Computer Lab 2

Martynas Lukosevicius, Alejo Perez Gomez, Zahra Jalil Pour

10/11/2020

Question 1: Optimizing parameters

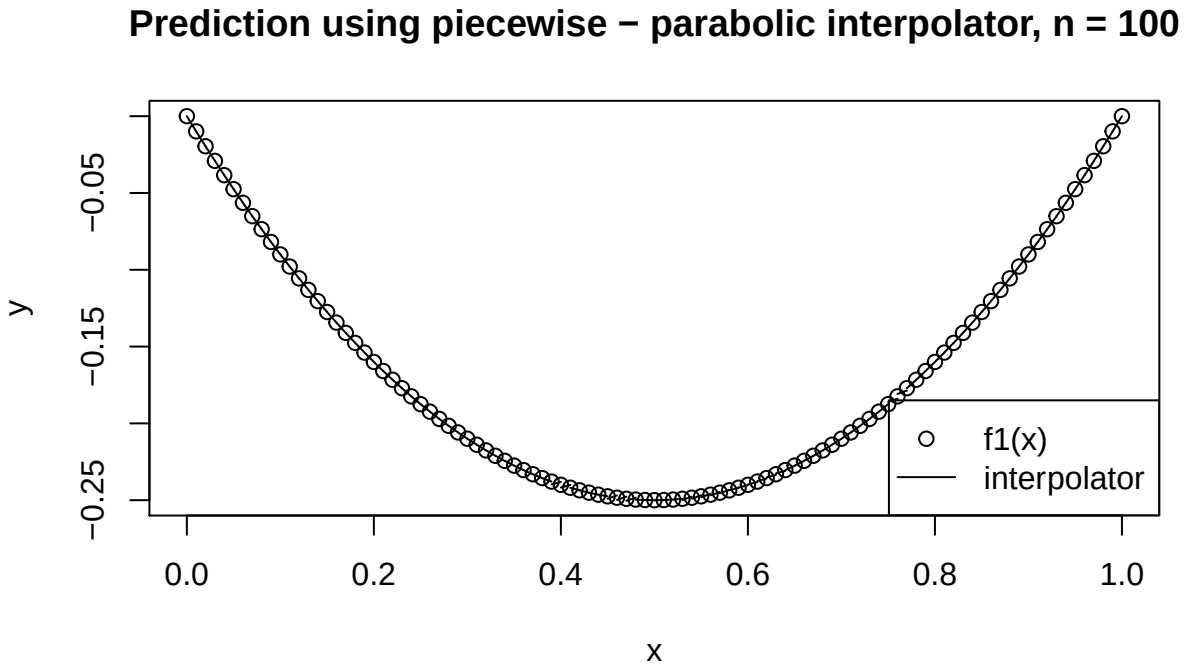
1

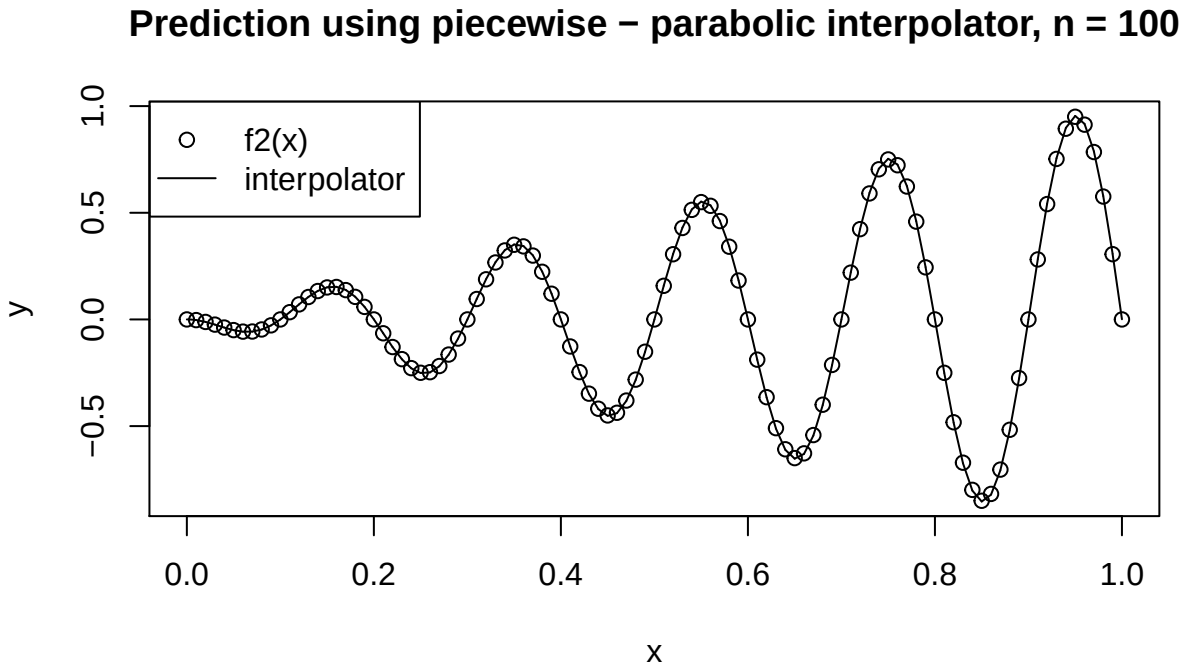
```
interpolator <- function(a, x){
  X <- as.matrix(c(1, x, x^2), ncol = 1 )
  return(as.vector(a %*% X))
}

SSE <- function(x, a, method1, method2){
  real <- sapply(x, method1)
  pred <- sapply(x, method2, a = a)
  return(sum((real-pred)^2))
}

optimiser <- function(x, method){
  aInit <- c(0, 0, 0)
  res <- optim(aInit, SSE, x = x, method1 = method, method2 = interpolator)
  return(res$par)
}
```

2

```
aproximate <- function(n, method){
  scale <- 1/n
  midVal <- scale / 2
  result <- list()
  for (i in 1:n) {
    x <- c((scale*i)-scale, (scale * i) - midVal, scale * i)
    result <- append(result, list(optimiser(x, method)))
  }
  return(result)
}
```

3





From graphs we can see that piecewice - parabolic interpolator predicted values almost without any error.

Question 2

1

```
load("data.RData")
```

2

The sampled data is normally distributed with μ and σ

$$y \sim N(\mu, \sigma^2)$$

Likelihood of the model:

$$L(p(\mu, \sigma^2|y)) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-} \frac{(y-\mu)^2}{2\sigma^2} = \frac{1}{(\sqrt{2\pi\sigma^2})^n} e^{-} \sum_{i=1}^{n} \frac{(y_i-\mu)^2}{2\sigma^2}$$

log - likelihood of the model:

$$lnL(p(\mu, \sigma|y)) = -\frac{n}{2} ln(2\pi\sigma^2) - \sum_{i=1}^{n} \frac{(y_i-\mu)^2}{2\sigma^2}$$

MLE estimators are partial derivatives of - log-likelihood

$\hat{\mu}_{MLE}$ estimator:

$$\frac{\partial lnL(p(\mu, \sigma|y))}{\partial \mu} = -\frac{1}{2\sigma^2} \frac{\partial(\sum y_i^2 - 2\mu \sum y_i + n\mu^2)}{\partial \mu} = -\frac{1}{2\sigma^2}(0 - 2\sum y_i + 2n\mu) = \frac{\sum y_i - n\mu}{\sigma^2}$$

$$MLE => \frac{\sum y_i - n\mu}{\sigma^2} = 0$$

$$\hat{\mu}_{MLE} = \frac{\sum y_i}{n}$$

$\hat{\sigma}_{MLE}$ estimator:

$$\frac{\partial lnL(p(\mu, \sigma|y))}{\partial \sigma} = -\frac{n}{\sigma} + \frac{1}{\sigma^3} \sum_{i=1}^{n} (y_i - \mu)^2$$

$$MLE => -\frac{n}{\sigma} + \frac{1}{\sigma^3} \sum_{i=1}^{n} (y_i - \mu)^2 = 0$$

$$\hat{\sigma}^2_{MLE} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \mu)^2$$

mean	variance
1.276	2.006

3

```
minusLogLikelihood <- function(x){
  n <- length(data)
  data <- data
  mu <- x[1]
  sigma <- x[2]
  part1 <-  ((n/2) * log(2*pi * (sigma^2)))
  part2 <- (sum((data - mu)^2)) / (2 * sigma^2)
  return(part1 + part2)
}
```

The function `optim()` minimizes the function by default in R. Thus we will optimize the minus log-likelihood function in order to find the maximum of the function. We will perform two types of algorithms, Conjugate Gradient and BFGS, both with gradient specified and without, to optimize the minus log-likelihood function. It is a better idea to maximize the log-likelihood than maximize the likelihood. This is due to the large values that occurs in the likelihood, which are numerically unstable, so it is preferable to take the logarithm which gives us a better scale to work with. Also, differentiating the log-likelihood function is more computationally convenient, since the product is replaced by a sum (see Question 2.2).

4

	converge	mean	variance	n. of functions	n. of gradients
minus loglikelihood CG	Yes	1.27552771909709	2.00597650338868	208	35
minus loglikelihood CG gradient	Yes	1.27552759112531	2.00597647249389	53	17
minus loglikelihood BFGS	Yes	1.27552755151932	2.00597696486639	41	15
minus loglikelihood BFGS gradient	Yes	1.27552755040258	2.00597654945241	39	15

Questions

- Do we have to estimate for sigma squared or sigma
- Our parabolic function approximation looks weird, are we missing sth?
- are our partial derivatives well calculated?, therefore MLE estimates for mu and sigma?
- Are the gradients of our function gradient well expressed? minus sign?
- Our optim function calls in #3 are returning such mismatched values, one parameter huge and the other really small, our parameters are not similar as the estimates
  - How should we apply this optim, to minimaze or maximize MLL function?