

Q4 GP

Xiner Ning

4/14/2020

Part 1

1. Fit a model using the radial basis function to the data in kernel_regression_1.csv

```
# https://katbailey.github.io/post/gaussian-processes-for-dummies/
# Kernel matrix
library(tidyverse)
```

```
## -- Attaching packages -----
----- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.0      v purrr   0.3.3
## v tibble  3.0.0      v dplyr   0.8.5
## v tidyr   1.0.2      v stringr 1.4.0
## v readr   1.3.1      v forcats 0.5.0
```

```
## -- Conflicts -----
----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(ggplot2)
library(MASS)
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##      select
```

```

# import the data
mydata<- read.csv("kernel_regression_1.csv")

# kernel function
K = function(x,x_prime,l){
  d = sapply(x, FUN = function(x_in)(x_in - x_prime)^2)
  return(t(exp(-1/(2*l^2) *d)))
}

X= mydata$x
Y=mydata$y
l=1

mu = mean(mydata$y)
mu_star = 0

x_prime = seq(min(mydata$x)-1, max(mydata$x)+1, length.out = length(mydata$x))

K_f = K(mydata$x, mydata$x, l) + diag(var(Y), length(X))

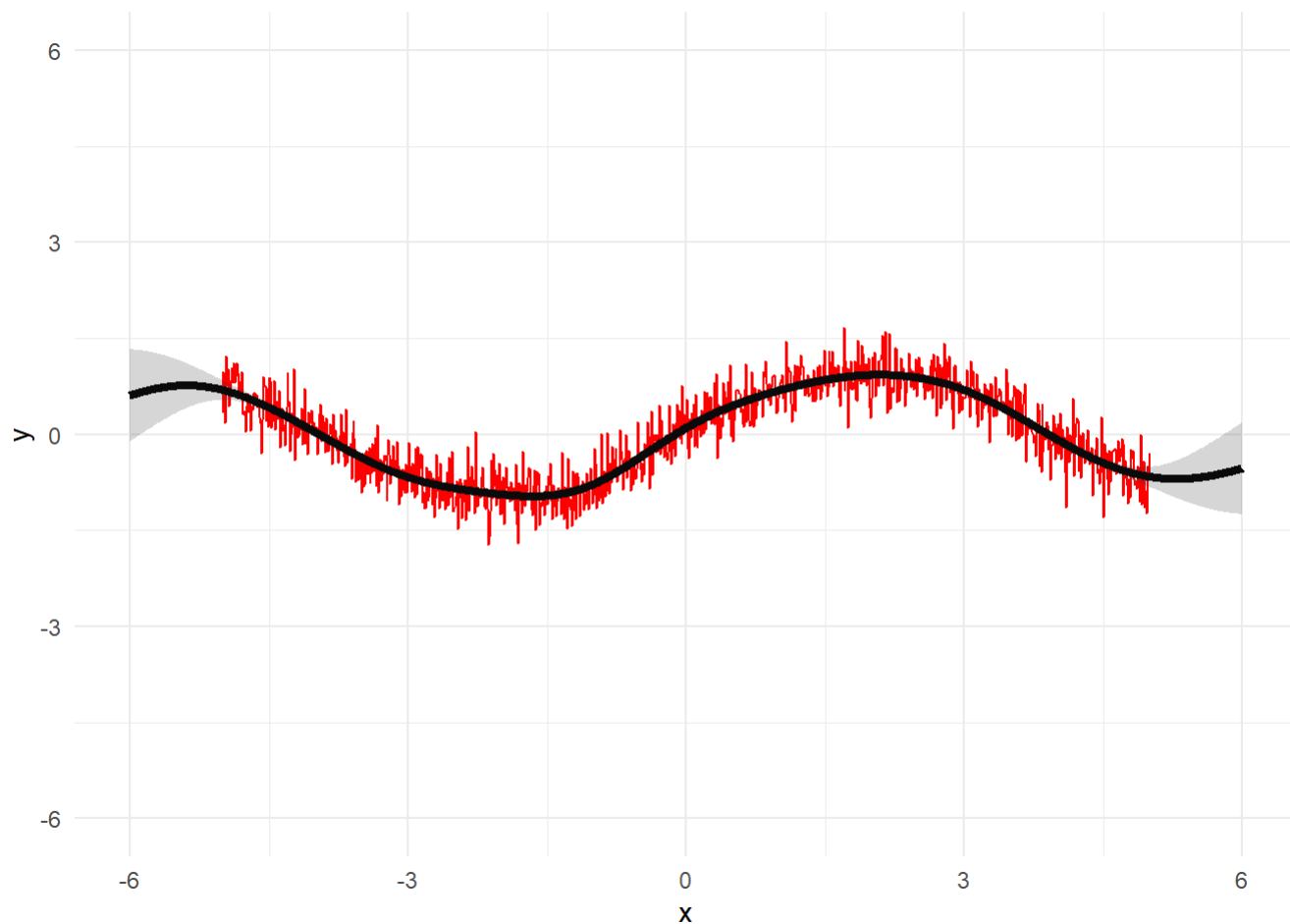
K_star = K(X,x_prime,l)
K_starstar = K(x_prime,x_prime,l)

mu_star = mu_star + t(K_star) %*% solve(K_f) %*% (Y - mu)
Sigma_star = K_starstar - t(K_star)%*% t(solve(K_f)) %*% K_star

df_plot = tibble(x = x_prime,
                  y = mu_star %>% as.vector(),
                  sd_prime = sqrt(diag(Sigma_star)))

ggplot() +
  geom_line(data = mydata, aes(x=x , y=y), color = 'red', size=0.5) +
  geom_line(data = df_plot, aes(x = x, y = y), size= 1.5) +
  geom_ribbon(data = df_plot, aes(x=x, y=y, ymin = y-sd_prime, ymax = y+sd_prime), alpha = 0.2)
+
  xlim(c(-6,6))+ylim(c(-6,6))+ theme_minimal()

```



Plot the mean function with a 95% confidence interval

```
ggplot() +  
  geom_line(data = mydata, aes(x=x , y=y), color = 'red', size=0.5) +  
  geom_line(data = df_plot, aes(x = x, y = y), size= 1.5) +  
  geom_ribbon(data = df_plot, aes(x=x, y=y, ymin = y-2*sd_prime, ymax = y+2*sd_prime), alpha =  
0.2) +  
  xlim(c(-6,6))+ylim(c(-6,6))+ theme_minimal()
```

