The University of Nottingham

SCHOOL OF MATHEMATICAL SCIENCES

SEMESTER SEMESTER 2024-2025

MATH3027 - OPTIMIZATION

Coursework 1

Deadline: 8 am, Thursday 21/11/2024

Your neat, clearly-legible solutions should be submitted electronically as a pdf file via the MATH3027 Moodle page by the deadline indicated there. A scan of a handwritten solution is acceptable. As this work is assessed, your submission must be entirely your own work (see the University's policy on Academic Misconduct).

Submissions up to five working days late will be subject to a penalty of 5% of the maximum mark per working day.

Deadline extensions due to Support Plans and Extenuating Circumstances can be requested according to School and University policies, as applicable to this module. Because of these policies, solutions (where appropriate) and feedback cannot normally be released earlier than 10 working days after the main cohort submission deadline.

You should submit a report electronically as a pdf file with the computational code in the appendix. You may be requested to submit your code (read instructions on Moodle). Comment on all your results.

PROBLEM 1: REGULARIZED LEAST SQUARES

In this exercise, you will work with a noisy dataset that approximates an exponential decay function. Your task is to fit a regularized polynomial to approximate the derivative of the underlying function. You will explore the effect of regularization on the polynomial approximation by experimenting with different values of the regularization parameter λ. Thus, for the given ODE $x'(t) = f(x)$, the objective of this problem is to determine $f(x)$ that models $x'(t)$. That is,

$$\min_f |x'(t) - f(x)|^2. \tag{1}$$

Here, $x'(t)$ is not explicitly given. Instead, we have noisy data for $x(t)$.

INSTRUCTIONS

Data Generation

- Read the time vector `t_data`, 100 points long, with values between 0 and 2.5 along with the data vector `x_data` of same length.
- (For plotting comparison) Define the true function $x(t) = e^{-t}$ (which results from solving the ordinary differential equation $x'(t) = f(x)$).

Use the following code to read the noisy data:

```python
import numpy as np

# Load the data from the CSV file
data = np.loadtxt('t_data_x_noisy.csv', delimiter=',', skiprows=1)

# Split the columns into time and noisy data
t_data = data[:, 0]
x_noisy = data[:, 1]
```

Listing 1: Load dataset

Spline Approximation

- Fit a smooth spline to the noisy data.
- Compute the derivative of the spline to approximate the derivative $x'(t)$.

```python
from scipy.interpolate import UnivariateSpline

# Fit the noisy data using a spline to approximate x'(t)
spline_fit = UnivariateSpline(t_data, x_noisy, s=0.5)  # Smoothing spline
spline_derivative = spline_fit.derivative()            # Derivative of the spline

# Get the derivative values from the spline approximation
x_prime_approx = spline_derivative(t_data)  # This is the approximation of x'(t)
```

Listing 2: Fit data

Least Squares Problem with Regularization

You should solve the following regularized least squares problem:

$$\min_c \left(\|Ac - b\|^2 + \lambda \int (f''(x))^2 \, dx \right),$$

where $f(x) \approx Ac$, $b \approx x'(t)$, and the integral term is approximated by a Riemann sum:

$$\int (f''(x))^2 \, dx \quad \approx \quad \sum_{i=1}^{m} (f''(x_i))^2 \, \Delta x.$$

For simplicity, we will neglect Δx and write the integral as

$$\int (f''(x))^2 \, dx \approx \|Dc\|^2.$$

a) Write a code to fit a polynomial of degree n to the approximated derivative $x'(t)$ using a regularized least squares approach. The regularization should be the second derivative of the polynomial (w.r.t. x) with parameter λ.

b) For a polynomial of degree 5, test different values of λ: 0, 1, 100, and 10,000. Generate a plot that compares the polynomial approximations for different values of λ (0, 1, 100, and 10,000) against the true derivative $x'(t)$. Compute and present the condition number of the normal equations for polynomial degree 5 and each regularization parameter.

Analysis

c) Comment on how the regularization parameter λ affects the fit of the polynomial to the derivative $x'(t)$. How does increasing or decreasing λ influence the smoothness of the approximation?

ADDITIONAL GUIDANCE

- You will need to construct a design matrix for the polynomial ’fitting’ (LSQ) and solve the regularized normal equations using `numpy.linalg.solve`.
- To penalize the derivative of the polynomial w.r.t. x, compute the derivative of the basis functions for the polynomial and add this term to the regularized normal equation.

Challenge: Explain why regularizing the derivative of the polynomial is necessary in this context.

PROBLEM 2: GAUSS-NEWTON METHOD

Weevils are small beetles with elongated snouts. As herbivorous pests, they damage crops and stored food by feeding on seeds and grains. Their rapid reproduction and resistance to pest control can cause significant agricultural and economic losses.

Experiments are conducted to study the influence of weevil population density on their fertility. For $i = 1, 2, \dots, 14$, let t_i represent the number of weevil couples in the first generation, and Y_i represent the corresponding average number of adult weevils in the second generation in repeated trials. The experiments yield the following data:

Table 1:

i	t_i	Y_i
1	1	77.5
2	2	136.2
3	4	240.4
4	8	356.0
5	16	505.6
6	24	643.2
7	32	700.8
8	48	720.0
9	64	710.4
10	96	748.8
11	128	666.0
12	192	614.4
13	256	588.8
14	384	153.6

Consider the following model that links Y_i and t_i,

$$Y_i = f(t_i; \mathbf{x}) := x_1 t_i \exp(-x_2 t_i^{x_3})$$

for some constant vector $\mathbf{x} = (x_1, x_2, x_3)^\top \in \mathbb{R}^3$. Let $r_i(x) = f(t_i; x) - Y_i$ and $r(x) = (r_1(x), r_2(x), \dots, r_{14}(x))^\top$.

a) Produce a scatter plot of the data and plot f with the following vectors $\mathbf{x}$ on the same diagram:

$$\mathbf{x}_1 = (200, 0.3, 0.5)^\top, \mathbf{x}_2 = (120, 0.25, 0.5)^\top, \mathbf{x}_3 = (180, 0.2, 0.7)^\top$$

b) Suppose that a Gauss-Newton method is used to solve the optimization problem

$$\min_{x \in \mathbb{R}^3} \left\{ g(x) := \|r(x)\|^2 \right\}.$$

Following the notations in the notes, write down $J(x^k)$ in the Gauss-Newton method explicitly.

c) Using a stopping rule of $\|\nabla g(x^{k+1})\| \leq 10^{-6}$, implement pure Gauss-Newton method starting with x_1, x_2 and x_3 respectively. Print the number of iterations in each run. Write down your observation.

d) With the same stopping rule, implement a damped Gauss-Newton method with constant stepsizes of 0.8, 0.6 and 0.4 respectively, for the starting point that did not produce a convergent sequence in part c). Write down your observation.

e) What could we learn from the previous parts regarding the selection of starting point and stepsize?