Andrew Wang

CS 155 FINAL

(1) Multiple Choice:

A: True

B: B

C: k_2, k_1, k_3 (order of images)

D: 2

E: B

F: False

G: A

H: False

I: True

J: False

K: B

L: True

M: True

N: True

(2) Naive Bayes:

1. $P(\text{Grade}=A \mid \text{Happy?}=\text{yes}) = \frac{1+3}{2+4} = \frac{4}{6} = \frac{2}{3}$

$P(\text{Grade}=C \mid \text{Happy?}=\text{yes}) = \frac{1+1}{2+4} = \frac{2}{6} = \frac{1}{3}$

$P(\text{year}=\text{senior} \mid \text{Happy?}=\text{yes}) = \frac{1+3}{2+4} = \frac{2}{3}$

$P(\text{year}=\text{Frosh} \mid \text{Happy}=\text{yes}) = \frac{1+1}{2+4} = \frac{1}{3}$

$P(\text{Grade}=A \mid \text{Happy}=\text{No}) = \frac{1+1}{2+4} = \frac{1}{3}$

$P(\text{Grade}=C \mid \text{Happy}=\text{No}) = \frac{1+3}{2+4} = \frac{2}{3}$

$P(\text{year}=\text{senior} \mid \text{Happy}=\text{no}) = \frac{1+2}{2+4} = \frac{1}{2}$

$P(\text{year}=\text{Frosh} \mid \text{Happy}=\text{no}) = \frac{1+2}{2+4} = \frac{1}{2}$

	P(happy)
Happy = Yes	0.5
Happy = No	0.5

	Year = Freshman	Grade : A
Happy = yes	1/3	2/3
Happy = NO	1/2	1/3

2. P(year = Freshman, Grade = C, Happy? = No) = P(happy = No) P(Grade = C | Happy = No) · P(Year = Freshman | Happy = No)

$= 0.5 \cdot \frac{2}{3} \cdot \frac{1}{2} = \frac{1}{2} \cdot \frac{2}{3} \cdot \frac{1}{2} = \boxed{\frac{1}{6}}$

3. (HAPPY, YEAR, GRADE represent the variables of the model)

```
let sampleY = random()
If sampleY > P(HAPPY = NO) then
    HAPPY = yes
else
    HAPPY = NO

let sampleYear = random()
If sampleYear > P(Freshman | HAPPY)
    YEAR = Senior
else
    YEAR = Freshman

let sampleGrade = random()
If sampleGrade > P(A | HAPPY) then
    GRADE = C
else
    GRADE = A
return (GRADE, YEAR, HAPPY)
```

Scanned by CamScanner

③ Data Transformations

1) We want:

$$w^T x = \tilde{w}^T \tilde{x}$$

we know $\tilde{x} = Ax$ so:

$$w^T x = \tilde{w}^T A x$$

$$w^T x \cdot x^{-1} = \tilde{w}^T A x \cdot x^{-1}$$

$$w^T = \tilde{w}^T A$$

$$w = A^T \tilde{w} \quad \text{and} \quad \tilde{w} = (A^T)^{-1} w$$

2) $$\text{argmin}_w \ \frac{\lambda}{2} \|(A^T)^{-1} w\|^2 + \sum_i (y_i - w^T x_i)^2$$

3) Compared to standard ridge regression, we have a $(A^T)^{-1}$ coefficient to consider. B/c we have scaled our data points x_i our w will also scale so that the $\sum_i (y_i - w^T x_i)^2$ part of the minimization equation is minimized (because y_i stays the same with scaling). To reflect ridge regression of this scaled w, we have the $\frac{\lambda}{2} \|(A^T)^{-1} w\|^2$ part of the minimization equation which differs from $\frac{\lambda}{2}\|w\|^2$ which is used for minimization of regular w before scaling.

(4) Latent Markov Embedding

1) We use a complexity argument. The dual-point model is strictly more complex than the single-point model. Every song that is modeled by the single-point model can be modeled by the dual-point model by setting $U=V=X$ for the particular song. In this case where $U=V=X$, $P(S)$ using the dual point model $=$ $P(S)$ using the single point model.

Now, given that optimal choices for U, V, X are selected to maximize $P(S)$, from the argument above, we know that at the "worst case," the likelihood for the two models will be the same. However, b/c the dual-point model can achieve any likelihood that the single-point model achieves AND more, the likelihood for the dual-point model can be better than that of the single-point model, assuming optimal choices for U and V are picked and that U and V are not equal to X. Thus, $P(S)$ for the dual point model is never less than the data likelihood for the single-point model.

2) This implies that $U=V=X$.

Scanned by CamScanner

(5) Neural Net Backprop Gradient Derivation

1. $$\frac{d}{dw_{11}}(y-f(x))^2 = \underbrace{\frac{d(y-f(x))^2}{df(x)}}_{(1)} \cdot \underbrace{\frac{df(x)}{d\left(\sum_{i=1}^{2} u_i h_i(x)\right)}}_{(2)} \cdot \underbrace{\frac{d\left(\sum_{i=1}^{2} u_i h_i(x)\right)}{dh_i}}_{(3)} \cdot \underbrace{\frac{dh_i}{d\left(\sum_{j=1}^{2} w_{j1} x_j\right)}}_{(4)} \cdot \underbrace{\frac{d\left(\sum_{j=1}^{2} w_{j1} x_j\right)}{w_{11}}}_{(5)}$$

For (1): $\frac{d(y-f(x))^2}{df(x)} = -2(f(x)-y)$

For (2): $= \sigma\left(\sum_{i=1}^{2} u_i h_i(x)\right) \cdot \left(1-\sigma\left(\sum_{i=1}^{2} u_i h_i(x)\right)\right)$

For (3): $= u_1$

For (4): $\sigma\left(\sum_{j=1}^{2} w_{j1} x_j\right)\left(1-\sigma\left(\sum_{j=1}^{2} w_{j1} x_j\right)\right)$

For (5): x_1

* more general: $\frac{d}{dw_{11}} L(y, f(x)) = \frac{dL(y,f(x))}{df(x)} \cdot \frac{df(x)}{dw_{11}}$

So final expression:

$$\boxed{\frac{d}{dw_{11}} L(y, f(x)) = 2(f(x)-y) \cdot \sigma\left(\sum_{i=1}^{2} u_i h_i(x)\right) \cdot \left(1-\sigma\left(\sum_{i=1}^{2} u_i h_i(x)\right)\right) \cdot u_1 \cdot \sigma\left(\sum_{j=1}^{2} w_{j1} x_j\right)\left(1-\sigma\left(\sum_{j=1}^{2} w_{j1} x_j\right)\right) \cdot x_1}$$

2. For (1): $= 2(0.55209 - 0.73) = -0.3958$

For (2): $= \sigma(0.2091) \cdot (1-\sigma(0.2091)) = 0.24728$

For (3): $= 0.5$

For (4): $= \sigma(0.05)(1-\sigma(0.05)) = 0.2498$

For (5): $= 0.1$

So product: $-0.3958 \cdot 0.24728 \cdot 0.5 \cdot 0.2498 \cdot 0.1 = \boxed{-0.00122}$

3. Although expanded out in previous parts:

in general, we can write

$$\frac{\partial L(y, f(x))}{\partial W_{11}} = \frac{\partial L(y, f(x))}{\partial f(x)} \cdot \frac{\partial (f(x))}{\partial W_{11}}$$

We see that $\boxed{\frac{\partial f(x)}{\partial W_{11}}}$ can be decomposed into a product of many partial derivatives from the output layer $f(x)$ ^potentially ranging to the input layer (in this case X, b/c we are taking the derivative w.r.t. W_{11}).

From this, we can easily see that when we have more layers, we will have more terms in the product for calculating $\frac{\partial f(x)}{\partial W_{11}}$. Also, b/c we are using $\sigma(s)$ ... saturating non-linearities, the derivative of the non-linearity will always be less than 1, so w/ many terms multiplied together, the gradient "vanishes"!!