Introduction to Machine Learning

Lab 5

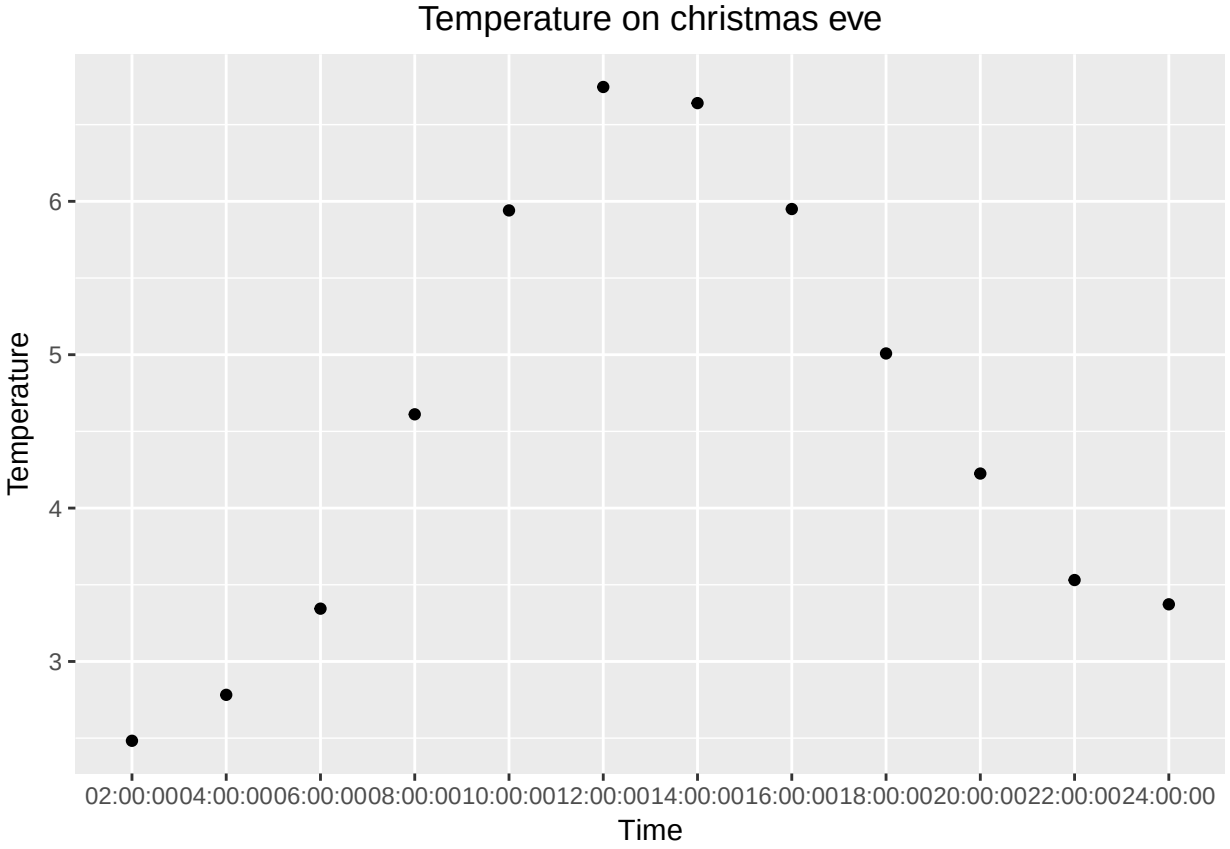
Anton Persson, Emil Klasson Svensson, Mattias Karlsson, Rasmus Holm

2016-12-12

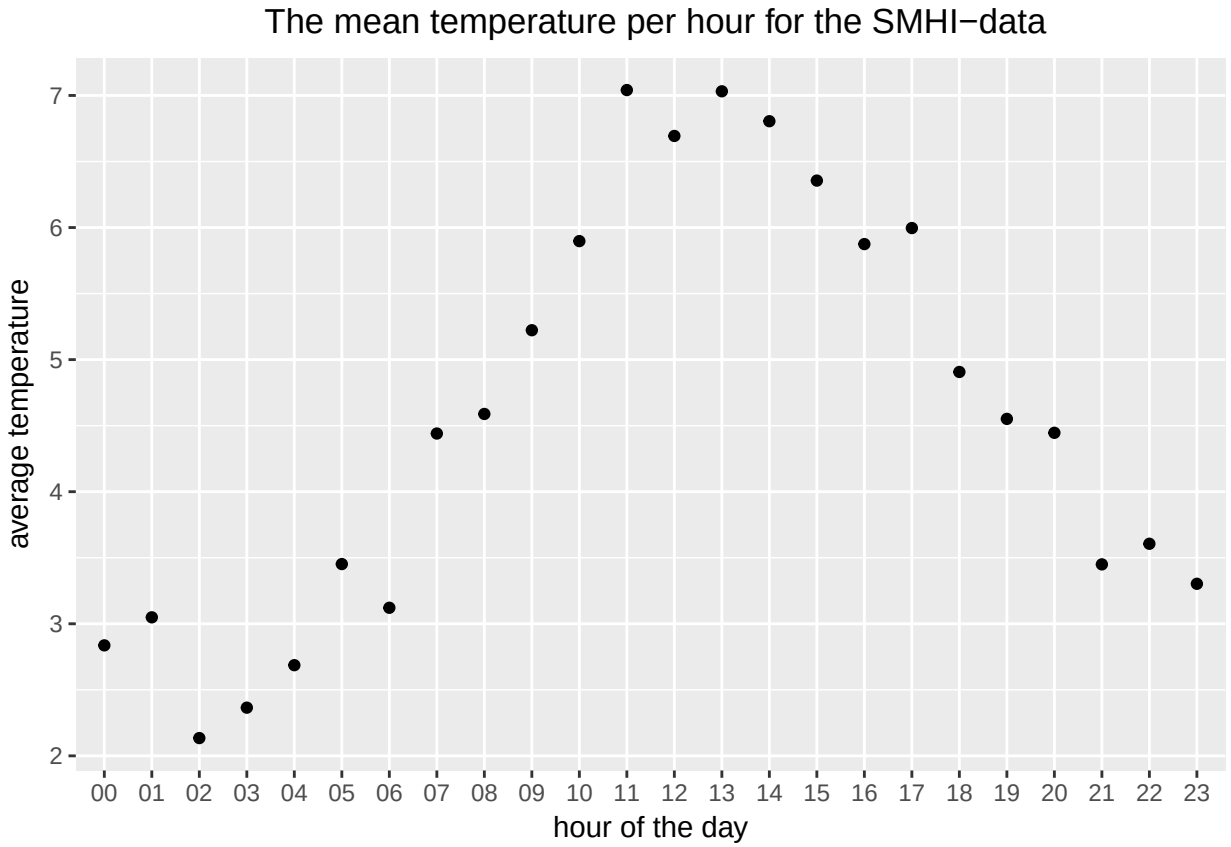
Contents

Assignment 1

In this assignment we were supposed to model air temperature using three Gaussian kernels that consider the geographical distance, the distance in days, and the distance in hours. Below is the predicted air temperatures for Christmas Eve this year in Vadstena. We can see that the shape is reasonable with colder temperatures in the morning/evening and it follows the typical bellshape. However, the actual values for the predicted temperatures are not very realistic and it turns out that it does not really matter which time of the year we predict, similar values will be estimated. We believe this is due to the independence of the kernels since with add them up and that results in the date kernel not being very influential in the decision making but it is the most important kernel in order to detect seasonal trends, i.e. warmer in the summer and colder in the winter.



Below are the average temperatures for every hour in the day from the data and it has the same shape as our estimate. This indicates that the time kernel play a big part in the prediction. We choose the smoothing factor for time to be 2 since it is reasonable to assume that $\pm$ 2 hours are similar to the current time.



For the distance kernel we choose 100000 to convert the meters to 100 kilometers which we thought seemed reasonable due to the size of Sweden and the regional determines the temperature quite a lot. It is usually colder in the north compared to the south. For the last kernel, time, we choose 7 to let the observations within the last week have the highest weights and it is the only kernel that can detect seasonal trends. However, since we added the kernels together its impact is severely reduced and therefore the model does not capture seasonal trends particular well. The estimates are basically an average of the temperatures on that particular hour for all days/years in the data located around the specified location.

To make the model a better predictor we propose to either multiply the kernels together making them dependent on each other or create new types of kernels, for instance day/week/month kernels similar to the time kernel to find the seasonal trends.

Appendix

Code for Assignment 1

```
set.seed(1234567890)
library(geosphere)
library(ggplot2)

stations <- read.csv("../data/stations.csv",header = TRUE,
                     stringsAsFactors=FALSE,
                     fileEncoding="latin1")
temps <- read.csv("../data/temps50k.csv")
st <- merge(stations,temps,by="station_number")

my_magic_kernel <- function(data ,time, date,
                            longlat = c(59.4446, 13.3374),
                            h_days = 6, h_time = 4,
                            h_distance = 100000){

    ## Defining the kernel
    gk <- function(x, xi){
        ## for the days
        if( all(class(x) == "Date")) {
            xi <- as.Date(factor(xi),format = "%Y-%m-%d")
            return(exp(-((abs( as.numeric(x - xi) )^2) / (h_days) )))
        }

        ## For the hours
        if(class(x)  == c("difftime") ) {
            xi <- strptime(xi,"%H:%M:%S")
            return(exp(-((abs( as.numeric(x) )^2) / (h_time))))
        }

        ## For long and lat
        return(exp(-((abs( (x - xi))^2) / (h_distance))))
    }

    ## Initiatin objects for loop.
    predictions <- data.frame(time=1,temp=1)
    i <- 1

    for (timme in times){
        mdate = strftime(paste(date,timme))
        data<-subset(st,  strftime(paste(st$date,st$time)) < mdate)

        ## Longitude and Latitude distances.
        dmat <- geosphere::distHaversine(p1 = cbind(data$latitude,data$longitude) ,
                                         p2 = longlat)
        gkdmat<- gk(dmat,0)

        ## datum
        datevec <- as.Date(st$date)
        gkdate<-gk(datevec,date)
```

```
        ##timme
        difftimes<-difftime(strptime(data$time,format="%H:%M:%S"),
                            strptime(timme,format = "%H:%M:%S"),
                            units = "hours")
        gktime<-gk(difftimes,0)

        alltemps <- rowSums(cbind(gkdmat,gktime,gkdate)*data$air_temperature) /
            sum((gkdmat + gkdate + gktime))
        predictions[i,] <- c(timme,sum(alltemps))
        i <- i + 1

    }

    predictions[,1]<- as.factor(predictions[,1])
    predictions[,2]<- as.numeric(predictions[,2])
    return(predictions)
}


a <- 58.4274
b <- 14.826
times <- c(paste0("0",seq(2,9,2),":00:00"),paste0(seq(10,24,2),":00:00"))

as <- my_magic_kernel(data = st ,time = times,
                      date = "2016-12-24",longlat = c(b,a),
                      h_days = 7, h_time = 2,
                      h_distance = 100000)

as[,1] <- as.factor(as[,1])
as[,2] <- as.numeric(as[,2])
ggplot(data = as, aes(x= time,y=temp))+geom_point() +
    labs(x= "Time",y= "Temperature",
         title = "Temperature on christmas eve") +
    theme(plot.title=element_text(hjust=0.5))

tempagg<- aggregate(st$air_temperature,
                    list(factor(substr(st$time,start= 1,stop =2))),
                    FUN = mean)

ggplot(data = tempagg, aes(x = Group.1, y =x )) + geom_point() +
    labs(x= "hour of the day", y = "average temperature",
         title = " The mean temperature per hour for the SMHI-data") +
    theme(plot.title=element_text(hjust=0.5))
```

Contributions

We divided the work into two parts and discussed/compiled the results in pairs. Then we all discussed our findings together as a whole group and checked that everyone had similar/understood the results.