Support Vector Mechine

Consider the diabetes dataset. As there, we will take Outcome as the binary response, the remaining 8 variables as predictors, and all the data as training data. For all the models below, use 10-fold cross-validation to compute the estimated test error rates and also to tune any hyperparameter that requires tuning.

a) Support vector classifier was fitted with cost parameter chosen optimally via tuning. Misclassification rate using LOOCV is reported in Table 2.

- Optimal cost value is chosen as 4.6.
- Number of Support Vectors: 1051 (without diabities: 524 and with diabities: 527)

```
library(e1071)
library(ISLR)

diab<-read.csv("diabetes.csv")
head(diab)
```

```
  Pregnancies.. Glucose.. BloodPressure.. SkinThickness.. Insulin.. BMI..
1             2       138              62              35         0  33.6
2             0        84              82              31       125  38.2
3             0       145               0               0         0  44.2
4             0       135              68              42       250  42.3
5             1       139              62              41       480  40.7
6             0       173              78              32       265  46.5
  DiabetesPedigreeFunction.. Age.. Outcome
1                      0.127    47       1
2                      0.233    23       0
3                      0.630    31       1
4                      0.365    24       1
5                      0.536    21       0
6                      1.159    58       0
```

```
diab$Outcome<-as.factor(diab$Outcome)
names(diab)<-c("Pregnancies","Glucose","BP","Thickness","Insulin","BMI","DPB","Age","Outcome")
str(diab)
```

```
'data.frame':   2000 obs. of  9 variables:
 $ Pregnancies: int  2 0 0 0 1 0 4 8 2 2 ...
 $ Glucose    : int  138 84 145 135 139 173 99 194 83 89 ...
 $ BP         : int  62 82 0 68 62 78 72 80 65 90 ...
 $ Thickness  : int  35 31 0 42 41 32 17 0 28 30 ...
 $ Insulin    : int  0 125 0 250 480 265 0 0 66 0 ...
 $ BMI        : num  33.6 38.2 44.2 42.3 40.7 46.5 25.6 26.1 36.8 33.5 ...
 $ DPB        : num  0.127 0.233 0.63 0.365 0.536 ...
 $ Age        : int  47 23 31 24 21 58 28 67 24 42 ...
 $ Outcome    : Factor w/ 2 levels "0","1": 2 1 2 2 1 1 1 1 1 1 ...
```

```
# Calculating misclassification rate using 10 fold cv
set.seed(1)
svmfit <- svm(Outcome ~ ., data = diab, kernel = "linear", cost = 4.6, scale = TRUE,cross=10)
summary(svmfit)
```

```
Call:
svm(formula = Outcome ~ ., data = diab, kernel = "linear", cost = 4.6,
```

```
    cross = 10, scale = TRUE)


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  linear
       cost:  4.6

Number of Support Vectors:  1051

 ( 524 527 )


Number of Classes:  2

Levels:
 0 1

10-fold cross-validation on training data:

Total Accuracy: 77.45
Single Accuracies:
 80.5 77 81 81 75 76 77 79.5 75.5 72
```

b) Support vector machine with a polynomial kernel of degree two was fitted with cost parameter chosen optimally via tuning. Misclassification rate using LOOCV is reported in Table 2.

- Optimal cost value is chosen as 4.9.
- Number of Support Vectors: 1055 (without diabities: 527 and with diabities: 528)

```r
 # Fit SV classifiers and perform cross-validation (default: 10-fold CV)
set.seed(1)
tune.out.pol <- tune(svm, Outcome ~ ., data = diab, kernel = "polynomial",degree=2,
                 ranges = list(cost = c(4.6,4.7,4.8,4.9,5,5.1,5.2,5.3,5.5,6)), scale = TRUE)
summary(tune.out.pol$best.model)
```

```
Call:
best.tune(METHOD = svm, train.x = Outcome ~ ., data = diab, ranges = list(cost = c(4.6,
    4.7, 4.8, 4.9, 5, 5.1, 5.2, 5.3, 5.5, 6)), kernel = "polynomial",
    degree = 2, scale = TRUE)


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  polynomial
       cost:  4.9
     degree:  2
     coef.0:  0

Number of Support Vectors:  1217

 ( 600 617 )


Number of Classes:  2

Levels:
 0 1
```

```r
# Calculating misclassification rate using 10 fold cv
set.seed(1)
svmfit1 <- svm(Outcome ~ ., data = diab, kernel = "polynomial",degree=2, cost = 4.9, scale = TRUE,cross=10)
summary(svmfit1)
```

```
Call:
svm(formula = Outcome ~ ., data = diab, kernel = "polynomial", degree = 2,
    cost = 4.9, cross = 10, scale = TRUE)


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  polynomial
       cost:  4.9
     degree:  2
     coef.0:  0

Number of Support Vectors:  1217

 ( 600 617 )


Number of Classes:  2

Levels:
 0 1

10-fold cross-validation on training data:

Total Accuracy: 73.1
Single Accuracies:
 73.5 72 71.5 74.5 74.5 71.5 74.5 74 71 74
```

c) Support vector machine with radial kernel was fitted with cost parameter chosen optimally via tuning. Misclassification rate using LOOCV is reported in Table 2.

- Optimal γ value is chosen as 4.5 as for any cost value $\gamma = 4.5$ gives the lowest misclassification rate.
- Optimal cost value is chosen as 0.8.
- Number of Support Vectors: 1024 (without diabities: 507 and with diabities: 517)

```r
set.seed(1)
tune.out.rad <- tune(svm, Outcome ~ ., data = diab, kernel = "radial",
    ranges = list(cost = c(0.5,0.6,0.7,0.8,0.9,1,1.1,1.2), gamma = 4.5))
summary(tune.out.rad$best.model)
```

```
Call:
best.tune(METHOD = svm, train.x = Outcome ~ ., data = diab, ranges = list(cost = c(0.5,
    0.6, 0.7, 0.8, 0.9, 1, 1.1, 1.2), gamma = 4.5), kernel = "radial")


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  radial
       cost:  0.8

Number of Support Vectors:  1024
```

```
 ( 507 517 )


Number of Classes:  2

Levels:
 0 1
```

```r
svmfit2 <- svm(Outcome ~ ., data = diab, kernel = "radial", cost = 0.8, gamma = 4.5, cross=10)
summary(svmfit2)
```

```
Call:
svm(formula = Outcome ~ ., data = diab, kernel = "radial", cost = 0.8,
    gamma = 4.5, cross = 10)


Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  radial
       cost:  0.8

Number of Support Vectors:  1024

 ( 507 517 )


Number of Classes:  2

Levels:
 0 1

10-fold cross-validation on training data:

Total Accuracy: 98.8
Single Accuracies:
 99 98.5 99 99 100 96.5 100 97.5 99.5 99
```

d) Support vector machine with radial kernel gives the lowest misclassification rate and support vector machine with polynomial gives the highest misclassification rate. Therfore I would recommend Support vector machine with radial kernel. According to the previous project I recommended knn with k=6 as the best model as it had the lowest misclassification rate of 0.1665. However Support vector machine with radial kernel seems better than knn as it has the lower misclassification rate. Therefore, I would recommend Support vector machine with radial kernel.

	svm linear	svm polynomial	svm radial
misclassification rate	0.2255	0.2690	0.012

Table 1: *Summary of misclassification*