Homework 3

Ma Jingchun, 2020111235

我们已经看到，可以用非线性核拟合SVM，以便使用非线性决策边界执行分类。我们现在将看到，我们还可以通过使用特征的非线性变换执行逻辑回归来获得非线性决策边界。

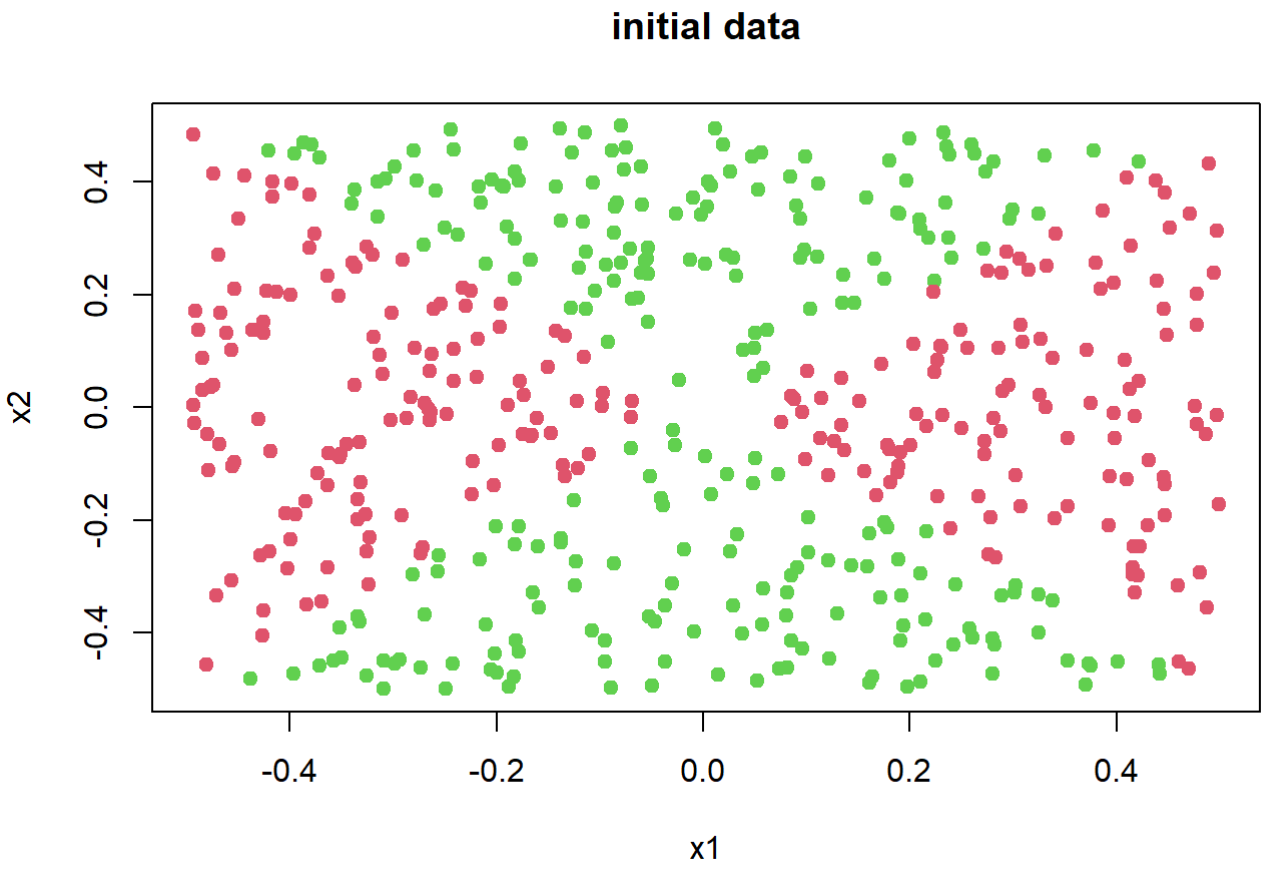
```
library(e1071)
```

1.生成一个n=500且p=2的数据集，使得观测值属于两个类，它们之间具有二次决策边界。

```
x1 <- runif(500) - 0.5
x2 <- runif(500) - 0.5
y <- 1 * (x1^2 - x2^2 > 0)
DF <- data.frame(x1 = x1, x2 = x2, y = as.factor(y))
```

2.绘制观测值并按标签赋颜色。在x轴上标注X1，在y轴上标注X2。

```
plot( x1, x2, col=(3-y), pch=19, cex=1, xlab='x1', ylab='x2', main='initial data' )
```



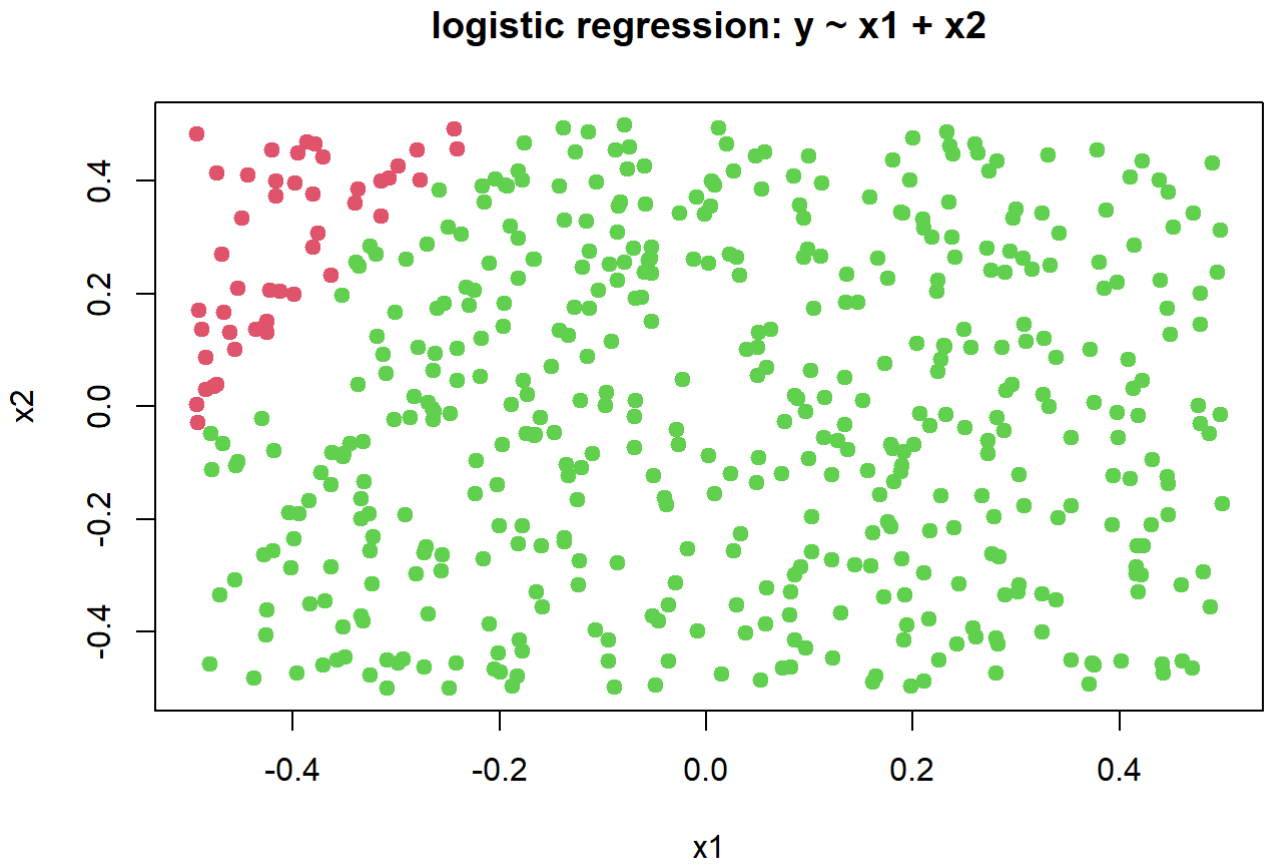
3.使用X1和X2作为预测变量，对数据拟合逻辑回归模型。

```
glm.fits <- glm(y ~ x1 + x2, data=DF, family = binomial)
summary(glm.fits)
```

```
##
## Call:
## glm(formula = y ~ x1 + x2, family = binomial, data = DF)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.185  -1.157  -1.130   1.193   1.230
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.04953    0.08957  -0.553    0.580
## x1          -0.10628    0.32106  -0.331    0.741
## x2           0.05231    0.30836   0.170    0.865
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 692.86  on 499  degrees of freedom
## Residual deviance: 692.72  on 497  degrees of freedom
## AIC: 698.72
##
## Number of Fisher Scoring iterations: 3
```

4.将该模型应用于训练数据集，以获得每个训练观测值的预测类标签。绘制观测值，并根据预测的类标签着色。决策边界应为线性。

```
glm.probs <- predict(glm.fits, newdata=data.frame(x1=x1,x2=x2), type = "response")
glm.labels <- as.numeric(glm.probs>=0.5)

plot(x1, x2, col=(3-glm.labels), pch=19, cex=1.05, xlab='x1', ylab='x2', main='logistic regress
ion: y ~ x1 + x2' )
```



5. 使用X1和X2的非线性函数作为预测因子，用逻辑回归模型拟合数据（例如，X1^2，X1∗X2，log(X2),以此类推）

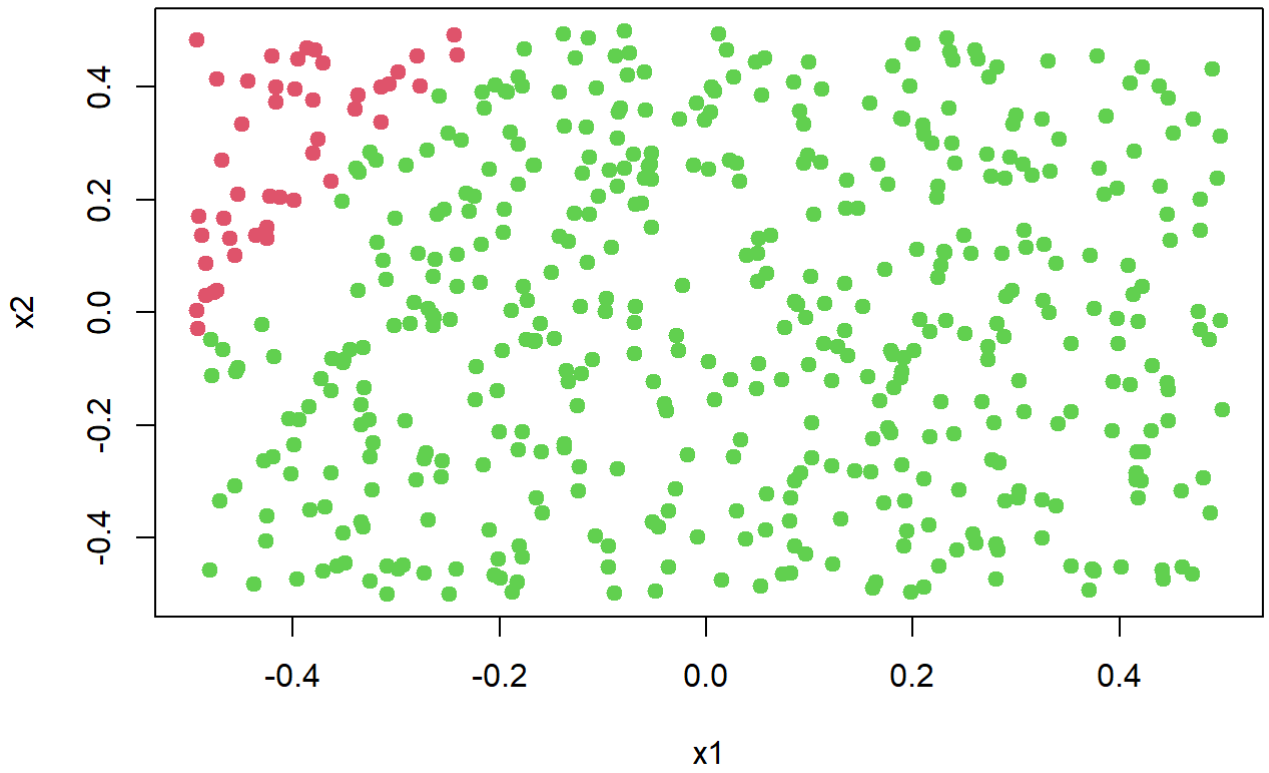
```
glm.fits1 <- glm(y ~ x1 + x1^2 + x2, data=DF, family = binomial)

glm.fits2 <- glm(y ~ x1 + x2 + log(x2 + 1), data=DF, family = binomial)

glm.fits3 <- glm(y ~ x1 + x2 + x1*x2, data=DF, family = binomial)

glm.fits4 <- glm(y ~ x1 + x2 + I(x1^2) + I(x2^2) + I(x1*x2), data=DF, family = binomial)
```

```
## Warning: glm.fit:算法没有聚合
```

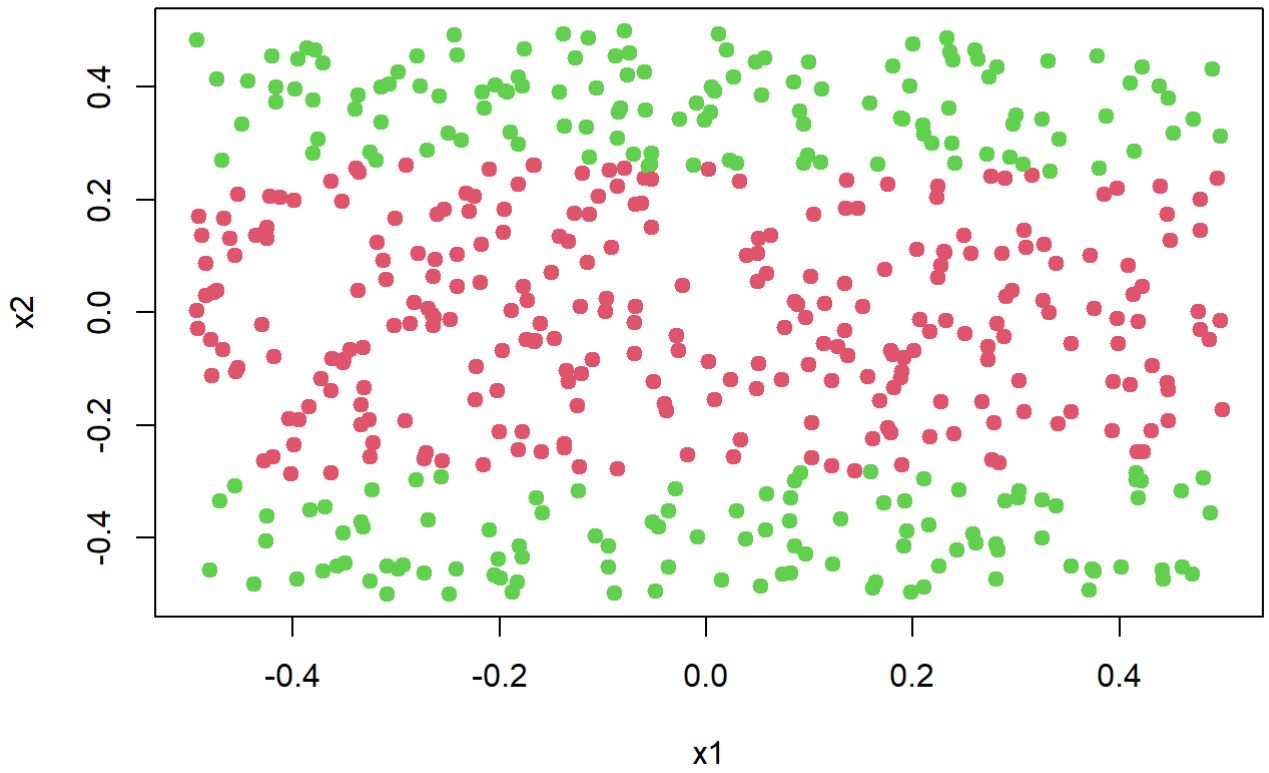
```
## Warning: glm.fit:拟合機率算出来是数值零或一
```

6. 将该模型应用于训练数据，以获得每个训练观测值的预测类标签。绘制观测值，根据类标签着色。决策边界应明显为非线性的。如果不是，那么重复(a)-(e)，直到找到一个预测的类标签明显是非线性的例子。

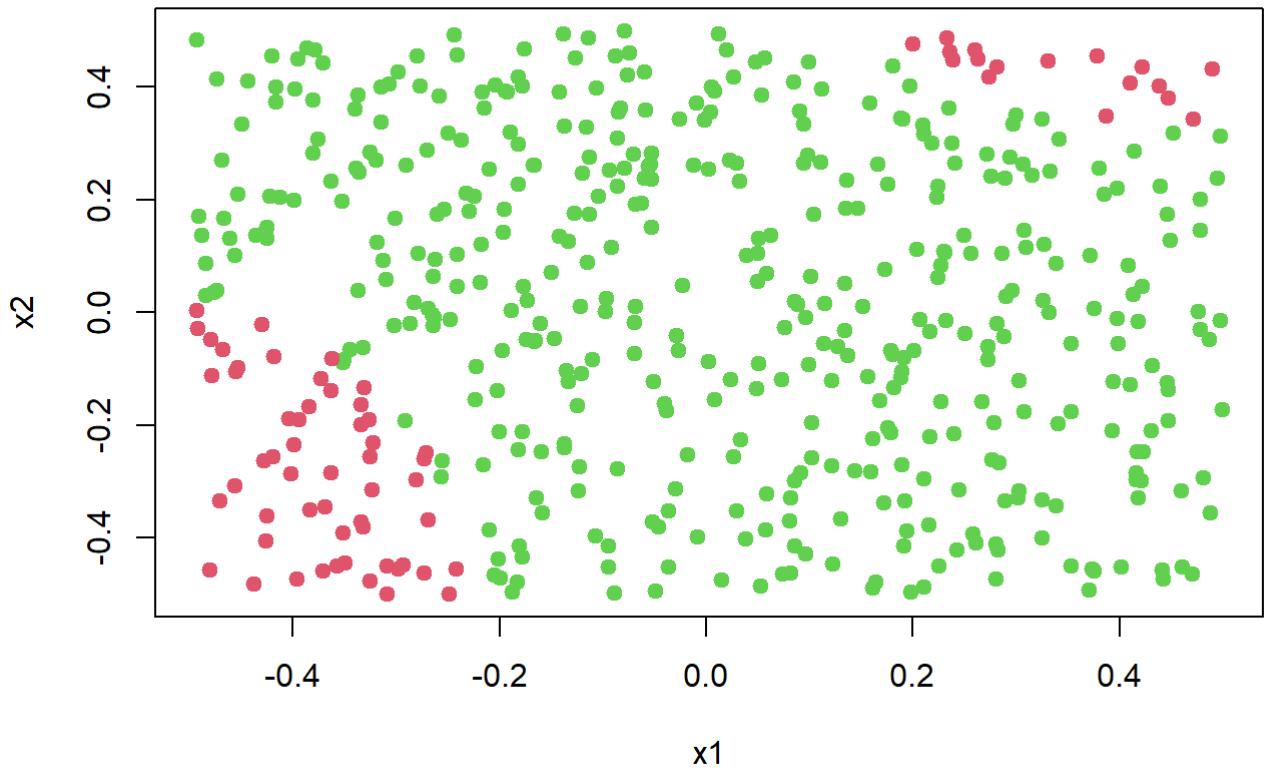
```
glm.probs1 <- predict(glm.fits1, newdata=data.frame(x1=x1,x2=x2), type = "response")
glm.labels1 <- as.numeric(glm.probs1>=0.5)
plot(x1, x2, col=(3-glm.labels1), pch=19, cex=1.05, xlab='x1', ylab='x2')
```



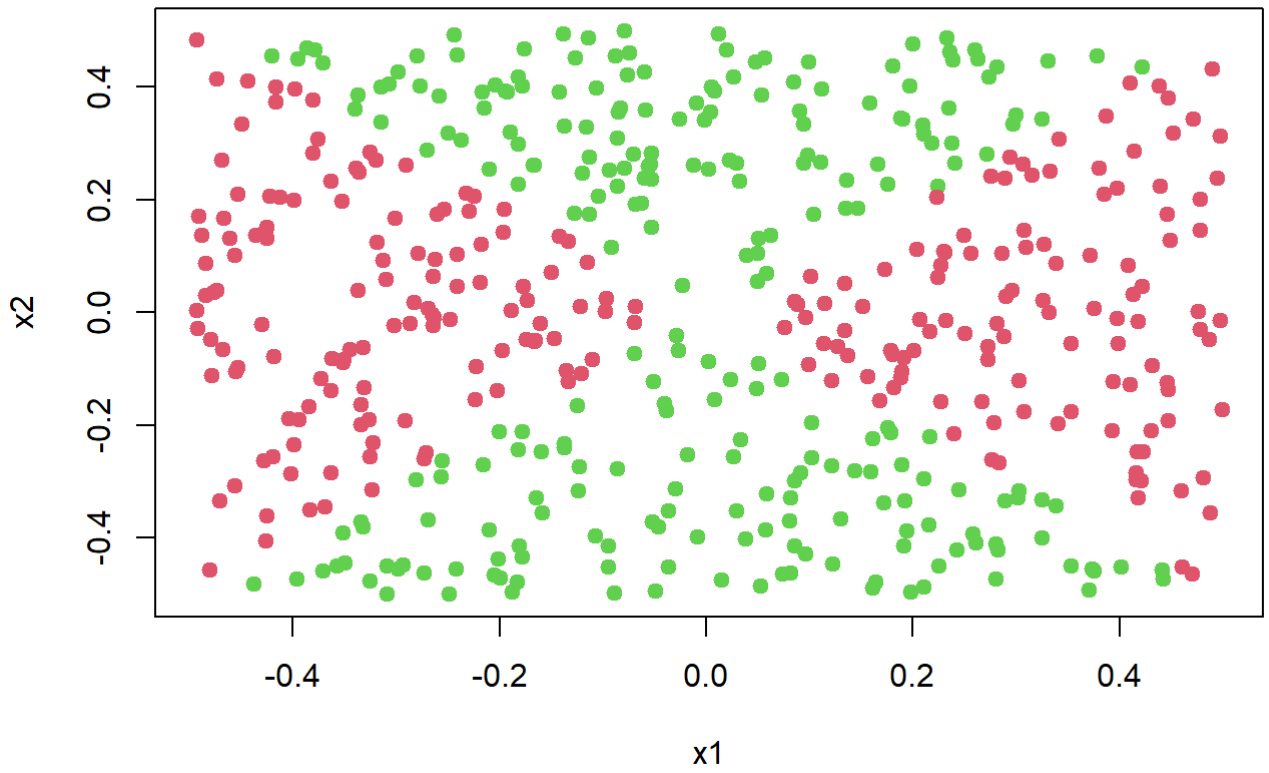

```
glm.probs2 <- predict(glm.fits2, newdata=data.frame(x1=x1,x2=x2), type = "response")
glm.labels2 <- as.numeric(glm.probs2>=0.5)
plot(x1, x2, col=(3-glm.labels2), pch=19, cex=1.05, xlab='x1', ylab='x2')
```




```
glm.probs3 <- predict(glm.fits3, newdata=data.frame(x1=x1,x2=x2), type = "response")
glm.labels3 <- as.numeric(glm.probs3>=0.5)
plot(x1, x2, col=(3-glm.labels3), pch=19, cex=1.05, xlab='x1', ylab='x2')
```




```
glm.probs4 <- predict(glm.fits4, newdata=data.frame(x1=x1,x2=x2), type = "response")
glm.labels4 <- as.numeric(glm.probs4>=0.5)
plot(x1, x2, col=(3-glm.labels4), pch=19, cex=1.05, xlab='x1', ylab='x2')
```

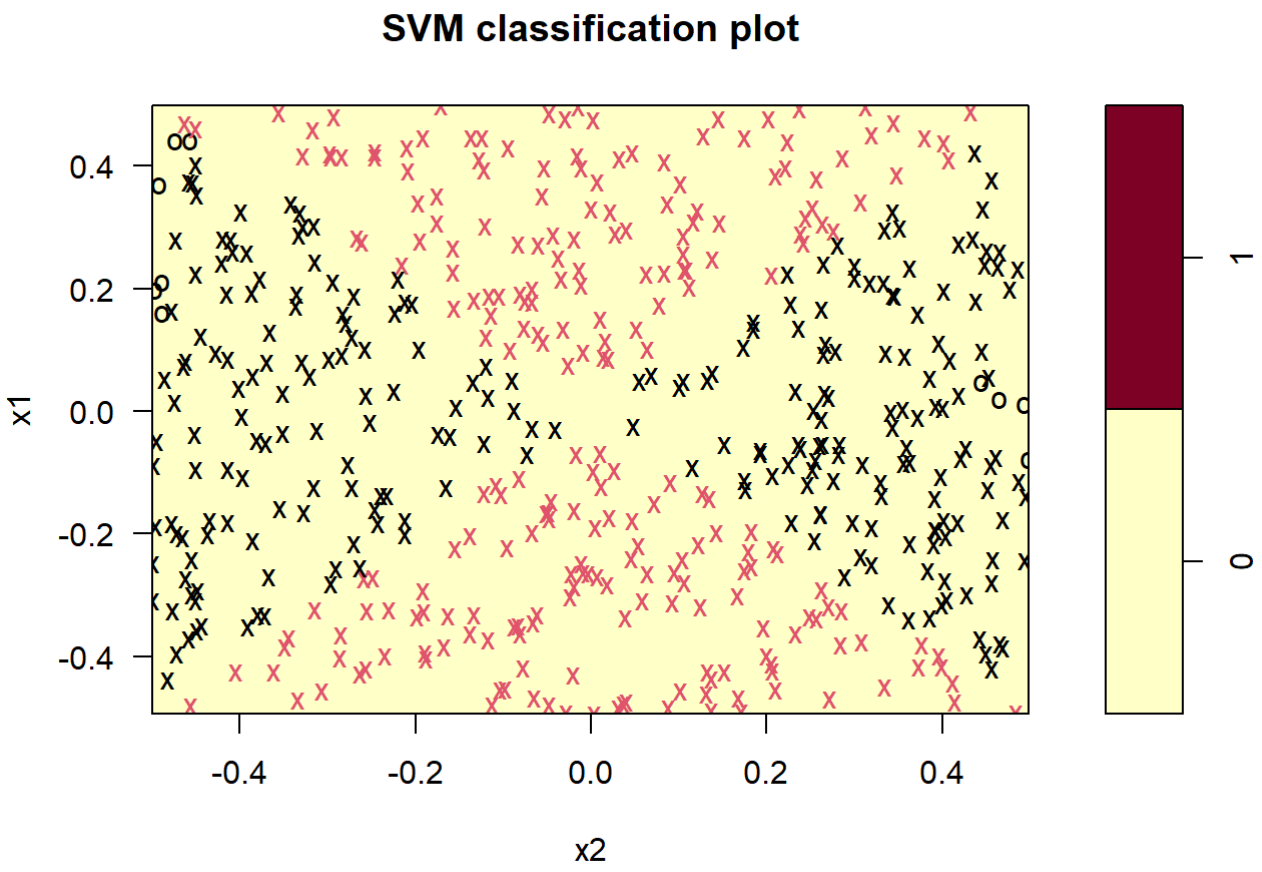


7. 用支持向量分类器拟合数据，并将X1和X2作为预测变量。获得每个训练观测值的分类预测。绘制观测值，并根据预测的类标签着色。

```
dat <- data.frame(x1=x1, x2=x2, y=as.factor(y))
tune.out <- tune(svm, y ~ ., data = dat,
    kernel = "linear",
    ranges = list(
      cost = c(0.01, 0.1, 1, 10, 100, 1000),
      gamma = c(0.5, 1, 2, 3, 4)
    )
  )
print(tune.out$best.model)
```

```
##
## Call:
## best.tune(METHOD = svm, train.x = y ~ ., data = dat, ranges = list(cost = c(0.01,
##     0.1, 1, 10, 100, 1000), gamma = c(0.5, 1, 2, 3, 4)), kernel = "linear")
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  linear
##        cost:  0.01
##
## Number of Support Vectors:  490
```

```
pred = predict(tune.out$best.model, newdata =
                 data.frame(x1=x1,x2=x2))
label = as.numeric(as.character(pred))
plot(tune.out$best.model, dat)
```

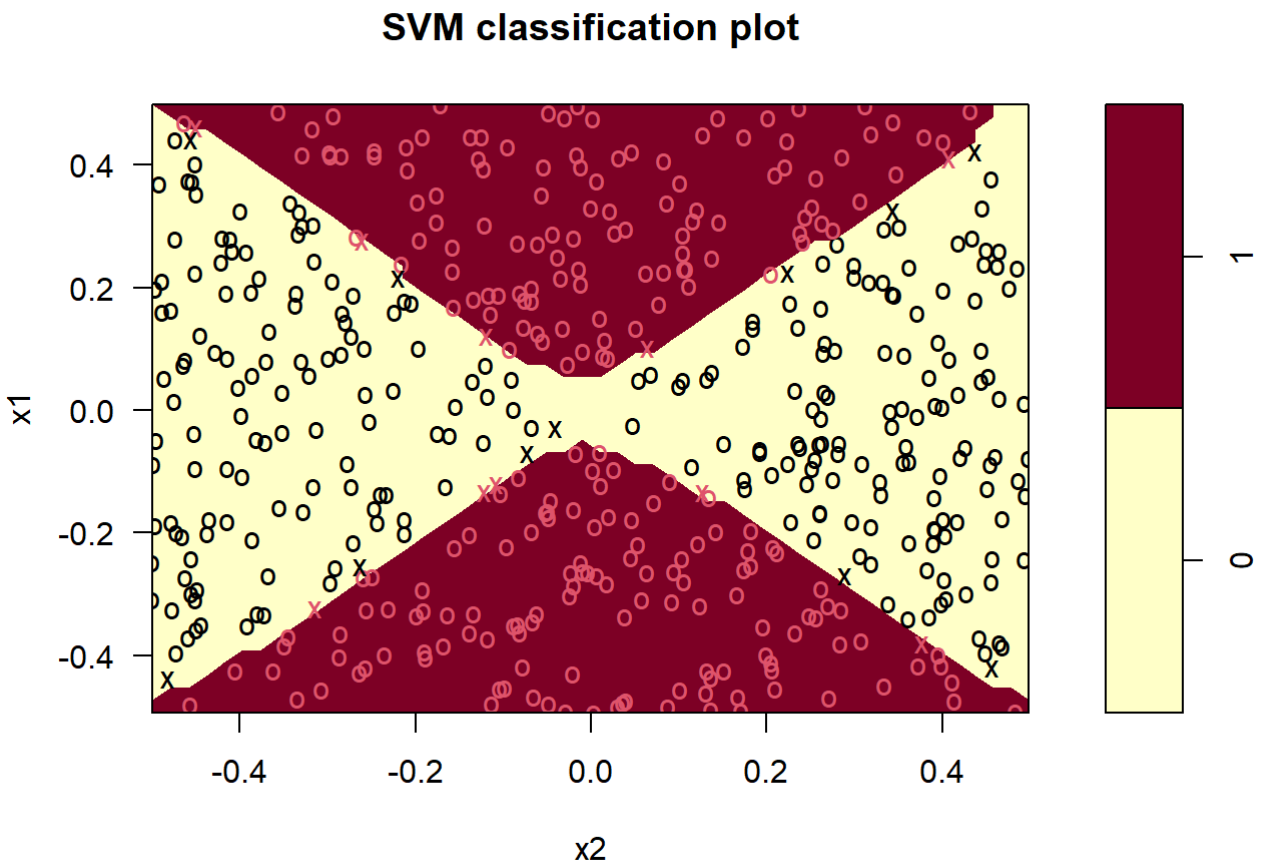


8. 使用非线性核的SVM拟合数据。获得每个训练观测值的分类预测。绘制观测值，并根据预测的类标签着色。

```
tune.out <- tune(svm, y ~ ., data = dat,
    kernel = "radial",
    ranges = list(
      cost = c(0.01, 0.1, 1, 10, 100, 1000),
      gamma = c(0.5, 1, 2, 3, 4)
    )
  )
print(tune.out$best.model)
```

```
##
## Call:
## best.tune(METHOD = svm, train.x = y ~ ., data = dat, ranges = list(cost = c(0.01,
##     0.1, 1, 10, 100, 1000), gamma = c(0.5, 1, 2, 3, 4)), kernel = "radial")
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  radial
##        cost:  1000
##
## Number of Support Vectors:  21
```

```
pred = predict(tune.out$best.model, newdata =
                 data.frame(x1=x1,x2=x2))
label = as.numeric(as.character(pred))
plot(tune.out$best.model, dat)
```



9. 描述你获得的结果。

非交互项逻辑回归和线性核支持向量机都无法找到决策边界。

将交互项添加到逻辑回归中可以找到非线性边界。

具有非线性核的支持向量机也可以找到非线性边界。