Q1

Yi Hung Chen

2022-12-12

Assignment 1

In this assignment, kernel method is preformed to predict the hourly temperatures for a certian date and place in Sweden. Two data files (stations.csv and temps50k.csv) are given, these data sets should be combined according to "station_number".

There are three different Gaussian kernels that are used in this assignment:

1. The physical distance from a station to the point of interest. For this purpose, use the function distHaversine from the R package geosphere.

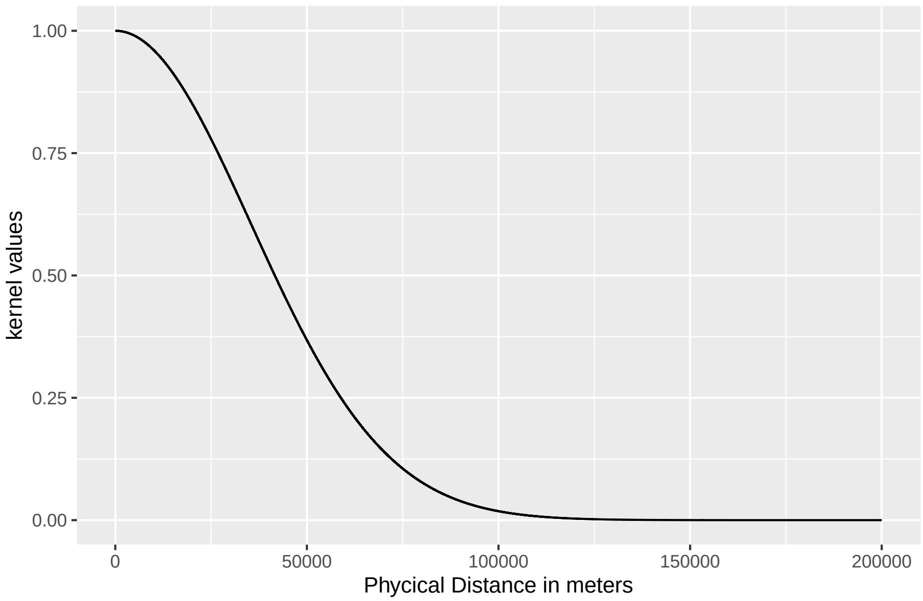
2. The distance between the day a temperature measurement was made and the day of interest.

3. The distance between the hour of the day a temperature measurement was made and the hour of interest.

To begin with, the first task is to choose an appropriate smoothing coefficient or width for each of three kernels above.
Since no cross-validation should be used so the width are manually chosen to have larger kernel values closer to the target point. Below are the plots showing the kernel value as a function of distance and the reason behind why the particular value is chose.

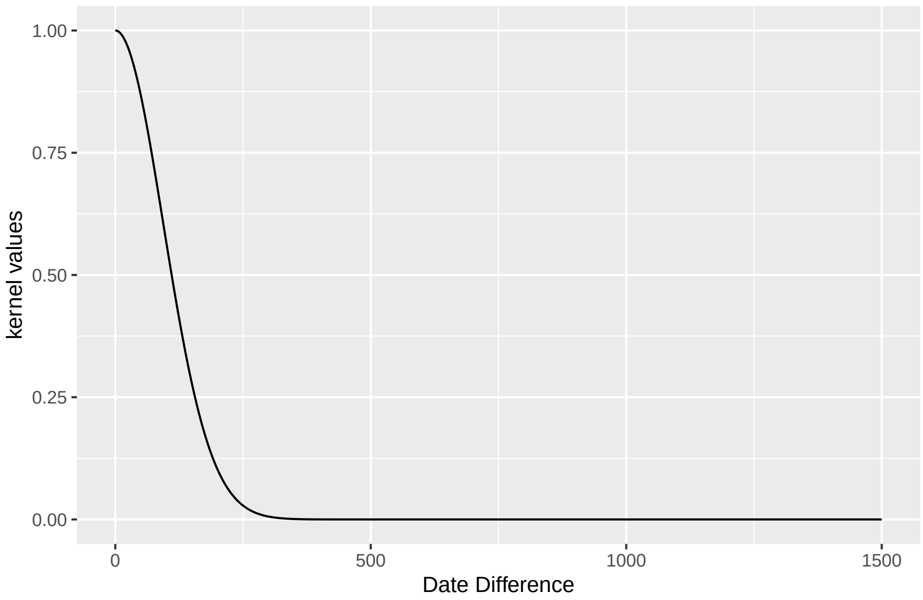
1. Kernel width of physical distance(h_distance=50000), which means 50 km. This is to gives larger kernel value to nearby stations(<50km). As Figure 1 shown below, if the data is from any station located more than 100km away, the data will not affect the prediction much. Note: In Figure 1, only kernel values within 200km are presented. As the true maximum distance to the target station (1186.7986499km) is way to far and the kernel value is already small after 150km.

Figure 1: Kernal value with respect to physical distance



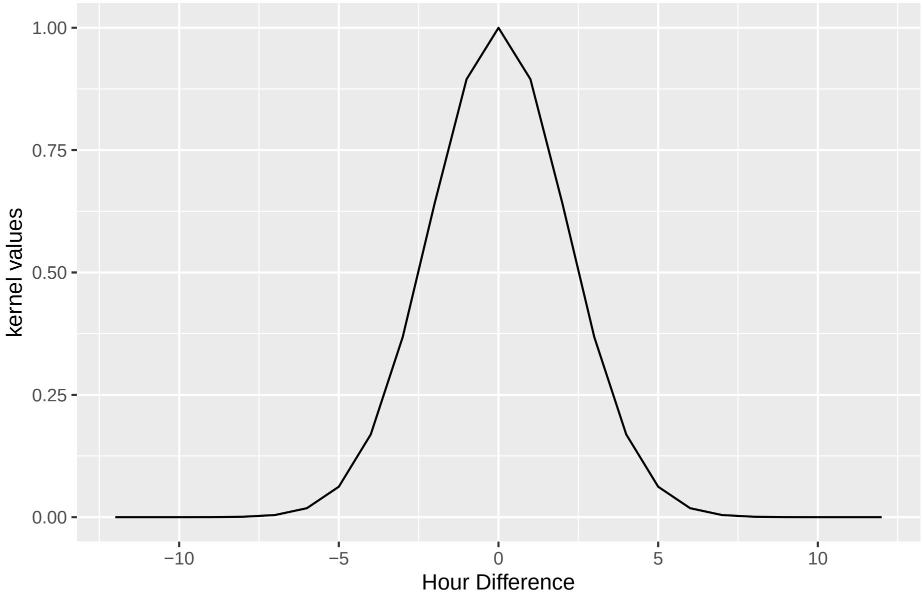
2. Kernel width of date difference(h_date <- 132.5). This is to gives larger kernel value to the observation within same year, especially for the month that is within half a year from target date. As Figure 2 shown below, as the date difference approach 500 days the kernel value become very small. However, this is not ideal in real life scenario, since the same date of different years should have larger impact on prediction than the date in different season. Note: In Figure 2, only kernel values within 1500 days are presented. As the true maximum date difference to the target date (2.6363×10^4 days) is way to big and the kernel value is already small after 500 days.

Figure 2:Kernal value with respect to date distance



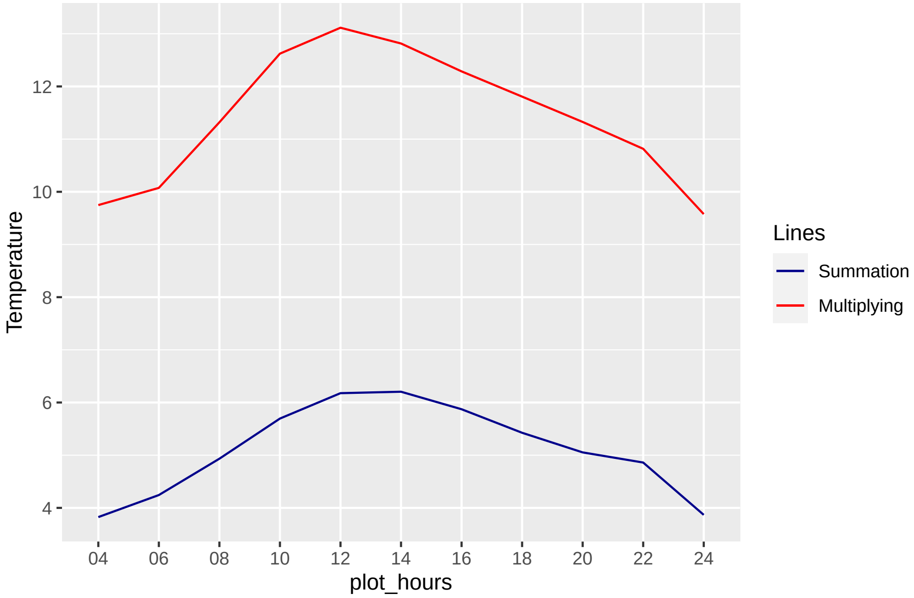
3. Kernel width of hour difference(h_time <- 3), which gives larger kernel value to the observation within 3 hours, and lower kernel value for more than 6 hours difference. This is to account for the difference between day time and night time.

Figure 3:Kernal value with respect to hour distance



With the kernels width being set. The next step is combining three Gaussian kernels by the summation and the multiplication of them. Then using these two approaches (summation, multiplication) to make prediction. Importantly, because the data includes measurements that are posterior to the day and hour of the target time. Filtering must be done before prediction being made.

Figure 4:Comparesion of Sumatinon and Multiplying kernels



According to Figure 4, the prediction of the multiplication approach has higher overall temperature than summation approach. Also the difference of temperature during the day is also larger in multiplication approach, in other word, the prediction of summation is flatter.

The reason behind of this is because when using summation, if any of three different gaussian kernels has large value, the weight will be large, resulting in a flatter prediction. On the other hand, in multiplication approach, if any one of three gaussian kernels produce small kernel value, it will result in smaller overall weight.

For example, if the other station is over 500km away but having a data at the same day and only a hour before target time. The summation approach will consider it has a larger weight because the date and hour kernel values are large, but the multiplication approach will result in near to 0 weight as the distance is too far away. It can be conclude that the multiplication approach will produce more reasonable prediction because of this behavior.

Appendix

```
#Question 1
library(geosphere)
library(ggplot2) #TA said ggplot is allowed for visualization

set.seed(1234567890)
stations <- read.csv("./stations.csv", fileEncoding = "latin1")
temps50k <- read.csv("./temps50k.csv")

data_full <- merge(stations, temps50k, by = "station_number")

#choose target location
set.seed(1378)
target_station <- 84260
target_loc <- stations[stations$station_number == target_station, c("longitude", "latitude")]
target_data <- data_full[data_full == target_station, ]
target_date <- "2013-11-04"

hours <- seq(as.POSIXct("2013-11-04 04:00"), as.POSIXct("2013-11-04 24:00"), by = "2 hour")
hours <- format(hours, format = "%H:%M:%S")
#find the max physical difference between the target station and the other station
max_distance <- max(distHaversine(target_loc, data_full[,  c("longitude", "latitude")]))

#gives the maximum date difference to "previous date"
max_date_diff <- max(as.vector(difftime(as.Date(target_date),as.Date(data_full$date), units = "day")))

# Check if the weight is suitable for our application
h_distance <- 50000 ; h_date <- 132.5; h_time <- 3
distance <- seq(0,200000,1)/h_distance
k_dist <- exp(-(distance)^2)
plotdf_distance <- data.frame(k_dist,seq(0,200000,1))

plot_distance <- ggplot(plotdf_distance,aes(x=seq(0,200000,1),y=k_dist))+
  geom_line()+
  ggtitle("Figure 1: Kernal value with respect to physical distance")+
  xlab ("Phycical Distance in meters")+
  ylab ("kernel values")

# Since many of the stations are simply too far away, limit the x-value to make observation easier


# kernel of date
max_date_diff <- max(as.vector(difftime(as.Date(target_date),as.Date(data_full$date), units = "day")))

#Here, calculate the max/min difference of date with target date, this doesn't account for season since
date <- seq(0,1500,1)/h_date
k_date <- exp(-(date)^2)
plotdf_date <- data.frame(k_date,seq(0,1500,1))

plot_date <- ggplot(plotdf_date,aes(x=seq(0,1500,1),y=k_date))+
  geom_line()+
  ggtitle("Figure 2:Kernal value with respect to date distance")+
```

```r
  xlab ("Date Difference")+
  ylab ("kernel values")


# we only present t closest 2000 day, since the day that is before that has too little affect to the va


# kernel of time
time <- seq(-12,12,1)/h_time
k_time <- exp(-(time)^2)
plotdf_time <- data.frame(k_time,seq(-12,12,1))

plot_hour <-ggplot(plotdf_time,aes(x=seq(-12,12,1),y=k_time))+
  geom_line()+
  ggtitle("Figure 3:Kernal value with respect to hour distance")+
  xlab ("Hour Difference")+
  ylab ("kernel values")


#Summation Kernel
i <- 1
temperature_sum <- vector(length = length(hours))
for(i in seq_along(hours)){
  #filter posterior data
  available_data <- data_full[(

      data_full$date < target_date|
      data_full$time <= hours[i] & data_full$date == target_date


  ), ]

  physcial_distance <- distHaversine(target_loc, available_data[,  c("longitude", "latitude")])
  date_distance <- difftime(as.Date(target_date), as.Date(available_data$date), units = "day")
  hour_distance <- difftime(as.POSIXct(hours[i], format = "%H:%M:%S"),
                    as.POSIXct(available_data$time, format = "%H:%M:%S"),
                    units = "hour")
  #Here we referred to page10 of "Lecture 3a Block 1:  Kernel Methods" Lecture Slide
  weight_sum <- exp(-(physcial_distance^2/h_distance^2/2)) + exp(-(as.numeric(date_distance)/h_date)^2/2
  temperature_sum[i] <- sum(available_data$air_temperature * weight_sum) / sum(weight_sum)
}

#Multiplying Kernel

temperature_multiply <- vector(length = length(hours))
for(i in seq_along(hours)){
   available_data <- data_full[(

      data_full$date < target_date|
      data_full$time <= hours[i] & data_full$date == target_date


  ), ]
  physcial_distance <- distHaversine(target_loc, available_data[,  c("longitude", "latitude")])



  date_distance <- difftime(as.Date(target_date), as.Date(available_data$date), units = "day")
  hour_distance <- difftime(as.POSIXct(hours[i], format = "%H:%M:%S"),
                    as.POSIXct(available_data$time, format = "%H:%M:%S"),
                    units = "hour")

  #Here we referred to page10 of "Lecture 3a Block 1:  Kernel Methods" Lecture Slide
  weight_multiply <- exp(-(physcial_distance^2/h_distance^2/2)) * exp(-(as.numeric(date_distance)/h_date


  temperature_multiply[i] <- sum(available_data$air_temperature * weight_multiply) / sum(weight_multiply
}
plot_hours <- c("04","06","08", "10", "12", "14", "16", "18", "20", "22" ,"24")
temperature_sum <- data.frame(cbind(plot_hours,temperature_sum))

plotdata <- data.frame(cbind(temperature_sum,temperature_multiply))
ggplot(data =plotdata)+
  geom_line(aes(x=plot_hours,y=as.numeric(temperature_sum),group=1,colour="Summation"))+
  geom_line(aes(x=plot_hours,y=as.numeric(temperature_multiply),group=1 ,colour="Multiplying"))+
  scale_color_manual(name = "Lines", values = c("Summation" = "darkblue", "Multiplying" = "red"))+
  ylab("Temperature")+
  ggtitle("Figure 4:Comparesion of Sumatinon and Multiplying kernels")
```