Lab3_Assignment2

Pontus Olsson,

2025-12-14

Assignment 1. Theory

1. What is the kernel trick?

A kernel is a function that takes two arguments from the same space and returns a scalar. By using a kernel as our input, we can compute the inner product between non-linear transformations ($\phi(\mathbf{x})^t\phi(\mathbf{x})$) without explicitly computing the transformation ($\phi(\mathbf{x})$).

This is called the kernel trick, it allows the user to increase the amount of features in our model without designing the feature space. (MLFC p.193-195)

2. What is the purpose of C

C is in a support vector a regularizing parameter. It's purpose is to penilize insignificant features in the model. (MLFC p.193-195)

[Comment: Double check this. This is just a scim-through]

Assignment 2 Kernel Methods

In this assignment you're suppose to predict the air temperature for an arbitrary date for an arbitrary weather station for every other hour between 04:00 to 24:00.

We choose to predict the temperature at station 96350 (in the city of *Västerås*, noted as one of the ugliest cities in Sweden).

A gaussian kernel were created for each input (date, time and distance) to calculate the distance. The kernel used is the following:

$$\mathbb{K}(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{||\mathbf{x}, \mathbf{x}'||_2^2}{2\mathrm{l}^2}\right)$$

where l is a parameter (MLFC p.195,207).

Out of these kernels to one kernel which is used as a weight.

$$\hat{y} = \frac{\sum_{i=1}^{n} \mathbb{K}_i \times y_i}{\sum_{i=1}^{n} \mathbb{K}_i}$$

The hyperparameter were used for each specific variable. Distance has a l = `1000`. Date has a l = `2`. Time has a l = `2`.

Date should be high priority since seasons, no matter the place or time, have a high correlation with the temperature. Time becomes relevant to measure the variation in temperature within a single day. Distance is the variable with the most variance, but the least effect on the temperature, hence the higher value.

The first kernel used is a sum out of all kernels.

$$\mathbb{K} = \mathbb{K}_{\text{Distance}} + \mathbb{K}_{\text{Date}} + \mathbb{K}_{\text{Time}}$$

The predicted value with the kernel is shown in figure 1.

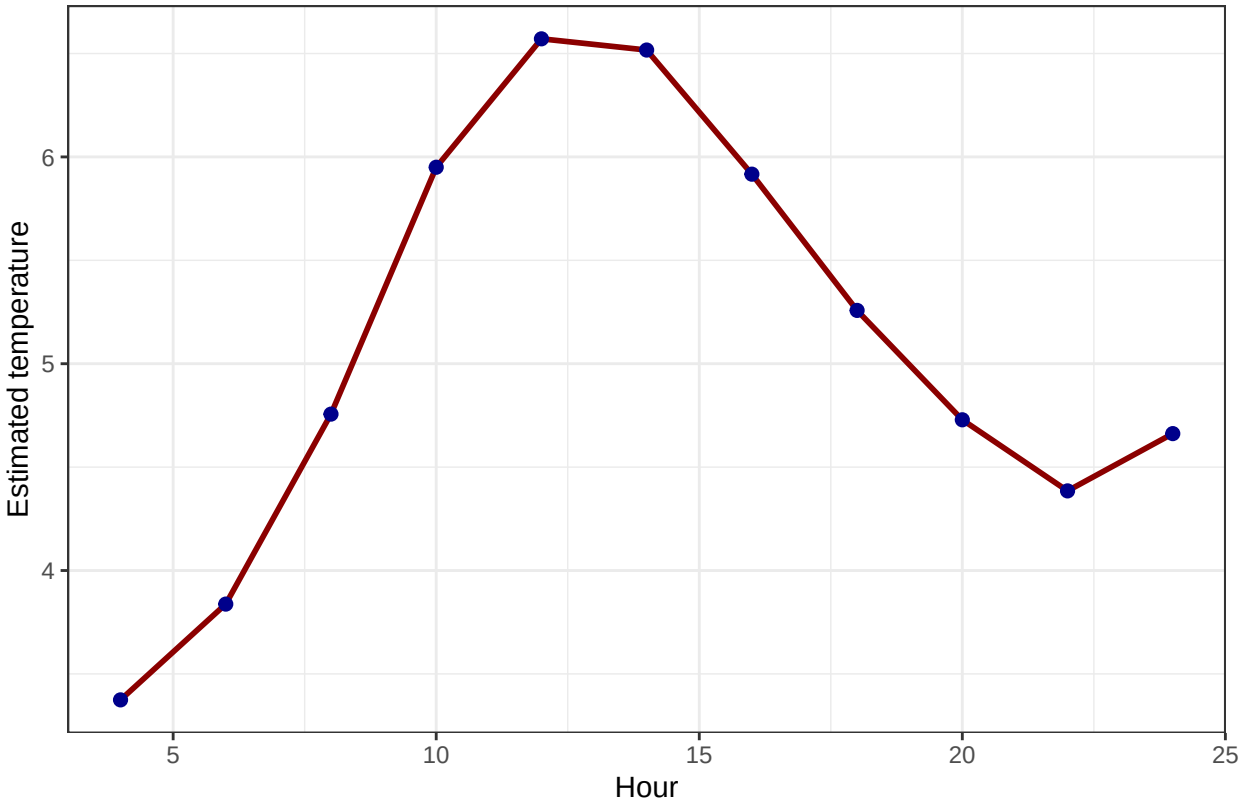

Figure 1: Summed kernel

The predictions do not reflect the temperature for a day in middle of May. This is only slightly higher than the mean of all the measurements through out the year and through out the country, which are shown in table 1.

Table 1: Mean value for all weather measurements

Hour	Temperature	Hour	Temperature	Hour	Temperature	Hour	Temperature
00	2.797617	06	3.065398	12	6.667048	18	4.865955
01	3.077366	07	4.421628	13	6.941809	19	4.558027
02	2.117997	08	4.529144	14	6.722578	20	4.354462
03	2.355427	09	5.147929	15	6.331536	21	3.399650
04	2.730655	10	5.815543	16	5.774832	22	3.614089
05	3.375202	11	7.025667	17	5.908738	23	3.232266

The second kernel used is a product out of all kernels.

$$\mathbb{K} = \mathbb{K}_{\text{Distance}} \times \mathbb{K}_{\text{Date}} \times \mathbb{K}_{\text{Time}}$$

The predicted value with the kernel is shown in figure 2.

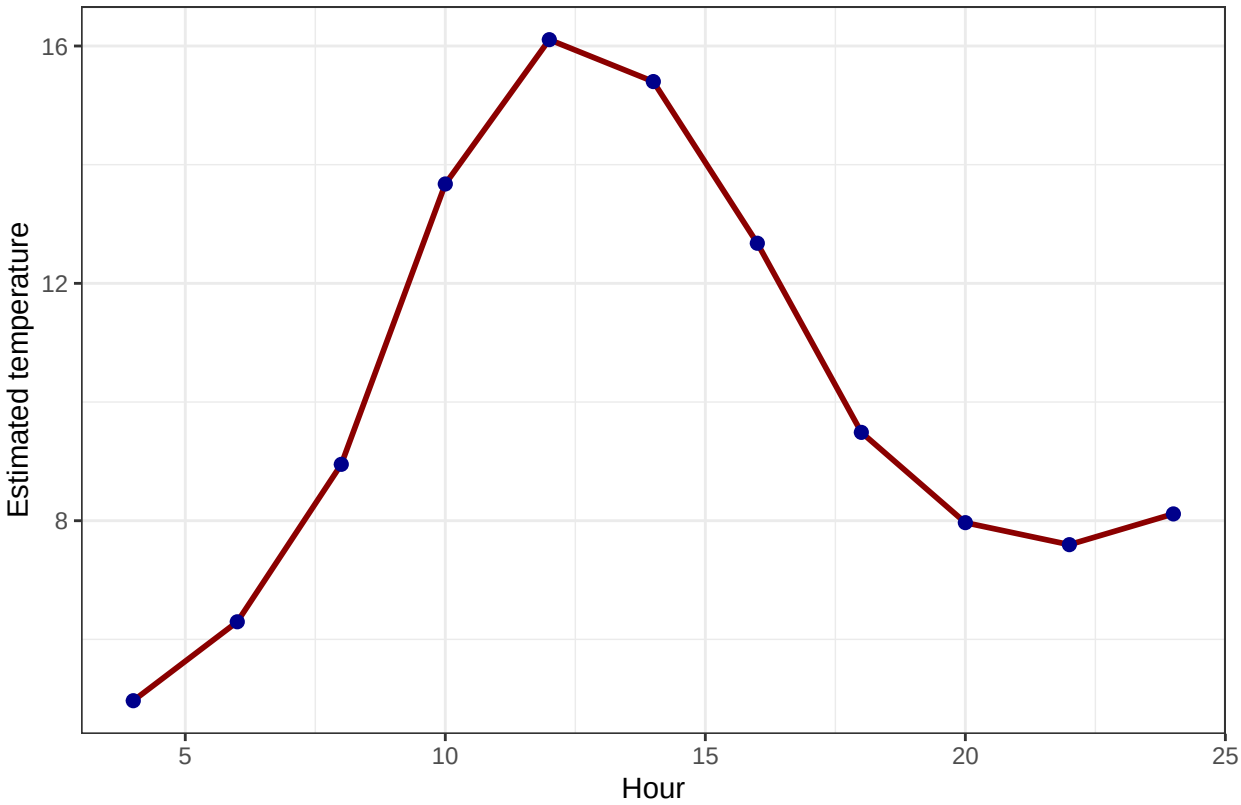

Figure 2: Product of kernel

The predictions reflect much better than the actual in the area than the previous kernel. This is shown in figure 2. The reason for this is that, since it is a summed kernel, an observation that is in winter can still be weighted in only because it is - for example - near the predicted hour. In the kernel that uses product, if value is 0, then it is not weighted in.

Table 2: Mean value for a day, areas around Västerås, May

Hour	Temperature	Hour	Temperature	Hour	Temperature	NA	NA
00	6.980	08	10.30000	14	15.900000	23	10.6
01	8.025	09	10.66667	15	15.600000	NA	NA
03	11.650	10	6.50000	16	13.450000	NA	NA
04	10.400	11	19.35000	17	15.700000	NA	NA
05	6.200	12	14.30000	18	7.300000	NA	NA
06	7.100	13	15.30000	20	5.733333	NA	NA

Appendix Assignment 2

```r
# Read libraries
library(geosphere)
library(knitr)
library(ggplot2)
library(lubridate)

# Read data ####
stations <- read.csv("stations.csv", fileEncoding = "latin1")
temps <- read.csv("temps50k.csv")

# Predictions values and merging data ####
# Station Västerås 2016-05-19
st <- merge(stations,temps,by="station_number")

a <- 16.6326 # longitude points of the city in Västerås
b <- 59.5976 # latitude
date <- "2015-05-19" # date that is used
times <- paste0(sprintf(fmt = "%02d", seq(from = 4, to = 24, by = 2)), ":00:00")

st <- st[as.Date(st$date) < "2015-05-19", ]

dist <- sapply(X = 1:nrow(st), FUN = function(i){
  distHaversine(c(a, b), c(st$longitude[i], st$latitude[i]))
})

day_diff <- sapply(X = 1:nrow(st), FUN = function(i){
  abs(as.POSIXlt(st$date[i])$yday - as.POSIXlt(date)$yday)
})

hour_diff <- sapply(X = 1:length(times), FUN = function(i){
  abs(difftime(c(paste0(c("2000-01-01 "), st$time)), c(paste0(c("2000-01-01 "), times[i])), units = "hou
})

# Selecting kernel ####
# distance plays way less of a role
h_distance <- 1000
# season plays a important role since days in winter are often colder than summers
h_date <- 2
# nighttime plays some role in the model
h_time <- 2

temp <- vector(length=length(times))


k_dist <- exp(-dist^2/(2*h_distance^2))
k_date <- exp(-day_diff^2/(2*h_date^2))
k_time <- exp(-hour_diff^2/(2*h_time^2))

# Kernel 1 ####
K <- k_dist+k_date+k_time

temp <- sapply(X = 1:ncol(K), FUN = function(i){
```

```r
  sum(K[, i]*st$air_temperature)/sum(K[, i])}
)


ggplot(mapping = aes(y = temp, x = seq(from = 4, to = 24, by = 2))) + geom_line(linewidth = 1, col = "da
# aggregate(st$air_temperature, by = list(substring(st$time, 1, 2)), FUN = mean)

## Table 1 ####
mean_values <- aggregate(st$air_temperature, by = list(substring(st$time, 1, 2)), FUN = mean)

output <- cbind(mean_values[1:6, ],
                mean_values[7:12, ],
                mean_values[13:18, ],
                mean_values[19:24, ])
colnames(output) <- rep(c("Hour", "Temperature"), times = 4)

kable(output, caption = "Mean value for all weather measurements", align = "c")


# Kernel 2 ####
K <- k_dist*k_date*k_time

temp <- sapply(X = 1:ncol(K), FUN = function(i){
  sum(K[, i]*st$air_temperature)/sum(K[, i])}
)


# Students' code here
ggplot(mapping = aes(y = temp, x = seq(from = 4, to = 24, by = 2))) + geom_line(linewidth = 1, col = "da
# Show https://www.vadret1.com/europe/sweden/vastmanland/vasteras?page=past-weather#day=19&month=5
# st[st$station_name == "Västerås" & month(st$date) == 5, c("station_name", "latitude", "longitude", "d
#
#
# st$air_temperature[st$station_name == "Västerås" & month(st$date) == 5]
#
# # Show https://www.vadret1.com/europe/sweden/vastmanland/vasteras?page=past-weather#day=19&month=5
# aggregate(st$air_temperature[st$station_name == "Västerås" & month(st$date) == 5], by = list(substrin

## Table 2 ####
lat_min <- 59.28
lat_max <- 59.80
lon_min <- 15.80
lon_max <- 17.26

inside <- function(lat, lon) {
  lat >= lat_min & lat <= lat_max &
    lon >= lon_min & lon <= lon_max
}

lon <- st$longitude
lat <- st$latitude

mean_values <- aggregate(st$air_temperature[inside(lat, lon) & month(st$date) == 5], by = list(substrin
```

```r
output <- cbind(mean_values[1:6, ],
                mean_values[7:12, ],
                mean_values[13:18, ],
                mean_values[19:24, ])
colnames(output) <- rep(c("Hour", "Temperature"), times = 3)

kable(output, caption = "Mean value for a day, areas around Västerås, May", align = "c")
```