11 / 09 / 23

- **Machine Learning**

  → Association

  → Supervised Learning

  - Classification » Labels : {1, 2, 3, ...} (categorical)
  - Regression » Labels : f(x)

    Targets

  → Unsupervised Learning » Grouping / Sorting

  → Reinforcement Learning

---

a) Supervised (Regression)

b) Unsupervised

c) Speaker ⇝ (Classification)

d) Supervised

Reinforcement

18 / 03 / 2023

Machine Learning

Supervised Learning { Regression, classification

Training set $X = \{\hat{x}^t, r^t\}_{t=1}^{N}$ ~ total samples

$\hat{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ ~ 2 features

$r = \begin{cases} 1 & \text{if } \hat{x} \in C \\ 0 & \text{else} \end{cases}$

Let x_1 = price & x_2 = engine power, then, class C annotates training samples to a class by:

$\therefore (p_1 \leq \text{price} \leq p_2) \wedge (e_1 \leq \text{engine power} \leq e_2)$

- input ~ algorithm ~ Logic ~ hypothesis $h(x)$

$$h(x) \triangleq \begin{cases} 1 & \text{if } h \text{ says } x \text{ is +ive} \\ 0 & \text{if } h \text{ says } x \text{ is -ive} \end{cases}$$

- error of h on $\mathcal{H}$:

$$E(h \mid X) = \sum_{t=1}^{N} 1(h(\hat{x}^t) \neq r^t)$$

- $h \in \mathcal{H}$; b/w S & G makes up version space
  - S → most specific
  - G → general most

Margin

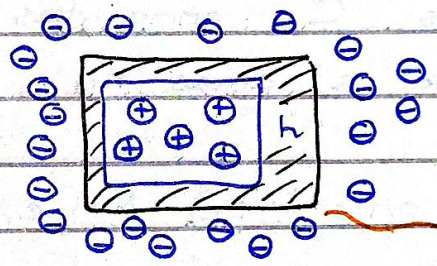
Choose h with largest margin

mean of S and G

Most {Specific General} Hypothesis

Probably Approximately Learning Correct (PAC)

How many sample N should we have such that h has error at most ε; with probability $1-\delta$ → $\delta \leq 1/2$, $\varepsilon > 0$



error with respect to hypothesis h (not class)

- Each side (strip) is at most $\varepsilon/4$
- P_r that we miss a strip $= 1 - \varepsilon/4$
- P_r that N instances miss a strip $= (1-\varepsilon/4)^N$
- P_r that N instances miss 4 strips $= 4(1-\varepsilon/4)^N$

$$4(1-\varepsilon/4)^N \leq \delta \text{, solve for } N$$

$$N \geq (4/\varepsilon)\log(4/\delta)$$

18 / 09 / 23

Machine Learning

Noise and Model Complexity

→ simple models are usually better to use than a highly complex one

→ irrelevant features to task ; such as, hair style for voice recognition

outliers generate → bias when true outlier un-classed (both)
→ error when noise

- if data lies outside predicted class box, reject it ; do not give wrong information
- if data lies inside two overlapping clusters, reject it as well

Multiclass Labels

$$\mathcal{X} = \{\hat{x}^t, r^t\}_{t=1}^N$$

$$r_i^t = \begin{cases} 1 & \text{if } \hat{x}^t \in C_i \\ 0 & \text{if } \hat{x}^t \in C_j, j \neq i \end{cases}$$

C_1 0 … 0 0 1
C_2 0 … 0 1 0
C_3 0 … 1 0 0
⋮ ⋮
C_K 1 … 0 0 0

$$h_i(x^t) = \begin{cases} 1 & \text{if } \hat{x}^t \in C_i \\ 0 & \text{if } \hat{x}^t \in C_j, j \neq i \end{cases}$$

Regression

$$r^t = f(x^t) + \epsilon$$

Linear Regression ⇝ $f(x^t) = w_1 x^t + w_0$

Model learns w_1 and w_0 for best fit

$$\text{MSE} \triangleq E(f|\mathcal{X}) = \frac{1}{N}\sum_{t=1}^{N}[r^t - g(x^t)]^2$$

MSE → Mean Squared Error

Triple Trade-off

- Complexity of $\mathcal{H}$ is $C(\mathcal{H})$
- Training set size, N
- Generalization Error, E, on new data

- Test Data $\subseteq \mathcal{D} \triangleq$ {Training Data $\cup$ Validation Data}

Practice

- We use mean (average) of S and G to minimize generalization error E.
- Yes, we can. Circle will have parameters {center, r}. Ellipse has {center, major axis, minor axis}.
- Outliers yield high error (~ variance formula); for robust regression, we opt for Huber / Pseudo-Huber loss which integrates a user specified parameter δ, and on its basis, we have L1 loss for outliers and L2 loss for inliers.

Sir ~: $f(x) = \left[1/\sigma\sqrt{2\pi}\right] e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$ → Gaussian Dist.
Assume distribution is Gaussian; MSE serves us well

25/09/23

Machine Learning

Bayesian Decision Theory

- Lack of info about origin of data compels us to believe that the problem is best modeled as statistical processes.
  stochastic

Coin Toss

- observable variable ≜ outcome
- unobservable variable ≜ temperature, humidity, etc.

⇝ z : unobserved x : observable

$x = f(z)$; $f(\cdot)$ is a deterministic function

We dont have $f(\cdot)$ so we use probability $P(X = x)$

Tossing a Coin $\in \{H, T\}$

- RV $X \in \{0, 1\}$
  - Bernoulli $P(X = 1) = p_o^x (1 - p_o)^{(1-x)}$

→ Finding the approximator p_o as $\hat{p}_o$

Estimation

$$\tilde{p}_o = \#\{H\} / \{N\}$$

no. of heads ↵ ; ↳ total tosses

$$= \sum_i x^t / N$$

Next toss is heads if $\hat{p}_o > 1/2$

Bayes' Rule

prior ↘ ; likelihood ↙

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

posterior ↖ ; evidence ↳

- Prior $P(C=0) + P(C=1) = 1$
- Evidence $P(x) = p(x|C=1)p(C=1) + p(x|C=0)p(C=0)$
- Posterior $p(C=0|x) + p(C=1|x) = 1$
- Likelihood $p(x|C=0) + p(x|C=1) \neq 1$

Bayes incorporates → initial belief and evidence to make revised decisions (posterior)

Example 1: $C = 1$; innocent

$C = 0$; guilty

$\hat{x}$ = blood matches

$p(C=1) = 0.4 \quad p(C=0) = 0.6$

$P(x \mid C=0) = 0.95$ $\quad P(x) = \text{Formula} = 0.61$

$P(x \mid C=1) = 0.1$

$P(C=1 \mid x) = (0.4)(0.1)/0.61 = 0.0655$

Example 2: $C=1$ +ive $\quad x$ = PCR +ive

$C=0$ -ive

$P(C=1) = 0.02 \quad P(C=0) = 0.98$

$P(x \mid C=1) = 0.85$

$P(x \mid C=0) = 0.05$

$P(x) = \text{Formula} = 0.066$

$P(C=1 \mid x) = 0.257$

Bayes Rule $K>2$

$$P(C_i \mid x) = \frac{P(x \mid C_i) P(C_i)}{P(x)}$$

$$= \frac{P(x \mid C_i) P(C_i)}{\sum_{k=1}^{K} P(x \mid C_k) P(C_k)}$$

$$\sum_{k=1}^{K} P(C_i) = 1$$

Choose C_i if $P(C_i \mid x) = \max_k P(C_k \mid x)$

26 / 09 / 23

Machine Learning

Losses and Risks

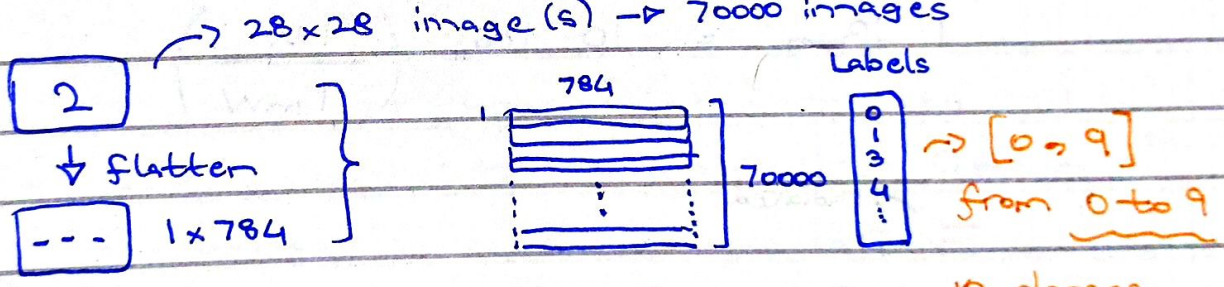
Actions $\triangleq \alpha_i$

Loss of α_i when state is $C_k \triangleq \lambda_{ik}$

Expected Risk $R(\alpha_i|x) = \sum_{k=1}^{K} \lambda_{ik} P(C_k|x)$

choose α_i if $R(\alpha_i|x) = \min_k R(\alpha_k|x)$

MNIST



Training : 60000 samples Test : 10000 samples

- Suppose 5000 labels of class '0' exist, then; ↳ in training

$$p(C=0) = \frac{5000}{60000} = 0.083$$

- Normalize the data [0 - 1] to map to pdf better; and to get the data to a same scale.

Machine Learning

$$p(x|C) = \frac{1}{\sqrt{2\pi}\,\sigma} \exp\left(-\frac{1}{2}\frac{(x-\mu_c)^2}{\sigma_c^2}\right)$$

$$model_c \sim [\{\mu_c, \sigma_c\}, p(C)]$$

$|\Sigma|^2$ ⇝ square of determinant of covariance matrix ⇝ replaces σ

↳ multivariate

Model ~ $model_c$

$$p(x|C) = \frac{1}{\sqrt{2\pi}^d |\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)\Sigma^{-1}(x-\mu)^T\right)$$

↳ d is shape of Σ $(d \times d)$

verify formula from web

$$Cov = \Sigma = \sum_i \left[(x-\mu_c^{(i)})(x-\mu_c^{(i)})^T\right] = 784 \times 784$$

$$\mu_c = 784 \times 1$$

- Association Rules

$$X \longrightarrow Y$$

{implies association and not necessarily causation}

→ Measures

- Support $\triangleq P(X,Y) = (X \text{ and } Y)/\#Number$
- Confidence $\triangleq P(Y|X) = (X \text{ and } Y)/X$
- Lift $\triangleq \frac{P(Y|X)}{P(Y)}$ or $\frac{P(X,Y)}{P(X)P(Y)}$

For cases such as, $X \to Y, Z$ and $X, Y \to Z$, we get;

$$c(X \to Y, Z) = \frac{X \text{ and } Y \text{ and } Z}{X}$$

} Do from next whereever written

$$c(\underbrace{X, Y \to Z}_{\text{all events are independent of each other}}) = \frac{X \text{ and } Y \text{ and } Z}{X \text{ and } Y}$$

→ Not confirmed .. | Search it up

Risks & Losses : Continued

Let $\lambda_{ik} = \begin{cases} 0 & \text{if } i \neq k \\ \lambda & \text{if } i = k+1 \\ 1 & \text{if } i = k \end{cases}$

$R(\alpha_{i+1} | x) = \lambda$

$R(\alpha_i | x) = 1 - P(C_i | x)$

→ $\lambda_{ik} \to \{\lambda_{01}, \lambda_{00}, \lambda_{10}, \text{etc.}\}$

where $i = 1, 2, 3, \ldots K$

$k = 1, 2, 3, \ldots K$

Actions { α_1 α_2 α_3 ... α_K

λ_{12}, λ_{13}, λ_{1K} etc.

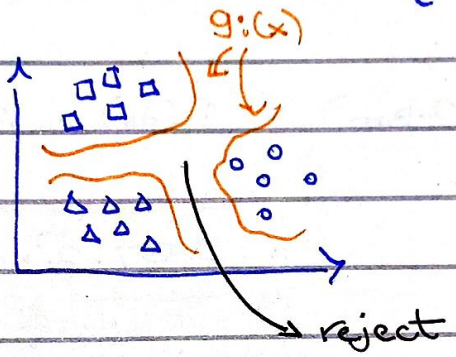
Classes { C_1 C_2 C_3 .... C_K

→ In 0/1 loss; $\lambda_{ik} = 0$ if $i = k$, 1 otherwise

Machine Learning

- Discriminant Functions

$$i = 1, 2, \cdots K \gg g_i(x) = \begin{cases} -R(\alpha_i|x) \\ P(C_i|x) \\ p(x|C_i)P(C_i) \end{cases}$$



$$\mathcal{R}_i = \{x \mid g_i(x) = \max_k g_k(x)\}$$

↳ Decision Regions

- Dichotomizer — $K = 2$

Polychotomizer $\{K \geq 3\}$

$$g(x) = g_1(x) - g_2(x)$$

Choose $\begin{cases} C_1 & \text{if } g(x) > 0 \\ C_2 & \text{otherwise} \end{cases}$

→ Log odds : $\log\left[\frac{(P(C_1|x))}{(P(C_2|x))}\right]$

gib me time sir

buh

Association Measures

Multiple Variables

$$c(X,Y \to Z) = \frac{P(X,Y,Z)}{P(X)\,P(Y)}$$

$$l(X,Y \to Z) = \frac{P(X,Y,Z)}{P(X)\,P(Y)\,P(Z)}$$

According to sir

Exercise 2

C = 0 No Disease C = 1 Disease

x = +ive test

$P(C=1) = 1/10^6$ $P(C=0) = 0.\overline{99}$

$P(x \mid C=1) = 0.99$

$P(x \mid C=0) = 1/10^3$

$$\Rightarrow P(x) = P(C=1)\,P(x \mid C=1) + P(C=0)\,P(x \mid C=0)$$

$$= 1e-3$$ ⟶ Verify

$$\Rightarrow P(C=1 \mid x) = \frac{P(C=1)\,P(x \mid C=1)}{P(x)}$$

$$= .9.89e-4$$ ⟶ Verify

Exercise 3

Given : $\log\left[\frac{P(C_1|x)}{P(C_2|x)}\right] \gg \log\left[\frac{P(x|C_1)\,P(C_1)}{P(x|C_2)\,P(C_2)}\right]$

$$g(x) = \log\left[\frac{P(x|C_1)}{P(x|C_2)}\right] + \log\left[\frac{P(C_1)}{P(C_2)}\right]$$

Exercise 4

$\lambda_{11} = \lambda_{22} = 0$; $\lambda_{12} = 10$, $\lambda_{21} = 5$

Write the optimal decision rule.

	C_1	C_2
α_1	0	10
α_2	5	0

→ Risk $R(\alpha_i|x) = \sum_{k=1}^{K} \lambda_{ik} P(C_k|x)$

$R(\alpha_1|x) = \lambda_{11} P(C_1|x) + \lambda_{12} P(C_2|x)$

$= 0 + 10\, P(C_2|x)$

$R(\alpha_2|x) = \lambda_{21} P(C_1|x) + \lambda_{22} P(C_2|x)$

$= 5\, P(C_1|x) + 0$

→ Known : $P(C_2|x) = 1 - P(C_1|x)$

$R(\alpha_1|x) = 10\,[1 - P(C_1|x)]$

- Choose C_1 if $R(\alpha_1 | x) < R(\alpha_2 | x)$
  → $10 - 10P(C_1|x) < 5P(C_1|x)$
  → $P(C_1|x) > 2/3$

Univariate and Multivariate Analysis

Parameter Estimation
└ Mean and variance are enough parameters to estimate an underlying distribution of data.

- Likelihood
$$l(\theta|\mathcal{X}) = p(\mathcal{X}|\theta) = \prod_t p(x^t|\theta)$$
(→ independent)

- Log Likelihood
$$\mathcal{L}(\theta|\mathcal{X}) = \log l(\theta|\mathcal{X}) = \sum_t \log p(x^t|\theta)$$

- Maximum Likelihood Estimator
$$\theta^* = \underset{\theta}{\text{argmax}}\ \mathcal{L}(\theta|\mathcal{X})$$

10 / 10 / 23

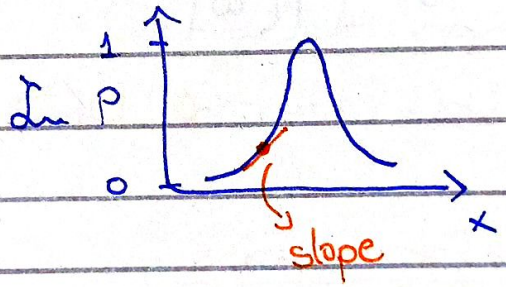
Machine Learning

- Bernoulli ; $P(x) = p^x (1-p)^x$

Log Likelihood $\{\mathcal{L}(p(x)) = \prod_t \log p^{x^t} (1-p)^{(1-x^t)}\}$ Important (Probably wrong)

$$\text{Mean} = E(x) = \sum_x x\, p(x) = 1 \cdot p + 0(1-p)$$
$$= p$$

$$\text{Var} = \text{VAR}(x) = \sum_x (x - E(x))^2 p(x) = p(1-p)$$



$$\frac{\partial \mathcal{L}}{\partial p} = 0 \gg \text{MLE } \hat{p} = \frac{\sum_t x^t}{N}$$

→ For multinomial, $K > 2$ states ;

$$x_i^t = \begin{cases} 1 & \text{if exp } t \text{ chooses state } i \\ 0 & \text{otherwise} \end{cases}$$

$$\text{MLE} \Rightarrow \hat{p}_i = \frac{\sum_t x_i^t}{N}$$ → inclusion of state i

16 / 10 / 23

Machine Learning

Parametric Classification

$$P(C_i|x) = \frac{P(C_i)\,P(x|C_i)}{\sum_k P(C_k)\,P(x|C_k)}$$

$$g_i(x) = \log(P(C_i)) + \log(P(x|C_i))$$

- If distribution is Gaussian;

$$g_i(x) = -\frac{1}{2}\log(2\pi) - \log\sigma_i \cdots$$

$$- \frac{(x-\mu_i)^2}{2\sigma_i^2} + \log(P(C_i))$$

→ Given $\mathcal{X} = \{x^t, r^t\}_{t=1}^N$

ML Estimates ⇝ "estimates"

$$\hat{P}(C_i) = \sum_t r_i^t / N$$

$$m_i = \sum_t x^t r_i^t / \sum_t r_i^t$$

$$s_i^2 = \frac{\sum_t (x^t - m_i)^2 r_i^t}{\sum_t r_i^t}$$

m_i: mean estimate

s_i^2: variance estimate

- Actual mean ~ $E(X)$, where X is an RV
  (expectation)

$g_i(x)$ → excluding constants

↳ $-(x - m_i)^2$

Choose C_i if $[x - m_i] = \min_k [x - m_k]$

→ Suppose $p(x|C_1)$ and $p(x|C_2)$ are gaussians; the best discriminant point is the inter-crossing of the two gaussians.

- When the means are different and the variances are identical, there will only be one discriminant point.
- But if variances are different as well, there will be two discriminant point.
- Choose the discriminant with higher area in a particular region.

Exercise (Class)

$$g_1(x) = -\frac{1}{2}\log(2\pi) - \log s_1 - \frac{(x - m_1)^2}{2s_1^2} + \log P(C_1)$$ ignore

$$g_2(x) = -\frac{1}{2}\log(2\pi) - \log s_2 - \frac{(x - m_2)^2}{2s_2^2} + \log P(C_2)$$ ignore

Let $s_1 = s_2 = s_0$

$$\Rightarrow g_1(x) = g_2(x)$$

$$-\cancel{\log s_0} - \frac{(x-m_1)^2}{2s_0^2} = -\cancel{\log s_0} - \frac{(x-m_2)^2}{2s_0^2}$$

$$-(x-m_1)^2 = -(x-m_2)^2$$

$$-(x^2 - 2xm_1 + m_1^2) = -(x^2 - 2xm_2 + m_2^2)$$

$$-\cancel{x^2} + 2xm_1 - m_1^2 = -\cancel{x^2} + 2xm_2 - m_2^2$$

$$2x(m_1 - m_2) = m_1^2 - m_2^2$$

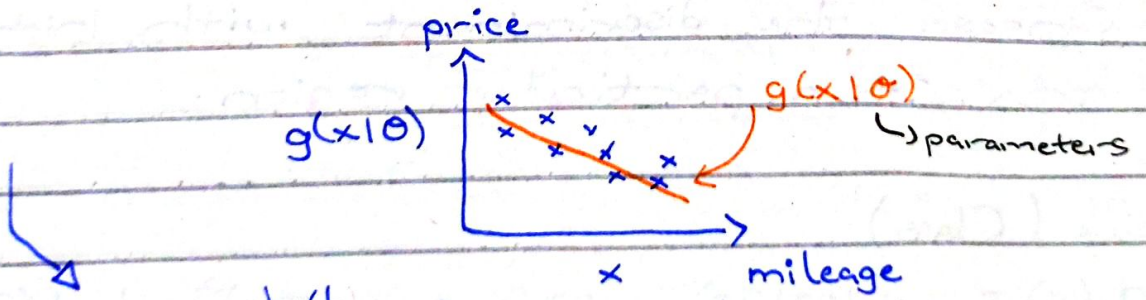
$$x = \frac{m_1^2 - m_2^2}{2(m_1 - m_2)}$$

$$\boxed{x = \frac{1}{2}(m_1 + m_2)}$$

Regression

- $r = f(x) + \varepsilon$

  $\varepsilon \sim N(0, \sigma^2)$



now, we don't know what the class likelihood $P(x|C)$ is;

mean is our estimator

$$p(r|x) \sim N(g(x|\theta), \sigma^2)$$

$$\mathcal{L}(\theta \mid \mathcal{X}) = \log \prod_t p(x^t, r^t)$$

$$\therefore p(x, y) = p(y \mid x)\, p(x)$$

Linear Regression

$$g(x^t \mid w_1, w_0) = w_1 x^t + w_0$$

$$\sum_t r^t \Big\} \text{sum of labels} = N w_0 + w_1 \sum_t x^t \qquad (i)$$

$$x^t = \begin{bmatrix} \vdots \\ \vdots \\ N \end{bmatrix}$$

$$\sum_t r^t x^t = w_0 \sum_t x^t + w_1 \sum_t (x^t)^2 \qquad (ii)$$

$$A = \begin{bmatrix} N & \sum_t x^t \\ \sum_t x^t & \sum_t (x^t)^2 \end{bmatrix} \quad w = \begin{bmatrix} w_0 \\ w_1 \end{bmatrix} \quad y = \begin{bmatrix} \sum_t r^t \\ \sum_t r^t x^t \end{bmatrix}$$

$$w = A^{-1} y$$

→ non-iterative, mathematical solution

17/10/23

Machine Learning

Linear Regression

$$X = [x^t]_{t=1}^{N} \quad W = \begin{bmatrix} w_0 \\ w_1 \end{bmatrix}$$

$$g(x^t | w_1, w_0)_{t=1}^{N} = (w_1 x^t + w_0)_{t=1}^{N}$$

Polynomial Regression

$$g(x^t | k_k, w_{k-1} \dots w_1, w_0) = w_k x^t + \dots + w_1 x^t + w_0$$

~ After solving ;

$$\hat{w} = (D^T D)^{-1} D^T r$$

Multivariate Data

$$X = \begin{bmatrix} X_1^1 & \dots & X_d^1 \\ X_1^2 & & X_d^2 \\ \vdots & & \vdots \\ X_1^N & \dots & X_d^N \end{bmatrix} \begin{matrix} \leftarrow \text{Row} = X^1 \\ X^2 \\ X^N \end{matrix}$$

$$\mu = [m_1, \dots m_d]^T \quad \text{Column wise mean}$$

Covariance : $\sigma_{ij} = Cov(X_i, X_j)$

$X - \mu$ { Replicate μ such that row-wise subtraction occurs }

$$\text{Cov} = E[(x-\mu)^T (x-\mu)]$$

$$\text{Corr} = \rho_{ij} = \frac{\sigma_{ij}}{\sigma_i \sigma_j}$$

- If Missing Values exist ; take column wise mean and replace the value missing
  - can also take mode
  - can also ignore { not suitable if dataset size is small }

$$x \sim N(\mu, \Sigma)$$

$$p(x) = \frac{1}{(2\pi)^{d/2} |\Sigma|^{1/2}} \exp\left[-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)\right]$$

→ shapes : $x\,(d \times 1)$, $\mu\,(d \times 1)$, $\Sigma\,(d \times d)$

Result as above is (1×1)

» $p(x^1)$ = scalar
⋮
$p(x^N)$ = scalar

} N values

Machine Learning

- When features are independent along with samples:
  - auto covariance (diagonal) remain non-zero
  - off diagonals (cross covariance) are zero

(Naive Bayes)

$$p(x) = \prod_{i=1}^{d} p_i(x_i)$$

↳ (Joint probability when independence)

Dimensionality Reduction

Curse of Dimensionality { when $d > N$ }

$$\begin{array}{c} \\ 1 \\ \vdots \\ N \end{array} \begin{array}{c} \begin{array}{cc} 1 & \cdots\cdots \quad d \end{array} \\ \begin{bmatrix} x_1^1 & x_d^1 \\ & \\ x_1^N & x_d^N \end{bmatrix} \end{array}$$

N should be at least 3 times more than d for there to be reasonable decisions

Principal Component Analysis (PCA)

↳ simple yet effective

30/10/23

Machine Learning

$y = mx + c$; $m = r \frac{\sigma_y}{\sigma_x}$ where r = correlation coefficient

$$r = \frac{[\hat{x} - \bar{x}][\hat{y} - \bar{y}]}{\sqrt{(\hat{x} - \bar{x})^2 (\hat{y} - \bar{y})^2}}$$

$\hat{y} = m\hat{x} + c$; $c = \bar{y} - m\bar{x}$

$\hat{y} = m(\hat{x} - \bar{x}) + \bar{y}$

Principal Component Analysis

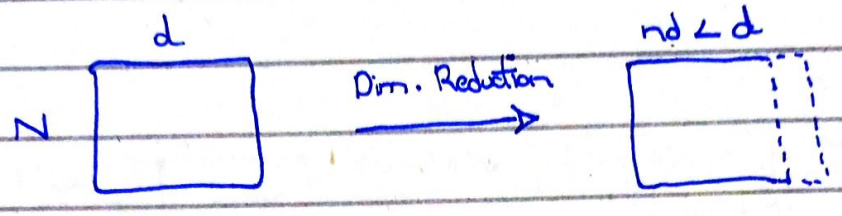
- Find mean of features
- Subtract for mean centering
- PC1 [Principal Component] passes through mean in - towards the largest variance
- PC2 is orthogonal to PC1 ; Project the 3D points onto the 2D plane

PCA gives unique (independent) features { will result in features that do not intermingle

- The largest variance value in the data is represented by largest eigenvalue.

31/10/23

Machine Learning



In PCA, covariance is for identifying those unique eigenvalue to yield highest class deentangling.
Dimensionality Reduction

Linear Discriminant Analysis (LDA)
- makes use of class information

Steps:
- Find the separation of the classes

$$S_B = \sum_{i=1}^{g} N_i (x_i - \bar{x})(\bar{x}_i - \bar{x})^T$$

S_B ← between class scatter matrix

- Find the separation within the classes

$$S_W = \sum_{i=1}^{g} (N_i - 1) S_i$$

$$= \sum_{i=1}^{g} \sum_{j=1}^{N_i} (x_{i,j} - \bar{x}_i)(x_{i,j} - \bar{x}_i)^T$$

└ Maximize b/w class variance
Minimize in-class variance

Neighbourhood Component Analysis

- Given X $[N \times d]$
- Initialize mapping matrix A $[d \times nd]$ (nd ← new dimension)
- where $nd \leq d$

└ on transformed space, find stochastic nearest neighbours

$$p_{ij} = \frac{\exp(-\|Ax_i - Ax_j\|^2)}{\sum_{k \neq i} \exp(-\|Ax_i - Ax_k\|^2)}, \quad p_{ii} = 0$$

∴ Probability that point 'j', the neighbour of point 'i' belongs to the same class as 'i'.

$$p_i = \sum_{j \in C_i} p_{ij}$$

$$f(A) = \sum_i p_i \quad \Big| > \partial f / \partial A = 0$$

↳ & find optimum A

14/11/23

Machine Learning

Gaussian Mixture Method {GMM}
Model

$d \triangleq$ dimensionality

$K \triangleq$ number of Gaussians

→ GMM is a weighted set of Gaussians summed to form a single distribution.

$$p(x) = \sum_{k=1}^{K} \pi_k N(x \mid \mu_k, \Sigma_k),$$

with $\pi_k \triangleq$ the mixing coefficients.

$\sum_{k=1}^{K} \pi_k = 1$ Constraint and $\pi_k > 0 \ \forall k$

→ of the form $p(x|\theta)$

log-likelihood ; $L = \ln p(X \mid \pi, \mu, \Sigma)$

$$= \sum_n \ln \left(\sum_k \pi_k N(x^{(n)}; \mu_k, \Sigma_k) \right)$$

↳ n as X is a vector of all data points $x^{(n)} \in X$

wrt. $\theta = \{\pi_k, \mu_k, \Sigma_k\}$

- Find θ using:

$$\frac{\partial L}{\partial \pi_k} = 0 \ ; \ \frac{\partial L}{\partial \mu_k} = 0 \ ; \ \frac{\partial L}{\partial \Sigma_k} = 0$$

~ in-class

$$p(x) = \sum_k \pi_k N(x \mid \mu_k, \Sigma_k)$$

Let z_k be the clusters ; ~ gaussians

$$p(x) = \sum_k p(x, z = k) = \sum_k \underbrace{p(x \mid z = k)}_{N(x \mid \mu_k, \Sigma_k)} \underbrace{p(z = k)}_{\pi_k}$$

↳ Take Ln ⟶ Solve and we will get expression of L

20/11/23

Machine Learning

Linear Regression

$$E = \frac{1}{2} \sum_n \left[r^{(n)} - w_1 x^n - w_0 \right]^2$$

$$\frac{\partial E}{\partial w_0} = \frac{2}{2} \left[\sum_n \left(r^{(n)} - w_1 x^{(n)} - w_0 \right) \right] \left[-w_0 N \right] = 0$$

$n = t$ both

$$\sum r^t = N w_0 + w_1 \sum_t x^{(t)} \quad \text{— i}$$

$$\frac{\partial E}{\partial w_1} = \frac{2}{2} \left[\sum_t \left(r^{(t)} - w_1 x^{(t)} - w_0 \right) \right] \left[-\sum_t x^t \right] = 0$$

$$\sum_t r^{(t)} x^{(t)} = w_0 \sum_t x^{(t)} + w_1 \sum_t \left[x^{(t)} \right]^2 \quad \text{— ii}$$

Multivariate Case

Model $\quad d$ = features

$$g_b(x \mid w_0, w_1, \ldots, w_d) = w_0 + w_1 x_1^t \ldots\ldots + w_d x^t$$

The rest is the same as $w = (X^T X)^{-1} X^T r$

Moore-Penrose Inverse

GMM $\quad p(x) = \sum_{k=1}^{K} \pi_k N(x \mid \mu_k, \Sigma_k)$

Maximum height of $p(x)$ is one.

$$\therefore \sum_{k=1}^{K} \pi_k = 1$$

$$\hookrightarrow p(x) = \prod_n \left(\sum_k \pi_k \mathcal{N}(x^{(n)} \mid \mu_k, \Sigma_k) \right)$$

$$\text{log likelihood } L = \sum_n \ln \left(\sum_k \pi_k \mathcal{N}(x^{(n)} \mid \mu_k, \Sigma_k) \right)$$

$$\frac{\partial L}{\partial \pi_k} = 0 \; ; \; \frac{\partial L}{\partial \mu_k} = 0 \; ; \; \frac{\partial L}{\partial \Sigma_k} = 0$$

» 3k equations

» Latent Variable

$$\underbrace{p(x)}_{\text{evidence}} = \sum_{k=1}^{K} p(x, z=k) = \sum_{k=1}^{K} \underbrace{p(z=k)}_{\pi_k} \underbrace{p(x \mid z=k)}_{\mathcal{N}(x \mid \mu_k, \Sigma_k)}$$

⇝ Parameters

$$\mu_k = \frac{\sum_m [z=k] \, x^{(m)}}{M} = \frac{\sum_{n=1}^{N} \mathbb{1}[z^{(n)}=k] \, x^{(n)}}{\sum_{n=1}^{N} \mathbb{1}[z^{(n)}=k]}$$

$$\pi_k = \frac{1}{N} \sum_{n=1}^{N} \mathbb{1}[z^{(n)}=k]$$

$$\Sigma_k = \frac{\sum_{n=1}^{N} \mathbb{1}[z^{(n)}=k] \left(x^{(n)} - \mu_k\right)\left(x^{(n)} - \mu_k\right)^T}{\sum_{n=1}^{N} \mathbb{1}[z^{(n)}=k]}$$

Expectation - Maximization (EM)

(EM) Algorithm

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

- **E Step**
  Compute posterior probability of/over z with respect to k (initial model) } responsibility
  (fancy way to say posterior ^.^)
- **M Step** Maximize the probability that it would generate the data it is responsible for.

E Step

$$\gamma_k = p(z=k|x) = \frac{p(x|z=k)\,p(z=k)}{p(x)} = \frac{\pi_k N(x|\mu_k, \Sigma_k)}{\sum_{j=k} \pi_j N(x|\mu_j, \Sigma_j)}$$

M Step

$$\mu_k, \pi_k, \Sigma_k$$

→ Equations from slides

→ use M-step parameters for evaluating Log-likelihood for convergence

28/11/23

Machine Learning

Speaker Recognition

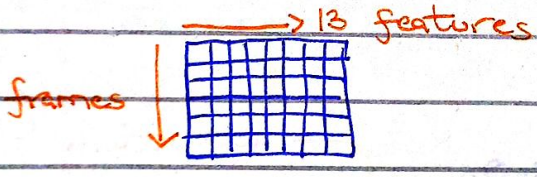
10 files | 10 files | ... | 10 files

S1 | S2 | ... | S10

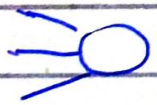
For Each Speaker → 10 files → MFCC .npy → 0.2 train/test split • cont.

• cont. split → train (8 files), test (2 files)

MFCC { Mel frequency cepstral coefficients }



Perceptron



~> oke nvm, just do slides
did the same shit in CV

bias := offset parameter ~> weights can not control offsets

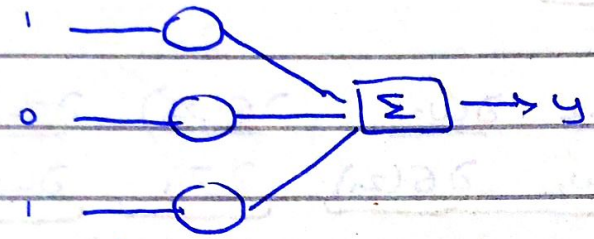
~~trainable~~ parameter as well

4/12/23

Machine Learning

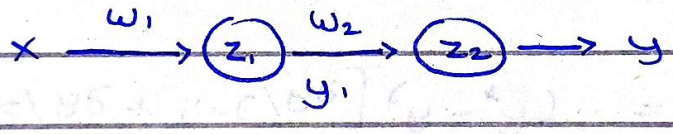
$w_{new} := w_{old} + \alpha(\text{desired} - \text{output}) / \text{input}$

↳ learning rate



one perceptron

multi-layer perceptron [one hidden layer]



$z_1 = x w_1$

$y_1 = f(z_1) = \sigma(z_1)$ $\quad f(\cdot) = \text{sigmoid}$

$z_2 = y_1 w_2$

$y = f(z_2) = \sigma(z_2) = \sigma(\sigma(x w_1) w_2)$

Let ε error (MSE) $= \frac{1}{2}[y^* - y]^2$

↳ desired / target

$\partial\varepsilon/\partial w_1 = ?$ $\quad \partial\varepsilon/\partial w_2 = ?$

$\gg \partial\varepsilon/\partial w_j = -(y^* - y)\frac{\partial y}{\partial w_j}$ $\quad -A$

$$\frac{\partial y}{\partial w_2} = \frac{\partial \sigma(z_2)}{\partial z_2} \frac{\partial z_2}{\partial w_2}$$

$$= \frac{\partial \sigma(z_2)}{\partial z_2} \frac{\partial (y_1 w_2)}{\partial w_2} = \sigma(z_2)[1-\sigma(z_2)]\, y_1$$

$$= \underbrace{y(1-y)}_{\delta_2} y_1 \quad - B$$

$$\frac{\partial y}{\partial w_1} = \underbrace{\frac{\partial \sigma(z_2)}{\partial z_2}}_{\sigma(z_2)(1-\sigma(z_2))} \cdot \underbrace{\frac{\partial z_2}{\partial w_2}}_{y(1-y)} \cdot \underbrace{\frac{\partial \sigma(z_1)}{\partial z_1}}_{w_2} \cdot \underbrace{\frac{\partial z_1}{\partial w_1}}_{\sigma(z_1)(1-\sigma(z_1)) \times} \quad y_1(1-y_1) \times$$

$$= \underbrace{y(1-y)\, w_2\, y_1(1-y_1)\, x}_{\delta_1} \quad - C$$

$$-A: \quad \frac{\partial E}{\partial w_j} = -(y^* - y)\left[\underbrace{\partial y/\partial w_1}_{\delta_1} + \underbrace{\partial y/\partial w_2}_{\delta_2}\right]$$

- $w_2 := w_2 - \alpha \delta_2 y_1$
- $w_1 := w_1 - \alpha \delta_1 x$

12/11/23

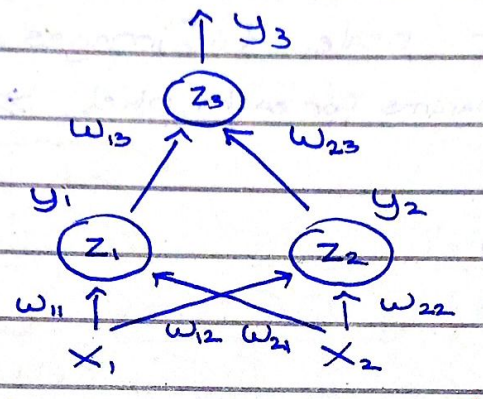
Machine Learning

Training a NN

desired : y^*

activation

sigmoid



Key

$y_1 = \sigma(z_1)$

$z_1 = x_1 w_{11} + x_2 w_{21}$

$y_2 = \sigma(z_2)$

$z_2 = x_1 w_{12} + x_2 w_{22}$

$y_3 = \sigma(z_3)$

$z_3 = y_1 w_{13} + y_2 w_{23}$

$$E = \frac{1}{2}[y^* - y_3]^2$$

$\bar{n}$ neurons will generate $\bar{n}$ δs

$$\frac{\partial E}{\partial w} = ?$$

$$y \equiv y_3$$

Output Layer

$$\frac{\partial E}{\partial w_{i3}} = \frac{\partial E}{\partial w_{13}} + \frac{\partial E}{\partial w_{23}}, \quad i = 1, 2$$

$$\frac{\partial E}{\partial w_{i3}} = -(y^* - y)\left[\frac{\partial y}{\partial w_{13}} + \frac{\partial y}{\partial w_{23}}\right]$$

$$\frac{\partial y_3}{\partial z_3}\frac{\partial z_3}{\partial w_{13}} = y_3(1-y_3)y_1 \qquad \frac{\partial y_3}{\partial z_3}\frac{\partial z_3}{\partial w_{23}} = y_3(1-y_3)y_2$$

δ_3

$$\frac{\partial E}{\partial w_{i3}} = -(y^* - y)\, y_3(1-y_3)\,[y_1 + y_2]$$

Hidden Layer

$$\frac{\partial E}{\partial w_{ij}} = -(y^* - y)\left[\frac{\partial y}{\partial w_{12}} + \frac{\partial y}{\partial w_{11}}\right]$$

$$\frac{\partial y_3}{\partial z_3}\frac{\partial z_3}{\partial y_2}\frac{\partial y_2}{\partial z_2}\frac{\partial z_2}{\partial w_{12}} \qquad \frac{\partial y_3}{\partial z_3}\frac{\partial z_3}{\partial y_1}\frac{\partial y_1}{\partial z_1}\frac{\partial z_1}{\partial w_{11}}$$

$$y_3(1-y_3)\,w_{23}\,y_2(1-y_2)\,x_1 \qquad y_3(1-y_3)\,w_{13}\,y_1(1-y_1)\,x_1$$

$\frac{\partial E}{\partial w_{ij}} = -(y^* - y)$ ~~ OKE Ignore

Sir's Way ^^

$$\frac{\partial E}{\partial w_{i2}} = -(y^* - y)\left[\frac{\partial y}{\partial w_{12}} + \frac{\partial y}{\partial w_{22}}\right]$$

$$\frac{\partial y_3}{\partial z_3}\frac{\partial z_3}{\partial y_2}\frac{\partial y_2}{\partial z_2}\frac{\partial z_2}{\partial w_{12}} \qquad \frac{\partial y_3}{\partial z_3}\frac{\partial z_3}{\partial y_2}\frac{\partial y_2}{\partial z_2}\frac{\partial z_2}{\partial w_{22}}$$

$$y_3(1-y_3)\,w_{23}\,y_2(1-y_2)\,x_1 \qquad y_3(1-y_3)\,w_{23}\,y_2(1-y_2)\,x_2$$

$$\frac{\partial E}{\partial w_{i2}} = \underbrace{-(y^* - y)\,y_3(1-y_3)\,w_{23}\,y_2(1-y_2)}_{\delta_2}\,[x_1 + x_2]$$

Similarly, $\frac{\partial E}{\partial w_{i1}} = \underbrace{-(y^* - y)\,y_3(1-y_3)\,w_{13}\,y_1(1-y_1)}_{\delta_1}\,[x_1 + x_2]$

Updates

$w_{13} = w_{13} - \gamma \delta_3 y_1$
$w_{23} = w_{23} - \gamma \delta_3 y_2$

} The rest is same as well

new old lr error inputs

Weight update equation decides the trajectory of E-curves. {should avoid local minimas}

Linear activation := Identity (·) func.

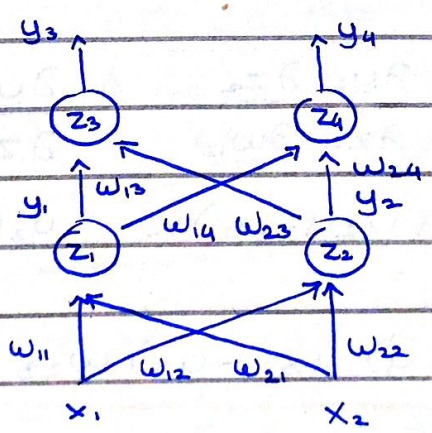
$x_{in} \xrightarrow{\theta} (z) \longrightarrow y$ $\boxed{y = z}$

$z = \theta x_{in}$

More Practice

not sure about the negative sign

$E = \frac{1}{2}\left[(y_3^* - y_3)^2 \cdots + (y_4^* - y_4)^2\right]$



Activation:
Sigmoid

$\delta_4 = -y_4(1-y_4)(y_4^* - y_4)$
$\delta_3 = -y_3(1-y_3)(y_3^* - y_3)$
$\delta_2 = y_2(1-y_2)\delta_3 w_{23}$
$y_2(1-y_2)\delta_4 w_{24}$
$= y_2(1-y_2)[\delta_3 w_{23} \cdots + \delta_4 w_{24}]$
$\delta_1 = y_1(1-y_1)[\delta_3 w_{13} \cdots + \delta_4 w_{14}]$

[δ_1 and δ_2 shouldnt have -ive but sir says otherwise
Check directly]

Weight Updates

- $w_{11} := w_{11} - \gamma \delta_1 x_1$
- $w_{12} := w_{12} - \gamma \delta_2 x_1$
- $w_{22} := w_{22} - \gamma \delta_2 x_2$
- $w_{21} := w_{21} - \gamma \delta_1 x_2$
- $w_{13} := w_{13} - \gamma \delta_3 y_1$
- $w_{14} := w_{14} - \gamma \delta_4 y_1$
- $w_{23} := w_{23} - \gamma \delta_3 y_2$
- $w_{24} := w_{24} - \gamma \delta_4 y_2$

12/12/23

Machine Learning

→ Replace dot product of FNN with * convolution

: $z_1 = x * w_1$

$y_1 = \sigma(z_1)$

$z_2 = y_1 * w_2$

$y_2 = \sigma(z_2)$

$z_2 = y_1 * w$

why did i write it again?
am i high?
or am i just drunk in love? ^^
lmao no, im emotionally dead

→ $z_1 = \sum_{k=-\infty}^{\infty} x(k)\, w_1(k-n)$

1/1/24

Machine Learning

On CNNs & Shapes

Input Layer : 5x5x3 | width x height x chan.

↓

Conv 2D Layer : $\frac{I - F + 2P}{S} + 1$

P: padding
F: filter
I: image

Example
- I = 5x5x3
- F = 3x3
- P = 2x2x2x2
- S = 2

$O = \frac{5 - 3 + 2(2)}{2} + 1 = 4$

↓

Pooling Layer : $\frac{I - F}{S} + 1$

↳ only stride controls output shape

Example
- F = 2x2
- S = 2 → half S = 1 → no effect

Perform ceil() on decimal values

Global Average Pooling

↳ if Input : 7×7 × 3 (average of all) | 3 is channels

Output : 1×1×3 | 1×1×# chan.

Batch Normalization

↳ No change in dimension

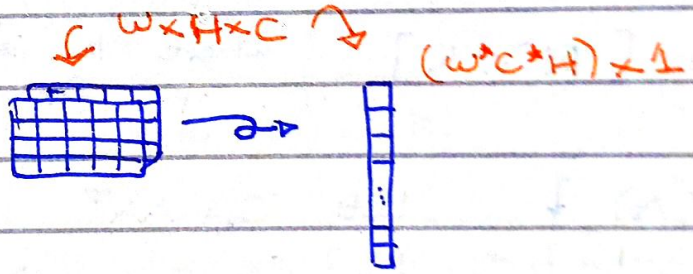
$Data' = \frac{[Data - \mu_D]}{\sigma_D}$ | Normalized: 0 mean, 1 variance / STD

Dropout Layer

↳ random dropping

Flatten Layer

↳



---

epochs ∝ 1/l.r.