STA 623 homework 5

Lingyun Shao

Oct. 17, 2018

Problem Statement

(1) Simulate data of the type described $(x_i, y_i), i = 1, ..., 1000$. $y_i \in \{0, 1\}, f(x_i|y_i = y)$ is univariate Gaussian.

To separate our rule building process and test process, I decide to derive 2 datasets, training data and test data. While training data are for building a decision rule, our test data chould be used to calculate FPR, FNR and draw the ROC curves.

Each sample consists of x_i, y_i, where y_i has a probability of 0.4 being 0 and a probability of 0.6 being 1, i.e. $Y_i \sim Bernoulli(0.6)$. $X_i|Y_i = 0 \sim N(0, 0.5^2)$ and $X_i|Y_i = 1 \sim N(2, 2)$

```
n = 1000 * 2
set.seed(1)
u = runif(n, 0, 1)
Y = u>0.4
X = rnorm(n, 2*Y, 2*Y + 0.5*!Y)
dt = cbind(X, Y)
ind = sample(1:n, n/2, replace = FALSE)
dt.train = dt[ind,] %>% data.frame()
dt.test = dt[-ind,] %>% data.frame()
```

(2) Calculate the FPR, FNR and draw ROC curve for different values of τ for the 3 decision functions(just threshold, logistic regression and discriminant analysis).

1. Just threshold

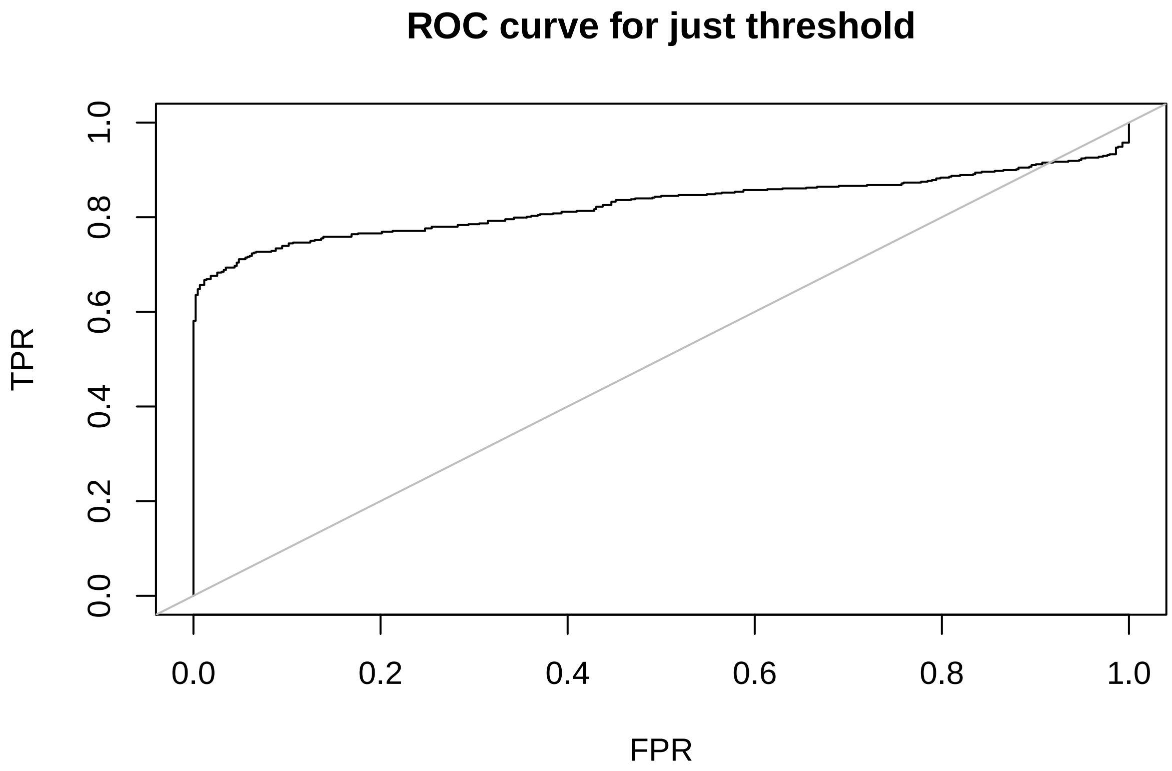
Suppose our classifier is

$$\delta_\tau(x) = \begin{cases} 1, & if\ x > \tau \\ 0, & otherwise \end{cases}$$

Since we don't have to learn any parameter in this decision model, we will just use the test data. For different τ, we can compute FPR, FNR and draw a ROC curve.

```
tau = seq(-10, 10, by = 0.01)
# defining function to compute fpr and fnr
comp = function(pred, tau) {
  x = pred
  y = dt.test[,2]
  fpr = mean(x[y==0] > tau)
  fnr = mean(x[y==1] <= tau)
  r = c(fpr, fnr)
  names(r) = c('FPR', 'FNR')
  return(r)
}
res1 = purrr::map(tau, ~ comp(pred = dt.test[,1], tau = .))
```

```r
res1 = matrix(unlist(res1), ncol = 2, byrow = TRUE)
plot(res1[,1], 1-res1[,2], xlim = c(0,1), ylim = c(0,1), type = 's',
     xlab = 'FPR', ylab = 'TPR', main = 'ROC curve for just threshold')
abline(a=0, b=1, col='gray')
```



```r
tb1 = map(-5:5, ~ comp(pred = dt.test[,1], tau = .)) %>%
  unlist %>%
  matrix(nrow = 2)
rownames(tb1) = c('FPR', 'FNR')
colnames(tb1) = paste('tau=',-5:5, sep='')
kable(tb1, digits = 3, caption = 'FPR and FNR of just threshold for different tau')
```

Table 1: FPR and FNR of just threshold for different tau

	tau=-5	tau=-4	tau=-3	tau=-2	tau=-1	tau=0	tau=1	tau=2	tau=3	tau=4	tau=5
FPR	1.000	1.000	1.000	1.000	0.981	0.512	0.032	0.0	0.000	0.00	0.000
FNR	0.002	0.004	0.014	0.028	0.067	0.155	0.315	0.5	0.692	0.84	0.923

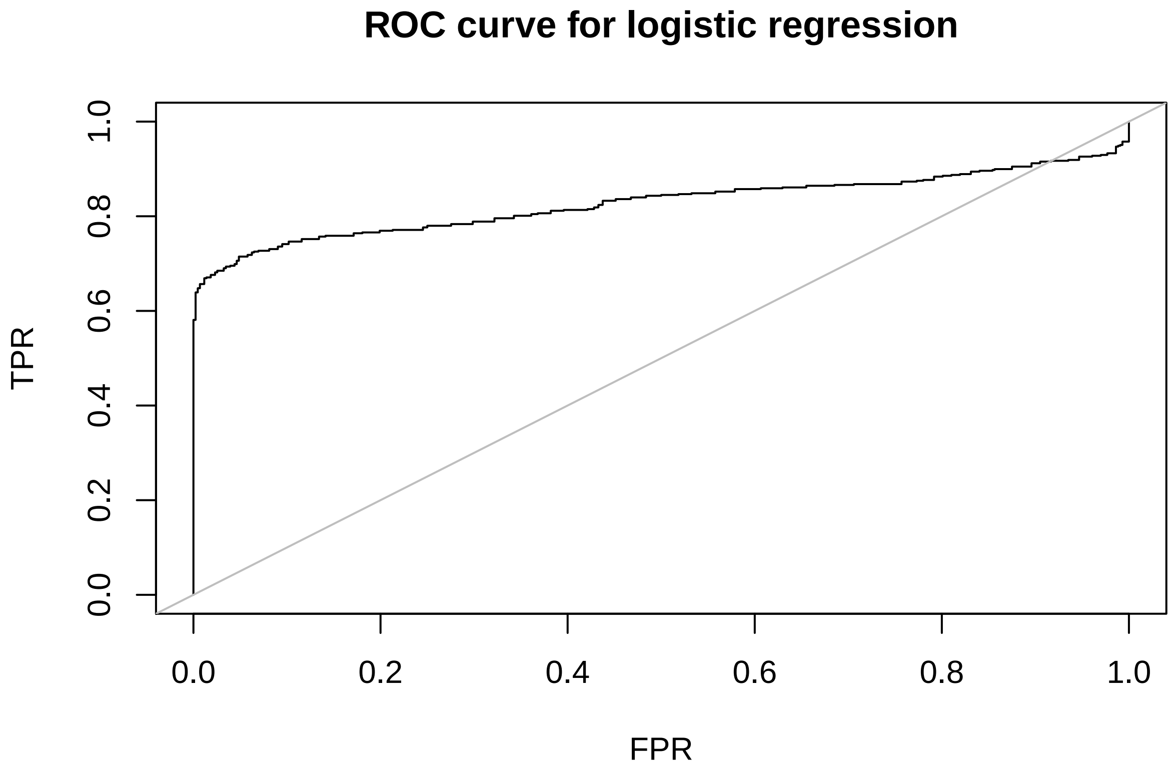
2. Logistic Regression

We can also build a logistic regression decision rule using the training data. By setting different threshold τ to predicted probabilities, we get τ-dependent decision rule.

$$\delta_\tau(x) = \begin{cases} 1, & if \quad \pi(x,\hat{\theta}) > \tau \\ 0, & otherwise \end{cases}$$

Where $\pi(x_i,\hat{\theta}) = Pr(y_i = 1|x_i) = \frac{1}{1+e^{-x_i^{\mathrm{T}}\hat{\theta}}}$ We can calculate FPR, FNR and draw a ROC curve based on the test dataset.

```
lr = glm(data = dt.train, Y ~ X, family = 'binomial')
lr.pred = predict.glm(lr, newdata = dt.test, type = 'response')
tau = seq(0, 1, 0.005)
res2 = purrr::map(tau, ~ comp(pred = lr.pred, tau = .))
res2 = matrix(unlist(res2), ncol = 2, byrow = TRUE)
plot(res2[,1], 1-res2[,2], xlim = c(0,1), ylim = c(0,1), type = 's',
     xlab = 'FPR', ylab = 'TPR', main = 'ROC curve for logistic regression')
abline(a=0, b=1, col='gray')
```



```
tb2 = map(seq(0, 1 ,by = 0.1), ~ comp(pred = lr.pred, tau = .)) %>%
  unlist %>%
  matrix(nrow = 2)
rownames(tb2) = c('FPR', 'FNR')
colnames(tb2) = paste('tau=',seq(0, 1, by =0.1), sep='')
kable(tb2, digits = 3, caption = 'FPR and FNR of logistic regression for different tau')
```

Table 2: FPR and FNR of logistic regression for different tau

	tau=0	tau=0.1	tau=0.2	tau=0.3	tau=0.4	tau=0.5	tau=0.6	tau=0.7	tau=0.8	tau=0.9	tau=1
FPR	1	1.000	0.991	0.889	0.620	0.343	0.090	0.007	0.000	0.000	0

	tau=0	tau=0.1	tau=0.2	tau=0.3	tau=0.4	tau=0.5	tau=0.6	tau=0.7	tau=0.8	tau=0.9	tau=1
FNR	0	0.026	0.051	0.095	0.141	0.204	0.269	0.352	0.468	0.653	1

3. Discriminant Analysis

Suppose we have $\Pr(y_i = 1) = \lambda$, then by Bayes rule, we have

$$
\begin{aligned}
Pr(y_i = 1|x_i, \theta) &= \frac{\lambda f(x_i|y_i = 1)}{\lambda f(x_i|y_i = 1) + (1-\lambda) f(x_i|y_i = 0)} \\
&= \frac{\lambda f(x_i|\mu_1, \sigma_1)}{\lambda f(x_i|\mu_1, \sigma_1) + (1-\lambda) f(x_i|\mu_0, \sigma_0)}
\end{aligned}
$$

Then for different threshold τ, we have a τ-dependent decision rule

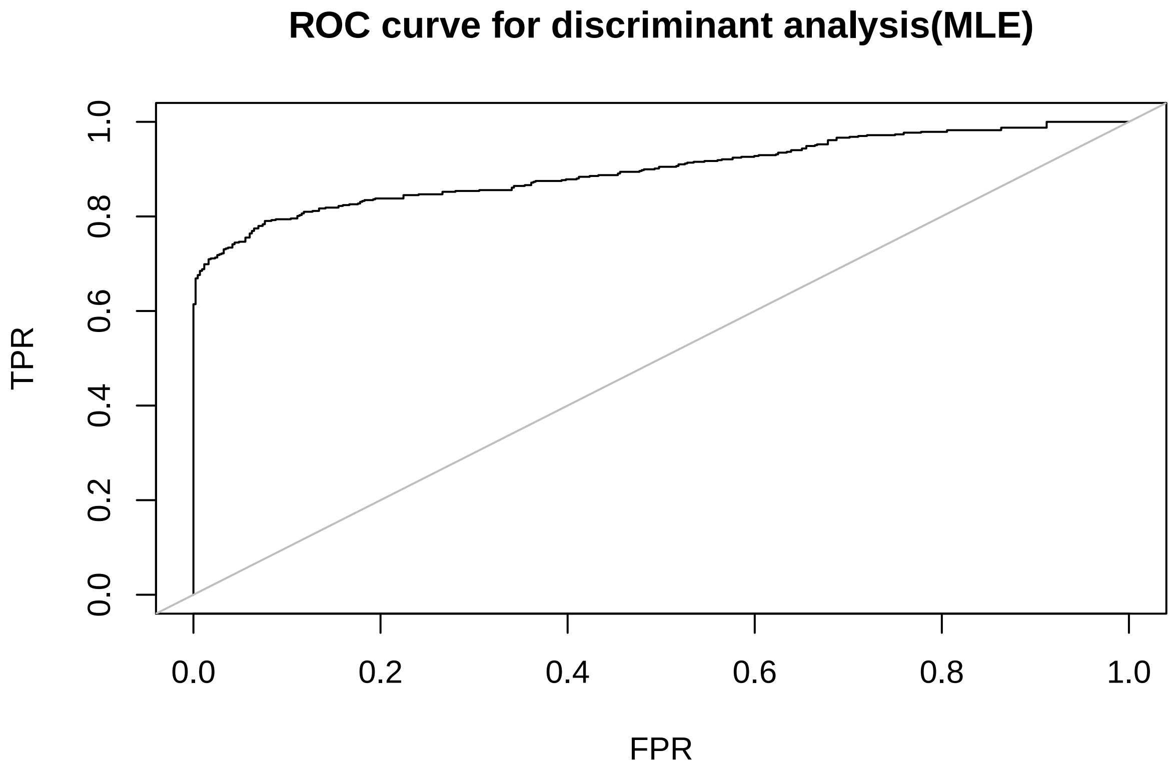
$$
\delta_\tau(x) = \begin{cases} 1, & if \quad \Pr(y_i = 1|x_i, \theta) > \tau \\ 0, & otherwise \end{cases}
$$

To get the information about $\theta = (\mu, \sigma)$, we need to use the training data.

a) MLE plug-in

one possible way is to get MLE from training data and plug in the estimates. Here I derived the MLE's, $\hat{\mu}_0, \hat{\mu}_1, \hat{\sigma}_0, \hat{\sigma}_1$ and get the likelihood and probability of being 1 for each case in the test dataset.

```r
ind1 = dt.train[,2] == 1
sigma0 = sd(dt.train[!ind1,1])
sigma1 = sd(dt.train[ind1,1])
mu0 = mean(dt.train[!ind1,1])
mu1 = mean(dt.train[ind1,1])
pi1 = sum(ind1)/nrow(dt.train)
pi0 = 1 - pi1
X.test = dt.test[,1]
set.seed(1)
# the prob of predicting x_i to be 1
Pr1 = pi1 * dnorm(X.test, mu1, sigma1)/ (pi1 * dnorm(X.test, mu1, sigma1) + pi0 * dnorm(X.test, mu0, sig
tau = seq(0, 1, 0.001)
res3 = purrr::map(tau, ~ comp(pred = Pr1, tau = .))
res3 = matrix(unlist(res3), ncol = 2, byrow = TRUE)
plot(res3[,1], 1-res3[,2], xlim = c(0,1), ylim = c(0,1), type = 's',
     xlab = 'FPR', ylab = 'TPR', main = 'ROC curve for discriminant analysis(MLE)')
abline(a=0, b=1, col='gray')
```



```
tb3 = map(seq(0, 1 ,by = 0.1), ~ comp(pred = Pr1, tau = .)) %>%
  unlist %>%
  matrix(nrow = 2)
rownames(tb3) = c('FPR', 'FNR')
colnames(tb3) = paste('tau=',seq(0, 1, by =0.1), sep='')
kable(tb3, digits = 3, caption = 'FPR and FNR of DA(MLE) for different tau')
```

Table 3: FPR and FNR of DA(MLE) for different tau

	tau=0	tau=0.1	tau=0.2	tau=0.3	tau=0.4	tau=0.5	tau=0.6	tau=0.7	tau=0.8	tau=0.9	tau=1
FPR	1	1	0.72	0.343	0.190	0.104	0.065	0.044	0.019	0.007	0
FNR	0	0	0.03	0.139	0.165	0.206	0.229	0.259	0.290	0.315	1

b) Bayesian Methods

If we have prior belief about the parameter μ, say we believe $\mu_0 \sim N(0,1), \mu_1 \sim N(5,1)$, then we can use Bayesian updating and Monte Carlo approximation to get the likelihood. To simplify the problem, we may treat σ as known here, which is given by its sample estimate $\hat{\sigma}$.

The sampling model is normal, the prior is conjugate for the sampling model (normal), so the posterior is also normal, which can be derived by updating the parameters.

The parameter μ can be updated by taking the weighted averaged

$posterior\ mean = \frac{prior\ precision*prior\ mean+sample\ precision*MLE}{prior\ precision+sample\ precision}$

$posterior\ precision = prior\ precision + sample\ precision$

Then we cam derive samples $\mu_i^{(1)}, ..., \mu_i^{(S)}$ from the posterior distribution of μ_i, compute the Monte Carlo approximation for the likelihood.

$$L_0 := L(x_i|\hat{\sigma}_0, x_{train}) = \int L(x_i|\mu_0, \hat{\sigma}_0, x_{train}) f(\mu_0|\hat{\sigma}_0, x_{train}) d\mu_0 \approx \frac{1}{S}\sum_{s=1}^{S} L(x_i|\mu_0^{(s)}, \hat{\sigma}_0) := \hat{L}_0$$

Where $\hat{\sigma}_0$ is given by training sample estimate to simply the problem. And $\hat{L}_1$ is similarly derived.

Then the probability is

$$Pr(y_i = 1|x_i) = \frac{Pr(y_i = 1)L_1}{Pr(y_1 = 1)L_1 + Pr(y_i = 0)L_0} \approx \frac{\hat{\lambda}\hat{L}_1}{\hat{\lambda}\hat{L}_1 + (1 - \hat{\lambda})\hat{L}_0}$$

where $\hat{\lambda} = \frac{\sum I_{(y_i=1)}}{n_{train}}$ is the estimated proportion of $y_i = 1$ in the training data.

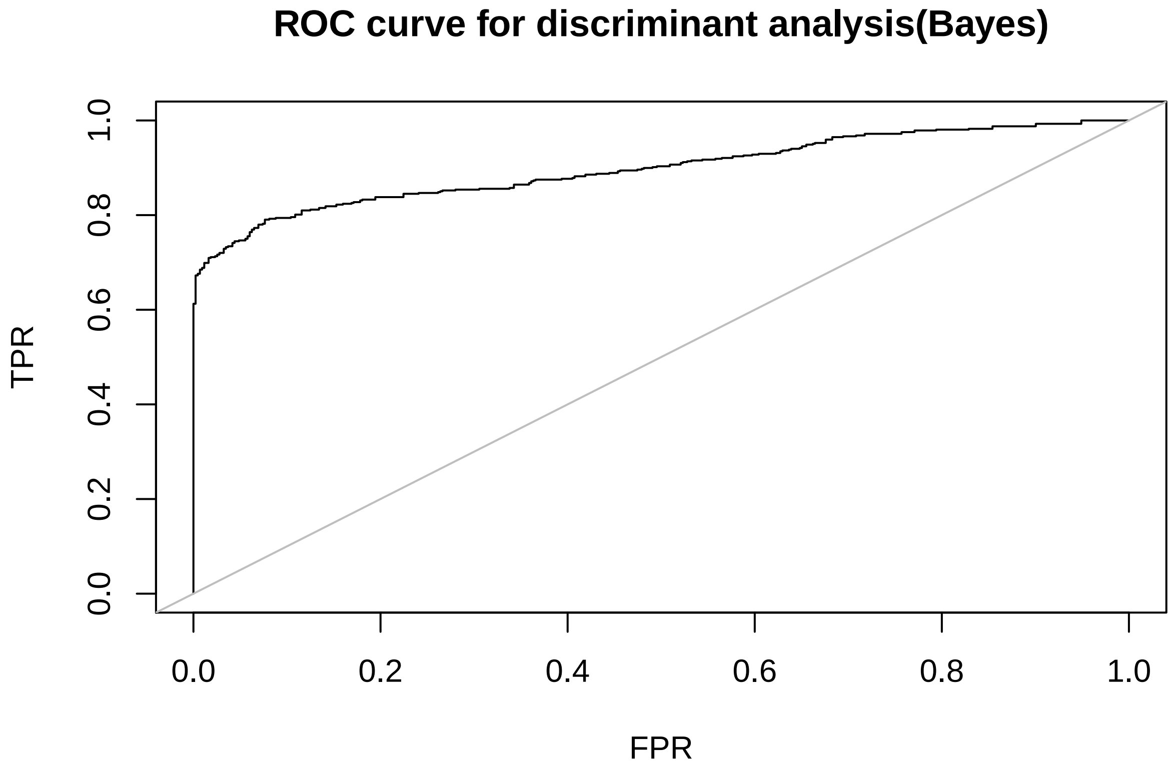
```
ind1 = dt.train[,2] == 1
# sample estimates
n0 = sum(ind1)
n1 = sum(!ind1)
sigma0 = sd(dt.train[!ind1,1])
pre0 = n0/sigma0^2
sigma1 = sd(dt.train[ind1,1])
pre1 = n1/sigma1^2
mu0 = mean(dt.train[!ind1,1])
mu1 = mean(dt.train[ind1,1])
# prior hyperparameters
mu0.pri = 0
mu1.pri = 5
sigma0.pri = sigma1.pri = 1
pre0.pri = pre1.pri = 1/1^2 #precision
# posterior parameters
pre0.pos = pre0.pri + pre0
mu0.pos = (mu0.pri*pre0.pri + mu0*pre0)/pre0.pos
pre1.pos = pre1.pri + pre1
mu1.pos = (mu1.pri*pre1.pri + mu1*pre1)/pre1.pos

# sample from posterior
nsam = 10000
set.seed(1)
sam0.pos = rnorm(nsam, mu0.pos, sqrt(1/pre0.pos))
sam1.pos = rnorm(nsam, mu1.pos, sqrt(1/pre1.pos))

# Monte Carlo approximation to get likelihood
l_approx = function(x, sam, sig) {
  mean(dnorm(x, sam, sig))
}
l0 = map_dbl(X.test, ~l_approx(., sam0.pos, sigma0))
l1 = map_dbl(X.test, ~l_approx(., sam1.pos, sigma1))

# Calculating posterior probability
pi1 = sum(ind1)/nrow(dt.train)
pi0 = 1 - pi1
Pr1.B = pi1 * l1/ (pi1 * l1+ pi0 * l0)
tau = seq(0, 1, 0.001)
```

```
res4 = purrr::map(tau, ~ comp(pred = Pr1.B, tau = .))
res4 = matrix(unlist(res4), ncol = 2, byrow = TRUE)
plot(res4[,1], 1-res4[,2], xlim = c(0,1), ylim = c(0,1), type = 's',
     xlab = 'FPR', ylab = 'TPR', main = 'ROC curve for discriminant analysis(Bayes)')
abline(a=0, b=1, col='gray')
```



```
tb4 = map(seq(0, 1 ,by = 0.1), ~ comp(pred = Pr1.B, tau = .)) %>%
  unlist %>%
  matrix(nrow = 2)
rownames(tb4) = c('FPR', 'FNR')
colnames(tb4) = paste('tau=',seq(0, 1, by =0.1), sep='')
kable(tb4, digits = 3, caption = 'FPR and FNR of DA(Bayes) for different tau')
```

Table 4: FPR and FNR of DA(Bayes) for different tau

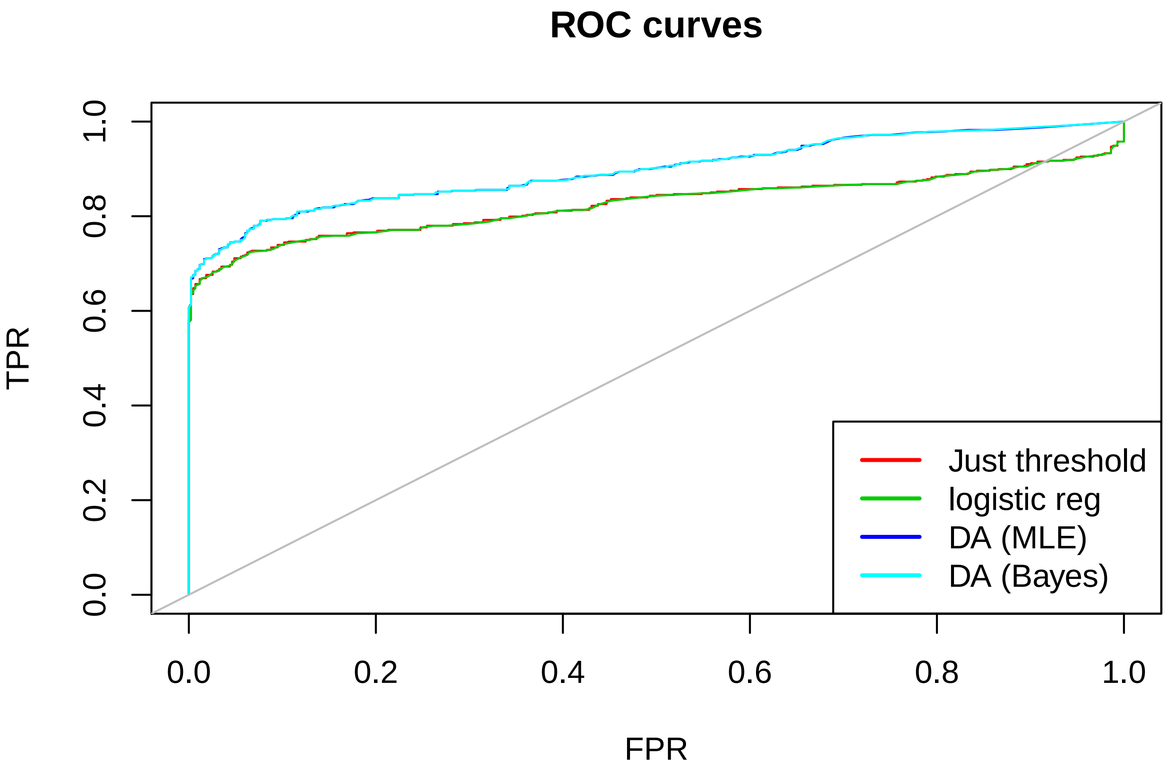
	tau=0	tau=0.1	tau=0.2	tau=0.3	tau=0.4	tau=0.5	tau=0.6	tau=0.7	tau=0.8	tau=0.9	tau=1
FPR	1	1	0.694	0.340	0.181	0.100	0.065	0.042	0.016	0.007	0
FNR	0	0	0.035	0.143	0.167	0.206	0.229	0.261	0.292	0.315	1

```
# comp(pred = dt.test[,1], tau = 1)
# comp(pred = lr.pred, tau = 0.5)
# comp(pred = Pr1, tau = 0.5)
# comp(pred = Pr1.B, tau = 0.5)
```

Comparing methods

We can compare 4 ROC curves and exam the performance of each method.

```
plot(res1[,1], 1-res1[,2], xlim = c(0,1), ylim = c(0,1), type = 's',
     xlab = 'FPR', ylab = 'TPR', main = 'ROC curves',col=2)
lines(res2[,1], 1-res2[,2],col=3)
lines(res3[,1], 1-res3[,2],col=4)
lines(res4[,1], 1-res4[,2],col=5)
abline(a=0,b=1,col='gray')
legend('bottomright',c('Just threshold','logistic reg','DA (MLE)','DA (Bayes)'), col=2:5, lwd=2)
```



As is shown in the plot, **Just threshold** and **logistic reg** have almost the same ROC curve. Since there is only one predictor in the logistic regression model and its coefficient is positive, these two methods will have the exact same ROC curve.

Two **DA** methods using MLE and Bayes have the same ROC curve, which is due to the huge training sample size here. When the sample size is too large, then the prior belief is overwhelmed by the data. Discriminant analysis using Bayesian methods are asymptotically the same as discriminant analysis using MLE.

DA has better performance than **Just threshold** and **logistic reg** in classifying our simulated test data since **DA** takes into consideration variance while the other two do not. However, I did find that if I set the variance to be the same in our data generating distribution, i.e. $X_i|Y_i = 0 \sim N(0, 2)$ and $X_i|Y_i = 1 \sim N(2, 2)$, then all 4 methods will have the same ROC curve.