CSE 512 Machine Learning

HW #2

Enbo Yu

113094714

23 SEP (1 day extension)

1. (a) the PDF of Normal distribution is $\frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2}$;

in this question, we got mean x, $\mu=70$, variance $\sigma^2=6^2=36$, $\sigma=6$;

so $f_{Arbok} = \frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2} = \frac{1}{6\cdot\sqrt{2\pi}}e^{-\frac{1}{2}(\frac{x-70}{6})^2} = \frac{1}{6\sqrt{2\pi}}e^{-\frac{(x-70)^2}{72}}$

so $P(Ardok>72) = \int_{72}^{\infty} f_{Ardok}\, dx$

$= \int_{72}^{\infty}\frac{1}{6\sqrt{2\pi}}e^{-\frac{(x-70)^2}{72}}dx =$ (by wolframalph.com) $\approx 0.369441 \approx 0.369$

so the probability is 0.369.

(b) the PDF of Bulbasaur: $f_{Bulbasaur} = \frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2} = \frac{1}{6\sqrt{2\pi}}e^{-\frac{(x-60)^2}{72}}$

and we got $Pr(Bulb>Arbok) = \int Pr(x>A)\,Pr(x=B)\,dx = \int_{-\infty}^{\infty} F_A^{cdf}(x) f_B^{pdf}(x)\,dx$

$=\int_{-\infty}^{\infty}\frac{1}{2}[1+erf(\frac{x-\mu_a}{\sigma\sqrt{2}})]\frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{1}{2}(\frac{x-\mu_b}{\sigma})^2}dx = \int_{-\infty}^{\infty}\frac{1}{2}[1+erf(\frac{x-70}{6\sqrt{2}})]\frac{1}{6\sqrt{2\pi}}e^{-\frac{1}{2}(\frac{x-60}{6})^2}$

= (by wolframalph.com) $\approx 0.119296\ldots \approx 0.119$.

So the probability is 0.119.

(c) i. $\mathcal{L}(accept, fulfill) = -10$;

$\mathcal{L}(accept, \neg fulfill) = 1$;

$\mathcal{L}(reject, fulfill) = 25$;

$\mathcal{L}(reject, \neg fulfill) = 0$.

the Bayes risk R of accept and reject is:

$R_{accept} = \mathcal{L}(accept, fulfill)\cdot Pr(fulfill \mid accept) + \mathcal{L}(accept, \neg fulfill)\cdot Pr(\neg fulfill \mid accept)$

$= (-10)\cdot 0.5 + 1\cdot 0.5 = -4.5$;

$R_{reject} = \mathcal{L}(reject, fulfill)\cdot Pr(fulfill \mid reject) + \mathcal{L}(reject, \neg fulfill)\cdot Pr(\neg fulfill \mid reject)$

$= 25\cdot 0.5 + 0\cdot 0.5 = 12.5$

ii. Tell me to accept because the min risk is -4.5.

(d) No, it doesn't fulfill the critiria.

2.(a). i. $\begin{cases} \text{widge or gadget is defective}: +1; \\ \text{no defective}: 0. \end{cases}$

the Bayes risk R of using a red press is:

$R_{red} = \mathcal{L}(red, warped, defective) \cdot Pr(defective \mid red, warped) +$
$\mathcal{L}(red, \neg warped, defective) \cdot Pr(defective \mid red, \neg warped)$

$= 30\% \cdot 10\% \cdot 1 + 5\% \cdot (1-10\%) \cdot 1$

$= 0.03 + 0.045 = 0.075$

the Bayes risk R of using a blue press is.

$R_{blue} = \mathcal{L}(blue, warped, defective) \cdot Pr(defective \mid blue, warped) +$
$\mathcal{L}(blue, \neg warped, defective) \cdot Pr(defective \mid blue, \neg warped)$

$= 85\% \cdot 10\% \cdot 1 + 0\% \cdot (1-10\%) \cdot 1 = 0.085$

So using red press.

ii. $R_{red} = \mathcal{L}(red, defective) \cdot MAX(Pr(defective \mid red, warped), Pr(defective \mid red, \neg warped))$

$= 1 \cdot 0.3 = 0.3$

$R_{blue} = \mathcal{L}(blue, defective) \cdot MAX(Pr(defective \mid blue, warped), Pr(defective \mid blue, \neg warped))$

$= 1 \cdot 0.85 = 0.85$

So using a red press.

iii. After removing all warped disks:

$R_{red} = \mathcal{L}(red, \neg warped, defective) \cdot Pr(defective \mid red, \neg warped) = 5\% \cdot 1 \cdot 1 = 0.05$

$R_{blue} = \mathcal{L}(blue, \neg warped, defective) \cdot Pr(defective \mid blue, \neg warped) = 0\% \cdot 1 \cdot 1 = 0$

so using blue press.

iv. After removing all warped disks.

$R_{red} = \mathcal{L}(red, \neg warped, defective) \cdot Pr(defective \mid red, \neg warped) = 5\% \cdot 1 \cdot 1 = 0.05$

$R_{blue} = \mathcal{L}(blue, \neg warped, defective) \cdot Pr(defective \mid blue, \neg warped) = 0\% \cdot 1 \cdot 1 = 0$

so using blue press.

(b) i. Bayes reward of blue and red press:

$$R_{Blue} = [\mathcal{L}(blue, sold, defective) \cdot Pr(sold, defective \mid blue) + \mathcal{L}(blue, sold, \neg defective) Pr(sold, \neg defective \mid blue)$$
$$+ \mathcal{L}(blue, \neg sold, defective) \cdot Pr(\neg sold, defective \mid blue) + \mathcal{L}(blue, \neg sold, \neg defective) Pr(\neg sold, \neg defective \mid blue)] \; Pr(inspection)$$
$$+ [\dots \dots] \; Pr(\neg inspection) + \mathcal{L}(inspection) \cdot Pr(inspection)$$
$$= \dots \dots$$
$$= -99.93x + 0.83.$$

$$R_{red} = [\mathcal{L}(red, sold, defective) Pr(sold, defective \mid red) + \mathcal{L}(red, sold, \neg defective) \cdot Pr(sold, \neg defective \mid red)$$
$$+ \mathcal{L}(red, \neg sold, defective) \cdot Pr(\neg sold, defective \mid red) + \mathcal{L}(red, \neg sold, \neg defective) \cdot Pr(\neg sold, \neg defective \mid red)] \; Pr(inspection)$$
$$+ [\dots \dots] \cdot Pr(\neg inspection) + \mathcal{L}(inspection) \cdot Pr(inspection)$$
$$= -100.04x + 0.85$$

ii. the arg max reward for x is 0, which means that do no inspection for both presses. I would recommend red press because

$$Reward_{red} > Reward_{blue}.$$

(C) i. Bayes reward of blue press:

$$R_{blue} = [\mathcal{L}(blue, sold, defective) \cdot Pr(sold, defective \mid blue) + \mathcal{L}(blue, sold, \neg defective) \cdot Pr(\neg defective, sold \mid blue)$$
$$+ \mathcal{L}(blue, \neg sold, defective) \cdot Pr(\neg sold, defective \mid blue) + \mathcal{L}(blue, \neg sold, \neg defective) \cdot Pr(\neg defective, \neg sold \mid blue)] \; Pr(inspection)$$
$$+ [\dots \dots] \cdot Pr(\neg inspection) + \mathcal{L}(inspection) Pr(inspection) = \dots \dots = 742.5x - 392.5$$

Bayes reward of red press:

$$R_{red} = [\mathcal{L}(red, sold, defective) \cdot Pr(sold, defective \mid red) + \mathcal{L}(red, sold, \neg defective) \cdot Pr(\neg defective, sold \mid red)$$
$$+ \mathcal{L}(red, \neg sold, defective) \cdot Pr(\neg sold, defective \mid red) + \mathcal{L}(red, \neg sold, \neg defective) \cdot Pr(\neg defective, \neg sold \mid red)] \; Pr(inspection)$$
$$+ [\dots \dots] \cdot Pr(\neg inspection) + \mathcal{L}(inspection) \cdot Pr(inspection) = \dots \dots = 165x - 287.5$$

ii. do all disks inspection using blue press.

3. K-NN

```python
def get_dist(Xtrain,zquery):
    distances = -2*Xtrain@zquery + np.sum(zquery**2) + np.sum(Xtrain**2, axis = 1)
    distances = distances**.5
    return distances

print(get_dist(Xtrain,Xtrain[0,:])[0])
print(get_dist(Xtrain,Xtest[0,:])[10])
print(get_dist(Xtrain,Xtest[10,:])[50])
```

```
0.0
2463.6278127996525
2379.441951382719
```

```python
import scipy.stats as ss

m = 100
K = 3

Xtrain_small, ytrain_small = get_small_dataset(Xtrain,ytrain,m)

def pred(zquery,Xtrain,ytrain, K):
    distances = get_dist(Xtrain, zquery)
    indices = np.argsort(distances)
    distances = np.sort(distances)
    topk = ytrain_small[indices[0:K]]
    return ss.mode(topk)[0]

ytest_pred = ytest + 0
for k in range(Xtest.shape[0]):
    z = Xtest[k,:]
    ytest_pred[k] = pred(z,Xtrain_small, ytrain_small, K)

print(ytest_pred[:20])
print(ytest[:20])
```

```
[7 2 1 0 4 1 4 4 6 9 0 0 9 0 1 9 7 7 3 4]
[7 2 1 0 4 1 4 9 5 9 0 6 9 0 1 5 9 7 3 4]
```

```python
import time
for m in [100,1000, 2500]:
    Xtrain_small, ytrain_small = get_small_dataset(Xtrain,ytrain,m)
    for K in [1,3,5]:

        start = time.time()
        ytest_pred = ytest + 0
        for k in range(Xtest.shape[0]):
            z = Xtest[k,:]
            ytest_pred[k] = pred(z,Xtrain_small, ytrain_small, K)

        print(m,K,get_accuracy(ytest,ytest_pred), time.time()-start)
```

```
100 1 0.6794 9.281357049942017
100 3 0.6476 9.205874681472778
100 5 0.6232 9.351280927658081
1000 1 0.869 54.075950622558594
1000 3 0.8622 48.400837898254395
1000 5 0.8582 47.38647818565369
2500 1 0.9136 126.55778646469116
2500 3 0.9146 126.62799572944641
2500 5 0.9101 125.87638521194458
```

Comment: It is not feasible for the full 60000 training dataset. And it is not advisable too. Although more dataset can improve the accuracy, the cost will be increased. And it doesn't mean that larger k is better, which might not be desirable. We should chang value of k from low to high and keep checking all value of accuracy until we find an appropriate.

4. (a)

```
[9] # Calculate the acc of 1 word.
positive = 0
for i in range(len(corpus) - 1):
    if pred_2gram(corpus[i])[0] == corpus[i+1]:
        positive += 1
print("The accuracy:", positive / (len(corpus)-1))
```

```
The accuracy: 0.2453493423910897
```

(b)

```
[13] # Calculate the acc of 2 word.
positive = 0
for i in range(len(corpus) - 2):
    if pred_3gram(corpus[i], corpus[i + 1])[0] == corpus[i+2]:
        positive += 1
print("The accuracy:", positive / (len(corpus)-2))
```

```
The accuracy: 0.5047397108776365
```

```
[7] # Using the likelihoods computed from the bigram classiffer, and starting with a seed word "alice",
# generate the next 25 words by always picking the most likely next word.
word = "alice"
article = word
for i in range(25):
    word, _ = pred_2gram(word)
    article = article + " " + word
print(article)
```

//a word starting with "alice"

```
alice was a little thing i can remember ever saw in a little thing i can remember ever saw in a little thing i can remember
```

```
[8] # Using random choices method
word = "alice"
article = word
for i in range(25):
    likelihood = get_likelihood_2gram(word)
    word = random.choices(dictionary, weights = likelihood)[0]
    article = article + " " + word
print(article)
```

//one word random choice

```
alice went on muttering to be it might belong to the dormouse into a caucus is but she picked her as she said turning to rest
```

```
first_word = "alice"
second_word = "was"
article = first_word + " " + second_word
for i in range(25):
    new, _ = pred_3gram(first_word, second_word)
    article = article + " " + new
    first_word = second_word
    second_word = new
print(article)
```

//2-past-words Navie Bayes starting with "alice was"

```
alice was not easy to take this young lady tells us a story afraid i am i ah that the queen who was peeping anxiously into its
```

```
[12] # Using random choices method
first_word = "alice"
second_word = "was"
article = first_word + " " + second_word
for i in range(25):
    likelihood = get_likelihood_3gram(first_word, second_word)
    new = random.choices(dictionary, weights = likelihood)[0]
    article = article + " " + new
    first_word = second_word
    second_word = new
print(article)
```

//two words random choice

```
alice was not pale beloved snail but come to the duchess cook had disappeared mind said the caterpillar seemed to be trampled under its feet move she
```

In my view, the Naive Bayes is better than random choice.