Inverse Problem of Diffusion

Martin Outzen Berild

September 11, 2019

We consider an inverse problem with a basis in the following differential equation

$$\frac{du(x,t)}{dt} = \frac{d^2u(x,t)}{dx^2}, \;\; u(x,0) = h_0(x), \;\; x \in (0,1), t \geq 0. \tag{1}$$

Data is $u(x,t) = h_t(x)$ for a given time $t > 0$. The aim of the inverse problem is $h_0(x)$.

The forward model can be written as

$$u(x,t) = h_t(x) = \frac{1}{\sqrt{4\pi t}} \int e^{-(x-y)^2/(4t)} h_0(y)dy, \;\; t \geq 0. \tag{2}$$

Using discretization we get

$$\mathbf{h_t} = \begin{bmatrix} h_t(x_1) \\ h_t x_2 \\ \vdots \\ h_t x_N \end{bmatrix} = \mathbf{A} \begin{bmatrix} h_0 x_1 \\ h_0 x_2 \\ \vdots \\ h_0 x_N \end{bmatrix} = \mathbf{A h_0}, \tag{3}$$

where a regular grid of N = 100 points is used, such that $x_1 = 0$, $x_2 = 0.01$, $\dots$, $x_N = 0.99$. The sequence x is created in **R** by the the code

```
x = seq(from = 0, to = 0.99, by = 0.01)
```

The interval (0,1) is made into a circle, i.e. 1 corresponds to 0. The matrix A has elements

$$A(i,j) = \frac{0.01}{\sqrt{4\pi t}} e^{-|x_i - x_j|^2/(4t)}. \tag{4}$$

The distance $|x_i - x_j|$ is modular on the circle (0,1). The *createA* function in below calculates the matrix A for a given position x and time t

```
createA <- function(x,t){
  A = diag(0.01/sqrt(4*pi*t), nrow = length(x), ncol = length(x))
  for (i in seq(1,length(x)-1)){
    for (j in seq(i+1,length(x))){
      A[i,j] = 0.01/sqrt(4*pi*t)*exp(-(x[i]-x[j])^2/(4*t))
      A[j,i] = A[i,j]
    }
  }
  A
}
```

Measurements $\mathbf{y} = (\mathbf{y_1}, ..., \mathbf{y_N})'$ are acquired at time t = 0.001 (1ms):

$$y_i = h_t(x_i) + \epsilon_i, \;\; \epsilon_i \sim \mathcal{N}(0, 0.025^2), \;\; \text{iid}. \tag{5}$$

The observations y are downloaded, imported into **R** and converted to vector form.

```
y = read.delim2(file = "OppgA.txt", header = F, sep = "\n", dec = ".")[[1]]
```

The observations are presented in Figure 1

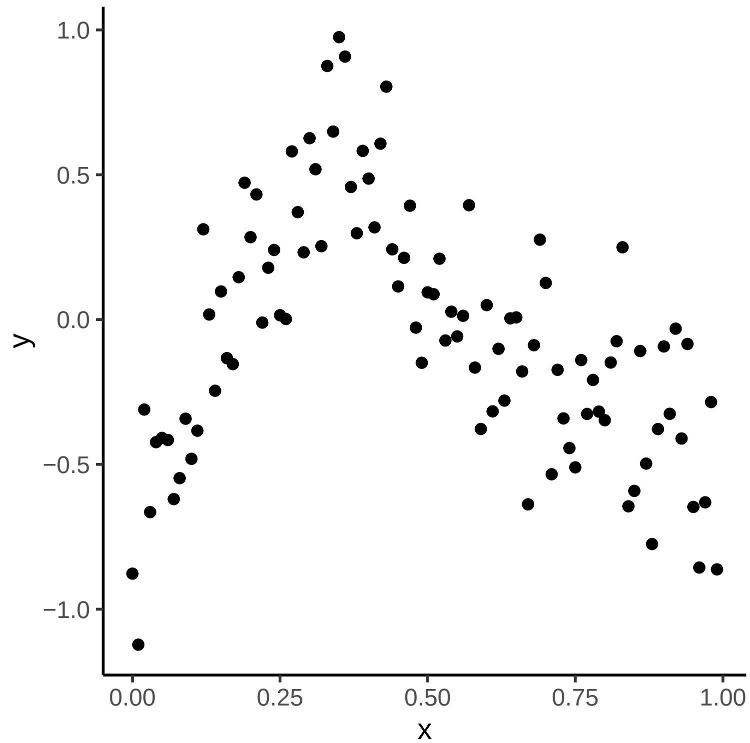


Figure 1: Observations $(y_1, ...y_{100})'$ that are informative of the latent process $h_t(x)$ at time $t = 1$ms.

Exercise a

We want to solve the inverse problem directly by $A^{-1}\mathbf{y}$. First we compute the eigenvalues of the matrix. The observations y are collected at time $t = 1$ms, and we firstly initialize the matrix A.

```
A = createA(x,t = 0.001)
```

The eigenvalues of A can easily be calculated in **R** and are shown in Figure 2.

```
S = eigen(A)[[1]]
```

The singular value decomposition can be found by finding the eigenvectors of A^TA and AA^T. Then since our matrix A is square we can use its eigenvalues in the formula

$$A = USV^T, \tag{6}$$

where U contains the eigenvectors of AA^T, V the eigenvectors of A^TA and S the eigenvalues of A.

```
U = eigen(A%*%t(A))[[2]]
V = eigen(t(A)%*%A)[[2]]
```

We want to approximate this solution using a filter. The approximation is given by

$$\hat{\mathbf{h}}_\mathbf{0} = \sum_{\{\mathbf{i}:\sigma_\mathbf{i}>\mathbf{0}\}} \phi_\mathbf{i}(\alpha)\frac{<\mathbf{u_i},\mathbf{y}>}{\sigma_\mathbf{i}}\mathbf{v_i}, \tag{7}$$

where $\phi_i(\alpha)$ is the filter applied. In our case we want to truncate the small eigenvalue of A, and this is done by the truncated singular value expansion which uses the filter $\phi_i(\alpha) = I\{\sigma_i > \alpha\}$. The choice of α which yields the best solution is not known however.

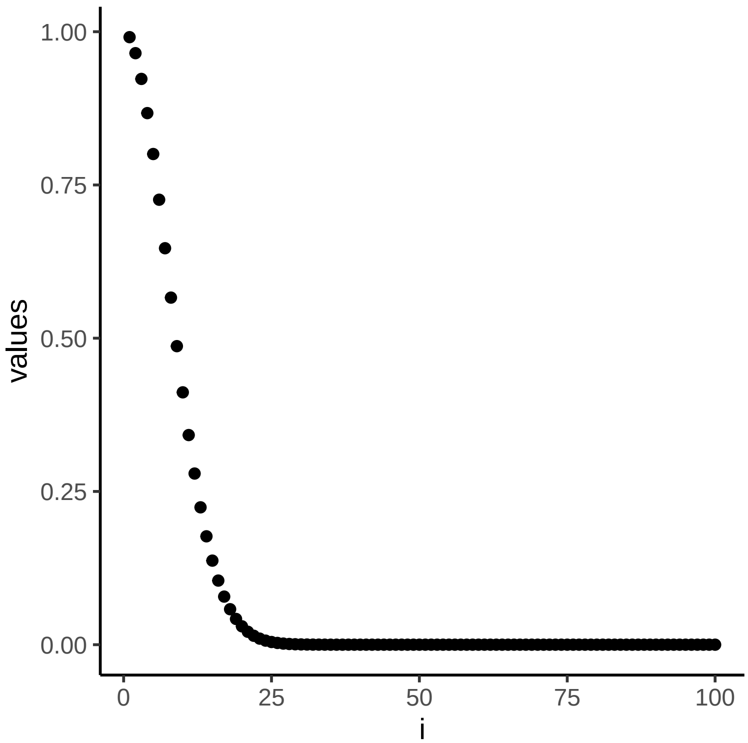
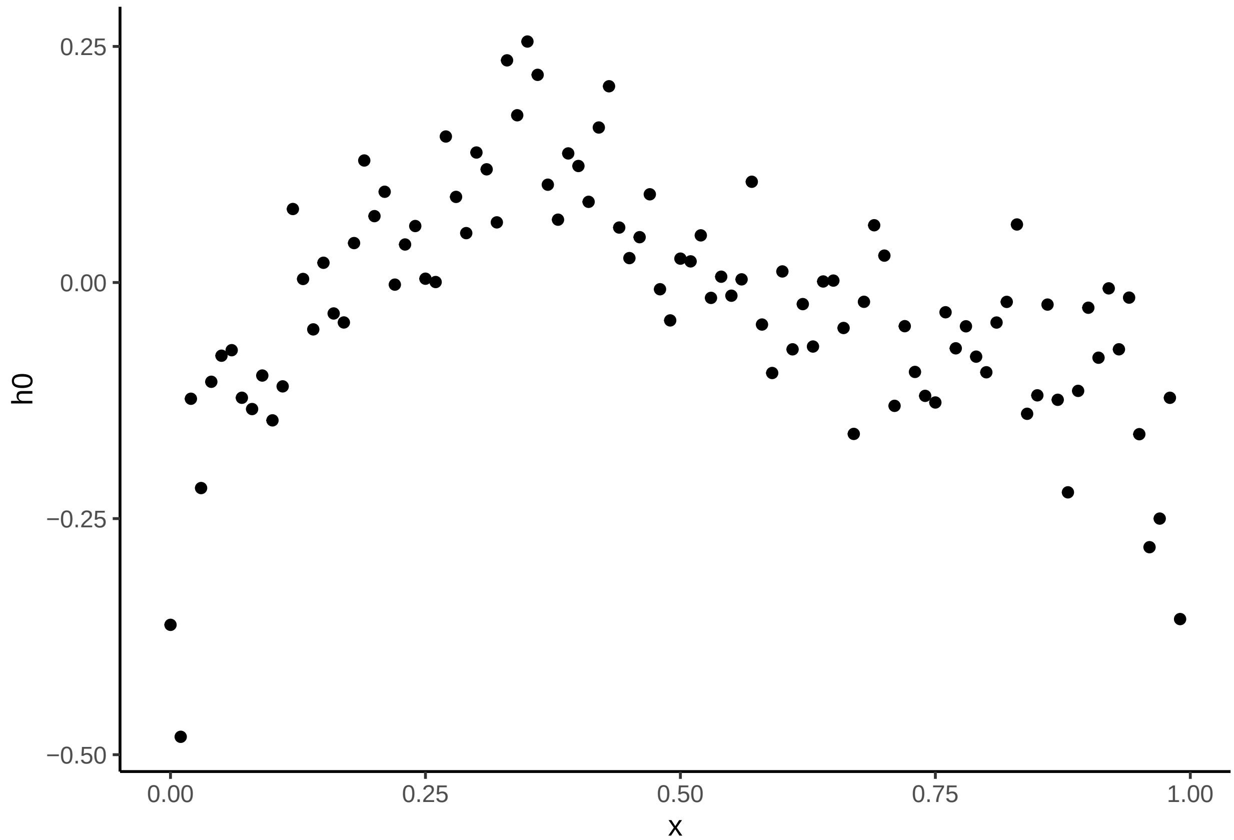
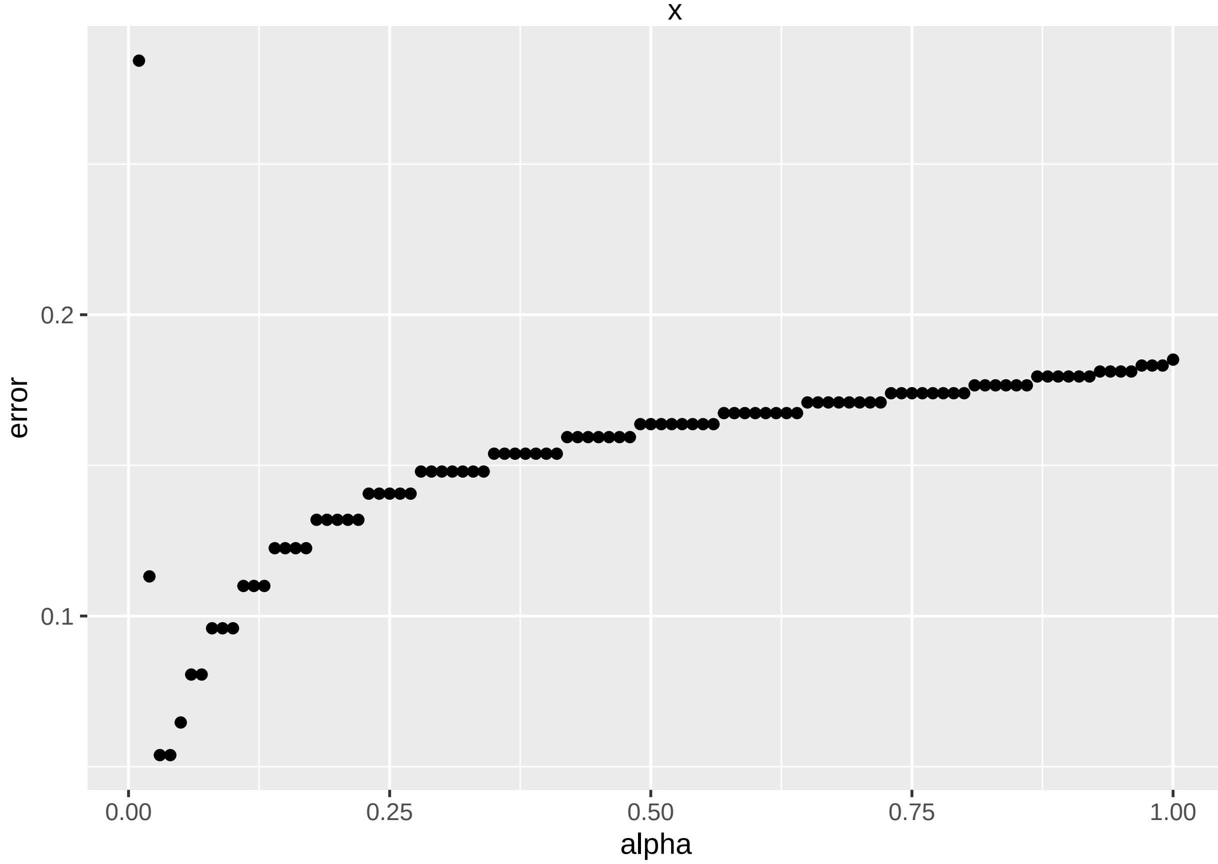


Figure 2: Eigenvalues of A at time $t = 1\text{ms}$

```
trunc.svd <- function(alpha,y,U,S,V){
  res = numeric(length(S))
  for (i in seq(length(S))){
    if(S[i]>alpha){
      res = res + (U[,i]*y)/S[i]*V[,i]
    }
  }
  res
}

h0 <- trunc.svd(alpha = 0.2, y,U,S,V)
```


0.25
0.00
−0.25
−0.50
h0
0.00
0.25
0.50
0.75
1.00
x


0.2
0.1
error
0.00
0.25
0.50
0.75
1.00
alpha

Exercise b

Exercise c