1

Let, $G_X(x)$ = CDF for $X$ and $F(x)$ = be the CDF for $X_1$ and $X_2$

$g(x)$ = PDF of $X$

Given, $X = \max(X_1, 2X_2)$

Now,

$$G_X(x) = pr(X \leq x)$$

$$= pr(\max(X_1, 2X_2) \leq x)$$

$$= pr(X_1 \leq x) \cdot pr(2X_2 \leq x)$$ [as $X_1$ and $X_2$ are Independent RVs]

$$= pr(X_1 \leq x)\, pr(X_2 \leq x/2)$$

$$\Rightarrow G_X(x) = F(x) \cdot F(x/2) \quad \text{——— (1)}$$

Now,

$$g(x) = \frac{d}{dx}[G_X(x)]$$

$$\Rightarrow g(x) = \frac{d}{dx}[F(x) \cdot F(x/2)]$$ [from equation (1)]

——— (2)

Now, as $F(x)$ is CDF of Independent Random variables $X_1$ and $X_2$ and they are uniformly distributed over $[0, 1]$, so,

Scanned by CamScanner

$$F(x) = \begin{cases} 0, & x \leq 0 \\ x, & 0 \leq x \leq 1 \\ 1, & x > 1 \end{cases}$$

$$F(x/2) = \begin{cases} 0, & x < 0 \\ x/2, & 0 \leq x \leq 1 \\ 1, & x > 1 \end{cases}$$

②

From equation - ②

$$g(x) = \frac{d}{dx}\left[x \cdot x/2\right]$$

$$= \frac{d}{dx}\left[\frac{x^2}{2}\right]$$

$$\Rightarrow g(x) = x \quad \text{———} \quad ③$$

①

1) $$E[X] = \int_{-\infty}^{\infty} x\, g(x)\, dx$$

$$= \int_0^1 x \cdot g(x)\, dx \quad , \text{as } 0 \leq x \leq 1$$

$$= \int_0^1 x \cdot x\, dx$$

$$= \int_0^1 x^2 dx$$

$$= \frac{1}{3}\left[x^3\right]_0^1$$

$$\Rightarrow \boxed{E[X] = \frac{1}{3}}$$

Answer

Scanned by CamScanner

1

2) $Var(x) = E[x^2] - (E[x])^2$

Now,

$$E[x^2] = \int_{-\infty}^{\infty} x^2 g(x)\, dx$$

$$= \int_{-\infty}^{\infty} x^2 \cdot x\, dx \quad \text{[by using equation 3]}$$

$$= \int_0^1 x^3 dx$$

$$= \left[\frac{x^4}{4}\right]_0^1$$

$$\Rightarrow E[x^2] = \frac{1}{4}$$

$$\Rightarrow Var(x) = \frac{1}{4} - \left(\frac{1}{3}\right)^2$$

$$= \frac{1}{4} - \frac{1}{9}$$

$$= \frac{9-4}{36}$$

$$\Rightarrow \boxed{Var(x) = \frac{5}{36}}$$ Answer

Scanned by CamScanner

① 3)

⑤

$\Rightarrow \text{cov}(X, X_1)$

from, Law of total probability:

$\text{cov}(X_1, \max(X_1, X_2))$

$= \text{cov}(X_1, X_1)\, Pr(X_1 = \max(X_1, X_2)) +$
$\text{cov}(X_1, X_2)\, Pr(X_2 = \max(X_1, X_2))$

$= \text{cov}(X_1, X_1)\, Pr(X_1 = \max(X_1, X_2)) + 0$

[as $X_1$ and $X_2$ are Independent so, $\text{cov}(X_1, X_2) = 0$]

$= \text{cov}(X_1, X_1)\, Pr(X_1 > X_2)$

$= \text{var}(X_1)\, Pr(X_1 > X_2)$ [as $\text{cov}(X, X) = \text{var}(X)$]

$= \text{var}(X_1) \cdot Pr(X_1 > X \mid X_2 = X)$

$= \text{var}(X_1)\, Pr(X_1 > X) \cdot Pr(X_2 = X)$ [as $X_1 \perp X_2$]

$= \text{var}(X_1)\, [1 - Pr(X_1 \leq X)] \cdot Pr(X_2 = X)]$

$= \text{var}(X_1)\, [\int_0^1 (1-x)\, dx]$

— ④

Scanned by CamScanner

Now,

$$var(x_1) = E[x_1^2] - (E[x_1])^2$$

$$= [\int_0^1 x^2 dx] - ([\int_0^1 x dx])^2$$

$$= [\frac{x^3}{3}|_0^1] - ([\frac{x^2}{2}|_0^1])^2$$

$$= \frac{1}{3} - (\frac{1}{2})^2$$

$$= \frac{1}{3} - \frac{1}{4} = \frac{4-3}{12} = \frac{1}{12}$$

and

$$\int_0^1 (1-x) dx = \int_0^1 dx - \int_0^1 x dx$$

$$= 1 - \frac{1}{2}$$

$$= \frac{1}{2}$$

from - 4

$$cov(x, x_1) = var(x_1) . [\int_0^1 (1-x) dx]$$

$$= \frac{1}{12} \cdot \frac{1}{2}$$

$$\Rightarrow \boxed{cov(x, x_1) = \frac{1}{24}}$$ Answer

Scanned by CamScanner

2.1

① Log-likelihood function of X given $\lambda$,

Likelihood function of X given $\lambda$ is

$$L(X|\lambda) = L(x_1, x_2, \ldots, x_n | \lambda)$$

$$= P(x_1|\lambda) \cdot P(x_2|\lambda) \cdots P(x_n|\lambda)$$

[because all the observation $x_i$ are independent]

$$= \frac{\lambda^{x_1}}{x_1!} e^{-\lambda} \cdot \frac{\lambda^{x_2}}{x_2!} e^{-\lambda} \cdots \frac{\lambda^{x_n}}{x_n!} e^{-\lambda}$$

$$L(X|\lambda) = \prod_{i=1}^{n} \frac{\lambda^{x_i}}{x_i!} e^{-\lambda} \quad \text{— ①}$$

$\Rightarrow$ Log-likelihood,

$$\ln(L(X|\lambda)) = \ln\left(\prod_{i=1}^{n} \frac{\lambda^{x_i}}{x_i!} e^{-\lambda}\right)$$

$$= \sum_{i=1}^{n} \ln\left[\frac{\lambda^{x_i}}{x_i!} e^{-\lambda}\right]$$ [by applying log]

$$= \sum_{i=1}^{n} \left[\ln(e^{-\lambda}) + \ln(\lambda^{x_i}) - \ln(x_i!)\right]$$

$$= \sum_{i=1}^{n} \left[-\lambda + x_i \ln(\lambda) - \ln(x_i!)\right]$$

Scanned by CamScanner

7

$$\boxed{\ln(L(X|\lambda)) = -n\lambda + \ln(\lambda)\sum_{i=1}^{n} x_i - \sum_{i=1}^{n} \ln(x_i!)}$$

— 2 (Answer)

2.1

2 compute MLE for $\lambda$:

$$\lambda_{MLE} = \underset{\lambda}{\operatorname{argmax}} \ln(L(X|\lambda)) \quad — 3$$

Now to get argmax, we need to take derivative of log-likelihood and set it to 0.

$$\frac{d}{d\lambda} \ln[L(X|\lambda)] = 0$$

$$\Rightarrow \frac{d}{d\lambda}\left[-n\lambda + \ln(\lambda)\sum_{i=1}^{n} x_i - \sum_{i=1}^{n} \ln(x_i!)\right] = 0$$

$$\Rightarrow -n + \frac{1}{\lambda}\sum_{i=1}^{n} x_i = 0$$

$$\Rightarrow \frac{1}{\lambda}\sum_{i=1}^{n} x_i = n$$

$$\Rightarrow \lambda = \frac{1}{n}\sum_{i=1}^{n} x_i$$

$$\Rightarrow \boxed{\lambda_{MLE} = \frac{1}{n}\sum_{i=1}^{n} x_i}$$ Answer

— 4

Scanned by CamScanner

2.1

3

8

Compute MLE for $\lambda$ for observed $x$

By using equation - 4

$$\lambda_{MLE} = \frac{1}{7}[4+5+3+5+6+9+3]$$

$$= \frac{35}{7}$$

$$\Rightarrow \boxed{\lambda_{MLE} = 5}$$ Answer.

2.2

1) Compute posterior distribution over $\lambda$ :-

Based on Bayes' theorem, the relationship between prior, the posterior and likelihood function is.

$$\cancel{P(\lambda|x) =} \quad P(\lambda|x) = \frac{P(x|\lambda) \cdot P(\lambda)}{\int P(x|\lambda) \cdot P(\lambda)}$$

$\cancel{P(x)}$

Scanned by CamScanner

Here,

Prior, $\Gamma(\lambda|\alpha,\beta) = \frac{\beta^{\alpha}}{\Gamma(\alpha)} \lambda^{\alpha-1} e^{-\beta\lambda}, \lambda > 0$

and

Poisson pmf $P(x|\lambda) = \frac{\lambda^{x}}{x!} e^{-\lambda}$

So, the posterior will be,

$$P(\lambda|x) = \left[\frac{e^{-\lambda}\lambda^{x}}{x!} * \frac{\beta^{\alpha}\lambda^{\alpha-1}e^{-\lambda\beta}}{\Gamma(\alpha)}\right] \Big/ \left[\int_0^{\infty}\left(\frac{e^{-\lambda}\lambda^{x}}{x!} * \frac{\beta^{\alpha}\lambda^{\alpha-1}e^{-\lambda\beta}}{\Gamma(\alpha)}\right)d\lambda\right] \quad \text{— (1)}$$

<u>Calculating the denominator:</u>

$$\int_0^{\infty} \frac{\beta^{\alpha}\lambda^{\alpha+x-1}e^{-\lambda(\beta+1)}}{x!\,\Gamma(\alpha)} d\lambda$$

$$= \frac{\Gamma(\alpha+x)}{x!\,\Gamma(\alpha)} \frac{\beta^{\alpha}}{(1+\beta)^{\alpha+x}} \int_0^{\infty} \frac{(\beta+1)^{\alpha+x}}{\Gamma(\alpha+x)} \lambda^{\alpha+x-1} e^{-\lambda(\beta+1)} d\lambda$$

$$= \frac{\Gamma(\alpha+x)}{x!\,\Gamma(\alpha)} \frac{\beta^{\alpha}}{(1+\beta)^{\alpha+x}} \int_0^{\infty} \text{Gamma}(\lambda|\alpha+x, \beta+1)\, d\lambda$$

$$= \frac{\Gamma(\alpha+x)}{x!\,\Gamma(\alpha)} \frac{\beta^{\alpha}}{(1+\beta)^{\alpha+x}} \quad \text{— (2)}$$

Scanned by CamScanner

Replacing denominator value from equation (2) to equation (1), we get,

$$P(\lambda|x) = \frac{e^{-\lambda}\cdot\cancel{\lambda^{x}}}{\cancel{x!}}\cdot\frac{\cancel{\beta^{\alpha}}\lambda^{\alpha-1}e^{-\lambda\beta}}{\cancel{\Gamma(\alpha)}}\cdot\frac{\cancel{x!}\,\cancel{\Gamma(\alpha)}}{\Gamma(\alpha+x)}\cdot\frac{(1+\beta)^{\alpha+x}}{\cancel{\beta^{\alpha}}}$$

$$\Rightarrow \boxed{P(\lambda|x) = \frac{(\beta+1)^{\alpha+x}}{\Gamma(\alpha+x)}\lambda^{\alpha+x-1}e^{-\lambda(\beta+1)}}$$

$$\boxed{P(\lambda|x) = \text{Gamma Distribution}(\lambda|\alpha+x,\beta+1)}$$

~~Answer~~

More generally,

$$P(\lambda|x_1,x_2,\ldots,x_n) = \text{Gamma}\left(\lambda\middle|\alpha+\sum_{i=1}^{n}x_i,\ \beta+n\right)$$

(2.2) $$\Rightarrow \boxed{P(\lambda|x_1,x_2,\ldots,x_n) = \Gamma\left(\lambda\middle|\alpha+\sum_{i=1}^{n}x_i,\ \beta+n\right)}$$

(Answer)

2) MAP estimate of $\lambda$ :-

Take the log of ~~pr~~ posterior,

~~$\ln(P(\lambda/x$~~

$$\ln(Pr(\lambda|x)) =$$

$$(\alpha+x)\ln(\beta+1) - \ln(\Gamma(\alpha+x)) + (\alpha+x-1)\ln(\lambda) - \lambda(\beta+1)$$

Scanned by CamScanner

Now, to find $\lambda_{MAP}$, we need to take derivative and set it to zero(0).

$$\frac{d}{d\lambda}\left[\ln(pr(\lambda|x))\right] = 0$$

$$\Rightarrow \frac{d}{d\lambda}\left[(\alpha+x)\ln(\beta+1) - \ln(\Gamma(\alpha+x)) + (\alpha+x-1)\ln(\lambda) - x(\beta+1)\right] = 0$$

$$\Rightarrow \frac{1}{\lambda}(\alpha+x-1) - (\beta+1) = 0$$

$$\Rightarrow \frac{\alpha+x-1}{\lambda} = (\beta+1)$$

$$\Rightarrow \boxed{\lambda_{MAP} = \frac{\alpha+x-1}{\beta+1}}$$

more generally,

$$\boxed{\lambda_{MAP} = \frac{\alpha+\sum_{i=1}^{n} x_i - 1}{\beta+n}}$$

Answer

Scanned by CamScanner

2.3

1) Given,

$$\eta = e^{-2\lambda}$$

$$\Rightarrow e^{2\lambda} = \eta^{-1}$$

$$\Rightarrow 2\lambda = -\ln(\eta)$$

$$\Rightarrow \lambda(\eta) = -\frac{1}{2}\ln(\eta) \quad \text{— (1)}$$

Now, if we put $\lambda$ value to the equation of poisson distribution then,

$$P(x \mid \lambda(\eta)) = \frac{\lambda^{x}(\eta)}{x!} e^{-\lambda(\eta)}$$

$$\Rightarrow P(x \mid \lambda(\eta)) = \frac{1}{x!} \cdot \left(-\frac{1}{2}\ln(\eta)\right)^{x} \cdot e^{\frac{1}{2}\ln(\eta)} \quad \text{— (2)}$$

By taking log of equation — (2)

$$\ln P(x \mid \lambda(\eta)) = x \ln\left(-\frac{1}{2}\ln(\eta)\right) + \frac{1}{2}\ln(\eta) - \ln(x!) \quad \text{— (3)}$$

To find the maximum, we will take derivative of equation — (2) with respect to $\eta$ and set it to zero (0).

Scanned by CamScanner

$$\frac{d}{d\hat{\eta}} \ln p(x \mid \lambda(\hat{\eta})) = 0$$

$$\Rightarrow \frac{d}{d\hat{\eta}} \left[ x \ln\left(-\frac{1}{2}\ln(\hat{\eta})\right) + \frac{1}{2}\ln(\hat{\eta}) - \ln(x!) \right] = 0$$

$$\Rightarrow \frac{x}{-\frac{1}{2}\ln(\hat{\eta})}\left(-\frac{1}{2\hat{\eta}}\right) + \frac{1}{2\hat{\eta}} = 0$$

$$\Rightarrow \frac{x}{\hat{\eta}\ln(\hat{\eta})} = -\frac{1}{2\hat{\eta}}$$

$$\Rightarrow -2x = \ln(\hat{\eta})$$

$$\Rightarrow \boxed{\hat{\eta} = e^{-2x}}$$

Answer

This proves that $\hat{\eta}$ is the maximum likelihood estimate of $\eta$.

Scanned by CamScanner

(23)

2) We know that,

$$\text{bias}(\hat{\eta}) = E[\hat{\eta}] - \eta \quad \text{—} \quad (1)$$

[from wikipedia page Bias-of-an-estimator]

Now,

$$E[\hat{\eta}] = \sum_{x \geq 0} \hat{\eta}\, p(x)$$

$$= \sum_{x \geq 0} e^{-2x} \frac{\lambda^x}{x!} e^{-\lambda}$$

$$= e^{-\lambda} \sum_{x \geq 0} \frac{(\lambda e^{-2})^x}{x!}$$

$$= e^{-\lambda} e^{\lambda e^{-2}}$$ [by using the given Taylor expansion series]

$$\Rightarrow \underline{E[\hat{\eta}] = e^{\lambda(e^{-2}-1)}}$$

Now, from equation — (1)

$$\text{bias}(\hat{\eta}) = e^{\lambda(e^{-2}-1)} - \eta$$

$$\Rightarrow \boxed{\text{bias}(\hat{\eta}) = e^{\lambda(1/e^{2}-1)} - e^{-2\lambda}}$$

<u>proved</u>

(answer)

Scanned by CamScanner

3) Lets assume,

$$\hat{\eta} = (-1)^x$$

$$\Rightarrow E[\hat{\eta}] = \sum_{x \geq 0} \hat{\eta}\, P(x)$$

$$= \sum_{x \geq 0} (-1)^x \frac{\lambda^x e^{-\lambda}}{x!}$$

$$= e^{-\lambda} \sum_{x \geq 0} \frac{(-\lambda)^x}{x!}$$

$$= e^{-\lambda} \cdot e^{-\lambda}$$ [by using Taylor expansion series]

$$\Rightarrow E[\hat{\eta}] = e^{-2\lambda}$$

By using bias formula,

$$\text{bias}(\hat{\eta}) = E[\hat{\eta}] - \eta$$

$$= e^{-2\lambda} - e^{-2\lambda}$$

$$\Rightarrow \boxed{\text{bias}(\hat{\eta}) = 0}$$ → This explains why this is a bad estimator.

As we can see the bias resulted as zero (0) so this estimator is <u>bad</u> and can't be used.

Scanned by CamScanner

Explanation :- The fundamental concept of machine learning is to reduce the mean squared Error (MSE). The MSE further can be decomposed into variance + (bias)$^2$. So, if there is small increase in bias that can result huge reduction of variance and as a result the overall MSE get decreased. But here the bias becomes '0' which is not reducing MSE. So, this estimator is a bad choice.

3

3.1 Given linear regression model,

$$y_i = w^T x_i + \epsilon_i$$

and each $\epsilon_i$ can be drawn from Gaussian Distribution:

$$\epsilon \sim N(0, \sigma_i^2)$$

So, each response becomes a draw from following Gaussian,

$$y_i \sim N(w^T x_i, \sigma_i^2)$$

Scanned by CamScanner

Now, probability of each response variable,

$$Pr(y_i | x_i, w) = \mathcal{N}(y_i | w^T x, \sigma_i^2)$$

$$= \frac{1}{\sqrt{2\pi}\,\sigma} \exp\left[-\frac{(y_i - w^T x_i)^2}{2\sigma_i^2}\right] \quad \text{— (1)}$$

Now, to estimate weight vector $w$, we need to take log-likelihood of equation-(1)

$$\Rightarrow \ln L(w) = \ln(P(y_i | x_i, w))$$

$$= \ln \prod_{i=1}^{n} P(y_i | x_i, w)$$ [for all samples under dataset $(x_1, y_1), (x_2, y_2), \ldots (x_n, y_n)$]

$$= \sum_{i=1}^{n} \ln P(y_i | x_i, w)$$

$$= \sum_{i=1}^{n} \ln \frac{1}{\sqrt{2\pi}\,\sigma_i} \exp\left[-\frac{(y_i - w^T x_i)^2}{2\sigma_i^2}\right]$$

$$\Rightarrow \ln L(w) = \sum_{i=1}^{n} \left[-\frac{1}{2} \ln(2\pi\sigma_i^2) - \frac{(y_i - w^T x_i)^2}{2\sigma_i^2}\right] \quad \text{— (2)}$$

Scanned by CamScanner

Now, to find Maximum-Likelihood solution,

$$W_{MLE} = \underset{w}{\operatorname{argmax}} \ln L(w)$$

$$= \underset{w}{\operatorname{argmax}} -\frac{1}{2\sigma_i^2} \sum_{i=1}^{n} (y_i - w^T x_i)^2$$

[from equation (2)]

$$\Rightarrow W_{MLE} = \underset{w}{\operatorname{argmin}} \frac{1}{2\sigma_i^2} \sum_{i=1}^{n} (y_i - w^T x_i)^2$$

if we consider, $s_i = \frac{1}{2\sigma_i^2}$ for all $i = 1, \dots, n$

then,

$$W_{MLE} = \underset{w}{\operatorname{argmin}} \ s_i \sum_{i=1}^{n} (y_i - w^T x_i)^2$$

$$\Rightarrow \boxed{W_{MLE} = \underset{w}{\text{minimize}} \sum_{i=1}^{n} s_i (y_i - w^T x_i)^2} \quad (3)$$

So, it is proved that MLE of $w$ can be found by solving weighted least square problem. Hence proved.

Answer

Scanned by CamScanner

## Derivation of closed form of MLE of $w$

From equation - 3, we get the MLE derivation for $w$

$$w_{MLE} = \underset{w}{\text{minimize}} \; \sum_{i=1}^{n} (y_i - w^T x_i)^2$$

$$\Rightarrow \boxed{w_{MLE} = \min_{w} (Y - Xw)^T (Y - Xw)}$$

This is the closed-form solution of MLE of $w$, where,

~~$w$ = Pseudo Inverse~~

$$\Rightarrow \boxed{w = (XX^T)^{-1} X Y}$$

and $(XX^T)^{-1} X$ is called the ~~Pseudo~~ Pseudo Inverse

and $X$ is invertible

Scanned by CamScanner

3.2

Using the previous concept

~~$P(y|w)$~~,

$$P(Y|X,w) = \prod_{j=1}^{n} \frac{1}{2b} \exp\left(-\frac{|y_j - \sum_i w_i h_i(x_j)|}{b}\right) \quad \text{— (1)}$$

The log likelihood of equation-(1) is,

$$\ln(Y|X,w) = \ln \prod_{j=1}^{n} \frac{1}{2b} \exp\left(-\frac{|y_j - \sum_i w_i h_i(x_j)|}{b}\right)$$

$$= \frac{-1}{2b} \sum_{j=1}^{n} \ln \exp\left(-\frac{|y_j - \sum_i w_i h_i(x_j)|}{b}\right)$$

$$= \frac{-1}{2b} \sum_{j=1}^{n} \frac{|y_j - \sum_i w_i h_i(x_j)|}{b}$$

$$\Rightarrow \ln(Y|X,w) = c \sum_{j=1}^{n} \frac{|y_j - \sum_i w_i h_i(x_j)|}{b} \quad \left[\text{where, } c = \frac{-1}{2b}\right] \quad \text{— (2)}$$

so, to get MLE of $w$

$$w^* = \underset{w}{\operatorname{argmax}} \sum_{j=1}^{n} \frac{|y_j - \sum_i w_i h_i(x_j)|}{b}$$

Scanned by CamScanner

$$\Rightarrow \boxed{w^* = \underset{w}{\operatorname{argmax}} \sum_{j=1}^{n} \left| y_j - \sum_i w_i h_i(x_j) \right|}$$

Answer

The above is the formulation for calculating MLE for $w$.

3.3

As we know, large residuals will be produced if data has noise. Now, if we consider Gaussian distribution, then the squared model (Eq) will have larger residuals. On the other hand, linear models with laplace distribution will have less residuals. So, the linear model with laplace distribution is more robust.

Scanned by CamScanner

4

4.1

1 Residual, $r_i = y_i - (w^T x_i + b)$

considering dot product of matrices:

$$X^T w$$

where, $X = \underbrace{\begin{bmatrix} x_1 & \cdots & x_n \\ x_2 & \cdots & x_n \\ \vdots & & \\ x_d & \cdots & x_n \end{bmatrix}}_{(d \times n)}$, $w = \underbrace{\begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_d \end{bmatrix}}_{(d \times 1)}$, $y = \underbrace{\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}}_{(n \times 1)}$

Using the above matrix notation or vectorized form,

$$\boxed{r = y - (X^T w + b} \quad \text{(Answer)}$$

<u>Complexity</u>:- Time complexity of computing residual, $r$ would be dominated by $(X^T w)$ as addition and substraction would be linear in time.

$\Rightarrow$ $O(X^T \cdot w)$ would have generally taken $O(n\,d)$ time, where, $d = \#$ of features and $n = \#$ of records

But since this is a <u>sparse matrix</u> with

Scanned by CamScanner

$\|X\|_0$ non-zero entries $(X^T w)$ takes time of $O(\|X\|_0)$ ——— ①

After this, we add scalar term 'b' and substract y (n×1 matrix). Both of this takes time $O(n)$ ——— ②

so, total time complexity to compute r is

$$\boxed{O(\|X\|_0 + n)}$$ — adding equation ① & ②

(Answer)

4.1

②

Let, $b_0^{(0)}$ be the original value

$b_0^{(1)}$ be the updated new value.

$$b_0^{(1)} = \frac{1}{n} \sum_{i=1}^{n} (y - w^T x_i) \quad \text{——— ①}$$

Now, predicted value,

$$\hat{y}_i = w^T x_i + b_0^{(0)}$$

$$\Rightarrow w^T x_i = \hat{y}_i - b_0^{(0)} \quad \text{——— ②}$$

Scanned by CamScanner

Replacing equation (2) in (1) we get,

$$b_0^{(1)} = \frac{1}{n} \sum_{i=1}^{n} y - (\hat{y}_i - b_0^{(0)})$$

$$= \frac{1}{n} \sum_{i=1}^{n} (y - \hat{y}_i) + \frac{\cancel{n} \cdot b_0^{(0)}}{\cancel{n}}$$

$$\Rightarrow b_0^{(1)} = \frac{1}{n} \sum_{i=1}^{n} (y - \hat{y}_i) + b_0^{(0)} \quad \text{—— (3)}$$

Equation - (3) can be written in matrix notation as:-

$$b_0^{(1)} = \frac{1}{n} \text{Sum}(y - \hat{y}_i) + b_0^{(0)}$$

$$\Rightarrow \boxed{b_0^{(1)} = \frac{1}{n} \text{Sum}(r) + b_0}$$ (Answer)

where, $r$ = residual
sum() = matlab function for matrices.

Complexity:-

time complexity of computing $r = O(n)$
adding $b_0$ is $O(1)$

$\boxed{\text{Total complexity} = O(n)}$ (ingoring $O(1)$ as this is a constant)

(Answer)

Scanned by CamScanner

3 Let, $r_i = y_i - \hat{y}_i$

where, $\hat{y}_i = (w^T x_i + b)$

considering, $b_0^{(1)}$ is new value of b and $b_0^{(0)}$ old value of b, then

$$\hat{y}^0 = x_i w^0 + b_0^{(0)} \quad \text{— ① [original prediction]}$$

and, $\hat{y}^{(1)} = x_i w^0 + b_0^{(1)}$ [updated $\hat{y}$]

$$= \hat{y}^0 - b_0^{(0)} + b_0^{(1)}$$

$$\Rightarrow r = \hat{y}^1 - \hat{y}^0$$

$$\Rightarrow \boxed{r = b_0^{(1)} - b_0^{(0)}} \text{ (Answer)}$$

$\longrightarrow$ update rule for r

<u>complexity</u>:- Since update complexity of b is already computed, the residual update has to be carried out for N elements. So, the complexity is:

$$\boxed{O(n)} \text{ (Answer)}$$

Scanned by CamScanner

Q.1

(4) Given,

$$c_k \leftarrow 2\sum_{i=1}^{n} x_{ik}\left(y_i - \left(b + \sum_{j\neq k} w_j x_{ij}\right)\right)$$

$$\Rightarrow c_k = 2\sum_{i=1}^{n} x_{ik}\left(y_i - b_0 - (\hat{y}_i - b_0) + w_k x_{ik}\right)$$

[as $\sum_{j\neq k} w_j x_{ij} = \sum_{j=1}^{d} w_j x_{ij} - x_{ik} w_k$

and

$\sum_{j=1}^{d} x_{ij} w_j = \hat{y}_i - w_0$]

$$\Rightarrow c_k = 2\sum_{i=1}^{n} x_{ik}\left(y_i - \hat{y}_i + w_k x_{ik}\right)$$

$$= 2\sum_{i=1}^{n} x_{ik}\left(r + w_k x_{ik}\right)$$

$$\Rightarrow \boxed{c_k = 2 * \text{sum}(x \cdot (r + x^T w_k))}$$

(Answer)

$x^T w$ = dot product of matrices

$x \cdot (r + \cdots)$ also dot product of matrices.

Scanned by CamScanner

<u>complexity</u>: Since $r$ and $X^T w$ are already calculated, the time complexity to compute $c_k$ is:-

$$O(X \cdot r \text{ computation}) = O(Z_k)$$

[considering sparse matrix]

So, from equation of $c_k$, we get,

$$O(2(Z_k + n))$$

$$= \boxed{O(Z_k + n)}$$ <u>Answer</u>

7.1

5

$$r_i = y_i - (w^T x_i + b)$$

$$\Rightarrow r_i = y_i - \hat{y}_i \text{ (given)}$$

Using, $w_k^{old}$ to $w_k^{new}$, we get,

$$r_i^{old} = y_i - \sum_{k=1}^{d} w_k^{old} x_{ik} + b \quad \text{— (1)}$$

$$r_i^{new} = y_i - \sum_{k=1}^{d} w_k^{new} x_{ik} + b \quad \text{— (2)}$$

Scanned by CamScanner

Subtracting equation 1 from 2,

$$r_i^{new} - r_i^{old} = \sum_{k=1}^{d} (w_k^{new} x_{ik} - w_k^{old} x_{ik})$$

$$= \sum_{k=1}^{d} x_{ik} (w_k^{new} - w_k^{old})$$

$$\Rightarrow \boxed{r_{update} = x_{ik} (w_k^{new} - w_k^{old})}$$

<u>complexity</u>:- considering the previous computation, the term $[x_{ik}(w_k^{new} - w_k^{old})]$ dominates.

As, $x$ is sparse matrix, the multiplication complexity is: $\boxed{O(Z_k)}$, where, $Z_k$ is the number of non-zero values in $k$-th row of $X$.

Scanned by CamScanner

6 Total complexity will be dominated by the most expensive (time) computation in the co-ordinate Descent Algorithm.

Below given complexity of each line execution per iteration.

| line # | complexity |
|---|---|
| 1 | $O(\|\|X\|\|_0 + n)$ |
| 2 | $O(n)$ |
| 3 | $O(n)$ |
| 4 | $O(dn + \|\|X\|\|_0)$ |
| 5 | $O(dn + \|\|X\|\|_0)$ |
| 6 | $O(dn + \|\|X\|\|_0)$ |

so, the longest complexity is: $O(dn + \|\|X\|\|_0)$

but $\|\|X\|\|_0 << dn$ as $X$ is a sparse matrix

$\Rightarrow$ $\boxed{\text{Total complexity} = O(dn)}$ [where, $d$ = # of features, $n$ = # of records]

(Answer)

Scanned by CamScanner

4.2

Implemented the co-ordinate descent to solve the Lasso.

please see attache code.

coordinateDescent_Lasso.py

4.3

1

Lasso predictor performs good for non-zero features.

I have noticed the following sparcity patterns

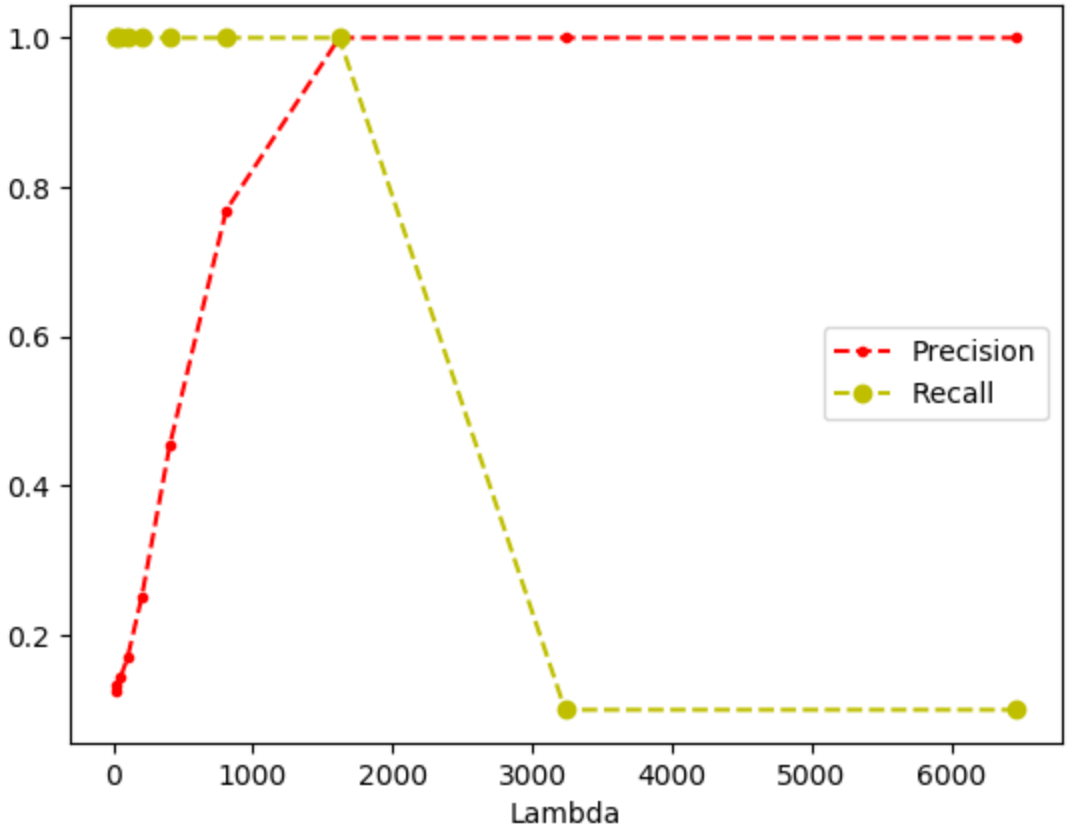
a) As we make progress through the execution, we decrease the $\lambda_{max}$ by a constant ratio. This increases the Recall parameter. As decreasing $\lambda$ more and more, makes feature weights non-zero. After all relevant features are recalled decreasing $\lambda$ further makes all other feature weights non-zero and this increases overfitting of our solution.

Scanned by CamScanner

b) The precission on the other hand, starts near one as initial features are identified as subset of true features. But as we keep on decreasing $\lambda$, more features which are not part of true feature, starts becoming non-zero due to overfitting and precision drops to zero and we reach least squared solution.

The graph below shows the same. This is generated using synthetic data.

Scanned by CamScanner


1.0
0.8
0.6
0.4
0.2
0
1000
2000
3000
4000
5000
6000
Lambda
Precision
Recall

(4.3)

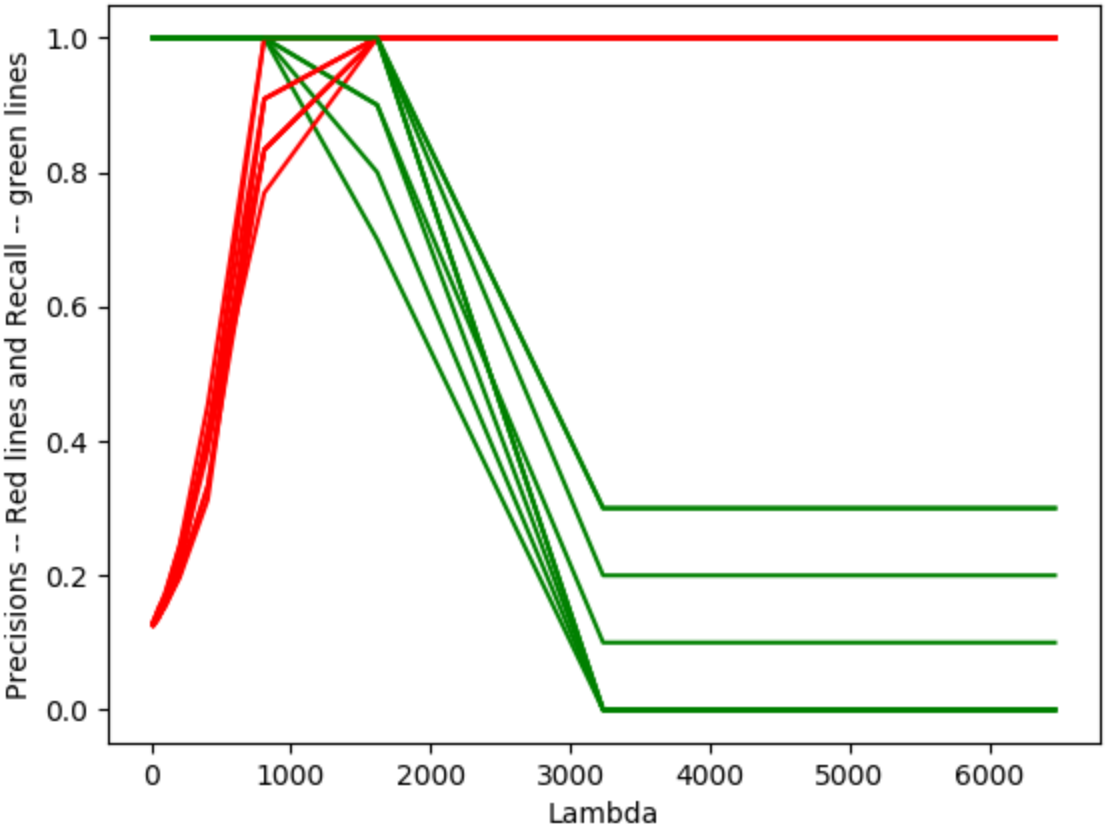
(2)

When we use $\sigma = 10$, we generally increasing the noise in the actual data a lot. If I use the values of $\lambda$ for which the part-(1) worked well we will notice that both precision an recall decreases. This is because more variance in data requires more complex model. But if we use $\lambda$ from part-(1), we not need the complex model and bias error of out solution increases.

To reduce the bias error we need to reduce the $\lambda$ parameter.

See the below graphs.

Scanned by CamScanner

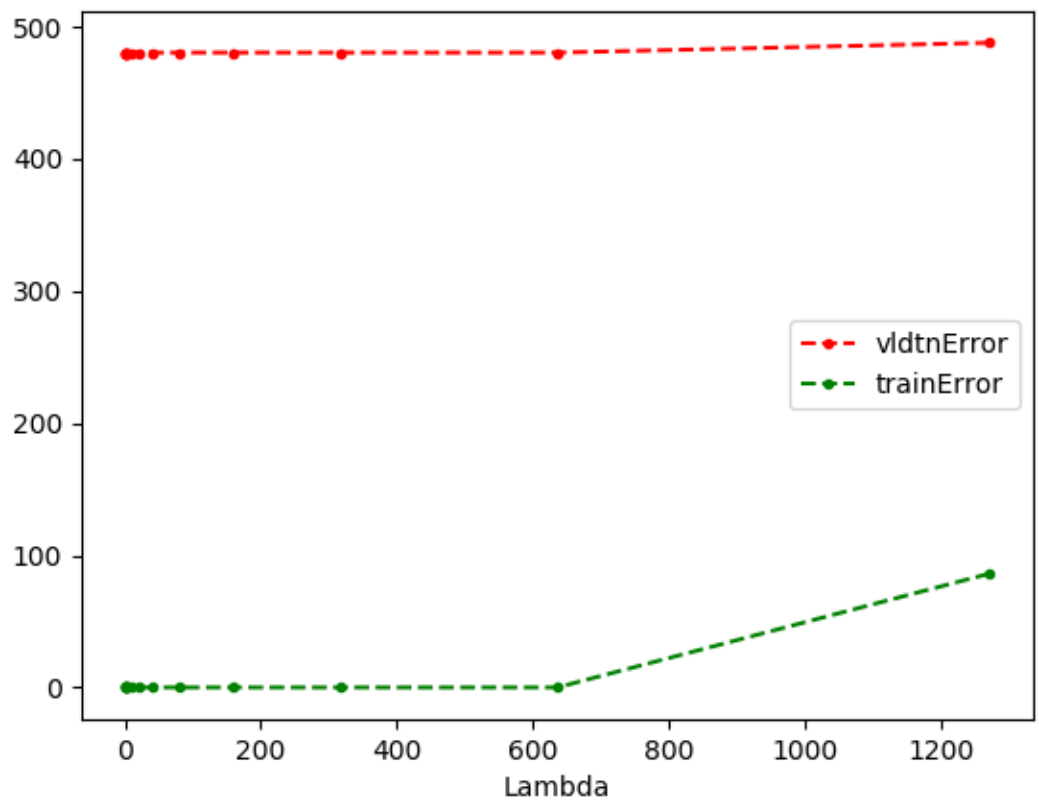

Precisions -- Red lines and Recall -- green lines
1.0
0.8
0.6
0.4
0.2
0.0
0
1000
2000
3000
4000
5000
6000
Lambda

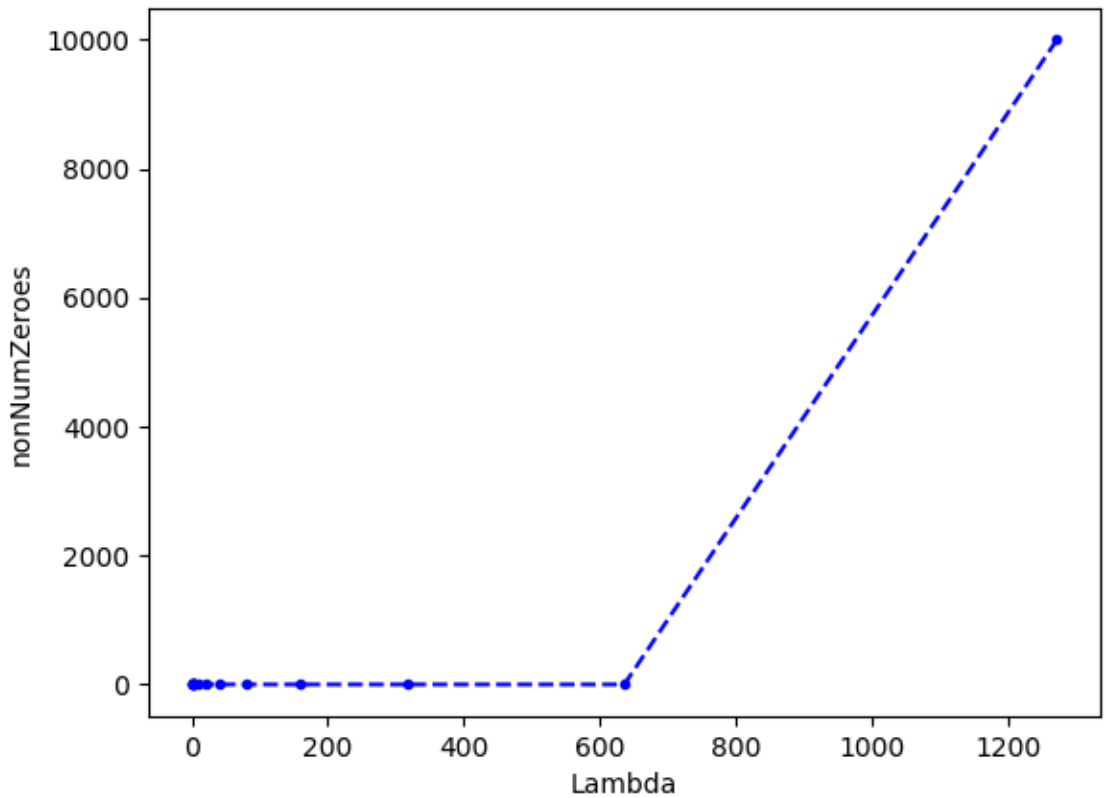
# 4.4

## 1.

The code for predicting wine review is implemented in WineReviewPredictionLasso.py file. And the generated plots are attached below and have explained the output.

The graphs for the solutions is attached, Lambda vs validation error and training error. As expected, the number of non-zero features increase with decreasing lambda. Also, the validation error decreases as we decrease lambda and then starts increasing (at which point we stop)






## 2.

Value of lambda that achieves the best validation performance is : 0.155257525757

**Top 10 features with lowest weights**
[('wine bone dry', 5.9318214190363169e-05, 5.9318214190363169e-05),
('napa', 9.1983735017532668e-05, -9.1983735017532668e-05),
('bit sweet', 0.00020323516825862853, -0.00020323516825862853),
('cherry liqueur', 0.00021658534024757324, -0.00021658534024757324),
('briary brambly', 0.00031003654417006103, -0.00031003654417006103),
('juice', 0.0003129714819800281, 0.0003129714819800281),
('creamy', 0.00046578855070757478, -0.00046578855070757478),
('fruit light', 0.0004791387226965081, -0.0004791387226965081),
('shows real', 0.00051918923866331945, -0.00051918923866331945),
('dose', 0.00052657423380290448, 0.00052657423380290448)]

The top 10 features like 'wine bone dry' / 'napa' / 'bit sweet' / …. / ‘dose' with lowest weights is used to decrease the review of the wine.

**Top 10 features with largest weights**
[('fairly', 5.4905000010124505, 5.4905000010124505),
('flavors just', 4.3500081581700911, 4.3500081581700911),
('zesty acidity', 2.8652331799625266, 2.8652331799625266),
('entirely', 2.2218616914719798, 2.2218616914719798),
('little soft', 2.1383519156238893, 2.1383519156238893),
('cherry coffee', 1.912186651959414, 1.912186651959414),
('flowery', 1.7181552522722956, 1.7181552522722956),
('house', 1.694516547737229, 1.694516547737229),
('thirds', 1.659191992654518, 1.659191992654518),
('plum flavors', 1.6038555297604002, 1.6038555297604002)]

The top 10 features like 'fairly' / 'flavors just' / 'zesty acidity' / …. / 'plum flavors' with largest weights is used to increase the positive review of the wine.

## 3.

I have uploaded the "predTestLabels.csv" file on kaggle competition and got the RMSE : 4.80227

## 4.

I have participated in kaggle completion by submitting the generated "predTestLabels.csv" file and its reflecting in kaggle leaderboard with the name of "gourabbhattacharyya".

| | | | | |
|---|---|---|---|---|
| new | **gourabbhattacharyya** | | 4.80227 | 4 |