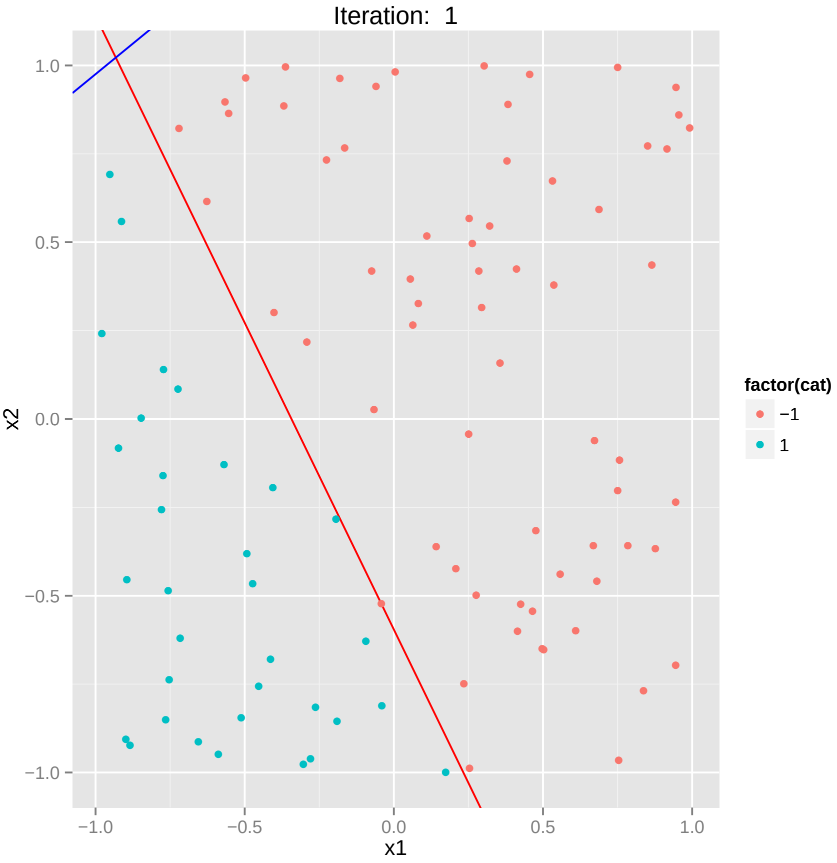
Perceptron Learning Algorithm

Segev Malool

December 23, 2015

The following code implements a version of the perceptron learning algorithm. To start the animation, click on it!



```
library(ggplot2)
set.seed(982)

pdf(file="perceptronTraining.pdf")
cumsum <- 0
cumerr <- 0
for (i in 1:10) {
n=100 ##number of training samples taken

##Target function:
x1 <- runif(n=2,min=-1,max=1)
x2 <- runif(n=2,min=-1,max=1)
slope <- (x2[2]-x1[2])/(x2[1]-x1[1])
```

```
intercept <- x1[2]-slope*x1[1]

y <- function(x){sign(slope*x[1]+intercept - x[2])}

##Simulated dataset
data <- data.frame(x1=runif(n=n,min=-1,max=1),
                   x2=runif(n=n,min=-1,max=1))
data$cat <- apply(data,1,y)
data$x0 <- rep(1,n)
data <- data[c(4,1,2,3)]


##Weight vector and initial hypotheses initialized to zero
w <- c(0,0,0)
data$HypCat <- rep(0,n)

##Train the perceptron:
miss <- data[data[,4] != data[,5],] ##misclassified points
iteration <- 1 ##to keep track of iterations

##training algorithm samples randomly from misclassified points and adds or subtracts the point to the
while( 1 ){
    miss <- data[data[,4] != data[,5],]
    if ( nrow(miss) == 0 ) {break}

    x <- as.numeric(rownames(miss))
    s <- ifelse(nrow(miss)==1, as.numeric(rownames(miss)), sample(x, size=1))

    if ( miss[as.character(s),4] == 1 ) {##condition on true category
        w <- w + miss[as.character(s),1:3] ##update weight vector
    }
    if ( miss[as.character(s),4] == -1 ) {
        w <- w - miss[as.character(s),1:3]
    }
    for ( i in 1:nrow(data) ) {
        data[i,5] <- sign(unlist(w)%*%unlist(data[i,1:3]))
    }

    x <- ggplot(data) +
        geom_abline(slope=slope,
                    intercept=intercept,
                    colour="red") +
        geom_abline(slope=-unlist(w[2])/unlist(w[3]),
                    intercept=-unlist(w[1])/unlist(w[3]),
                    colour="blue") +
        geom_point(aes(x1,x2,color=factor(cat))) +
        labs(title=paste("Iteration: ",iteration))

   print(x)
    iteration <- iteration + 1
}
cumsum <- cumsum + iteration
```

```
yhyp <- function(x){sign(-unlist(w[2])/unlist(w[3])*x[1]-unlist(w[1])/unlist(w[3]) - x[2])}

pred <- data.frame(x1=runif(n=1000,min=-1,max=1),
                   x2=runif(n=1000,min=-1,max=1))
pred$TrCat <- apply(pred,1,y)
pred$HyCat <- apply(pred,1,yhyp)
err <- sum(pred$TrCat!=pred$HyCat)
cumerr <- sum(cumerr, err)
}
cumerr/(1000*10)

## [1] 0.0219

cumsum/10

## [1] 135.8

dev.off()

## pdf
##   2
```