ISC3, Fall 2020 (A22)
Computer works report 002

Pierre Lague

October 7, 2022

1 Exercice 1 : Piecewise linear regression

From a dataset point cloud, we want to achieve a regression using the following function defined on $[0, 1]$:

$$\mathrm{f}_d(x) = \sum_{j=1}^{d} u_j \Lambda_j(x) \text{ where } \Lambda_j(x) = \max\left(0, 1 - (d-1)\left|x - \frac{j-1}{d-1}\right|\right). \tag{1}$$

We're provided with a function that defines a piecewise linear function implemented in `Scilab`.

Consider the following dataset generated by the following `Scilab` script

```
N = 100
d = 5
xi = rand(N, 1)
yi = sin(2*%pi*xi)+0.2*rand(N, 1, "normal")
```

1) Assembling the system matrix
In a `Scilab` script, assemble the matrix $A \in \mathscr{M}_{Nd}(\mathbb{R})$

```
A = zeros(N, d)
for i=1:N
//modifying the i-th column of the matrix
    A(:,i) = max(0, 1-(d-1)*abs(xi-(i-1)/(d-1)))
end
```

2) Solve a normal equation
Now solve the normal equation : $A^T A\boldsymbol{u} = A^T \boldsymbol{y}$

```
coefs = (A'*A)\(A'*yi)
```

This line of code gives us the coefficients of the regression equation.

3) By using the function piecewiselinear(), plot the resulting regression function in solid line. On the same graphics, plot also the point cloud (xi, yi)i=1,...,N with circles for each point. Check if the resulting function f˜(x) is a good regression function.

```
x = linspace(0, 1, 200)
y = piecewiselinear(x, d, coefs)
plot(x, y, 'r');
plot(xi, yi, 'o')
xgrid
title('5 points Piecewise regression')
xlabel('x')
ylabel('y')
legend(['Piecewise Reg.', 'Data Pts.'])
```

The script above gives us the following plot :

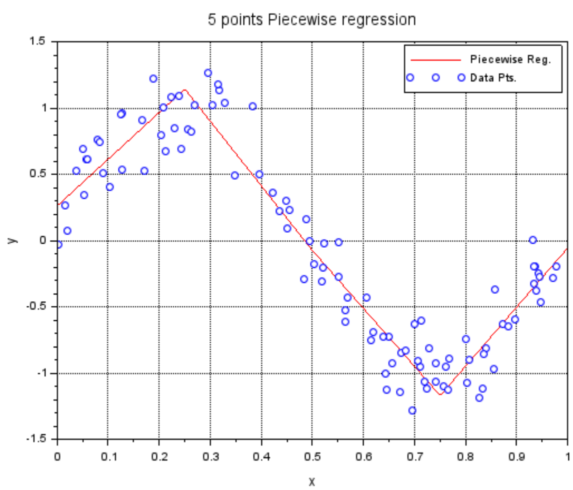

Figure 1: 5 points piecewise regression of our dataset, parameter d influences the quality of the regression.

1.1 Tykhonov Regularization Coefficient

Next, we would like to add a Tykhonov regularization term to the least square minimization problem, and study the effect of the regularization coefficient > 0.

Consider a set of regularization coefficients $\mu k = 10^k$**,** $k = [-8, 2]$**. For each** k **solve the regularized normal equations** $(A^T A + \mu_k * I)\boldsymbol{\mu}_k = A^T \boldsymbol{y}$

The following code answers the question :

```
for k=-8:2
    d = 10
    x = linspace(0, 1, 200)
    //tykhonov regularization coefficient
    mu = 10^k
    //system matrix (same use as ven der monde but we're not looking for a polynomial)
    A_reg = zeros(N, d+1)
    //identity matrix
    I = eye(d+1, d+1)
    //filing the matrixs columns
    for j=1:d+1
        A(:,j) = max(0, 1-(d-1)*abs(xi-(j-1)/(d-1)))
    end
    //solving the linear system to find coefficients.
    coefs_reg = (A'*A+mu*I)\(A'*yi)
    //computing the regression
    y_reg = piecewiselinear(x, d, coefs_reg)
    plot(x, y_reg, 'r');
end
plot(xi, yi, 'o')

xlabel('x')
ylabel('y')
legend(['Regression'])
title('Tykhonov Regularized Regression')
```

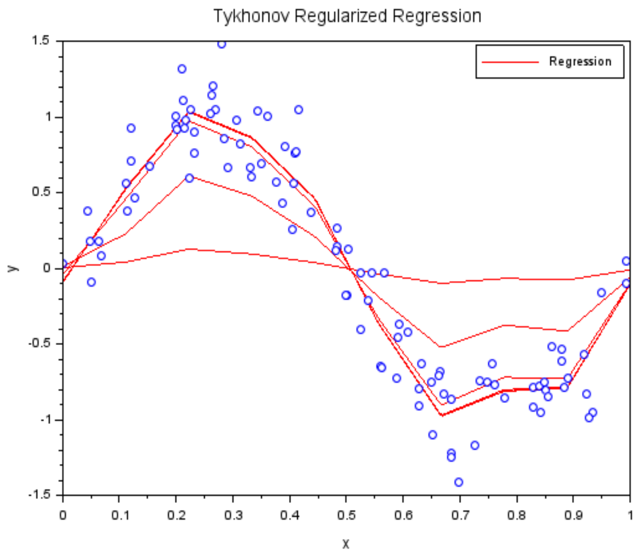

Figure 2: Multiple Tykhonov Regularized Regressions. It seems the smaller μ is the better the regression is.

2) On another graphic, plot the parametric curve $->$ Didn't really figure this one out...

$$\mu_k \mapsto (\|A\boldsymbol{u}_k - \boldsymbol{y}\|, \|\boldsymbol{u}_k\|)^T \quad (2)$$

```
y_reg = piecewiselinear(x, d, coefs_reg)
x_3 = norm((A*coefs_reg) - yi)
y_3 = norm(coefs_reg)
plot(log(x_3), log(y_3), 'o')

xlabel('x')
ylabel('y')
title('Parametric Curve')
```

The previous code gives the following plot:

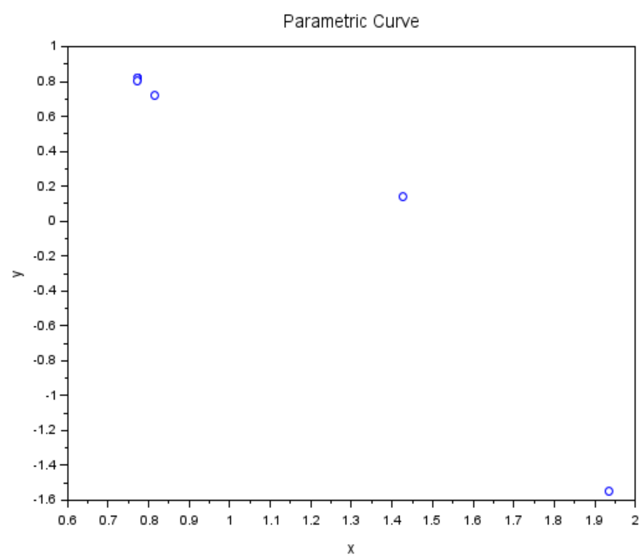

Figure 3: The error vs the coefficients.

This experiment is used to determine what is the best empirical value for μ_k in order to find the best regularized regression. The lower the error is for a μ_k given the better the regression is. In this plot we can see that the best regularization coefficient is $\mu_k ->$ $log(\mu_k)$≈-1.6.