Assignment for Research and Development / AI

Madhava Narayanan KJ

November 9, 2025

Objective

Given points lying on a parametric curve for $6 < t < 60$, identify the values of the unknown variables θ, M, and X in:

$$x = t\cos(\theta) - e^{M|t|}\sin(0.3t)\sin(\theta) + X$$
$$y = 42 + t\sin(\theta) + e^{M|t|}\sin(0.3t)\cos(\theta)$$

Step 1: Data Inspection

Loaded the data using pandas:

```
import pandas as pd
data = pd.read_csv("xy_data.csv")
print(data.head())
print(data.info())
```

The data consists of 1500 rows of (x, y) pairs with no missing values.

Step 2: Visualization

Plotted the raw data for inspection:

```
import matplotlib.pyplot as plt
plt.figure(figsize=(8, 5))
plt.plot(data['x'], data['y'], 'ro', label='Data Points')
plt.xlabel('x')
plt.ylabel('y')
plt.title('Observed Data Points (x vs y)')
plt.legend()
plt.show()
```

Step 3: Model Definition

Defined the parametric equations in Python:

```
import numpy as np
def model(t, theta, M, X):
    x_pred = t * np.cos(theta) - np.exp(M * np.abs(t)) * np.sin(0.3 * t) * np.sin(theta)
    y_pred = 42 + t * np.sin(theta) + np.exp(M * np.abs(t)) * np.sin(0.3 * t) * np.cos(t
    return x_pred, y_pred
```

Step 4: Loss Function

Used mean L1 loss as the objective:

```
def l1_loss(params, t, x_obs, y_obs):
    theta, M, X = params
    x_pred, y_pred = model(t, theta, M, X)
    return np.mean(np.abs(x_obs - x_pred) + np.abs(y_obs - y_pred))
```

Step 5: Parameter Estimation

Since t was not present, generated it uniformly:

```
n_points = len(data)
t_vals = np.linspace(6, 60, n_points)
x_obs = data['x'].values
y_obs = data['y'].values
```

Applied bounded optimization:

```
from scipy.optimize import minimize

bounds = [
    (np.deg2rad(0.01), np.deg2rad(50)), # theta in radians
    (-0.05, 0.05),                      # M
    (0, 100)                            # X
]
initial_guess = [np.deg2rad(25), 0.0, 50.0]

result = minimize(
    l1_loss,
    initial_guess,
    args=(t_vals, x_obs, y_obs),
    bounds=bounds,
    method='L-BFGS-B'
)
theta_best, M_best, X_best = result.x
```

Step 6: Results

- θ (degrees): **28.12**
- M: **0.0214**
- X: **54.90**
- Final L1 Loss: **25.24**

Step 7: Final Equations

The estimated parametric equations:

$$x = t\cos(28.12^\circ) - \exp(0.0214t)\sin(0.3t)\sin(28.12^\circ) + 54.90$$

$$y = 42 + t\sin(28.12^\circ) + \exp(0.0214t)\sin(0.3t)\cos(28.12^\circ)$$

Step 8: Visualization of Fit

```
x_pred, y_pred = model(t_vals, theta_best, M_best, X_best)
plt.figure(figsize=(8,5))
plt.plot(x_obs, y_obs, 'ro', label='Observed')
plt.plot(x_pred, y_pred, 'b-', label='Predicted', alpha=0.7)
plt.xlabel('x')
plt.ylabel('y')
plt.title('Curve Fit: Data vs Model')
plt.legend()
plt.show()
```

Conclusion

Optimization provided a close fit of model parameters to the observed data. The steps cover data inspection, modeling, loss minimization, and parameter reporting, as required by the assignment.