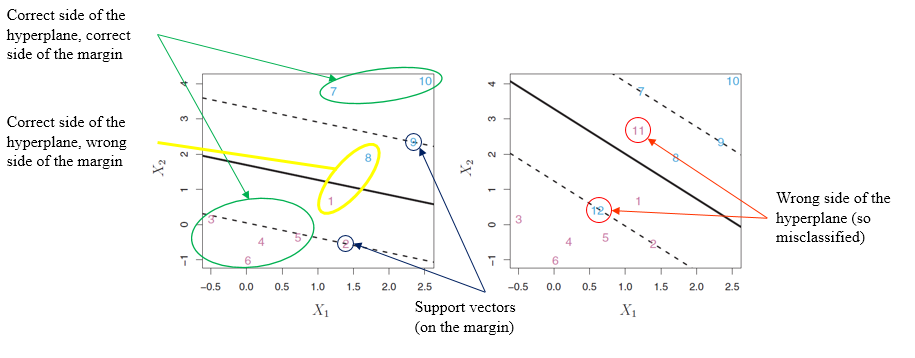
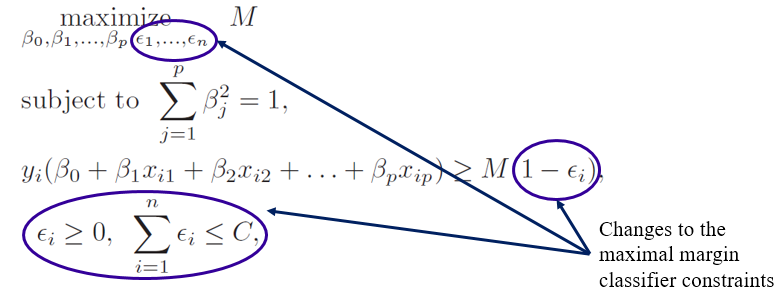
# Support Vector Classifiers

* Advantages of a classifier that is based on a hyperplane that does *not perfectly* separate the two classes:
  + Greater robustness to individual observations
  + Better classification of *most* of the training observations
* A **support vector classifier** does not seek the largest possible margin so that every observation is both on the correct side of the hyperplane and also on the correct side of the margin, but instead allows some observations to be on the incorrect side of the margin, or even the incorrect side of the hyperplane (thus the term “soft margin classifier”).   
  
* To find the desired hyperplane, we solve the following optimization problem**:**
* *C* is a nonnegative tuning parameter (more later)
* The are *slack variables* that allow individual observations to be on the wrong side of the margin or the hyperplane.
* As before, we classify the test observation based on the sign of
* Since *C* bounds the sum of the ’s, it determines the number and severity of the violations to the margin (and to the hyperplane) that we will tolerate. This is a tuning parameter that we must determine, typically with cross validation.
* We can think of *C* as a *budget* for the amount that the margin can be violated by the *n* observations.
  + As *C* increases, we become more tolerant of violations to the margin, and so the margin will widen.
  + As *C* decreases, we become less tolerant of violations to the margin and so the margin narrows.
* Observations that lie on the margin, or on the wrong side of the margin for their class, are known as *support vectors*. These are the only observations that affect the support vector classifier.
* This implies that *C* controls the bias-variance trade-off of the support vector classifier.
* When *C* is large, the margin is wide, many observations violate the margin, and so there are many support vectors, so many observations are involved in determining the hyperplane and variance is low, and conversely.
* **R (just the important stuff)**

rm(list=ls())

require(e1071)

set.seed(5082)

# Build toy data that is not perfectly separable

x <- matrix(rnorm(2000 \* 2), ncol=2)

y <- c(rep(-1, 1000), rep(1, 1000))

x[y == 1,] <- x[y == 1,] + 2

plot(x[, 2], x[, 1], col=(3 - y))

# Create a dataframe with the response variable   
# converted to a factor to indicate classification

dat <- data.frame(x=x, y=as.factor(y))

# Use cross validation to find best Cost

tune.out <- tune(svm,y ~ ., data=dat,

kernel="linear",

ranges=list(cost=c(0.001, 0.01, 0.1, 1, 5)))

summary(tune.out)

bestmod <- tune.out$best.model

summary(bestmod)

plot(bestmod, dat)

# Use predict() function to predict the class

# label on a set of test observations.

xtest <- matrix(rnorm(200 \* 2), ncol=2)

ytest <- sample(c(-1, 1), 200, rep=TRUE)

xtest[ytest==1,]=xtest[ytest==1,] + 2

testdat <- data.frame(x=xtest, y=as.factor(ytest))

ypred <- predict(bestmod, testdat)

my.table <- table(actual=testdat$y, predict=ypred)

print(paste("The test error rate is", mean(testdat$y != ypred)))