CS540 SP22 Midterm Note   Ruixuan Tu

Probability $P(E_1 \cup E_2) = P(E_1) + P(E_2) - P(E_1 \cap E_2)$ $P(X=a) = \sum_b P(X=a, Y=b)$

Expectation/mean $E[X] = \sum_a a P(X=a)$ Variance $Var[X] = E[(X-E[X])^2]$

Independence $P(X,Y) = P(X)P(Y)$ Conditional Prob $P(X=a \mid Y=b) = \frac{P(X=a, Y=b)}{P(Y=b)}$

Conditional Ind $P(X,Y \mid Z) = P(X \mid Z) P(Y \mid Z)$ Chain Rule $P(A_1, A_2, \ldots, A_n) = P(A_1) P(A_2 \mid A_1) P(A_3 \mid A_2, A_1) \ldots$

Bayes' Rule $\underset{\text{posterior}}{P(F \mid S)} = \frac{P(F,S)}{P(S)} = \frac{\overset{\text{likelihood}}{P(S \mid F)} \, P(F)}{P(S)}$ ← prior

PCA $\mu_X = \frac{1}{n} \sum_{i=1}^{n} \vec{X}_i$, $\vec{X}_i = \vec{X}_i - \mu_X$ so that new $\mu_X = 0$ → covariance matrix $S = \frac{1}{n-1} \sum_{i=1}^{n} \vec{X}_i \vec{X}_i^T = \frac{1}{n-1} X^T X = U \Lambda U^T$ (U: eigenvectors $\vec{u}_1, \ldots, \vec{u}_d$; Λ: eigenvalues)

high dim → low $X = X - \bar{X}$ (mean) Σ eigenvalues = Σ diagonal elements   ← new representation of $\vec{X}_i$ is $(\vec{u}_1^T \vec{X}_i, \ldots, \vec{u}_d^T \vec{X}_i)$

$X_{ij}^{\text{pro}} = \sum_{j=1}^{m} a_{ij} u_j$, $a_{ij}^{\mathbb{R}^m} = u_j^T X_i$, $\mathbb{R}^d$

Logic Precedence $\neg, \wedge, \vee, \Rightarrow, \Leftrightarrow$ Entailment $A \models B$ $\boxed{B \boxed{A}}$ iff in every interpretation where A true, B also true

Logical equivalences $(a \wedge b) \equiv (b \wedge a)$ $(a \vee b) \equiv (b \vee a)$ commutativity   XOR $A \oplus B = (A \wedge \neg B) \vee (\neg A \wedge B)$

$((a \wedge b) \wedge c) \equiv (a \wedge (b \wedge c))$ $((a \vee b) \vee c) \equiv (a \vee (b \vee c))$ associativity

$\neg(\neg a) \equiv a$ double negation $(a \Rightarrow b) \equiv (\neg b \Rightarrow \neg a)$ contraposition $(a \Rightarrow b) = (\neg a \vee b)$ implication

$(a \Leftrightarrow b) \equiv ((a \Rightarrow b) \wedge (b \Rightarrow a))$ biconditional $\neg(a \wedge b) \equiv (\neg a \vee \neg b)$ $\neg(a \vee b) \equiv (\neg a \wedge \neg b)$ de Morgan

$(a \wedge (b \vee c)) \equiv ((a \wedge b) \vee (a \wedge c))$ $(a \vee (b \wedge c)) \equiv ((a \vee b) \wedge (a \vee c))$ distributivity

CNF ① replace $\Leftrightarrow$ ② replace $\Rightarrow$ ③ move negation inward ④ apply distributivity of $\vee$ over $\wedge$

NLP $k=0$: Unigram $P(w_1, w_2, \ldots, w_n) = P(w_1) P(w_2) \ldots P(w_n)$ full independence assumption

$k=1$: Bigram $P(w_1, w_2, \ldots, w_n) = P(w_1) P(w_2 \mid w_1) P(w_3 \mid w_2) \ldots P(w_n \mid w_{n-1})$ Markov assumption

$k=n-1$: n-gram $P(w_i \mid w_{i-1}) = \frac{count(w_{i-1}, w_i)}{count(w_{i-1})} \xrightarrow{\text{smoothing}} \frac{count(w_{i-1}, w_i) + 1}{count(w_{i-1}) + V}$ (zero prob solved)   bigram (training count) $P(w_1, \ldots, w_n) = P(w_1) P(w_2 \mid w_1) P(w_3 \mid w_2 w_1) \ldots$

perplexity $PP(W) = P(w_1, w_2, \ldots, w_n)^{-\frac{1}{n}}$ lower is better   similarity $P(o \mid c) = \frac{\exp(u_o^T v_c)}{\sum_{w \in V} \exp(u_w^T v_c)}$ (o: context word, c: center) maximize this

Word2vec likelihood $L(\theta) = \prod_{t=1}^{T} \prod_{-a \le j \le a} P(w_{t+j} \mid w_t, \theta)$ (θ: word vector (variable/hypothesis); t: all positions; window length of $2a$)   two vectors per word v_w, u_w for center/context

ML Supervised: Classification (label discrete), regression (label continuous)   training set (multiset): $\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$

i.i.d. assumption: $(x_i, y_i) \sim P_{XY}$ (x_i: feature/covariance/input; y_i: label/output)   goal: learn mapping $f: X \to Y$

loss function: 0-1 loss for classification $\ell(f, \vec{x}, y) = \mathbb{1}_{[f(\vec{x}) \neq y]} = \begin{cases} 1 & \text{if } f(\vec{x}) \neq y \\ 0 & \text{otherwise} \end{cases}$ $f(x)$ predicts label y of future x

squared loss for regression $\ell(f, \vec{x}, y) = (f(\vec{x}) - y)^2$ empirical risk/training set error $\hat{R}(f) = \frac{1}{n} \sum_{i=1}^{n} \ell(f, \vec{X}_i, y_i)$ ← want $\hat{f} = \underset{f \in \mathcal{F}}{\text{argmin}} \, \hat{R}(f)$ from learning

Unsupervised: dim reduction, clustering (output: division into clusters with intra-cluster similarity and inter-cluster dissimilarity)   given: data with no label $x_1, \ldots, x_n$ goal: discover patterns and structures in data

Unsupervised I K-means ① select k cluster centers $c_1, \ldots, c_k$ ② for each point x, find closest center in Euclidean distance $y(x) = \underset{i=1:k}{\text{argmin}} \|x - c_i\|$ change x assignments $y(x)$ to centers

③ update cluster centers as centroids $c_i = \frac{\sum_{x: y(x)=i} x}{\sum_{x: y(x)=i} 1}$ change centers $c(d)$ ④ update by repeating ②, ③ until convergence

minimizes distortion $\sum_x \sum_{d=1 \ldots D} [x(d) - c_{y(x)}(d)]^2$ hill-climbing

decide k: $\underset{k}{\text{argmin}}$ distortion $+ \lambda D k \log N$ (D: #dim, k: #clusters, N: #points)

Hierarchical (no need k) input: points output: a hierarchy (binary tree)   maximum depth on n points: $n-1$

agglomerative: bottom-up / divisive: top-down

↳ repeat: get a closest pair of clusters and merge

↳ single-linkage $\min_{x_1 \in A, x_2 \in B} d(x_1, x_2)$

complete-linkage $\max_{x_1 \in A, x_2 \in B} d(x_1, x_2)$

average-linkage $\frac{1}{|A||B|} \sum_{x_1 \in A, x_2 \in B} d(x_1, x_2)$

Unsupervised II Part

Density Estimation goal: given samples $X_1, \dots, X_n$ from distribution P, estimate P

simplist idea: histograms define bins; count # of samples in each bin, normalize

Kernel Density Estimation density as combination of kernels

$$f(x) = \frac{1}{nh} \sum_{i=1}^{n} \underbrace{K}_{\text{kernel func, often Gaussian}}\left(\frac{x - x_i}{\underbrace{h}_{\text{width param}}}\right) \leftarrow \text{center at each point}$$

Linear Regression

goal: minimize square loss

train set: $(\vec{x}_1, y_1), (\vec{x}_2, y_2), \dots, (\vec{x}_n, y_n)$ notation: $f(x) = \theta_0 + x^T\theta$, let $x = \begin{bmatrix} 1 \\ x \end{bmatrix}$, then $f(x) = x^T\theta$

$X = \begin{bmatrix} x_1^T \\ \dots \\ x_n^T \end{bmatrix}$ ($n*(d+1)$ matrix) $y = \begin{bmatrix} y_1 \\ \dots \\ y_n \end{bmatrix}$ empirical risk: $\frac{1}{n}\|X\theta - y\|^2$ error/residual: $|y_i - f(x_i)| = |y_i - \hat{\theta}^T x_i|$

solution: $\hat{\theta} = (X^TX)^{-1}X^Ty$ $MSE = \frac{1}{m}\sum_{i=n+1}^{n+m}(\hat{\theta}^T x_i - y_i)^2$

logistic regression: $y \in \{0,1\}$ $\theta^T x \in [0,1]$ $P(y=1|x) = \frac{1}{1+\exp(-\theta^T x)}$

Classification

KNN input: train data $(\vec{x}_1, y_1), \dots, (\vec{x}_n, y_n)$; distance func $d(\vec{x}_i, \vec{x}_j)$; num of neighbors k, test data $\vec{x}^*$

① find k training instances $\vec{x}_{i1}, \dots, \vec{x}_{ik}$ closet to x^* under $d(\vec{x}_i, \vec{x}_j)$

② output y^* as the majority class of $y_{i1}, \dots, y_{ik}$; break ties randomly (for classification)

② (for regression) output the predicted $y^* = \frac{1}{k}(y_{i1} + \dots + y_{ik})$

distance: categorical features — Hamming ~~dif~~ distance: count (same position, diff items in 2 strs)

numerical features — p-norm ($p \geq 1$) $d(x, x') = \|x - x'\|_p = \left(\sum_{i=1}^{d} |x_i - x_i'|^p\right)^{\frac{1}{p}}$

used other situations: 2-norm = Euclidean $d(x, x') = \|x - x'\|_2 = \sqrt{\sum_{i=1}^{d}(x_i - x_i')^2}$

1-norm = Manhattan $d(x, x') = \|x - x'\|_1 = \sum_{i=1}^{n}|x_i - x_i'|$

pick (k) data: ① training, tuning, test shuffled & splitted randomly

② classify tuning with different k

③ pick k with least tuning error

④ report test error

Error/Accuracy: $\text{error} = \frac{1}{n}\sum_{i=1}^{n} 1[f(x_i) \neq y_i]$ accuracy = 1 − error

MLE labeled train data $(\vec{x}_1, y_1), \dots, (\vec{x}_n, y_n) \rightarrow$ learning algo $\rightarrow$ classifier $\hat{f}$

i.i.d. (fixed underlying distribution)

MLE (Maximum likelihood estimation): best fits data MAP (Maximum a posteriori): best fits data & prior assumption

$\theta_{MLE} = \arg\max_\theta P(X|\theta)$ likelihood func: $L(\theta) = \prod_i p(\vec{x}_i|\theta)$ $\theta_{MAP} = \arg\max_\theta P(X|\theta)P(\theta)$ $\theta^* = \arg\max \ell(\theta) = \frac{N_H}{N_T + N_H}$

↳ Bernoulli likelihood func $L_D(\theta) = \theta^{N_H} \cdot (1-\theta)^{N_T}$ Log-likelihood $\ell(\theta) = N_H \log\theta + N_T \log(1-\theta)$

Gaussian likelihood func $L(\mu, \sigma^2|X) = \prod_{i=1}^{n} p(x_i; \mu, \sigma^2) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x_i - \mu)^2}{2\sigma^2}\right)$

$\mu^*, \sigma^{2*} = \arg\max_{\mu, \sigma^2} \prod_{i=1}^{n} p(x_i; \mu, \sigma^2)$ $\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} x_i$ $\hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{\mu})^2 = E[(x-\mu)^2]$

Naive Bayes Assumption: $P(X_1, \dots, X_k|y)P(y) = \prod_{i=1}^{k} P(X_i|y)P(y)$

Perceptron

Linear: output $f = \langle \underset{\text{weight}}{\vec{w}}, \underset{\text{input}}{\vec{x}} \rangle + b$ Step function activation: $\sigma(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases}$

with activation: $o = \underset{\text{activation func}}{\sigma}(\langle \vec{w}, \vec{x} \rangle + b)$ Sigmoid/logistic activation: $\sigma(z) = \text{sigmoid}(z) = \frac{1}{1+\exp(-z)}$

tanh activation: $\sigma(z) = \tanh(z) = \frac{1-\exp(-2z)}{1+\exp(-2z)} = 2\,\text{sigmoid}(2z) - 1$

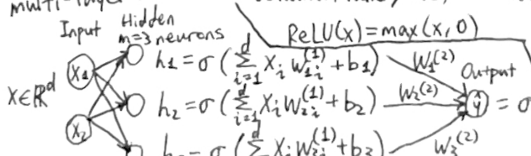
$ReLU(x) = \max(x, 0)$

regression: given $\{(\vec{x}_i, y_i)\}_{i=1}^{n}$, $\vec{x} \in \mathbb{R}^d$, $y = \{-1, +1\}$

MLE $\max_{\vec{w}} \sum_i \log \frac{1}{1+\exp(-y_i \vec{w}^T \vec{x}_i)}$

MAP $\min_{\vec{w}} \sum_i -\log \frac{1}{1+\exp(-y_i \vec{w}^T \vec{x}_i)} + \frac{\lambda}{2}\|\vec{w}\|_2^2$

multi-layer:



$h_1 = \sigma\left(\sum_{i=1}^{d} x_i W_{1i}^{(1)} + b_1\right)$

$h_2 = \sigma\left(\sum_{i=1}^{d} x_i W_{2i}^{(1)} + b_2\right)$

$h_3 = \sigma\left(\sum_{i=1}^{d} x_i W_{3i}^{(1)} + b_3\right)$

$y = \sigma\left(\sum_{i=1}^{M} h_i W_i^{(2)} + b'\right)$

softmax: $p(y|\vec{x}) = \text{softmax}(f) = \frac{\exp f_y(x)}{\sum_i \exp f_i(x)}$

Distribution

$E(aX+b) = aE[X] + b$ $E[g(x)] = \sum_k g(k)p_X(k)$ / $\int_{-\infty}^{\infty} g(x)f(x)dx$ $Var(X) = E[(X - E[X])^2] = E[X^2] - (E[X])^2$

$Var(aX+b) = a^2 Var(X)$ $X \sim Geom(p)$: $p_X(k) = p(1-p)^{k-1}$, $E = \frac{1}{p}$, $Var = \frac{1-p}{p^2}$

$X \sim Ber(p)$: $p_X(0) = 1-p$, $p_X(1) = p$, $E = p$, $Var = p(1-p)$ $X \sim Unif[a,b]$: $f_X(t) = \frac{1}{b-a}$, $E = \frac{a+b}{2}$, $Var = \frac{(b-a)^2}{12}$

$X \sim Bin(n,p)$: $p_X(k) = \binom{n}{k} p^k (1-p)^{n-k}$, $E = np$, $Var = np(1-p)$

$X \sim Normal(\mu, \sigma^2)$: $f_X(t) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(t-\mu)^2}{2\sigma^2}}$, $E = \mu$, $Var = \sigma^2$, $z = \frac{X - \mu}{\sigma}$

GD

① choose learning rate $\alpha > 0$ ② init model params W_0 ③ for $t = 1, 2, \dots$ update params until convergence $\vec{W}_t = \vec{W}_{t-1} - \alpha \frac{1}{|D|}\sum_{x \in D} \frac{\partial \ell(\vec{x}_i, y_i)}{\partial \vec{W}_{t-1}}$

loss of NN: $\frac{1}{|D|}\sum_i \ell(\vec{x}_i, y_i)$

CS540 SP22 Final Note   Ruixuan Tu

BP (14.20)

CNN and DL (15-18)

2DConv Input $\begin{bmatrix}1&2&3\\4&5&6\\7&8&9\end{bmatrix} * $ Kernel $\begin{bmatrix}0&1\\2&3\end{bmatrix} = $ Output $\begin{bmatrix}19&25\\37&43\end{bmatrix}$ $\vec{Y} = \underbrace{\vec{X} * \vec{W}}_{\text{learnable}} + \underbrace{b}_{\text{scalar bias}}$

$\vec{X}: n_h \times n_w$ $\vec{W}: k_h \times k_w$ $\vec{Y}: (n_h - k_h + 1) \times (n_w - k_w + 1)$

Padding adds rows/cols around input

$\begin{bmatrix}0&0&0&0&0\\0&0&1&2&0\\0&3&4&5&0\\0&6&7&8&0\\0&0&0&0&0\end{bmatrix} * \begin{bmatrix}0&1\\2&3\end{bmatrix} = \begin{bmatrix}0&3&8&4\\9&19&25&10\\21&37&43&16\\6&7&8&0\end{bmatrix}$

Output shape: $(n_h - k_h + p_h + 1) \times (n_w - k_w + p_w + 1)$

Common: $p_h = k_h - 1$, $p_w = k_w - 1$,

top: $\lceil p_h/2 \rceil$, bottom: $\lfloor p_h/2 \rfloor$

stride of 3 for height (s_h) and 2 for width (s_w)

$\begin{bmatrix}0&0&0&0&0\\0&0&1&2&0\\0&3&4&5&0\\0&6&7&8&0\\0&0&0&0&0\end{bmatrix} * \begin{bmatrix}0&1\\2&3\end{bmatrix} = \begin{bmatrix}0&8\\6&8\end{bmatrix}$

Output shape: (Ultimate)

$\lfloor (n_h - k_h + p_h + s_h)/s_h \rfloor \times \lfloor (n_w - k_w + p_w + s_w)/s_w \rfloor$

Input: $(n_h/s_h) \times (n_w/s_w)$

Multiple input channels:

$\underbrace{\vec{Y}}_{m_h \times m_w \text{ output}} = \sum_{i=0}^{c_i} \underbrace{\vec{X}_{i,:,:}}_{c_i \times n_h \times n_w \text{ input}} * \underbrace{\vec{W}_{i,:,:}}_{c_i \times k_h \times k_w \text{ kernel}}$

Multiple output channels:

$\underbrace{\vec{Y}_{i,:,:}}_{c_o \times m_h \times m_w} = \underbrace{\vec{X}}_{c_i \times n_h \times n_w} * \underbrace{\vec{W}_{i,:,:,:}}_{c_o \times c_i \times k_h \times k_w}$ for $i = 1, \ldots, c_o$

2DMaxPool   No learnable params

$\begin{bmatrix}0&1&2\\3&4&5\\6&7&8\end{bmatrix}$ 2x2 Max Pool $\begin{bmatrix}4&5\\7&8\end{bmatrix}$

Loss function: $\frac{1}{|D|} \sum_i \ell(\vec{X}_i, y_i)$

Per-sample loss: (cross-entropy/softmax)

$\ell(\vec{X}, y) = \sum_{j=1}^{K} -y_j \log p_j$

LeNet < AlexNet < VGG (Lec 16) < ResNet

← residual connections (e.g, every 2 layers for ResNet34) → easier learning with BP, Batch Normalization

Avoid Overfitting: Data Augmentation (one regulation) – transform & add new samples (crop, color, rotation), can also in text by thesaurus

Classic regulation $\min_\theta \frac{1}{n} \sum_{i=1}^{n} \underbrace{\ell(f_\theta(x_i), y_i)}_{\text{standard loss}} + \underbrace{\lambda}_{\text{regulation param}} \underbrace{R(f_\theta)}_{\text{regularizer}}$ Review: Lec 18

Game (19-20)

Goal - Max reward   Properties: N of players, action space (finite/inf), deterministic/random, Sum of payoff (zero-sum: win-lose, General-sum: prison's dilema), Sequential (turns, payoff ated) / Simultaneous (same time)

Rational Player at DSE if exists

Absolute Best → Dominant Strategy Equilibrium

Nash Equilibrium: Best response to each other

$u_i(a_i^*, a_{-i}^*) \geq u_i(b, a_{-i}^*)$ Mixed: $u_1(p, q) = u_1(f_1, q) = u_1(f_2, q)$

Minimax: max of children max's turn, min at min, root and leaf nodes; heuristics; normal form

Search (21-23)

Problem: State space S, Initial state $I \in S$, Goal state $G \in S$, Successor function $Succ(s) \in S$, $Cost(s, s')$

BFS, DFS   Uniform-cost search: use PQ in BFS   Iterative deepening: DFS with depth limit and repeat

Informed A* $0 \leq \underbrace{h(s)}_{\text{heuristic}} \leq \underbrace{h^*(s)}_{\text{real}}$

(PQ) open ← S ⇒ open empty, fail ⇒ remove n which $f(n)$ min from open to close ⇓ ⇐ n goal, exit ⇐ for each n' (succ) of n ⇐

- n' not in open/closed, estimate $h'(n)$, $g(n') = g(n) + c(n, n')$, $f = g(n) + h(n)$; $f(n') = g(n') + h(n')$
- n' in open/closed, $g(n')$ lower, update n' backward ptr to path of $g(n')$
- not lower $g(n')$: nothing

IDA*: with ID

Hill-Climbing (23.12): Local Optima, always pick max neighbor until descent

Simulated Annealing (23.20): $T = 1$, for $k = 0$ to K: $T \leftarrow T * 0.99$, pick random neighbor $t \leftarrow neighbor(s)$,

if $f(s) \leq f(t)$, then $s \leftarrow t$, else with prob $P(f(s), f(t), T)$ do $s \leftarrow t$

Genetic Algorithm (23.34): ① keep a population (a fixed no states) ② selection, cross-over, mutation According to p

$P_i = f(s_i) / \sum_j f(s_j)$ reproduction prob

RL (24-25)

MDP $M = (S, A, P, r, \mu, \gamma)$ State set S. initial s_0. Action set A. Reward function: $r(s_t, a_t)$,

State transition model $P(s_{t+1} | s_t, a_t)$ (Markov assumption). Policy $\pi(s)$. Discount factor $\gamma \in (0, 1]$

$V^\pi(s_0) = \sum_{seq} P(seq) U(seq)$; $U(seq) = \sum_{t \geq 0} \gamma^t r_t$

$V^*(s) = \max_a r(s, a) + \gamma \sum_{s'} P(s' | s, a) V^*(s')$

Geometric Series:

$\forall |r| < 1$, $\sum_{n=1}^{\infty} a r^{n-1} = \frac{a}{1-r}$, $\frac{1}{1-x} = \sum_{n=0}^{\infty} x^n$

Q-learning (25.21)

2-Prob 3-PCA 4-Stat 5-Logic 6-NLP 7-ML 8-Unsupervised L 10-Linear Reg 11-KNN, Naive Bayes classification

12-Perceptron 13-14 NN