# Homework #1

Yifan Zhang

zhan4372@umn.edu

# Problem 1

$E_{(x,y)}[l(f(x),y)] = \int_x\{\int_y l(f(x),y)p(y|x)dy\}p(x)dx$

## (a)

To find the optimal $f(x)$, we need to minimized the loss function.

According to the definition of loss function above, and since the loss function is independent from $x$, we need to minimize

$\int l(f(x),y)p(y|x)dy$.

Substitute $l(f(x),y) = (f(x)-y)^2$:

$$\frac{\partial E}{\partial f} = 2\int (f(x) - y)p(y|x)dy$$

$$let\ the\ dirivative\ be\ 0$$

$$2\int (f(x) - y)p(y|x)dy = 0$$

$$2\int f(x)p(y|x)dy = 2\int yp(y|x)dy$$

$$f(x) = E[y|x]$$

## (b)

To find the optimal $f(x)$, we need to minimized the loss function.

According to the definition of loss function above, and since the loss function is independent from $x$, we need to minimize

$\int l(f(x), y)p(y|x)dy$.

Substitute $l(f(x), y) = |f(x) - y|$:

$$
\begin{aligned}
\frac{\partial E}{\partial f} &= \frac{\partial E}{\partial f}\int |f(x)-y|p(y|x)dy \\
&= \frac{\partial E}{\partial f}\{\int_{-\infty}^{f(x)} (f(x)-y)p(y|x)dy - \int_{f(x)}^{\infty} (f(x) -
\end{aligned}
$$

$$
\begin{aligned}
&= \int_{-\infty}^{f(x)} p(y|x)dy - \int_{f(x)}^{\infty} p(y|x)dy \\
&let\ the\ dirivative\ be\ 0 \\
0 &= \int_{-\infty}^{f(x)} p(y|x)dy - \int_{f(x)}^{\infty} p(y|x)dy \\
\int_{-\infty}^{f(x)} p(y|x)dy &= \int_{f(x)}^{\infty} p(y|x)dy \\
f(x) &= median(Y|X=x)
\end{aligned}
$$

# Problem 2

$$
\begin{aligned}
||Aw-b+f||_2^2+g &= ||(Aw-b)+f||_2^2+g \\
&= ||Aw-b||_2^2+2(Aw-b)^Tf+||f||_2^2+g
\end{aligned}
$$

Then, we can get

$$
\begin{aligned}
2(Aw-b)^Tf+||f||_2^2+g &= c^Tw+d \\
2(Aw)^Tf-2b^Tf+||f||_2^2+g &= c^Tw+d
\end{aligned}
$$

Then

$$
\begin{aligned}
c^T w &= 2(Aw)^T f \\
c^T w &= 2(A^T f)^T w \\
c^T &= 2(A^T f)^T \\
c &= 2A^T f
\end{aligned}
$$

And

$$d = -2b^T f + ||f||_2^2 + g$$

To minimize the error, we can get

$$
\begin{aligned}
\frac{\partial}{\partial w} &= 2(Aw - (b - f))A \\
&\textit{let the derivative be } 0 \\
(Aw - (b - f))A &= 0 \\
A^T Aw - A^T(b - f) &= 0 \\
A^T Aw &= A^T(b - f) \\
w &= (A^T A)^{-1} A^T (b - f)
\end{aligned}
$$

# Problem 3

## (i)

Approaches:

Suppose X is the data set and y is the target set.

K is the number of classes

$m_k = \frac{1}{N_k} \sum_{i \in C_k} X_i$

$S_b = (m0 - m1) * (m_0 - m_1)^T$

$S_w = \sum_{i \in C_k} (X_i - m_i)(X_i - m_i)^T$

$f(x) = w^T x$ where $w = S_w^{-1}(m0-m1)$

**Threshold:**

$w_0 = w^T * m$ where $m = \frac{1}{N} \sum_{i=1}^{N} x_i$

If $f(x) <= threshold$, $p(C_1|x) = 1$.

If $f(x) > threshold$, $P(C_0|x) = 1$.

**result**

```
Boston50 with LDA1dThres:
performance on testing data:
fold 0: 0.098
fold 1: 0.2549
fold 2: 0.2157
fold 3: 0.1765
fold 4: 0.1961
fold 5: 0.1373
fold 6: 0.1
fold 7: 0.14
fold 8: 0.12
fold 9: 0.12
mean: 0.15585
standard deviation: 0.050070974626024604

performance on training data:
fold 0: 0.1516
```

```
fold 1: 0.1297
fold 2: 0.1341
fold 3: 0.1407
fold 4: 0.1363
fold 5: 0.1429
fold 6: 0.1491
fold 7: 0.1447
fold 8: 0.1447
fold 9: 0.1469
mean: 0.14207
standard deviation: 0.006555920987931446
```

# (ii)

**projection:**

Suppose X is the data set and y is the target set.

$X \in \mathbb{R}^{m \times n}$

K is the number of classes

$S_b = \sum_{k=1}^{K} N_k(\mu_k - \mu)(\mu_k - \mu)^T$

where $\mu_k = \frac{1}{N_k}\sum_{n \in C_k} x_n$ and $\mu = \frac{1}{N}\sum_{k=1}^{K} N_k \mu_k$.

$S_w = \sum_{k=1}^{K}\sum_{n \in C_k}(x_n - \mu_k)(x_n - \mu_k)^T$.

$v = eigenvector\ of\ S_w^{-1}S_b$.

$\lambda = eigenvalue\ of\ S_w^{-1}S_b$

Sort v according to $\lambda$ in descending order.

Then, $w = [v[0], v[1]]$, $w \in \mathbb{R}^{n \times 2}$

$newX = X * w, newX \in \mathbb{R}^{m \times 2}$

**bivariate gaussian generative model:**

$P(C_k|x) = \frac{exp(a_k)}{\sum_{j=1}^{K} exp(a_j)}$, which is the formula of soft max.

$a_k = w_k^T x + w_{k0}$

where

$w_k = \Sigma^{-1} \mu_k$

$w_{k0} = -\frac{1}{2} \mu_k^T \Sigma^{-1} \mu_k + log(p(C_k))$

$p(C_k) = \frac{N_k}{N}$

$\mu_k = \frac{1}{N_k} \sum_{n \in C_k} x_n$

$\Sigma = \sum_{k=1}^{K} \frac{N_k}{N} (\frac{1}{N_k} \sum_{x \in C_k} (x_i - \mu_k)(x_i - \mu_k)^T)$

**predict:**

The class of $x_i$ is $argmax_k \ p(C_k|x_i)$.

**result:**

```
Digits with LDA2dGaussGM:
performance on testing data:
fold 0: 0.3056
fold 1: 0.3389
fold 2: 0.3611
fold 3: 0.2778
```

```
fold 4: 0.3167
fold 5: 0.3611
fold 6: 0.3056
fold 7: 0.4078
fold 8: 0.3073
fold 9: 0.3743
mean: 0.33562
standard deviation: 0.03784290686509165

performance on training data:
fold 0: 0.2839
fold 1: 0.2938
fold 2: 0.274
fold 3: 0.2888
fold 4: 0.3142
fold 5: 0.2777
fold 6: 0.2919
fold 7: 0.3004
fold 8: 0.3331
fold 9: 0.28
mean: 0.29378000000000004
standard deviation: 0.017240986050687464
```

# Problem 4

## (i)

**muti-class gaussian generative model:**

Suppose X is the data set and y is the target set.

$X \in \mathbb{R}^{m \times n}$

K is the number of classes

$P(C_k|x) = \frac{exp(a_k)}{\sum_{j=1}^{K} exp(a_j)}$, which is the formula of soft max.

$a_k = w_k^T x + w_{k0}$

where

$w_k = \Sigma^{-1} \mu_k$

$w_{k0} = -\frac{1}{2} \mu_k^T \Sigma^{-1} \mu_k + log(p(C_k))$

$p(C_k) = \frac{N_k}{N}$

$\mu_k = \frac{1}{N_k} \sum_{n \in C_k} x_n$

$\Sigma = \sum_{k=1}^{K} \frac{N_k}{N} (\frac{1}{N_k} \sum_{x \in C_k} (x_i - \mu_k)(x_i - \mu_k)^T)$

By naive bayes, since $p(x|C_k) = \prod_{i=1}^{p} p(X_i|C_k)$,

The covariance matrix $\Sigma$ will become a diagonal matrix since $cov(x_i, x_j) = 0, i \neq j$.

**predict:**

The class of $x_i$ is $argmax_k \ p(C_k|x_i)$.

## (ii)

I used gradient descent to find the global minima, soft max for the prediction, and cross entropy error as the loss function.

Suppose X is the data set and y is the target set.

$X \in \mathbb{R}^{m \times n}$

K is the number of classes.

$\phi(X) = [1, X_1, X_2, ..., X_n]$.

$p(C_k|x) = \frac{exp(a_k)}{\sum_{j=1}^{K} exp(a_j)}$,

where

$a_k = w_k^T * \phi(x)$

$w \in \mathbb{R}^{d+1 \times k}$.

$w$ is initialized by uniform distribution between -0.01and 0.01.

The cross entropy error: $E = -\sum_{i=1}^{N} \sum_{k=1}^{K} y_{n,k} * ln(predict_{n,k})$

The gradient $\nabla E = \sum_{k=1}^K \sum_{n=1}^N (predict_{n,k} - y_{n,k})*x_n$

**parameters:**

step_size = 0.01

max_iteration = 2000

tolerance = 0.001

**convergence condition:**

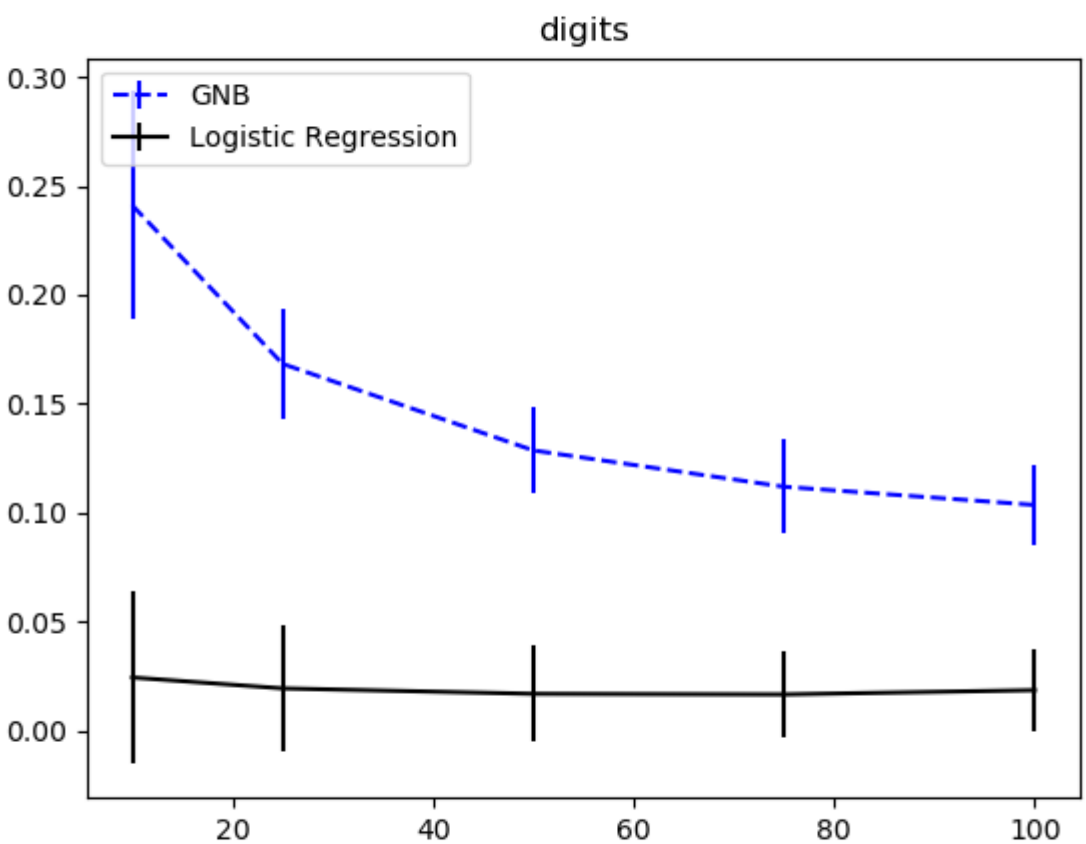
The cross entropy $E = 0$ or $E < tolerance$.

### Pseudo code

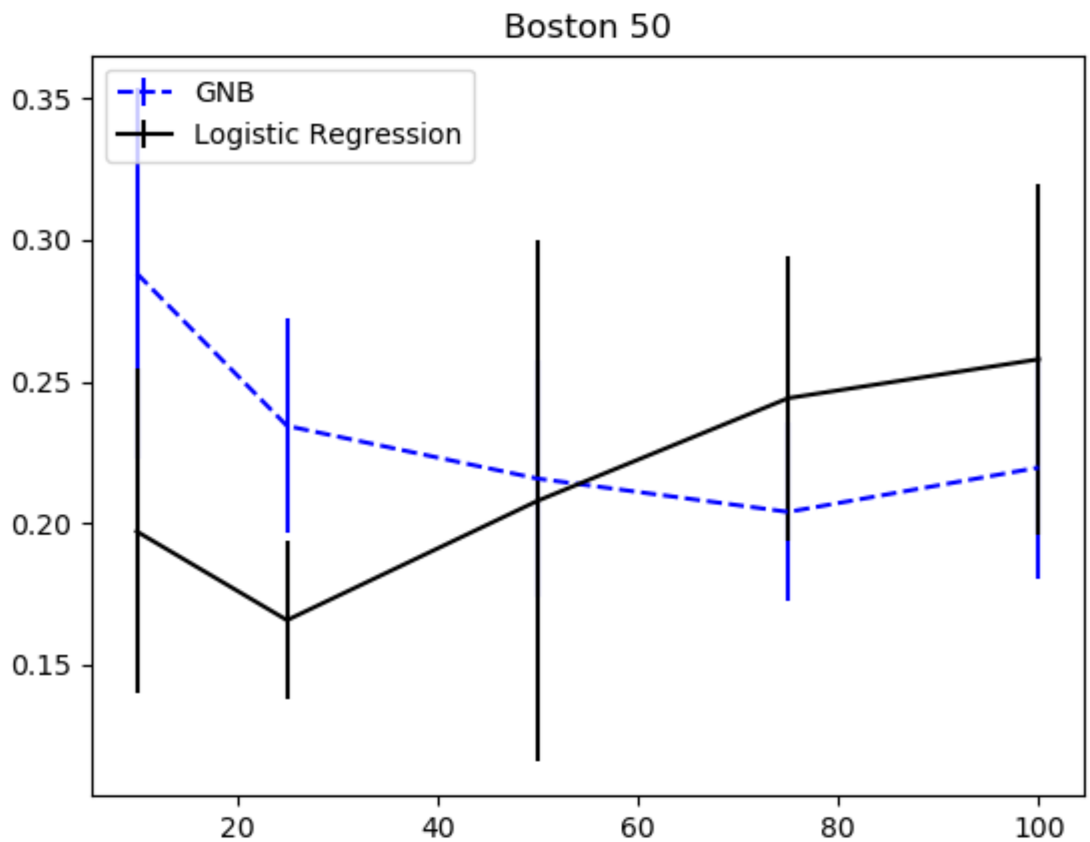
```
for each iteration:
    update delta_w
    w = w - step_size*delta_w
    calculate cross entropy error E

    if converge:
        stop
```

### performance:

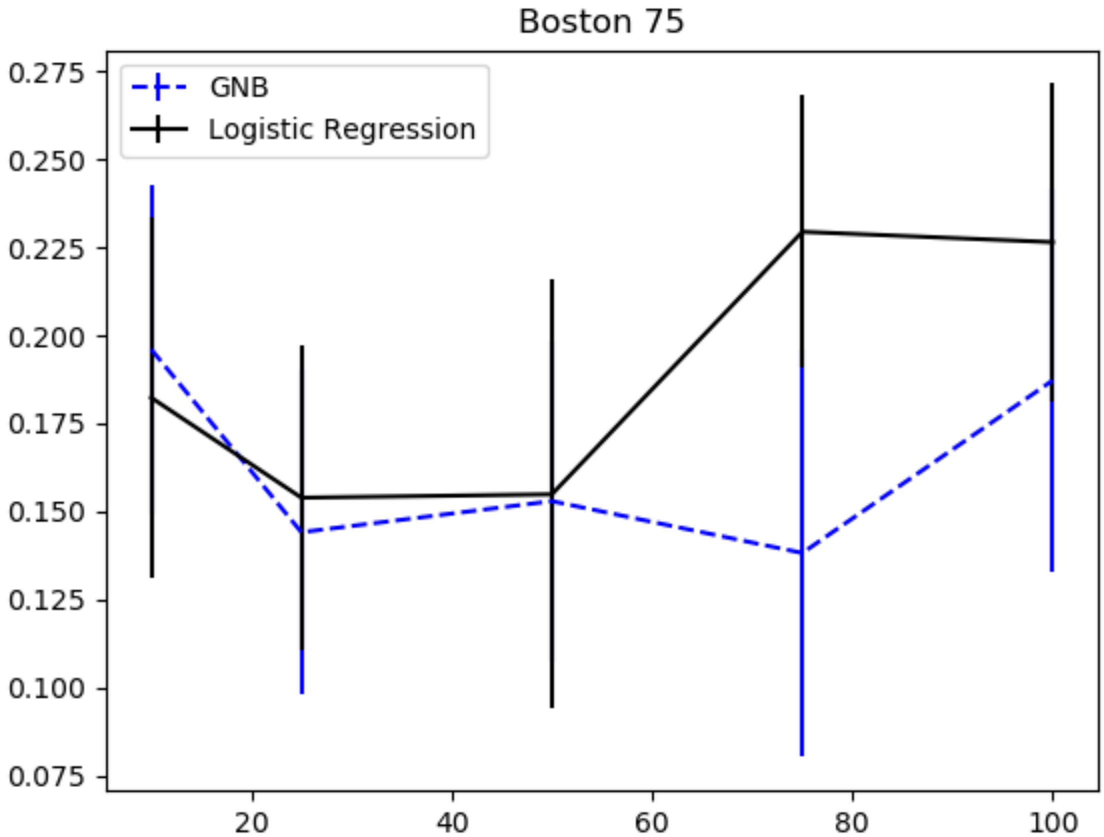
### Digits




### Boston 50


Boston 50
GNB
Logistic Regression
0.35
0.30
0.25
0.20
0.15
20
40
60
80
100

## Boston 75




# Acknowledge

Yiwen Xu

Jun-Jee Chao