02610 Optimization and Data Fitting

Exercises for Week 8

1 Exponential Fit (in Python)

In this exercise, we apply the variable projection method to solve the exponential fit problem in Exercise 3 from last week. We try to fit the data in `data_exe3.mat` with the function

$$\phi(\boldsymbol{x}, t) = x_1 e^{x_3 t} + x_2 e^{x_4 t}.$$

Note that the fit function is slightly different comparing with the one in Exercise 3 from last week.

1. Write a Python function to return the residual $\boldsymbol{r}(\boldsymbol{a}_k)$ (also called as the variable projection of $\boldsymbol{y}$), the Jacobian $J(\boldsymbol{a}_k)$, and the linear coefficients $\boldsymbol{c}(\boldsymbol{a}_k)$, where $\boldsymbol{a} = [x_3, x_4]^T$ and $\boldsymbol{c} = [x_1, x_2]^T$. You can write this function by completing the following Python code:

```
def fun_All(a, t, y):

# obtain F(a)

# compute c by calling linearLSQ

# compute the residual, i.e. the variable projection of y

# compute the Jacobian
```

2. Download the Python function `variable_projection` and save in the same folder as the function written in the previous question. Set the same starting point as in Exercise 3.3 from last week, i.e., $\boldsymbol{a} = [-1, -2]^T$, and call

```
xopt, stat = variable_projection_method(fun_All, a0, None, t, y)
```

   Do you get the same solution as applying Gauss-Newton method directly on the nonlinear data fitting problem?

3. Plot $\|\nabla f(\boldsymbol{a}_k)\|_2$ and $f(\boldsymbol{a}_k)$ as functions of the iteration number. Do you need more or less iterations than applying Gauss-Newton?

4. Plot all data as points and the fit function as a curve.

2 Minimization of a quadratic function

We consider the strictly convex quadratic optimization problem

$$\min_{(x_1,x_2)\in\mathbb{R}^2} f(x_1,x_2) = 3x_1^2 + 5x_2^2 + 2x_1x_2 + 7x_1 + 3x_2 + 5. \tag{1}$$

1. (By hand) Show that (1) can be expressed as

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T H\boldsymbol{x} + \boldsymbol{g}^T\boldsymbol{x} + \gamma$$

   where H is symmetric. What are the values of H, $\boldsymbol{g}$ and γ? What is the value of n?

2. (By hand) What is the gradient $\nabla f(\boldsymbol{x})$ and the Hessian $\nabla^2 f(\boldsymbol{x})$?

3. (In Python) Install `cvxopt`, and use the Python function `cvxopt.solvers.qp`, solve the unconstrained minimization problem (1). What is the solution?

4. (In Python) Use `cvxopt.solvers.qp` to solve the minimization problem (1) with the constraint $3x_1 + x_2 \leq -5$. What is the solution now?

3 Data fitting with different norms (in Python)

We try to fit the data shown in the following table

t_i	-1.5	-0.5	0.5	1.5	2.5
y_i	0.80	1.23	1.15	1.48	2.17

by a fit function in the form of $\phi(\boldsymbol{x}, t) = x_1 t + x_2$.

1. Call your Python function `linearLSQ` to find the least-squares fit solution $\boldsymbol{x}^*_{(2)}$, and calculate the objective function value $f_{(2)}(\boldsymbol{x}^*_{(2)}) = \|\boldsymbol{r}(\boldsymbol{x}^*_{(2)})\|_2$.

2. Call the Python function `scipy.optimize.linprog` to find the l_1 regression solution $\boldsymbol{x}^*_{(1)}$, and calculate the objective function value $f_{(1)}(\boldsymbol{x}^*_{(1)}) = \|\boldsymbol{r}(\boldsymbol{x}^*_{(1)})\|_1$.

   In fact the l_1 regression solution is not unique. All

$$\boldsymbol{x}_{(1)} = \begin{bmatrix} 0.227 \\ 1.140 \end{bmatrix} + \alpha \begin{bmatrix} 2 \\ 3 \end{bmatrix} \quad \text{for } 0 \leq \alpha \leq 0.0579$$

   give you the same minimal value $f_{(1)}$. Please pick up one value for α and check if it gives the same minimal value as what you got with $\boldsymbol{x}^*_{(1)}$.

3. Call the Python function `scipy.optimize.linprog` to find the l_∞ regression solution $\boldsymbol{x}^*_{(\infty)}$, and calculate the objective function value $f_{(\infty)}(\boldsymbol{x}^*_{(\infty)}) = \|\boldsymbol{r}(\boldsymbol{x}^*_{(\infty)})\|_\infty$.

4. Plot the data as points and all 3 fit functions as straight lines.

5. Download and run the Python script `contourplot.py`. It creates three contour plots for the objective functions $f_{(2)}$, $f_{(1)}$ and $f_{(\infty)}$. From the contour plots, can you see that for l_2 and l_∞ the minimizer is unique, but for l_1 the minimizer is not unique? In addition, the function with smoother contours is usually easier to solve. Then, which one would be the easiest to solve?

6. Now, we study how sensitive these three regression solutions to outliers. Change the data (t_5, y_5) from $(2.5, 2.17)$ into $(2.5, 4)$. Then, use the Python function `linearLSQ` and `scipy.optimize.linprog` to find the/a minimizer of l_2, l_1 and l_∞ regression solutions. Plot them in the same figure created in Question 4 but with dashed lines. Which one is the most robust with respect to the outliers?