Chapter 9 Support Vector Machines problems 4

Kartheek Raj

12/28/2019

4. Generate a simulated two-class data set with 100 observations and two features in which there is a visible but non-linear separation between the two classes. Show that in this setting, a support vector machine with a polynomial kernel (with degree greater than 1) or a radial kernel will outperform a support vector classifier on the training data. Which technique performs best on the test data? Make plots and report training and test error rates in order to back up your assertions.

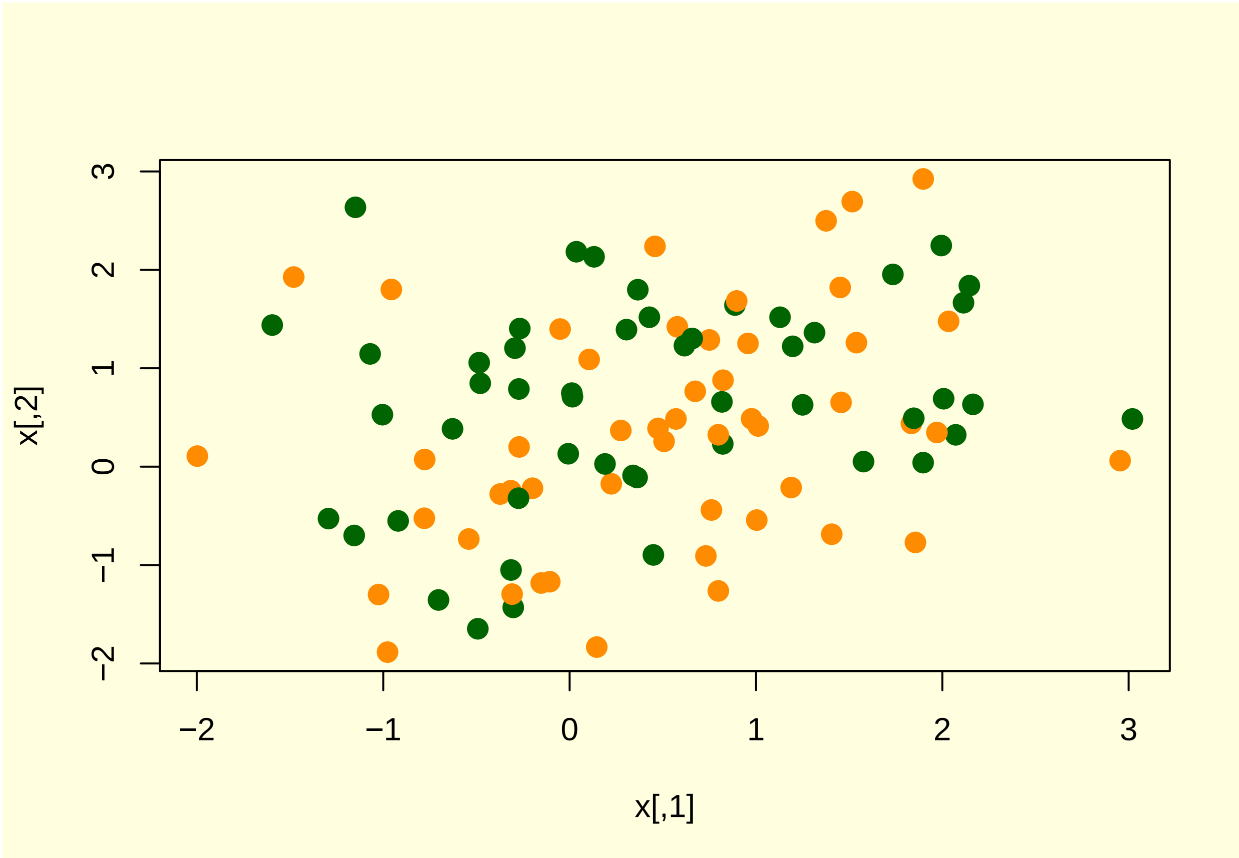
Answer Required Packages: e1071

Generating the 100 observations and plotting those data points

```
set.seed(12)
par(mfrow=c(1,1),bg="lightyellow")
x<-matrix(rnorm (100*2), ncol=2)
length(x)
```

```
## [1] 200
```

```
y<-c(rep(-1,50), rep(1,50))
x[y==1,]<-x[y==1,] + 1
plot(x, col=c("darkorange","darkgreen"),type="p",cex=1.5,pch = 16)
```



train and test data split for support vector classifiers with various types

```
dat<-data.frame(x=x, y=as.factor(y))
colnames(dat)<-c("x1","x2","y")
sd<-sample(nrow(dat),0.60*nrow(dat),replace = F)
train<-dat[sd,]
test<-dat[-sd,]
```

Applying svm() to train data to build various svm classifiers and summaries of those

```
library(e1071)
svmfit.poly<-svm(y~., data=train , kernel ="polynomial", cost=10, scale=F,degree = 2)
svmfit.lin<-svm(y~., data=train , kernel ="linear", cost=10, scale=F)
svmfit.rad<-svm(y~., data=train , kernel ="radial", cost=10, scale=F)
summary(svmfit.poly)
```

```
##
## Call:
## svm(formula = y ~ ., data = train, kernel = "polynomial", cost = 10,
##     degree = 2, scale = F)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  polynomial
##        cost:  10
##      degree:  2
##      coef.0:  0
```

```
##
## Number of Support Vectors:  40
##
##  ( 20 20 )
##
##
## Number of Classes:  2
##
## Levels:
##  -1 1
```

```r
summary(svmfit.lin)
```

```
##
## Call:
## svm(formula = y ~ ., data = train, kernel = "linear", cost = 10,
##     scale = F)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  linear
##        cost:  10
##
## Number of Support Vectors:  28
##
##  ( 14 14 )
##
##
## Number of Classes:  2
##
## Levels:
##  -1 1
```

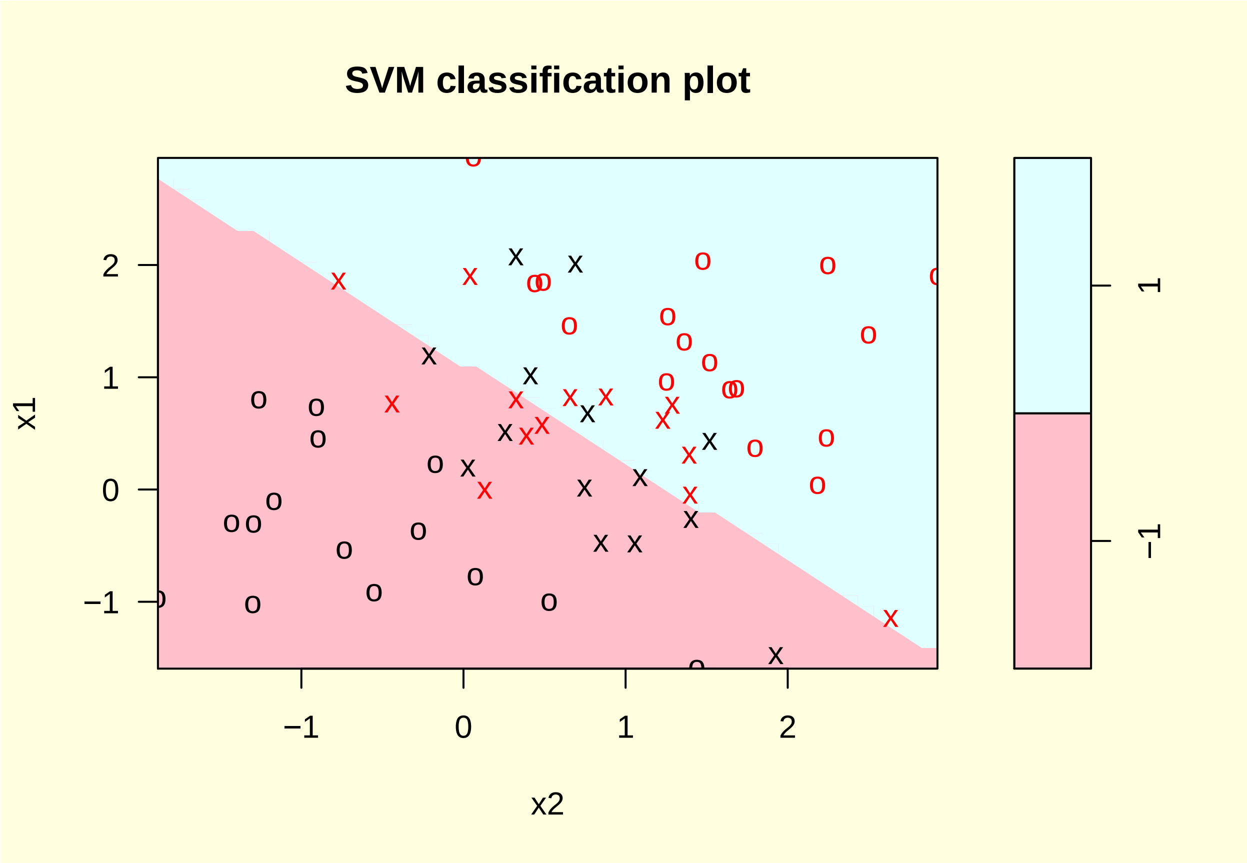
```r
summary(svmfit.rad)
```

```
##
## Call:
## svm(formula = y ~ ., data = train, kernel = "radial", cost = 10,
##     scale = F)
##
##
## Parameters:
##    SVM-Type:  C-classification
##  SVM-Kernel:  radial
##        cost:  10
##
## Number of Support Vectors:  33
##
##  ( 18 15 )
##
##
## Number of Classes:  2
##
## Levels:
```

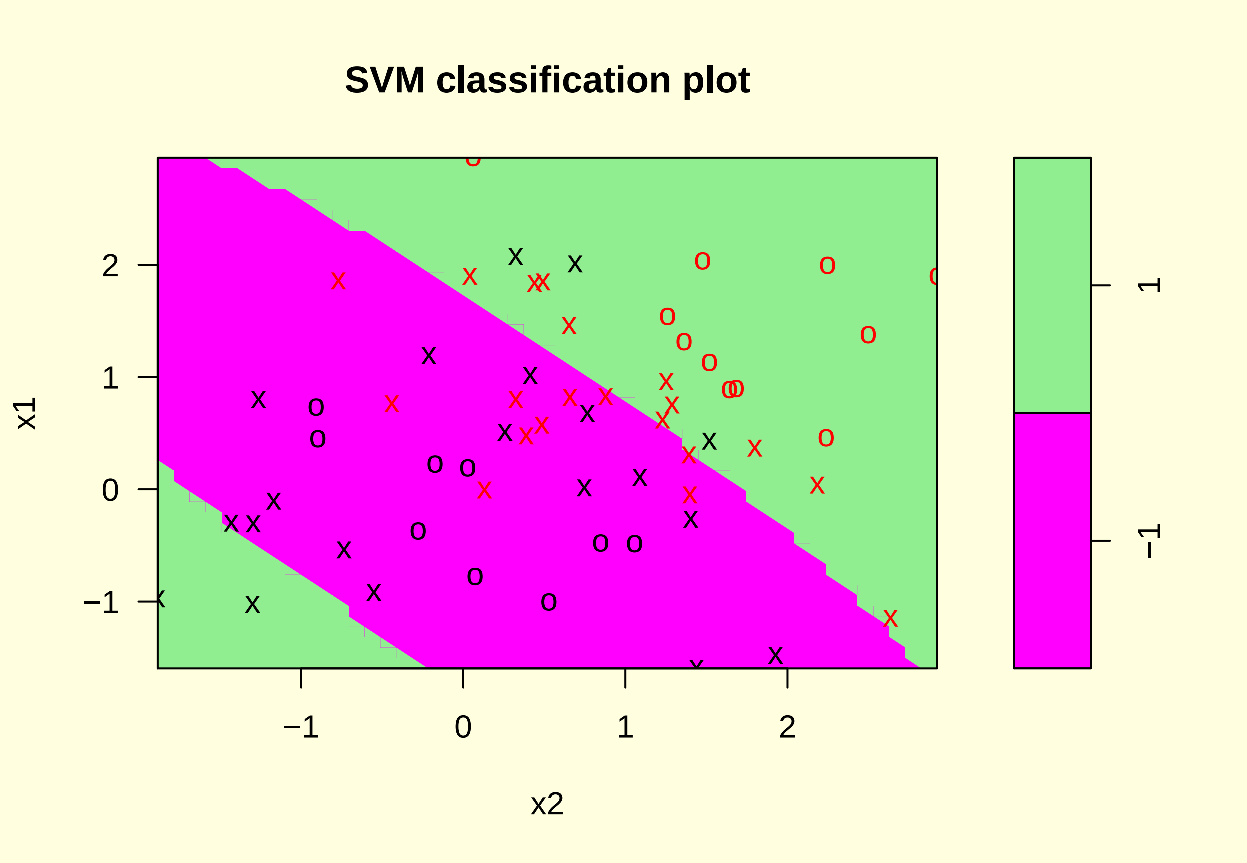
```
##   -1 1
```

Plotting svm classifiers previously build.

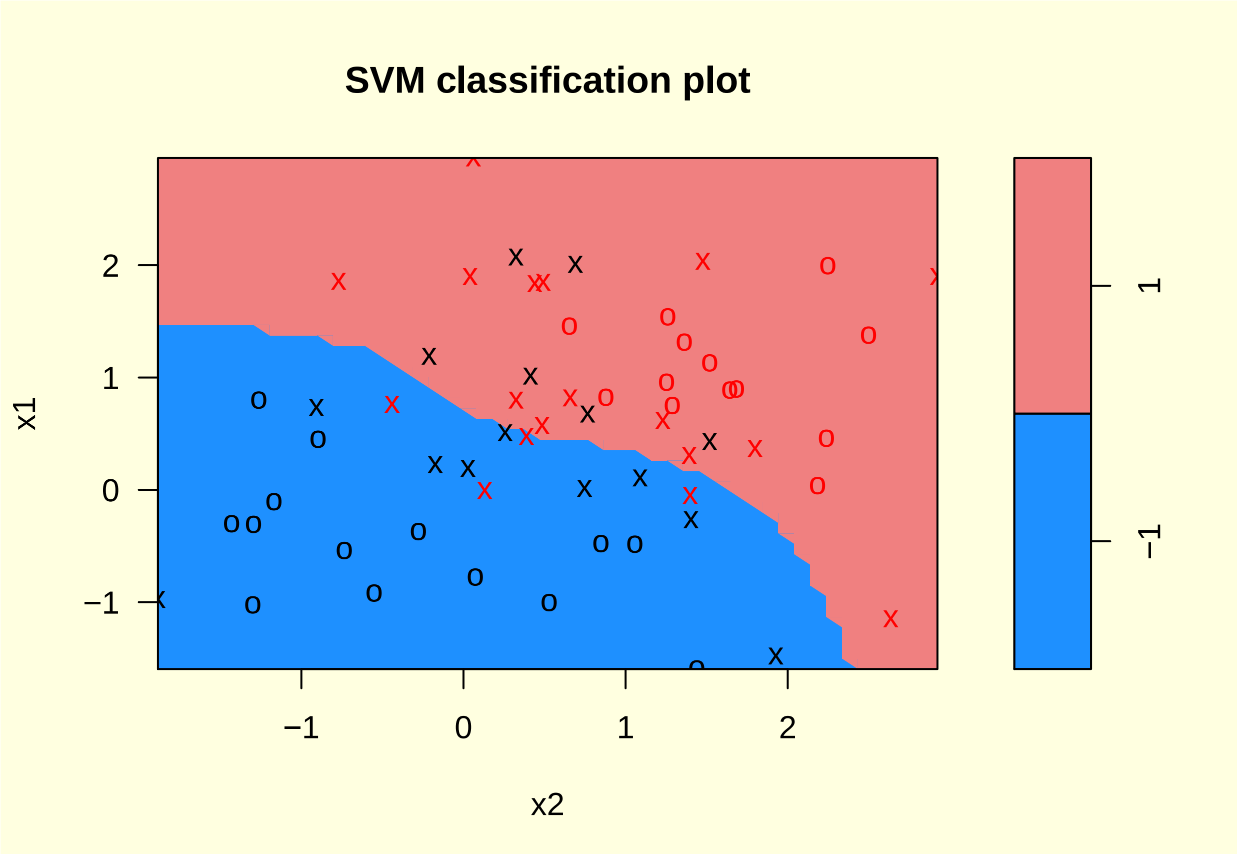
```
par(mfrow=c(1,1),bg="lightyellow")
library(e1071)
plot(svmfit.lin,data=train,col=c("pink","lightcyan"),cex=1.5)
```




```
plot(svmfit.poly,data=train,col=c("magenta","lightgreen"),cex=1.5)
```




```
plot(svmfit.rad,data=train,col=c("dodgerblue","lightcoral"),cex=1.5)
```



Predicting on the test data using previously build svm classifiers

```
pre.poly<-predict(svmfit.poly,test)
pre.lin<-predict(svmfit.lin,test)
pre.rad<-predict(svmfit.rad,test)
```

Test error rate of SVM Classifiers

```
paste0("Polynomial SVM test error rate is ",mean(pre.poly!=test$y))
```

```
## [1] "Polynomial SVM test error rate is 0.4"
```

```
paste0("Linear SVM test error rate is ",mean(pre.lin!=test$y))
```

```
## [1] "Linear SVM test error rate is 0.25"
```

```
paste0("Radial SVM test error rate is ",mean(pre.rad!=test$y))
```

```
## [1] "Radial SVM test error rate is 0.2"
```

Train error rate of svm classifiers, model trained on the train data and predict on the same.

```
pre.polyt<-predict(svmfit.poly,train)
pre.lint<-predict(svmfit.lin,train)
pre.radt<-predict(svmfit.rad,train)

paste0("Polynomial SVM train error rate is ",mean(pre.polyt!=train$y))
```

```
## [1] "Polynomial SVM train error rate is 0.25"
```

```r
paste0("Linear SVM train error rate is ",mean(pre.lint!=train$y))
```

```
## [1] "Linear SVM train error rate is 0.15"
```

```r
paste0("Radial SVM train error rate is ",mean(pre.radt!=train$y))
```

```
## [1] "Radial SVM train error rate is 0.15"
```

Insight : Radial SVM Classifiers yields better predictions. Linear models yields close to radial classifiers.