

1.

a) Model 1:  $y = ax + e$  | Model 2:  $y = ax + bu^c + e$

Fitsse paru ki mukelito, por mu qnto implo con normal equations:

$$X^T X w = X^T y, \text{ note case:}$$

$$X = \begin{bmatrix} x_1 & x_1^2 \\ x_2 & x_2^2 \\ \vdots & \vdots \\ x_n & x_n^2 \end{bmatrix}, w = \begin{bmatrix} a \\ b \end{bmatrix}, y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

Can an usual question:

$$\begin{bmatrix} x_1 & x_2 & \dots & x_n \\ x_1^2 & x_2^2 & \dots & x_n^2 \end{bmatrix} \begin{bmatrix} x_1 & x_1^2 \\ x_2 & x_2^2 \\ \vdots & \vdots \\ x_n & x_n^2 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} x_1 & x_2 & \dots & x_n \\ x_1^2 & x_2^2 & \dots & x_n^2 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

$n \times n$        $n \times 2$        $2 \times 1$        $n \times 1$

$$\begin{bmatrix} x_1^2 + x_2^2 + \dots + x_n^2 & x_1^3 + x_2^3 + \dots + x_n^3 \\ x_1^3 + x_2^3 + \dots + x_n^3 & x_1^4 + x_2^4 + \dots + x_n^4 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} x_1 y_1 + x_2 y_2 + \dots + x_n y_n \\ x_1^2 y_1 + x_2^2 y_2 + \dots + x_n^2 y_n \end{bmatrix}$$

$2 \times n$        $n \times n$        $2 \times 1$        $n \times 1$

$$\begin{bmatrix} (x_1^2 + x_2^2 + \dots + x_n^2)a + (x_1^3 + x_2^3 + \dots + x_n^3)b \\ (x_1^3 + x_2^3 + \dots + x_n^3)a + (x_1^4 + x_2^4 + \dots + x_n^4)b \end{bmatrix} = \begin{bmatrix} x_1 y_1 + x_2 y_2 + \dots + x_n y_n \\ x_1^2 y_1 + x_2^2 y_2 + \dots + x_n^2 y_n \end{bmatrix}$$

Fica o difere:

$$(x_1^2 + x_2^2 + \dots + x_n^2) a + (x_1^3 + x_2^3 + \dots + x_n^3) b = x_1 y_1 + x_2 y_2 + \dots + x_n y_n$$

$$(x_1^4 + x_2^4 + x_3^4 + \dots + x_n^4) a + (x_1^5 + x_2^5 + \dots + x_n^5) b = x_1^2 y_1 + x_2^2 y_2 + \dots + x_n^2 y_n$$

Resumindo se temos  $n$  pontos ou valores em y, temos de obter os valores de  $a$  e  $b$ .

==

b)

ii) Anthony I haven't seen the data, I would say that a polynomial repair with order 2 would be a better option than a linear repair with order 1. The first would probably underfit the data (RMSE would be higher).

c)

ii) Again the model 2 model is more likely, but we could have seen overfitted in our data which could lead to a higher generalization error.

2.

a) O KNN, também chamado "lazy learner", é mais rápido que a SVM, pois é agil e faz o implemente memorizar os dados de treino e os respectivos "labels". A SVM, contudo, leva mais tempo de treino pois tem de aprender a que é que os support vectors que são os que maximizam a margem de separação entre as classes para poder aprender o modelo. Os tempos, contudo, invertem-se na fase de teste, uma vez que, na fase de classificação, a SVM é muito mais rápida, pois terá de aplicar distância o modelo que aprendeu. Por outro lado, o KNN na teste, tem de analisar a distância do test set a todos os samples do training set, perceber os majority votes e atribuir os labels, o que irá demorar bastante tempo se forem muitos.

b) Let's recall the primal form of the SVM:

$$\min \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \sum_{i=1}^m \xi_i \quad \text{See:}$$

(10f - 60f - midterm.pdf)  
(final 2008f. solution(1).pdf)

subject to  $y_i (\mathbf{w}^T \mathbf{x}_i + b) \geq 1 - \xi_i \quad \forall i$

$$\xi_i \geq 0, \forall i$$

This is a constrained quadratic optimization problem, where  $C$  is the regularization parameter, which balances the size of the margin (*i.e.*, smaller  $\|\mathbf{w}\|$ ) vs the violation of the margin (*i.e.*, smaller  $\sum_{i=1}^m \xi_i$ ).

So, for  $C=0$ , this means that we only care of the size of the margin, we would have a solution which would inverse the size of the support vectors, and a larger margin. For  $C=\infty$ , this means that we only care for the violation of the margin, which means that we would only care about having the less support vector as possible, resulting in a thin margin.

### 3. Naive Bayes:

$d$  is odd: this means that  $d(A) = \frac{P(A)}{1-P(A)}$  not whole

and

$$P(A) = \frac{d(A)}{1+d(A)}$$

In Bayes decision we can make

$d$  is an odd (impr.) number

Decide  $\begin{cases} c_1 & \text{if } P(c_1) > P(c_2) \\ c_2 & \text{otherwise} \end{cases}$

Equally probable class

$$\begin{aligned} P(x) &= P(x|c_1)P(c_1) + P(x|c_2)P(c_2) \\ &= 0.5p + (1-p)(0.5) \\ &= 0.5p + 0.5 - 0.5p \\ &= 0.5 \end{aligned}$$

$$P(c_1) = P(c_2) = 0.5$$

We want to compute a margin

$$P_{11} = P(x|c_1) - p, \text{ for } p > \frac{1}{2}$$

$$P_{12} = P(x|c_2) - 1-p$$

$$P(c_j | x) = \frac{P(x|c_j) P(c_j)}{P(x)} , \text{ esehler } G \cap P(c_1|x) > P(c_2|x)$$

latako calculateo:

$$P(c_1|x) = \frac{p}{0.5} = p , \text{ lakerre } p > \frac{1}{2}$$

$$P(c_2|x) = \frac{(1-p)}{0.5} = 1-p , \text{ entso } 1-p < \frac{1}{2}$$

Entso

$$p > 1-p$$

$$2p > 1$$

$$\boxed{p > \frac{1}{2}}$$

Tendo a a pini prob:

clau label:

$$(1) P(c=c_1) \prod_i^d P(x_i|c_1) = 0.5 \times p_{x_1} \times \dots \times p_{x_d}$$

$$(2) P(c=c_2) \prod_i^d P(x_i|c_2) = 0.5 \times (1-p_{x_1}) \times \dots \times (1-p_{x_d})$$

$$(1) > (2) \\ 0.5 \times p^m > 0.5 (1-p)^m \Leftrightarrow p^m > (1-p)^m \quad \boxed{p > \frac{1}{2}}$$

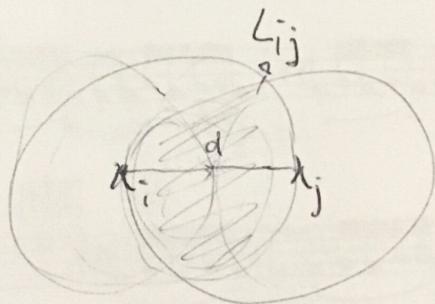
Para  $D > \frac{1}{2}$ , o risco de fechar artifical é menor que o risco de fechar artifical a 2:

Então se  $d_{ij} < d$  é o risco de fechar a 1 impr.

$\sum_{i=1}^n n_i > \frac{d}{2}$  para as escadas de grau 1, dado que em 20

1 ou 0, daí da alternativa em que caiam. Se  $d_{ij} < d$  fone par, havendo de dar um empate e a escada de grau menor da escada de grau maior.

4.



$L_{ij}$  is  
a set  
of all  
points

Ora este algoritmo é extremamente simples car 1-NN, que funciona sempre com 100% de accuracy no teste e teste set.

Poderia querer que se  $L_{ij}$  for sempre contendo pelo menos um ponto todo se conta 1 vizinho, entao, se  $L_{ij}$  não tem nenhum outro ponto. Por isso, o resultado é trivialismente certo.

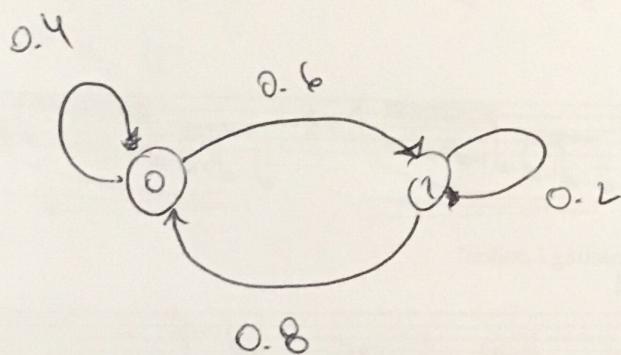
## 5. HMM

a) Two states: 0, 1

Probability Matrix:

$$\pi_{00} = \begin{bmatrix} 0 & 1 \\ 0.3 & 0.7 \end{bmatrix}$$

$$\text{Transition Matrix} = \begin{bmatrix} 0 & 1 \\ 0.4 & 0.6 \\ 0.8 & 0.2 \end{bmatrix}$$



Emission Matrix:

$$= \begin{bmatrix} A & B \\ 0 & 0.9 & 0.1 \\ 1 & 0.5 & 0.5 \end{bmatrix}$$

Forward Algorithm:

$$P(x_1, o_1=A, o_2=B)$$

A type observed e: A B

Python Code:

0.1548
--------

$$b) P_n(x=1 | O_1=A, O_2=B)$$

Linha

$$= P_n(x_1=1) \times P(A|1) \times P(1|1) \times P(B|1) \times P(1|1)$$

$$= 0.7 \times (0.5) \times (0.2) \times (0.5) \times (0.2)$$

$$= 7 \times 10^{-3}$$

====

6. Neste caso, tudo em conta o trigo srt, podemos modelar o PNT Bayes a probabilidade de ter um intérino positivo.

Neste caso, o nosso beneficiário não possui a doença com ilhas com os intérinos positivos. Portanto ele pode ser o "morder o malho" rascando que é positivo. Daí só se bota haver um intérino positivo para a raca ser positiva, isto aplica-se quer a raca tenha 1 ou 100 intérinos. De fato se for possível a raca de trigo srt morder os probabilidade para os positivos só é, porque que se torna utilizável com input, utilizá-lo

Neste caso é um SNT que fazendo a probabilidade de intérino positivo menor que , todo o resto o risco de intérinos.