Lab 4 - Gaussian Processes

David Björelind

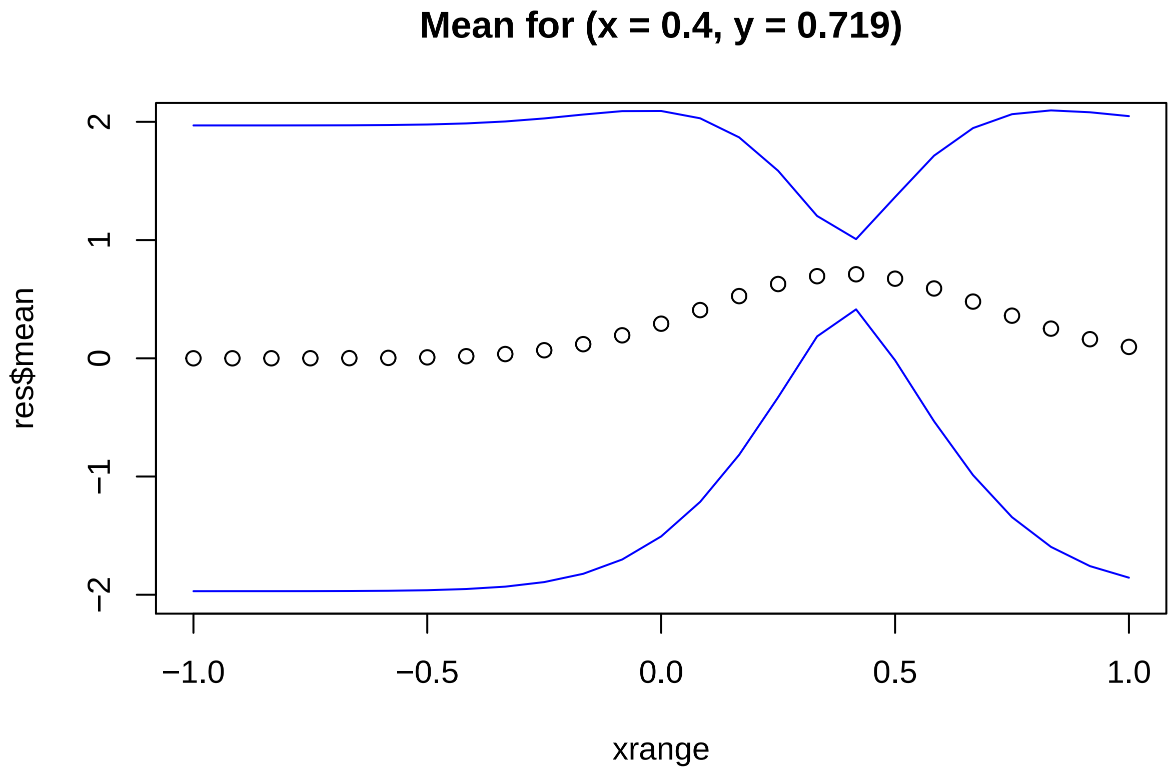
10/18/2020

2.1) (1) Implementing GP Regression

```
# Simulating from posterior distribution of f
posteriorGP = function(X, y, XStar, sigmaNoise, k, sigmaF, l){

  K = k(X,X, sigmaF, l)
  n = length(XStar)
  L = t(chol(K + sigmaNoise*diag(dim(K)[1])))
  kStar = k(X,XStar, sigmaF, l)
  alpha = solve(t(L), solve(L,y))

  FStar = t(kStar) %*% alpha
  v = solve(L, kStar)
  vf = k(XStar, XStar, sigmaF, l) - t(v)%*%v + sigmaNoise*diag(n) #Adding sigma for noise
  #print(k(XStar, XStar, sigmaF, l))
  #print(diag(t(v)%*%v))
  logmarglike = -t(y)%*%alpha/2 - sum(diag(L)) - n/2*log(2*pi)


  # Returns a vector with the posterior mean and variance
  return(list("mean" = FStar,"variance" = vf,"logmarglike" = logmarglike))
}
```

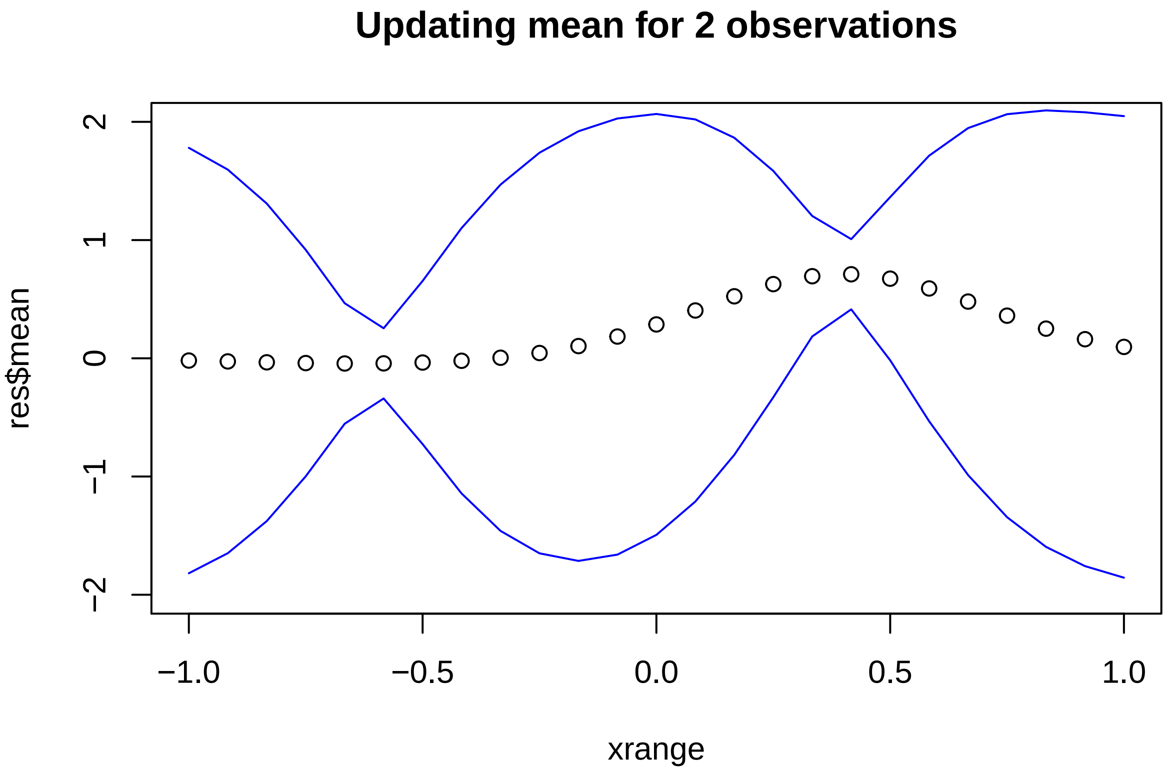
2.1) (2) GP Regression with kernlab

```
sigmaf = 1^2
sigman = 0.1^2
l = 0.3
x = 0.4
y = 0.719
xrange = seq(-1,1, length=25)

res = posteriorGP(x, y, xrange, sigman, SquaredExpKernel, sigmaF = sigmaf, l=l)

# (2) Plotting posterior mean and 95% interval bands
plot(x = xrange, y = res$mean, ylim = c(-2,2), main = "Mean for (x = 0.4, y = 0.719)")
lines(x = xrange, y = res$mean + sqrt(diag(res$variance))*1.96, col="blue", type="l")
lines(x = xrange, y = res$mean - sqrt(diag(res$variance))*1.96, col="blue",  type="l")
```



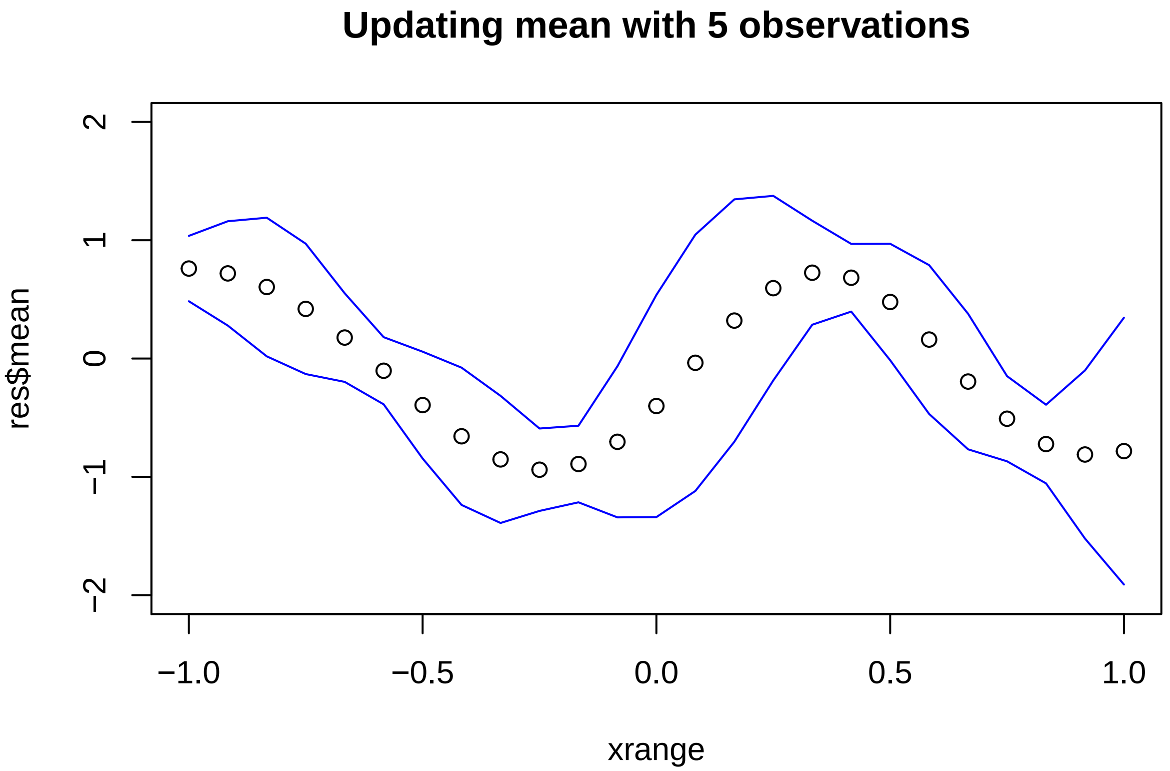
2.1) (3)

```r
# (3)
x = c(0.4,-0.6)
y = c(0.719,-0.044)

res = posteriorGP(x, y, xrange, sigman, SquaredExpKernel, sigmaF = sigmaf, l=l)
plot(x = xrange, y = res$mean, ylim = c(-2,2), main = "Updating mean for 2 observations")
lines(x = xrange, y = res$mean + sqrt(diag(res$variance))*1.96, col="blue", type="l")
lines(x = xrange, y = res$mean - sqrt(diag(res$variance))*1.96, col="blue",  type="l")
```



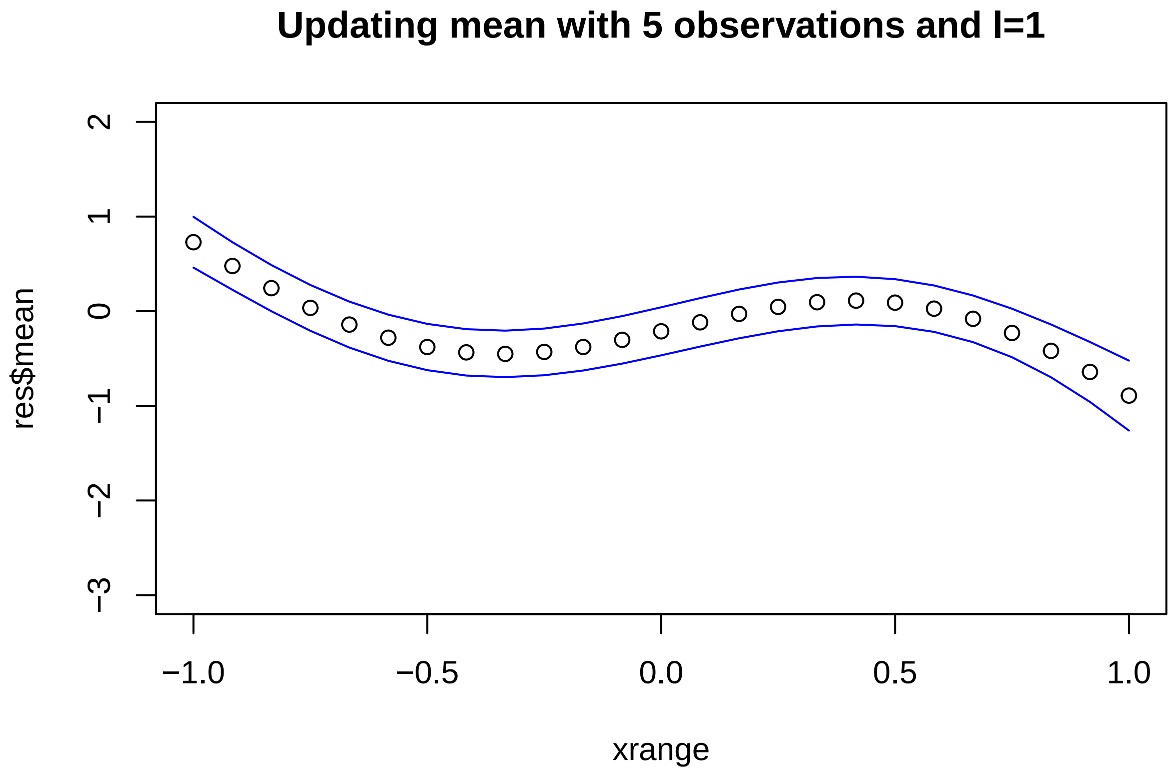
2.1) (4)

```
# (4)
x = c(-1.0, -0.6, -0.2, 0.4, 0.8)
y = c(0.768, -0.044, -0.940, 0.719, -0.664)
sigmaf = 1^2
l = 0.3
res = posteriorGP(x, y, xrange, sigman, SquaredExpKernel, sigmaF = sigmaf, l=l)
plot(x = xrange, y = res$mean, ylim = c(-2,2), main = "Updating mean with 5 observations")
lines(x = xrange, y = res$mean + sqrt(diag(res$variance))*1.96, col="blue", type="l")
lines(x = xrange, y = res$mean - sqrt(diag(res$variance))*1.96, col="blue",  type="l")
```



## 2.1) (5)

```
# (5)
x = c(-1.0, -0.6, -0.2, 0.4, 0.8)
y = c(0.768, -0.044, -0.940, 0.719, -0.664)
sigmaf = 1^2
l = 1
res = posteriorGP(x, y, xrange, sigman, SquaredExpKernel, sigmaF = sigmaf, l=l)

plot(x = xrange, y = res$mean, ylim = c(-3,2), main = "Updating mean with 5 observations and l=1")
lines(x = xrange, y = res$mean + sqrt(diag(res$variance))*1.96, col="blue", type="l")
lines(x = xrange, y = res$mean - sqrt(diag(res$variance))*1.96, col="blue",  type="l")
```



Since **l=1**, the produced function will be more smooth compared to (4). We also see that the bands created are much more narrow compared to plot produced in (4).

2.2 (1) GP Regression with kernlab

2.2 (2)

```
# (3)
```

Another chunk

```
# (3)
```

Another chunk

```
# (3)
```