2 = x^T w_2 + y$

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

as $g(z)$ is convex

$f(w)$ is convex

Lipschitz Continuity Smoothness

L is a constant

- L-Lipschitz continuous:

$\|f(x_1) - f(x_2)\|_2 \leq L\|x_1 - x_2\|_2$

- L-Lipschitz smooth:

$\|\nabla f(x_1) - \nabla f(x_2)\|_2 \leq L\|x_1 - x_2\|_2$

FACT $\nabla^2 f(x) \preceq L I$ 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:

$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|$

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

$= \|\nabla f(x) - \nabla f(y)\|$

$\leq L\|x - y\|$ (L-smooth)

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

from Cauchy-Schwarz inequality (upper bound)

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

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

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

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

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

$\Rightarrow \nabla^2 f(z) \preceq L I$

Cauchy-Schwarz inequality: $|\langle a, b\rangle| \leq \|a\|\|b\|$

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

- $z(t) = x + t(y - x)$, $g(t) = f(z(t))$

$\therefore g(0) = f(x)$, $g(1) = f(y)$

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

$g'(0) = \nabla f(x)^T (y - x)$

- fundamental theorem

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

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

$\leq \int_0^1 \|g'(t) - g'(0)\|\,dt$

- $\|g'(t) - g(0)\|$

$= \|\nabla f(z(t))^T (y - x) - \nabla f(x)^T (y - x)\|$

$= \|[\nabla f(z(t)) - \nabla f(x)]^T (y - x)\|$ → Lo.

$\leq L\|z(t) - x\| \cdot \|y - x\|$

$= L\|x + ty - tx - x\| \cdot \|y - x\|$

$= tL\|y - x\|^2$

- $\int_0^1 \|g'(t) - g'(0)\|dt \leq \int_0^1 tL\|y - x\|^2 dt$

$= L\|y - x\|^2 [\frac{1}{2}t^2]_0^1$

$= \frac{1}{2}L\|y - x\|^2$

- $\therefore g(1) - g(0) - g'(0)$

$= f(y) - f(x) - \nabla f(x)^T (y - x)$

$\leq \frac{1}{2}L\|y - x\|^2$

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

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

proof

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

$\geq f(w^t) + \nabla f(w^t)^T d + \frac{L}{2}\|d\|^2$

$\geq f(w^t + d)$

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

$\therefore$ 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

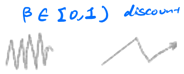
$V_t = (1-\beta)\nabla f(W_t) + \beta(1-\beta)\nabla f(W_{t-1}) + \beta^2(1-\beta)\nabla f(W_{t-2}) + \dots$ (similar to IIR)

@ 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

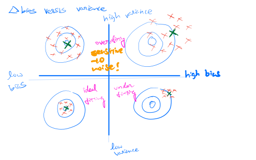
$\begin{bmatrix} w_{t+1}^1 \\ w_{t+1}^2 \\ \vdots \\ w_{t+1}^d \end{bmatrix} \leftarrow \begin{bmatrix} w_t^1 \\ w_t^2 \\ \vdots \\ w_t^d \end{bmatrix} - \begin{bmatrix} \frac{\alpha}{(\epsilon + G_t^{(1,1)})^{0.5}} & & 0 \\ & \frac{\alpha}{(\epsilon + G_t^{(2,2)})^{0.5}} & \\ 0 & & \ddots \end{bmatrix}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

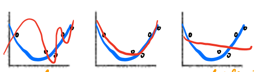
- $g_t \approx \frac{1}{b}\sum_{i=1}^{b} f_i(w)$
- $G_t = \sum_{\tau=1}^{t} g_\tau g_\tau^T$

Regularization

- L1-norm
- L2-norm

to prevent over-/under-fitting

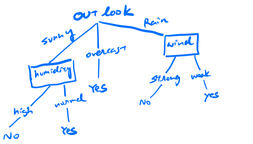




overfitting $\lambda = 0$ → underfitting $\lambda \uparrow$

Tree-based Models

- Play Baseball?



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

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

$H(S) = -(\frac{3}{8}\log_2\frac{3}{8} + \frac{3}{8}\log_2\frac{3}{8} + \frac{2}{8}\log_2\frac{2}{8}) = 1.58$ (2 as base) bad split

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

we want $H(S) \downarrow$!

- averaged entropy of a split

$S \to S_1, S_2$

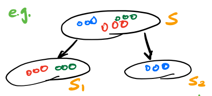
$\frac{|S_1|}{|S|}H(S_1) + \frac{|S_2|}{|S|}H(S_2)$

- 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.



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

$= 1.58$

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

$= 1$

$H(S_2) = -(\frac{1}{3}\log_2(1))$

$= 0$

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.

$H(S) = 1.58$

$H(S_1) = -(\frac{1}{3}\log\frac{1}{3} + \frac{3}{9}\log\frac{3}{9} + \frac{1}{3}\log\frac{1}{3})$

$= 1.52$

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

$= 1.5$

I.G.: $H(S) - (\frac{|S_1|}{|S|}H(S_1) + \frac{|S_2|}{|S|}H(S_2))$

$= 1.58 - (\frac{5}{9} \cdot 1.52 + \frac{4}{9} \cdot 1.5)$

$= 1.58 - 1.51 = 0.07$

- 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: $\nabla f(w) \approx \frac{1}{B}\sum_{n=1}^{B}\nabla f_n(w)$

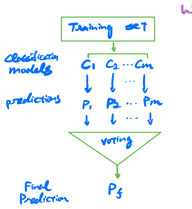
$|B|$ = batch size

Adam

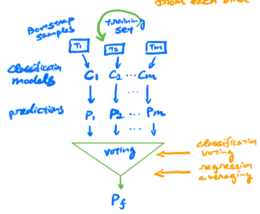
- Adagrad + Momentum
  - $g_t \leftarrow \nabla f_i(w_t)$
  - $m_t \leftarrow \beta_1 m_{t-1} + (1-\beta_1)g_t$
  - $v_t \leftarrow \beta_2 v_{t-1} + (1-\beta_2)g_t^2$
  - $\hat{m}_t \leftarrow m_t/(1-\beta_1^t)$
  - $\hat{v}_t \leftarrow v_t/(1-\beta_2^t)$
  - $w_t \leftarrow w_{t-1} - \alpha \cdot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

Ensemble Methods

- Stronger Together ensembles of weak learner!!!



- Bagging (Bootstrap aggregation) ↑ independent from each other



eg. Random Forest

can converge on this dataset — perform better on other dataset

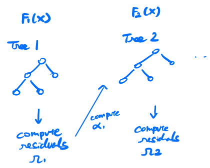
low bias → reduced variance

high variance (start) — train — might perform poorly on other dataset (end)

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

- Boosting (Gradient Boosting)

Dataset (X, Y)

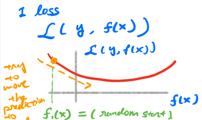


$x \to F(x)$

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

1 loss

$\mathcal{L}(y, f(x))$



2 Get $\hat{r}_i$

$\hat{r}_{mi} = -\frac{\partial L(y_i, \hat{y}_i)}{\partial \hat{y}_i}\Big|_{\hat{y}_i = F_m(x_i)}$

∀ 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?

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

$\approx F_m(x) + \gamma_m \hat{r}_m$

$= F_m(x) + \gamma_m \frac{\partial L(y, F_m(x))}{\partial F_m(x)}$

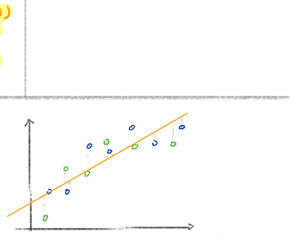
⇒ Hey! It is Gradient Descent!

Bias v. Variance !!!

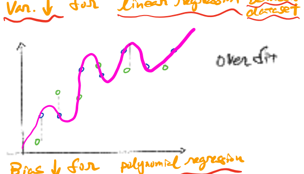
- Bias: unable to capture real relation
- Variance: difference in fits between datasets

ML target: Bias ↓ Var. ↓

method: bagging / boosting / regularization



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



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

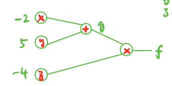
Neural Network

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



- $q = g(x, y)$ — $f = f(q, z)$

$\Rightarrow 3 = g(-2, 5)$ — $\Rightarrow -12 = f(3, -4)$

$\Rightarrow \frac{\partial q}{\partial x} = 1$ — $\Rightarrow \frac{\partial f}{\partial z} = q = 3$

$\frac{\partial q}{\partial y} = 1$ — $\frac{\partial f}{\partial q} = z = -4$

- $\frac{\partial f}{\partial x} = \frac{\partial q}{\partial x}\frac{\partial f}{\partial q} = 1 \cdot -4 = -4$

$\frac{\partial f}{\partial y} = \frac{\partial q}{\partial y}\frac{\partial f}{\partial q} = 1 \cdot -4 = -4$

$\frac{\partial f}{\partial z} = 3$

Activation function

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

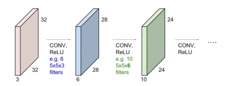
CNN kernel

eg. 1

- 32x32 input * 6x 5x5x1 convolution, 0 padding, 1 stride

output? 6x 28x28

e.g. 2



input 32x32x3 * conv 6 5x5x3

⇒ output? 28x28x6

CNN Pooling

- to down sampling

6x28x28 * max pool stride 2

output? 6x14x14

CNN Padding

- padding 2 ▢▢▢▢···▢▢▢

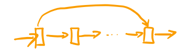
Overfitting

- 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

Theory of Generalization

FACT

△ with distribution in
training data $\mathcal{D}$
testing data
⇒ low training error
⫫
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 $x_1 \dots x_N$

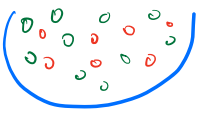
△ Testing error:
- $E_{te}(h) = \frac{1}{N}\sum_{n=1}^{N} e(h(x_n), f(x_n))$
where $x_1, \dots x_n$ sampled from $\mathcal{D}$
- h is independent from $x_1 \dots x_N$

△ 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 (sampled) could be applied to unseen data; w/ sufficient $E(h)$ & small $E_{tr}(h)$)
and
$E_{tr}(h) \approx 0$ → Training

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

FACT Hoeffding's inequality



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

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

△ $v \to u$?
perhaps

△ Hoeffding's inequality
(抽样的 ~ 真实的)
$$P[|v-u| > \epsilon] \le 2e^{-2\epsilon^2 N}$$
annote: v & u 的差距, 比 ϵ 大的 P, 很小多小? → 比 $2e^{-2\epsilon^2 N}$ 还小

△ statement $u = v$ is
probably approximately correct
(PAC !)

$v \approx u$ probably approximately correct

FACT

△ $P[|v-u| > \epsilon] \le 2e^{-2\epsilon^2 N}$
- valid for N
- $\epsilon > 0$
- independent from u (real probability)

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

↓

△ in learning:
- given a function h
- we randomly draw $x_1, \dots, x_n$] independent
- generalization error
$E(h) = E_{x\sim\mathcal{D}}[h(x) \neq f(x)] \Leftrightarrow u$ unknown
sample data error
$E_{tr}(h) = \frac{1}{N}\sum_{n=1}^{N}[h(x_n) \neq y_n] \Leftrightarrow v$ known

$P[|v-u| \ge \epsilon] \le 2e^{-2\epsilon^2 N}$
$P[|v-u| > \epsilon] \le 2e^{-2\epsilon^2 N}$

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

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

△ summary

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

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

NOTE More on Hoeffding's inequality

inferred hypothesis (假設現在我說我的模型 i.e., 不是學出來的) → 對來到我手上的資料 D1, D2 ... DN. inferred的 h on D 可能導致 "Bad" 亦即 Ein(h) ≠ Eout(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 就是在 fit 我 h... upper bound 是嗎?

∴ $P_D[\text{BAD } D]$

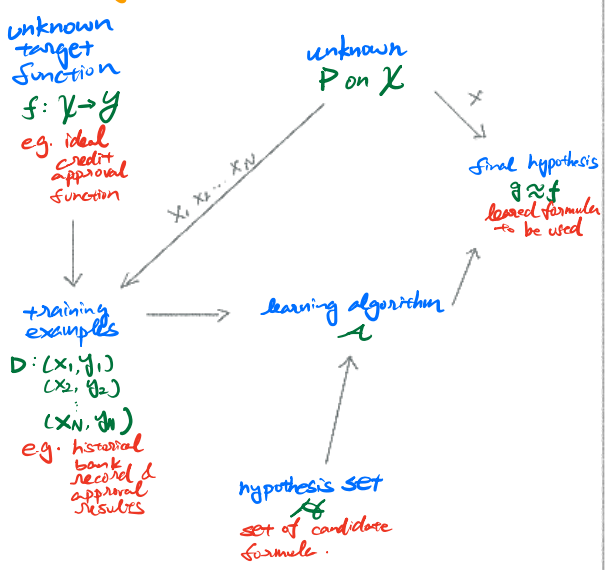
$= P_D[\text{BAD } D \text{ for } h_1 \text{ or BAD } D \text{ for } h_2 \cdots \text{or BAD } D \text{ for } h_M]$

$\le P_D[\text{BAD } D \text{ for } h_1] + P_D[\text{BAD } D \text{ for } h_2] + \cdots + P_D[\text{BAD } D \text{ for } h_M]$ (union bound)

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

- finite-bin version of Hoeffding
- & hope... Ein(g) = Eout(g) is PAC.
  → A will pick hm w/ min. Ein(hm) as g

△ summary: statistical learning flow



& 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 $|\mathcal{H}|$ (一當 hypothesis set 小 - P[BAD] 發生小 - 但小 |H|, 我訓練找到 h 去 fit Ein(h) 的 Ein(h) 小)
  $P[\text{BAD}] \le 2|\mathcal{H}| e^{-2\epsilon^2 N}$
  small! great!
  but |H| too little
  Ein(g) ↑
- large $|\mathcal{H}|$ (我 |H| 大 到時找得到 Ein(h) ≈ Eout(h) 但 P[BAD] 大!)
  Ein(g) → 0
  small error! great!
  but |H| too large
  P[BAD] ↑

註: 我們如何找 finite |H| (|H| is ∞ no doubt but can control the number!)

FACT establish a finite quantity replace |H|

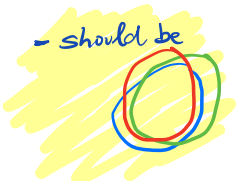
let |H| replaced by $m_{\mathcal{H}}$ s.t.

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

FACT |H| is over-estimated for BAD events

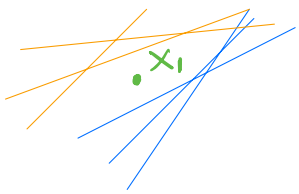
- BAD events B_m : $|E_{in}(h_m) - E_{out}(h_m)| > \epsilon$
- over-lapping for similar hypothesis $h_1 \approx h_2$
- as ① $E_{out}(h_1) \approx E_{out}(h_2)$

② for most D $E_{in}(h_1) \approx E_{out}(h_2)$

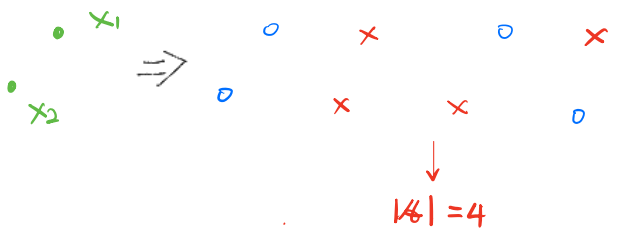
- should be  instead of 

- so: can we group similar kinds?

eg. H in R^2 $|\mathcal{H}| \to \infty$



→ |H| = 2



|H| = 4

N = 3 |H| = 8 but if on same line, different

N = 4 |H| = 14 but if on same line, different

FACT observation: effective |H| ≤ 2^N

perhaps can replace |H| by effective |H| ?

need more rigorous proof

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

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

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

△ growth function:

remove dependence by taking max of all possible $(x_1, x_2 \ldots, x_N)$

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

△ 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 開始，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)

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

△ study minimum break point

eg linear case break point $k=4$

note: 4 ↑ 無法被 shattered

FACT conjecture:

△ no break point : $m_{\mathcal{H}}(N) = 2^N$

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

proof ?

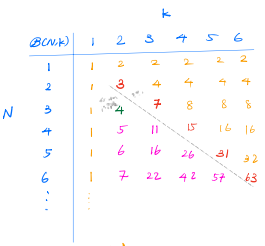
FACT $m_{\mathcal{H}}(N) \le$ maximum possible $m_{\mathcal{H}}(N)$ given k $\le$ poly (N)

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

- 從上面結論：
  - 在 break point $= k$ ⇝ 露出一線曙光！
  - 任 k 維度，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

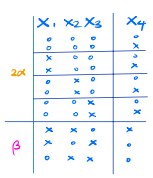
△ $k>N$, max. dichotomies = 2^N

△ $k=N$, max. dichotomies = 2^N-1

△ $k<N$? PINK

PINK:

△ $B(4,3) = 2\alpha+\beta$



△ if only consider $x_1 x_2 x_3$ we have 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 have 2^3

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

∴ $\alpha+\beta \le B(3,3)$

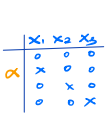
- $B(N,k) \le \sum_{i=0}^{k-1} \binom{N}{i}$
- Growth Function $\le$ Bounding Function $\le \sum_{i=0}^{k-1} \binom{N}{i}$

e.g. $B(4,3) \le \binom{4}{0}+\binom{4}{1}+\binom{4}{2} = 1+4+6 = 11$

只要有 break point 有界无！

可被 polynomial 上限住！

△



∵ x_3 provide dichotomies "o" & "x"

∴ within $x_1 x_2$, we cannot shatter any 2 pts

∴ $\alpha \le B(3,2)$

△ ∴ $2\alpha+\beta \le B(3,2)+B(3,3)$

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

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

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

$E_{out}(h)$ it 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'

- $\mathcal{H} = \{\mathcal{H}(x_1,\dots,x_N,x_1',\dots,x_N')\}$
- ∵ 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 $N \ge 2, k \ge 3$ $m_{\mathcal{H}}(N) \le B(N,k) = \sum_{i=0}^{k-1}\binom{N}{i} \le N^{k-1}$
- 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

△ $N > d_{vc} \Rightarrow k$ does not shatter N points (k)

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

eg 2D ∴

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

let $Xw = y$

∴ $w = X^{-1}y$

- 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

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

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

⇒ $w^T x_4$ must be (+) cannot be (−) therefore : not shattered.

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

$X = \begin{bmatrix} -x_1^T- \\ -x_2^T- \\ -x_{d+2}^T- \end{bmatrix}$ ⇒ rows → cols

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

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

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

cannot be shattered !