DD2421 Machine Learning Exam

Jacob Hedén Malm TMAIM

980405-1499

A-1

a - 6

b - 3

c - 9

d - 2

e - 7

DD 2421 Machine Learning Exam

Jacob Hedén Malm TMAIM

19980405-1499

A-2

a) i i

b)

$$E(X) = -\sum_{i}^{n} P(X_i)\log(P(X_i))$$

$$\Omega : \{win, lose, draw\}$$

green die $\rightarrow Y \sim U(1,6)$

red die $\rightarrow X \sim U(1,6)$

$$P(win) = P(Y < X) = \frac{6}{36} = \frac{1}{6}$$

$$P(lose) = P(Y > X) = \frac{15}{36} = \frac{5}{12}$$

$$P(draw) = P(Y = X) = \frac{15}{36} = \frac{5}{12}$$

Y \ X=	1	2	3	4	5	6
1	draw	win	win	...	...	...
2	lose	draw	win	...	...	...
3	lose	...	draw	...	...	...
4	...	...	...	draw	...	...
5	...	...	...	...	draw	...
6	...	...	...	...	...	draw

Each square eq. Probability

36 total possible outcomes

6 → draw

15 → win

15 → lose

$$E(X) = -(E_{win} + E_{draw} + E_{lose})$$

$$= -\left(\frac{5}{12}\times\log_2\left(\frac{5}{12}\right) + \frac{5}{12}\times\log_2\left(\frac{5}{12}\right) + \frac{1}{6}\times\log_2\left(\frac{1}{6}\right)\right)$$

$$= -\left(2\times\frac{5}{12}\times\log_2\left(\frac{5}{12}\right) + \frac{1}{6}\times\log\left(\frac{1}{6}\right)\right) =$$

$$= \boxed{-1.483}$$

DD2427
Jacob Heldén Malm TMAIM
19980905- 1499

A-2 cont...

c) We combine the probabilities from each classifier for each class, then average them per class. We get a prediction by choosing the class with the largest ~~probability~~ combined probability.

$$P_{tot}(Red|X) = \frac{0.1 + 0.2 + 0.3 + 0.2}{4} = \frac{0.8}{4}$$

$$P_{tot}(Yellow|X) = \frac{0.5 + 0.4 + 0.4 + 0.2}{4} = \frac{1.5}{4}$$

$$P_{tot}(Green|X) = \frac{0.4 + 0.4 + 0.3 + 0.6}{4} = \frac{1.7}{4}$$

Thus, we label X as belonging to class green.

DD2421
Jacob Hedén Malm
19980405-1499

A-3

a) K-nn when $K=1$ has no error on the training set. Thus error $= 0\%$

b) Error on test set $=$ total Error $\times 2$ since train error $= 0$

$$\text{Tot Error} = 0.5 \times \text{train Error} + 0.5 \times \text{test Error}$$
$$= 0.5 \times \text{test Error}$$

2 tot Error $=$ test Error $\longrightarrow$ thus error on testset $= 3\% \times 2$
$= 6\%$

c) $4\% = 0.5 \times 3\% + 0.5 \times X$

$$4\% - 1.5\% = \frac{1}{2} X$$

$$2.5\% = \frac{1}{2} X$$

$$5\% = X$$

train error with subspace method $=$ ~~8~~ 5%

d) We want to pick the method that performed the best on unseen data, or the test set. Therefore we pick the subspace method.

DD 2421

Jacob Hedén Malm

1998 0405 - 1499

A-4)

a)

Variance measures how far off an instance of a classifier $\hat{f}_D(\vec{x})$ is to the average of all possible instances of the same classifiers due to differences in datasets:

$$V = E_D[(\hat{f}(\hat{x}) - E[\hat{f}(\vec{x})])^2]$$

Bias measures how far the average of all classifiers on all different sample sets diverges from the true underlying function. ~~Thus~~ bias is given by

$$E_D(\hat{f}(\vec{x}) - f(\vec{x}))$$

DD 2421 Jacob Hedén Malm
1998 0405 - 1499

A-4)

b)

The error of our function is $\hat{f}(x) - f(x)$

Squared error is $(\hat{f}(x) - f(x))^2$

$$= (\hat{f}(x) - E[\hat{f}(x)] + E[\hat{f}(x)] - f(x))^2$$

$$= (\hat{f}(x) - E[\hat{f}(x)])^2 + (E[\hat{f}(x)] - f(x))^2$$
$$+ 2(\hat{f}(x) - E[\hat{f}(x)])(E[\hat{f}(x)] - f(x))$$

Mean squared error = mean of squared error, or $E[(\hat{f}(x) - f(x))^2]$

$$MSE = E\left[(\hat{f}(x) - E[\hat{f}(x)])^2 + (E[\hat{f}(x)] - f(x))^2\right.$$
$$\left. + 2(\hat{f}(x) - E[\hat{f}(x)])(E[\hat{f}(x)] - f(x))\right]$$

$$= E\left[(\hat{f}(x) - E[\hat{f}(x)])^2\right] + E[\text{bias}^2]$$
$$+ E[2(\hat{f}(x) - E[\hat{f}(x)])(E[\hat{f}(x)] - f(x))]$$

DD2421 Jacob Hedén Malm
19980405 - 1499

$$MSE = \cancel{B}\ \text{Variance} + \text{bias}^2 + 2 \times \left(E\left[\hat{f}(x) - E[\hat{f}(x)]\right]\right)$$

$$\times\ E\left[\hat{f}(x) - f(x)\right]$$

$$= \text{Variance} + \text{bias}^2 + 2 \times (0) \times E\left[\hat{f}(x) - f(x)\right]$$

$$= \text{Variance} + \text{bias}^2$$

DD2421
Jacob Hedén Malm TMAIM
19980405 - 1499

A-5

a) ii)

b)

As we increase S, we are loosening the penalty, and allowing the slope to take on larger values. When S = 0, we cannot have any slope on the line, it must be flat. Thus increasing S is akin to increasing variance and decreasing bias. This means that we first expect the model to fit better on the test set, until we begin to overfit, where it begins to fit worse. Hence,

iv.

c) It allows us to push the weights of useless features to zero, increasing sparsity of the model.

DD2421 TMAIM

Jacob Hedén Malm

19980405 - 1499

B-1

a)

$$Y \sim N(\beta^T X_n, \sigma^2)$$

find β that maximizes $P_r(\bar{Y}|\bar{X}, \beta, \sigma^2)$

$$P(Y_n|X_n, \beta, \sigma^2) = \prod_n^N \left(\frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(Y_n - \beta^T X_n)^2}{2\sigma^2}} \right)$$

log P and P have same location of maximum.
Therefore we can solve for $\log P(Y_n|X_n, \beta, \sigma^2)$

$$\log P(Y_n|X_n, \beta, \sigma^2) = \sum_n^N -\frac{1}{2}\log(2\pi\sigma^2) - \frac{(Y_n - \beta^T X_n)^2}{2\sigma^2}$$

$$= \sum_n^N -\frac{1}{2}\left(\log(2\pi\sigma^2) + \frac{(Y_n - \beta^T X_n)^2}{\sigma^2}\right)$$

we wish to maximize this probability by choosing a suitable β

$$\beta_{ML} = \underset{\beta}{\arg\max} -\frac{1}{2}\sum_n^N (\log 2\pi\sigma^2) + \frac{(Y_n - \beta^T X_n)^2}{\sigma^2}$$

σ^2 is known and constant...

DD2421 TMAIM

Jacob Heldén Malm

19980405 -1499

B-1

a) cont...

$$\beta_{ML} = \underset{\beta}{\operatorname{argmax}} \underbrace{-\frac{1}{2} N \times \log(2\pi\sigma^2)}_{\text{constant w.r.t } \beta} + -\frac{1}{2}\sum_n^N \frac{(y_n - \beta^T x_n)^2}{\sigma^2}$$

$$\beta_{ML} = \underset{\beta}{\operatorname{argmin}} \underbrace{\frac{1}{\sigma^2}}_{\text{constant}} \sum_n^N (y_n - \beta^T x_n)^2$$

$$\beta_{ML} = \underset{\beta}{\operatorname{argmin}} \sum_n^N y_n^2 - 2y_n\beta^T x_n + (\beta^T x_n)^2$$

$$= \underset{\beta}{\operatorname{argmin}} \sum_n^N (\beta^T x_n)^2 - 2y_n\beta^T x_n$$

$$= \underset{\beta}{\operatorname{argmin}} \|\bar{X}\beta\|^2 - 2\bar{y}^T\bar{X}\beta$$

B - 1

b) 1 is the correct interpretation. We wish to minimize $||X\beta||^2$ and maximize $Y^T X\beta$. $||X\beta||$ is the euclidean length and $Y^T X\beta$ is the ~~length~~ projection of Y onto $X\beta$, which is maximized when they point in the same direction,

c) $$\beta_{MAP} = \underset{\beta}{\text{argmax}}\ P(\beta | \text{Data})$$

$$= \underset{\beta}{\text{argmax}}\ \frac{P(\beta) \times P(\text{Data} | \beta)}{P(\text{Data})}$$

$$= \text{argmax}_\beta\ P(\beta) \times P(\text{Data} | \beta)$$

$$= \underset{\beta}{\text{argmax}}\ \prod_p^P \tau^{\beta_p} (1-\tau)^{1-\beta_p} \times \prod_n^N \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(y_n - \beta^T x_n)^2}{2\sigma^2}}$$

$\downarrow$ log( ) $\downarrow$

$$= \text{argmax}\ \sum_p^P \beta_p \log(\tau) + (1-\beta_p)(\log 1-\tau) + \sum_n^N \frac{-(y_n - \beta^T x_n)^2}{2\sigma^2}$$

$$= \text{argmax} - 2\sigma^2 \times \sum_p^P \beta_p \log(\tau) + \log(1-\tau) - \beta_p(\log 1-\tau) - \beta_{ML}$$

constant w.r.t β

$$= \underset{\beta}{\text{argmax}} - 2\sigma^2 \times \sum_p^P \beta_p\ \log(\tau) - \log(1-\tau) + \log(1-\tau) - \beta_{ML}$$

c) cont.

$$\beta_{MAP} = \underset{\beta}{\text{argmax}} -2\sigma^2 \times \sum_p^P \beta_p \times \log\left(\frac{\gamma}{1-\gamma}\right) + \beta_{ML}$$

$$\beta_{MAP} = \underset{\beta}{\text{argmax}}\ \lambda \sum^P - \beta_p + \beta_{ML}$$

$$= \text{argmax}\ \lambda \, |\beta|_1 + 2y^T\beta X - \|X\beta\|_2^2$$

d) When γ is less than 0.5 λ will be negative. Thus we want to minimize $\lambda|\beta|_1$. Thus we benefit from β being sparse. 1 is right answer.

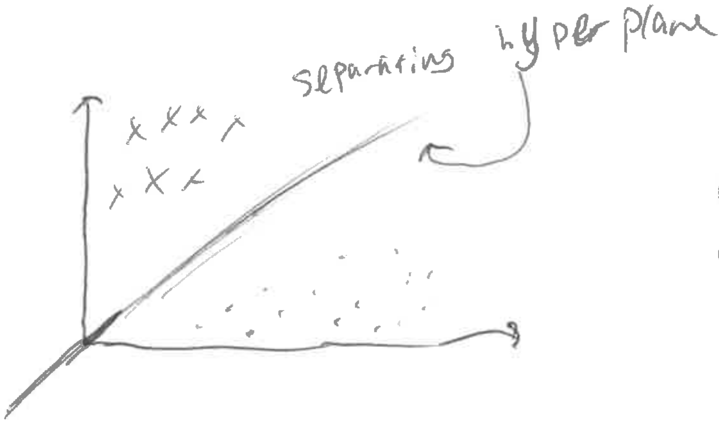
e) Nothing

f) It becomes more and more important to minimize the manhattan distance, and thus choose smaller values for $\beta_p = \{\beta_1, \beta_2 \ldots \beta_p\}$.

DD2421 TMAIM
Jacob Hedén Malm
19980405-1499

C-1

2)
a)



needs to pass through origin due to no bias.

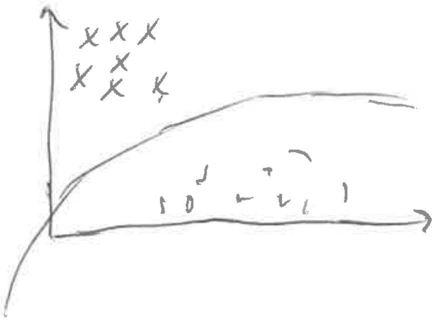
b)

Any separating hyperplane that does not pass through origin works.

c)

We choose the optimal ~~linear~~ separating hyperplane, maximizing distance to nearest points.

d)



With for example a quadratic kernel we can achieve non-linear ~~hyperpla~~ decision boundaries.

DD2421 TMA IM

Jacob Hedén Malm

19980405-1499

C-2

b) The first part of the nn can be represented as

$$(-x_1 + 2x_2 - 2) - 1 \quad + \quad -1(x_1 * 3 - x_2 * 4 - 1)$$

$$= 2x_2 - 2x_1 + 2$$

we can call this x_3

Then 2nd part of network is

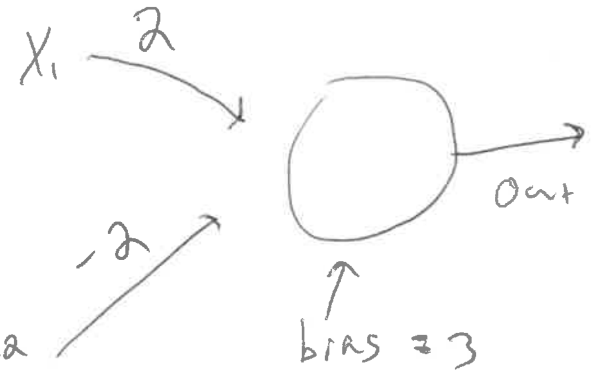
$$Out = (x_3 * 2 - 1) + (x_3 * 4 - 1) - (x_3 * 5 - 1) - 2(-1)$$

$$= 2x_3 - 1 + 4x_3 - 1 - 5x_3 + 1 + 2$$

$$Out = x_3 + 1 \rightarrow 2x_2 - 2x_1 + 3$$

Thus

a)



DD2421 TMAIM

Jacob Hedén Malm

19980405-1499

C-3 a. and b.

~~I~~ pick kernel 2 as this is the radial basis function kernel and it can generate non-linear decision boundaries, which is needed as this isn't a linearly separable data set. The parameter to tweak is ~~thes~~ ρ.

I cannot calculate the exact values of sigma, but it ~~cannot be~~ needs to be small, as this parameter controls the flexibility of the decision boundary. If sigma is 1, it will be able to find a boundary. If sigma is 100000 it will not, as this is a classifier only capable of linear decision bounds.