Introduction to Machine Learning

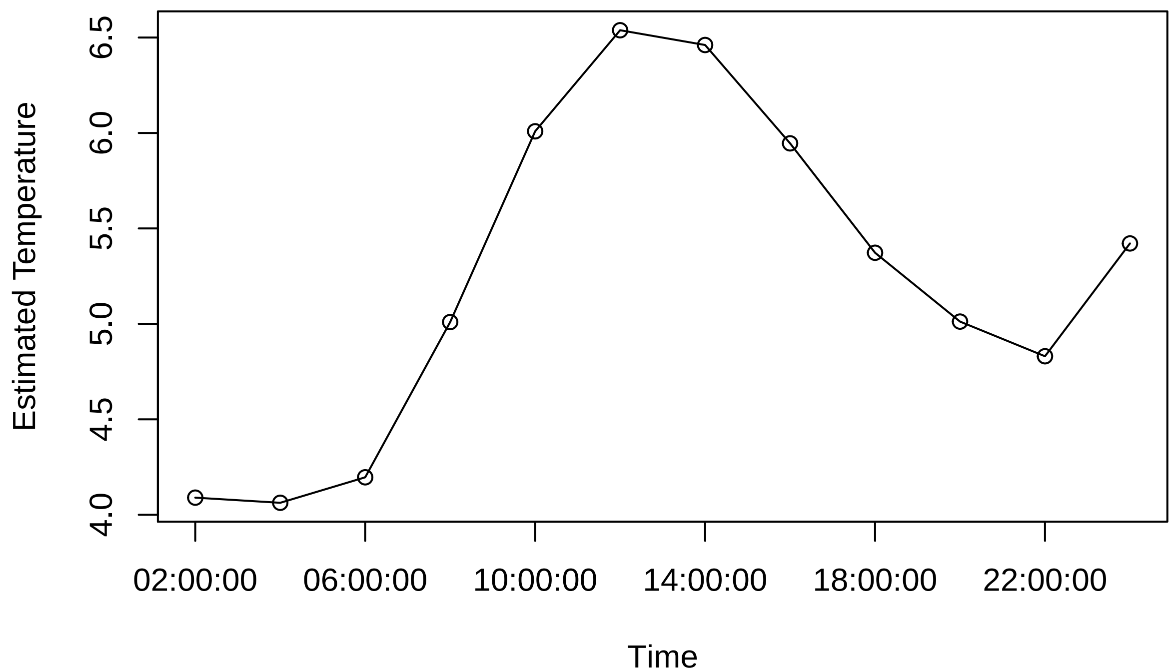
Lab 5

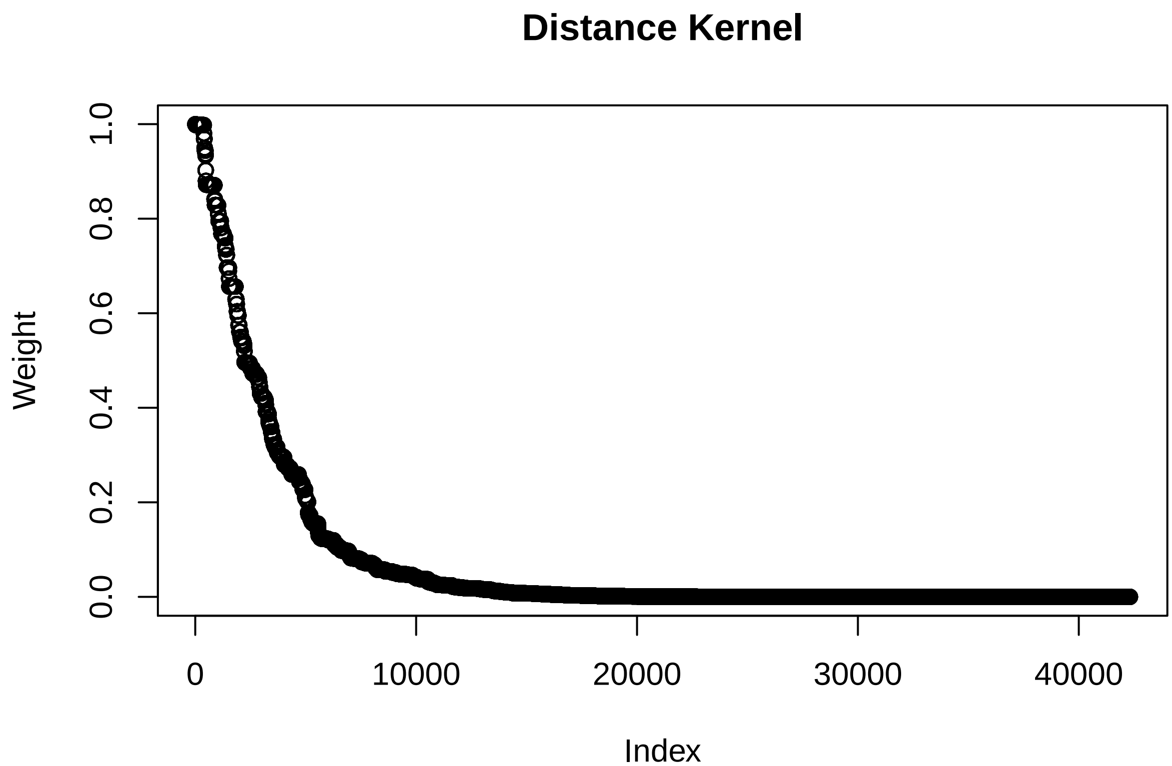
Rasmus Holm

2016-12-09

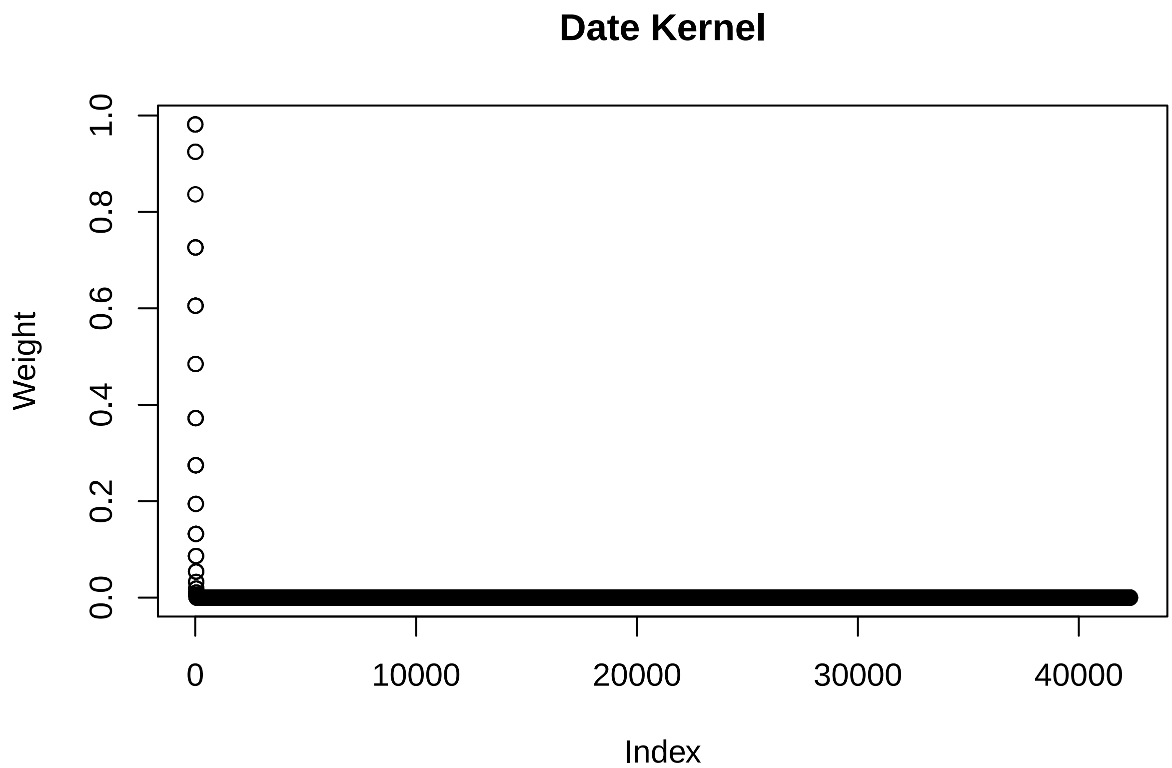
Contents

Assignment 1

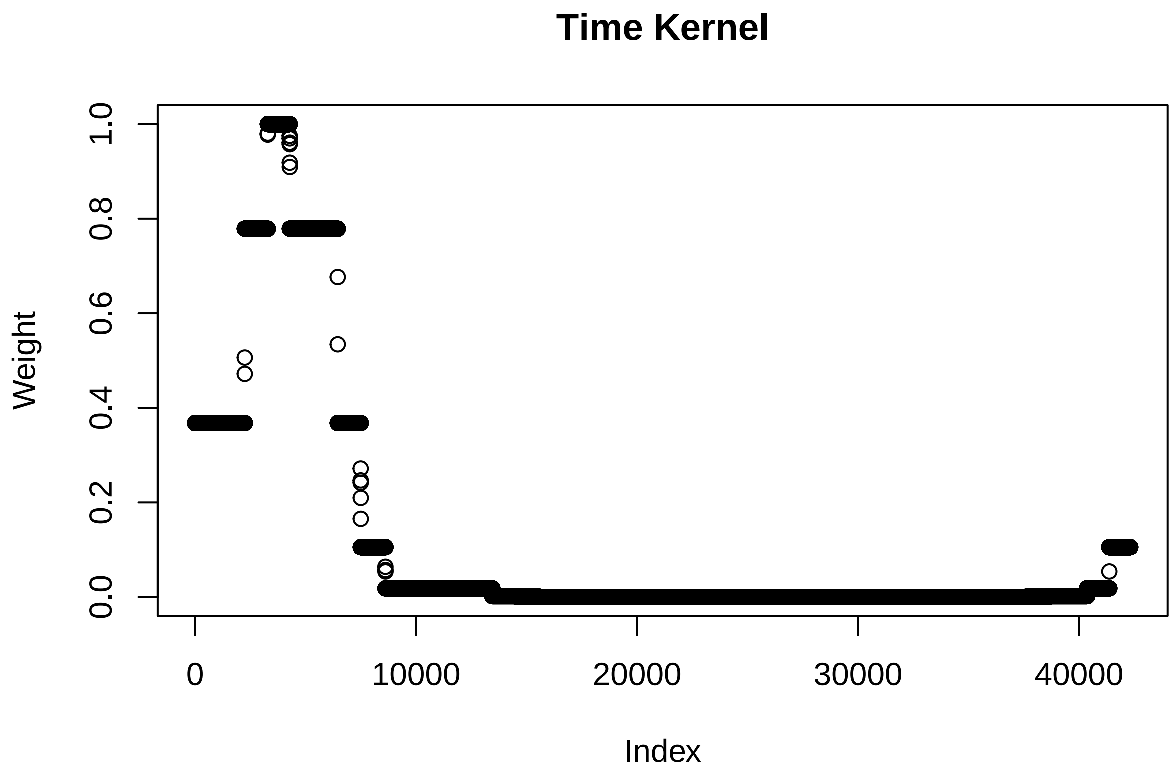




Distance Kernel
Weight
0.0
0.2
0.4
0.6
0.8
1.0
0
10000
20000
30000
40000
Index



Date Kernel
Weight
1.0
0.8
0.6
0.4
0.2
0.0
0
10000
20000
30000
40000
Index


Time Kernel
Weight
0.0
0.2
0.4
0.6
0.8
1.0
Index
0
10000
20000
30000
40000

Appendix

Code for Assignment 1

```
library(geosphere)

set.seed(1234567890)

stations <- read.csv("../data/stations.csv",
                     stringsAsFactors=FALSE,
                     fileEncoding="latin1")
temps <- read.csv("../data/temps50k.csv", stringsAsFactors=FALSE)

st <- merge(stations, temps, by="station_number")
data <- st[, c("longitude", "latitude", "date", "time", "air_temperature")]

gaussian.kernel <- function(u) {
    exp(-u^2)
}

distance.kernel <- function(X, lat, long, h) {
    distances <- distHaversine(X[, c("longitude", "latitude")],
                               c(long, lat))
    gaussian.kernel(distances / h)
}

## distance.pred(st_filtered, pred_latitude, pred_longitude, h_distance)

date.kernel <- function(X, date, h) {
    distances <- as.numeric(difftime(X$date, date, units="days"))
    gaussian.kernel(distances / h)
}

## date.pred(st_filtered, pred_date, h_date)

time.kernel <- function(X, time, h) {
    distances <- as.numeric(difftime(X$time, time, units="hours"))
    distances[distances > 12] <- 24 - distances[distances > 12]
    gaussian.kernel(distances / h)
}

## time.pred(st_filtered, pred_times[1], h_time)

filter_by_date <- function(X, date, time) {
    complete_dates <- paste(X$date, X$time)
    complete_dates <- as.POSIXct(complete_dates, format="%Y-%m-%d %H:%M:%S")

    complete_date <- paste(date, time)
    complete_date <- as.POSIXct(complete_date, format="%Y-%m-%d %H:%M:%S")

    idx <- which(complete_dates <= complete_date)

    X <- X[idx,]
```

```r
    X$time <- as.POSIXct(X$time, format="%H:%M:%S")
    X$date <- as.Date(X$date)
    X
}

kernel.model <- function(X, lat, long, h_dist, date, h_date, time, h_time) {
    X <- filter_by_date(X, date, time)
    date <- as.Date(date)
    time <- as.POSIXct(time, format="%H:%M:%S")
    kernel <- (distance.kernel(X, lat, long, h_dist) +
               date.kernel(X, date, h_date) +
               time.kernel(X, time, h_time))
    sum(kernel * X$air_temperature) / sum(kernel)
}
h_distance <- 100000 # These three values are up to the students
h_date <- 7
h_time <- 2

pred_latitude <- 58.409158 # The point to predict (up to the students)
pred_longitude <- 15.607452
pred_date <- "2010-12-24" # The date to predict (up to the students)
pred_times <- c("02:00:00", "04:00:00", "06:00:00", "08:00:00",
                "10:00:00", "12:00:00", "14:00:00", "16:00:00",
                "18:00:00", "20:00:00", "22:00:00", "24:00:00")

pred_temp <- vector(length=length(pred_times))

for (i in 1:length(pred_times)) {
    pred_temp[i] <- kernel.model(data, pred_latitude, pred_longitude, h_distance,
                                 pred_date, h_date, pred_times[i], h_time)
}

x_breaks <- seq(1, length(pred_times), 2)
plot(y=pred_temp, x=1:length(pred_times), type="o", xaxt = "n",
     xlab="Time", ylab="Estimated Temperature")
axis(1, at=x_breaks, labels=pred_times[x_breaks])
```