$L_1 \overset{\text{optim}}{\Leftrightarrow}$ Laplacian error

$L_2 \Leftrightarrow$ Gaussian.

Examination Aid Sheet

Faculty of Applied Science & Engineering

Both sides of the sheet may be used;
must be printed on 8.5" x 11" paper.

Subject: ECE 521

Candidate's name: King Fung

Candidate's signature:

Definitions

CDF(x) $= \int_S^x p(x')\,dx'$

CDF(∞) $= 1$

$E[x] = \int_S x P(x)\,dx$

$Var[x] = \int_S (x-E[x])^2 P(x)\,dx$

$= E[x^2] - E[x]^2$

$Var[\mathcal{N}] = \sigma^2$

Notation: S: sample space e.g. $S=\{1,2,3\}$, $S=\mathbb{R}$

E: Event: $E \subseteq S$ e.g. $E=\{1,2\}$, $E=[0,1)$

X: Random Variable: $X: S \to \mathbb{R}$

$X=6 \Rightarrow$ eqn where S st $X(S)=6$ is the sol^n

P: Probability: $P(S)=1$, $P(E)\geq 0$

Properties: $\sum_{S_x} P(x) = \int_{S_x} P_c(x) = 1$ (PDF) $\mathcal{N}$

Functions: $\mathcal{N}(x;\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\{-\frac{(x-\mu)^2}{2\sigma^2}\}$, $S_x \in \mathbb{R}$ | $Beta(x;\alpha,\beta) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{B(\alpha,\beta)}$, $S_x=[0,1]$

Joint Probability: $P(x,y)$:

Marginal $P(x)$, $P(y) \Rightarrow P(x) = \int_{S_y} P(x,y)\,dy$

Conditional $P(x|y) = \frac{P(x,y)}{P(y)}$

Total Probability Thm: $P(x) = \sum_{y\in S_y} P(y)P(x|y) = \int_{S_y} P(y)P(x|y)\,dy$

Bayes Theorem: $P(y|x) = \frac{P(x|y)P(y)}{P(x)} = \frac{P(x|y)P(y)}{\sum_{S_y} P(x|y)P(y)}$

Models: kNN, GPR, Lin/Log reg, Neural

Loss fns: $L_p = \|\hat{y}-y\|_p^p = \sum_{i=1}^N (|\hat{y}_i - y_i|)^p$

$p=1$: Taxicab $p=2$: Euclidean

Learning:
for some passes over data set (epochs):
for some iterations:
refine search dir
move in search dir

Just refinement:
steepest descent: $W \leftarrow W - \eta \frac{\partial \mathcal{L}}{\partial W}$

Convexity ↓

→ Local min = global min.

→ always. Jensen's inequality: $\forall \alpha \in [0,1]$:
$f(\alpha W_1 + (1-\alpha)W_2) \leq \alpha f(W_1) + (1-\alpha) f(W_2)$

KNN: Given M data pts $\{(x^{(m)}, t^{(m)})\}$ find the nearest k data pts $L_2(x^* - x^{(m)})$ dist & output the avg/majority (regress / class)

Momentum (Classical): $\Delta W_{NoMo}: -\eta \frac{\partial \mathcal{L}(t)}{\partial W}$ $\Delta W_{CMo}(t) = -\eta \frac{\partial \mathcal{L}(t)}{\partial W} + \alpha \Delta W(t-1)$, $\alpha \in [0,1)$

Nesterov's: t', $\alpha \Delta W(t-1)$, $-\eta\frac{\partial \mathcal{L}(t')}{\partial W}$ $\Delta W_{NMo} = -\eta \frac{\partial \mathcal{L}(t')}{\partial W} + \alpha \Delta W(t-1)$

Stochastic (+Minibatch) Gradient Descent:
$\mathcal{L}(W)$ expensive as $M \uparrow\uparrow$
$\frac{\partial \mathcal{L}(W)}{\partial W}$ more so.

Choose $M' < M$ samples:
$\Rightarrow \mathcal{L}(W) = \sum_{m}^{M'} \mathcal{L}(x^{(m)}, y^{(m)}, W)$
$\frac{\partial \mathcal{L}(W)}{\partial W} = \sum_{m}^{M'} \frac{\partial \mathcal{L}_m(x^{(m)}, y^{(m)}, W)}{\partial W}$

Key tricks:
divide $\mathcal{L}$ & $\frac{\partial \mathcal{L}}{\partial W}$ by batch size, subsample w/o replace
Learning rates:
$\sum_{t=0}^{\infty} \eta_t = \infty$ $\sum_{t=0}^{\infty} \eta_t^2 < \infty$ e.g. $\eta = \frac{1}{t}$

MLE vs MAP
$\frac{\partial P}{\partial \text{param}} = 0$ or $\frac{\partial \log P}{\partial \text{param}} = 0$
$P(\text{param}|\text{data}) = \frac{P(d|p)P(p)}{P(\text{data})}$ ↑ requires a prior

MLE for $\mathcal{N}(x|\mu,\sigma^2)$:
$\ln \mathcal{N}(x|\mu,\sigma^2) = -\frac{1}{2\sigma^2}\sum_n^N (x_n-\mu)^2 - \frac{N}{2}\ln\sigma^2 - \frac{N}{2}\ln 2\pi$
$\frac{\partial \ln \mathcal{N}}{\partial \mu} = \frac{1}{\sigma^2}\sum_{i=1}^N x_n - \mu = 0 \Rightarrow \mu_{ML} = \frac{1}{N}\sum_{i=1}^N x_n$

Multivariate $\mathcal{N}_D$: $X = (x_1 \cdots x_D)^T \Rightarrow$ i.i.d, gaussian
$\mathcal{N}_D(x|\mu,\Sigma) = \frac{1}{(2\pi)^{D/2}}\frac{1}{|\Sigma|^{1/2}}\exp\{-\frac{1}{2}\underbrace{(x-\mu)^T\Sigma^{-1}(x-\mu)}_{\Delta^2}\}$
D dim, $D\times D$

$E[x] = \mu$.
$Cov[x] = \Sigma$

$\frac{\partial \ln \mathcal{N}}{\partial \sigma^2} = \frac{1}{2(\sigma^2)^2}\sum_{i=1}^N (x_n-\mu_{ML})^2 - \frac{N}{2}\frac{1}{\sigma^2} = 0 \Rightarrow \sigma^2_{ML} = \frac{1}{N}\sum_{i=1}^N (x-\mu_{ML})^2$

MLE of $\mathcal{N}_D$: $\mu_{ML} = \frac{1}{N}\sum_{n=1}^N x_n$ $\Sigma_{ML} = \frac{1}{N}\sum_{n=1}^N (x_n-\mu_{ML})(x_n-\mu_{ML})^T$

LL: $\ln \mathcal{N}_D = -\frac{ND}{2}\ln(2\pi) - \frac{N}{2}\ln|\Sigma| - \frac{1}{2}\sum_{i=1}^N (x_n-\mu)^T\Sigma^{-1}(x_n-\mu)$

Precision Matrix & Disjoint Sets: $X \Rightarrow (x_a, x_b)$
$\mu \Rightarrow (\mu_a, \mu_b)$
$\Sigma = \begin{pmatrix}\Sigma_{aa} & \Sigma_{ab}\\ \Sigma_{ba} & \Sigma_{bb}\end{pmatrix}$

define $\Lambda = \Sigma^{-1}$, $\Lambda = \begin{pmatrix}\Lambda_{aa} & \Lambda_{ab}\\ \Lambda_{ba} & \Lambda_{bb}\end{pmatrix}$ (Λ_{aa} not Σ_{aa}^{-1})

COND:
$p(x_a|x_b) = \mathcal{N}(x_a|\mu_{a|b}, \Sigma_{a|b})$
$\Sigma_{a|b} = \Lambda^{-1} = \Sigma_{aa} - \Sigma_{ab}\Sigma_{bb}^{-1}\Sigma_{ba}$.
$\mu_{a|b} = \mu_a + \Sigma_{bb}^{-1}(x_b - \mu_b)$

Marg: $p(x_a) = \mathcal{N}(x|\mu_a, \Sigma_{aa})$

Least Squares.
Error: $\frac{1}{2}\sum_{i=1}^{N}(x_n^T w - t_n)^2 = \frac{1}{2}\mathcal{L}_2$.

Learning a Parametric Model
Max μ (i.i.d)
$P = \prod_{m=1} P(x^{(m)}, y^{(m)}|w)$ $\Rightarrow$

Max
$W^* = \arg\max \log P$
$= \arg\min_W -\log P$ ($\mathcal{L}(w)$)

Regularizing
$E = E_D(w) + E_W(w)\lambda$
$= \frac{1}{2}\mathcal{L}_2 + \frac{\lambda}{2}|w|^q$
$q=1$: lasso $\Rightarrow$ sparse.
$q=2$: quad $\Rightarrow$ std.

Disc: $P(x,y) = P(y|x,w)P(x)$
Gen: $P(x,y|w) = P(x|y,w)P(y)$
Disc $\Rightarrow$ outputs (find) Gen: inputs find.

This MAP $= \mathcal{L}_2$ penalization

MLE of gaussian:
$W = (X^TX)^{-1}X^TY$
↳ $\arg\min_{w,b} -\log P$

MAP: Assume. $W \in \mathcal{N}(w_{n,i}|0,\frac{1}{\lambda})$
↳ $\log P(W) = -\sum_{n=1}^{N}\frac{\lambda}{2}\|W_2\|_2^2 + c$

Parametric Dist:
$P(x,y|W)$: likelihood of x,y given W.

Loss Fcn.
L_2: $\mathbb{E}[\mathcal{L}_2] = \iint (y-\hat{y}(x))^2 P(x,y)\,dx\,dy$
↳ min when $\hat{y}(x) = E[y|x]$.

Minkowski
L_q $E[\mathcal{L}_q] = \iint \|y-\hat{y}(x)\|_q^q P \ldots$
minimize:
$q=0 \Rightarrow$ mode $q=2$ mean
$q=1 \Rightarrow$ median ($q<2$ ✓ noise)

Bias/Variance tradeoff:
↳ regularize: $\lambda\uparrow$ bias $\uparrow$, $\lambda\downarrow$ var $\uparrow$: $\mathbb{E}[L_2] = \int (\hat{y}(x) - \mathbb{E}[y|x])^2 P(x)\,dx$ ⇒ comp. to square.

$= \int (\mathbb{E}[\hat{y}(x)] - \mathbb{E}[y|x])^2 P(x)\,dx + \int [\hat{y}(x) - \mathbb{E}[\hat{y}(x)]]^2 P(x)\,dx$

Linear Basis Funct Model

$y(x,w) = \sum_{j=0}^{M-1} w_j \phi_j(x)$ ϕ_0 typ $=1$

e.g. $\phi_j(x) = x^j$

$\phi_j(x) = \exp\left(-\frac{(x-\mu_j)^2}{2s^2}\right)$

$\phi_j(x) = \sigma\left(\frac{x-\mu_j}{s}\right)$

$L_s(a) = \frac{1}{1+\exp(-a)}$

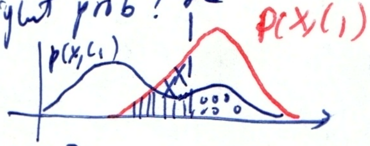
MLE: $w_{ML} = (\Phi^T \Phi)^{-1} \Phi^T t$ ← norm eqn

iterative: $w^{(\tau+1)} = w^{(\tau)} - \eta \nabla E = w^{(\tau)} - \eta\left(t_n - w^{(\tau)T}\phi(x_n)\right)\phi(x_n)$

↳ w/ quad penalization: $w^* = (\lambda I + \Phi^T\Phi)^{-1}\Phi^T t$.

Classification: given a sample, C_1 & C_2 class, correctly classify

$p(C_k|x) = \frac{p(x|C_k)}{\sum p(x|C_k)} = \frac{p(x|C_k)\,p(C_k)}{p(x)}$ ⇒ sample highest prob? yes



$p(\text{mistake}) = p(x \in R_1, C_2) + p(x \in R_2, C_1)$ ⇒

$\boxed{X}$

✱ can multiply err cost by const to weigh e.g. $\begin{bmatrix} 0 & 1000 \\ 1 & 0 \end{bmatrix}$

Decision theory:
can be simple equal weights, but not true: e.g.

$$\begin{array}{ccc} & \text{cancer} & \text{normal} \\ \text{cancer} & 0 & 1000 \\ \text{normal} & 1 & 0 \end{array}$$

(class'd) → truth ⇒ Hard to figure out these matrices

Log regression.

$\sigma(z) = \frac{1}{1+e^{-z}}$

$\sigma: \mathbb{R} \to (0,1)$

$\sigma(-z) = 1 - \sigma(z)$

$\sigma(0) = 0.5$

w/ L_2 loss, & $\hat{y} = \sigma(w^T x^{(m)} + b)$

$\frac{\partial L}{\partial w} = \sum_m (\hat{y}^{(m)} - t^{(m)})\,\hat{y}^{(m)}(1-\hat{y}^{(m)})\,x^{(m)}$ ⇒ $= \sum \frac{\partial L}{\partial \hat{y}^m}\frac{\partial y}{\partial z}\frac{\partial z}{\partial w}$

also, $\frac{\partial \sigma}{\partial z} = \sigma(z)(1-\sigma(z))$ near 0 ~~zero~~ 0 :(

decision bound drawback: all class1: $w\uparrow C_1(+)$; 50% C_1, 50% C_2 ⇒ $w \Rightarrow 0$

Too C_1 ⇒ bad MLE
↳ use MAP to constrain

other loss: CROSS ENTROPY

$L_x = \sum_m -t^{(m)}\log \sigma(z^{(m)}) - (1-t^{(m)})\log(1-\sigma(z^{(m)}))$

$\frac{\partial L}{\partial W} = \sum (\hat{y}^{(m)} - t^{(m)})\,x^{(m)}$

↳ $= \sigma(z^{(m)}) = \sigma(W^T x^{(m)} + b)$

Measuring distance in classification

$KL(Q\|P) = \sum Q(x) \log\frac{Q(x)}{P(x)}$ non neg, non sym

↳ if $P(t=1|x) = \hat{y}(x)$, $KL(Q\|P)$ ⇒ $-t\log\hat{y} - (1-t)\log(1-\hat{y})$

$t = Q(t=1|x)$ ∈

Softmax & Multiclass, $t \in \{0, 1 \ldots k\}$ $= -\sum_{k=1}^{K} I(k, t_{class}) \log P(t=k|x)$

↳ $P(t=k|x) = \frac{e^{z_k}}{\sum_j e^{z_j}}$ wh. z the out.

Neurons

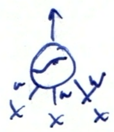
sign: $\begin{cases} 1 & z \geq 0 \\ 0 & z < 0 \end{cases}$ $\frac{\partial s}{\partial z} = 0\ \forall z \neq 0$ ~~$z = \sum w_n x_n + b$~~ $z = \sum w_n x_n + b$

$\sigma(z) = \frac{1}{1+e^{-z}}$ $\frac{\partial\sigma}{\partial z} = \sigma(z)(1-\sigma(z))$

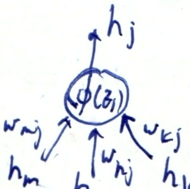
$f(z) = z$ $\frac{\partial\sigma}{\partial z} = 1$

ReLU $= \max(0,z)$ $\frac{\partial \text{ReLU}}{\partial z} = \begin{cases} 1 & z > 0 \\ 0 & z < 0 \end{cases}$

tanh x $1 - \tanh^2 x$



In general: a neuron.



Forward Prop:

$h_j = \phi(z_j) = \phi\left(\sum_n w_{nj} h_n + b_j\right)$

Back Prop

$\frac{\partial L}{\partial w_{nj}} = \frac{\partial L}{\partial h_j}\frac{\partial h_j}{\partial z_j}\frac{\partial z_j}{\partial w_{nj}}$

$= \frac{\partial L}{\partial z_j} h_n$!!!

input → ∂z_j

$\frac{\partial L}{\partial z_1} = \frac{\partial L}{\partial \hat{y}}\frac{\partial \hat{y}}{\partial z_2}\frac{\partial z_2}{\partial z_1}$

$\frac{\partial L}{\partial W_1} = \frac{\partial L}{\partial z_1} h_1^T$

+ z for i≥2+

Learning Multilayer:

$z^1 = w^1 x + b^1 \Rightarrow h^1 = \phi(z^1)$

$z^2 = w^2 h^{(1)} + b^2 \Rightarrow h^2 = \phi(z^2)$

$z^N = w^N h^{(N-1)} + b^N \Rightarrow \hat{y} = f(z^N)$

Back:

$\frac{\partial L}{\partial z^{(N)}} = \frac{\partial \hat{y}}{\partial z^N}\frac{\partial L}{\partial \hat{y}}$, $\frac{\partial L}{\partial W^N} = \frac{\partial L}{\partial z^N} h^{(N-1)T}$

$\frac{\partial L}{\partial z^{(n)}} = \frac{\partial h^{(n)}}{\partial z^n}\left(W^{(n+1)}\frac{\partial L}{\partial z^{(n+1)}}\right)$,

$\frac{\partial L}{\partial W^{n+1}} = \frac{\partial L}{\partial z^n} h^{(n-1)}$

Strategies:

Hyperparams: 2-3 layers w/ 500 units → validate. ReLu best practice.

Generalization: wide shallow > narrow deep

Init: (identity or $N(x)\,\sigma\frac{\pi}{\sqrt{in+o}}$)
↳ better
↳ or use pretrained weights

Overfitting:
↳ stop once valid err ↑

Dropout: randomly drop neurons for fwd/bwd prop, $E[w]$ used.

CNN: local connect, w/ DEPTH

RNN: use the same weights for multiple layers & whence