Homework 1 - Machine Learning

Maximilian Arrich, Florian Benkhalifa, Raphael Eigenmann

28 Februar 2019

Exercise 1

Autonomous weapons have been described as the third revolution in warfare, as autonomous weapons introduce a completely new dimension of possibilities on the battlefield, which are also accompanied by a high risk of abusion. While Artificial intelligence lowers the necessity to put a human life at risk, it also lowers the threshold to take part in a war. Clearly, this creates major imbalances between fractions with autonomous weapons versus fractions with old-fashioned techniques. A global AI arms race is probably the result of such intimidating imbalances. What makes abusion of AI so likely is that software is not costly to produce, easy to trade between counterparties on the official as well as on the black market, and once acquired it is very easy to handle. Even terrorist groups or small regimes will bet able to buy and use such a technology for assassinations or terror attacks. Because of the possible backlashes of abusive suppresion and possibly disastrous consequences, autonomous weapons could result in a public backlash against AI, which could come at the cost of tarnishing peaceful and societally beneficial applications.

Another danger of machine learning is its usage in the car industry. In a self-driving car, a software is making the decisions over the consequences of an accident (whether to hit one victim or the other). Apart from the ethically controversial question of how to program such a software, it will also be hard to later find the culprit of an accident and thus to draw legal consequences. Furthermore, any software can be hacked and hence criminals might find a way to use self-driving cars as weapons against civilians.

Exercise 2

The given functions are:

1. quadratic
2. linear
3. none of the two
4. quadratic
5. none
6. linear

Exercise 3

We prepare the working space:

```r
# Clean workspace and load packages
rm(list = ls())
library(MASS)
library(scales)

# Loading data
data(Boston)
attach(Boston) # to save "ink"
```

Part 1: Built ind lm on "crim"

```r
lm(crim~medv) # lm command
```

```
##
## Call:
## lm(formula = crim ~ medv)
##
## Coefficients:
## (Intercept)         medv
##     11.7965      -0.3632
```

Part 2: Built ind lm on "medv"

```r
lm(medv~crim) # lm command
```

```
##
## Call:
## lm(formula = medv ~ crim)
##
## Coefficients:
## (Intercept)         crim
##     24.0331      -0.4152
```

Part 3.1: Mean and Variances

```r
mean(crim) # mean function
```

```
## [1] 3.613524
```

```r
var(crim)  # variance function
```

```
## [1] 73.98658
```

```r
mean(medv)
```

```
## [1] 22.53281
```

```r
var(medv)
```

```
## [1] 84.58672
```

Part 3.2: Verification of lm function

```r
# Part 3.2: Verification of lm function

# creating linear regression function manually
lm_fun <- function(x,y){
```

```r
  x_bar <- mean(x)                                # mean of x
  y_bar <- mean(y)                                # mean of y
  coef <- sum((x-x_bar)*(y-y_bar))/
            (var(x)*(length(x)-1))         # computation beta coefficient with mean and var function
  intercept <- y_bar-coef*x_bar            # computation intercept
  mat <- matrix(c(intercept, coef), nrow=2,  # organizing output in matrix
         dimnames = list(c("Intercept", "Coefficient")))
  print(mat)
}


lm_fun(x=crim, y=medv) # coherent with the results from lm command
```

```
##                   [,1]
## Intercept   24.0331062
## Coefficient -0.4151903
```

```r
lm_fun(x=medv, y=crim) # coherent with the results from lm command
```

```
##                   [,1]
## Intercept   11.7965358
## Coefficient -0.3631599
```