Machine Learning Jiecao Chen ID:4716311
HOMEWORK #1

Question 1

a.

$$
\begin{aligned}
E[\mathfrak{l}(f(x),y)] &= \int\int (f(x)-y)^2 p(x,y)dxdy \\
&= \int\int \{(f(x)-E[y|x])^2 + (y-E[y|x])^2 + 2(f(x)-E[y|x])(y-E[y|x])\} p(x,y)dxdy
\end{aligned}
$$

note that $E[y|x]$ is a function of x, thus

$$\int\int 2(f(x)-E[y|x])(y-E[y|x])p(x,y)dxdy = \int 2(f(x)-E[y|x])(E[y|x]-E[y|x])p(x)dx = 0$$

so we have

$$E[\mathfrak{l}(f(x),y)] = \int\int \{(f(x)-E[y|x])^2 + (y-E[y|x])^2\} p(x,y)dxdy \geq \int\int (y-E[y|x])^2 p(x,y)dxdy$$

to minimize it, simply let $f(x) = E[y|x]$

b. our goal is to minimize

$$
\begin{aligned}
E[\mathfrak{l}(f(x),y)] &= \int\int |f(x)-y| p(x,y)dxdy \\
&= \int (\int |f(x)-y| p(y|x)dy) p(x)dx
\end{aligned}
$$

as for every x, the value of $f(x)$ could be independently chosen, thus we just need to minimize

$$\int |f(x)-y| p(y|x)dy$$

now calculate the derivative of above expression with respect to $f(x)$, and set it to zero, we have

$$0 = \int sign(f(x)-y)p(y|x)dy = \int_{f(x)}^{+\infty} p(y|x)dy - \int_{-\infty}^{f(x)} p(y|x)dy$$

which implies

$$\int_{f(x)}^{+\infty} p(y|x)dy = \int_{-\infty}^{f(x)} p(y|x)dy$$

which is the condition $f(x)$ must satisfy to minimize $E[\mathfrak{l}(f(x),y)]$

Question 2

let's try to minimize

$$
\begin{aligned}
L(f) = P(f(x) \neq y) &= \int\int \mathbf{1}(f(x) \neq y) p(x,y) dx dy \\
&= \int \Big(\int \mathbf{1}(f(x) \neq y) p(y|x) dy \Big) p(x) dx
\end{aligned}
$$

as $f(x)$ could independently chosen for every x, thus we just need to minimize

$$\int \mathbf{1}(f(x) \neq y) p(y|x) dy = \mathbf{1}(f(x) \neq 1) P(1|x) + \mathbf{1}(f(x) \neq 0) P(0|x) = (1)$$

let's discuss how to set $f(x)$ to minimize (1)

- case 1: $P(1|x) > 1/2, P(0|x) = 1 - P(1|x) < 1/2$,
  should choose $\mathbf{1}(f(x) \neq 1) = 1$ and $\mathbf{1}(f(x) \neq 0) = 0$ to minimize (1)
- case 2: $P(1|x) < 1/2, P(0|x) = 1 - P(1|x) > 1/2$,
  should choose $\mathbf{1}(f(x) \neq 1) = 0$ and $\mathbf{1}(f(x) \neq 0) = 1$ to minimize (1)
- case 3: $P(1|x) = P(0|x) = 1/2$,
  any assignment for $\mathbf{1}(f(x) \neq 1)$ and $\mathbf{1}(f(x) \neq 0)$ is OK

we note that $f^*(x)$ satifies above discussion thus is an optimal solution for

$$minimize\ :\ L(f)$$

so we have proved $L(f) \geq L(f^*)$

Question 3

Fisher's Linear Discriminant.

Idea. The main idea of **Fisher's Linear Discriminant** is to project the features from a high-dimention space to a low-dimention space when maximize the distinctions between different classes. And Fisher set the criterion as

$$J(w) = \frac{w^T S_B w}{w^T S_W w}$$

the value of $J(w)$ indicates the distance between different classes. To get the optimal w, we just need to select the D' largest eigenvalues of $S_W^{-1} S_B$ where D' is the dimention of the projected feature space. In the code **diagFisher**, we simply set $S_w = I$.

Experiment results. In our implementatin, the dimention of projected feature space D' was set as 3 for data "Iris.csv" and 9 for "Wine.csv", and in the projected space, **Gaussian Generative Model** will be used to do classification

==================================

```
$ python diagFisher.py 'Iris.csv' 10

Data:  Iris.csv
Error rate for cross_validation: 0.0333333333333
```

```
$ python Fisher.py 'Iris.csv' 10

Data:  Iris.csv
Error rate for cross_validation: 0.02

$ python diagFisher.py 'Wine.csv' 10

Data:  Wine.csv
Error rate for cross_validation: 0.0588235294118


$ python Fisher.py 'Wine.csv' 10

Data:  Wine.csv
Error rate for cross_validation: 0.0470588235294
```
====================================

we can see that both models are quite accurate, and the performence of *Fisher.py* is slightly better.

Least squares linear discriminant.

Idea. This model is quite simple and we just need to minimize the following criterion

$$E(W) = \frac{1}{2} Tr\{(Y - XW)^T(Y - XW)\}$$

The optimal solution would be

$$W = (X^T X)^{-1} X^T Y$$

where X is the matrix of features and Y is the vector of labels.

Experiment results. let's see our result

====================================

```
$ python SqClass.py 'Iris.csv' 10
Data:  Iris.csv
Error rate for cross_validation: 0.2

$ python SqClass.py 'Wine.csv' 10
Data:  Wine.csv
Error rate for cross_validation: 0.0235294117647
```
====================================

much worse than **Fisher's** method for data 'Iris.csv', but better for 'Wine.csv'