HW 01: Constraint Optimization
CPE 490/590 Machine Learning for Engineering Applications

Instructor: Rahul Bhadani

Due: April 05, 2024: 11:59 PM

Total Points: 25

---

1 Theory

1.1 Constraint Optimization (10 points)

Consider the function $f(x,y) = e^{xy}$ subject to constraint $x^3 + y^3 = 16$. Use Lagrange multipliers to find the coordinates (x,y) of any points subject to the given constraint where the function f could attain a maximum or minimum.

Answer

We wish to find the extreme values of the function f subject to the constraint $g(x,y) = x^3 + y^3 - 16$
Using Lagrange multiplier:

$$\begin{aligned} \nabla f &= \lambda \nabla g \\ \Rightarrow ye^{xy} &= 3\lambda x^2 \\ \Rightarrow xe^{xy} &= 3\lambda y^2 \end{aligned} \tag{1}$$

In this case, if with either $x = 0$ or $y = 0$, it would make $x = 0, y = 0$ based on Equation (1) and would violate our constraint $g(x,y) = x^3 + y^3 - 16$. Hence, clearly $x \neq 0$, $y \neq 0$.
Then

$$\lambda = \frac{ye^{xy}}{3x^2} = \frac{xe^{xy}}{3y^2}$$

based on Equation (1).
That gives as $x^3 = y^3 \Rightarrow x = y$. Putting back this in our constraint, we get $2x^3 = 16 \Rightarrow x = 2, \Rightarrow y = 2$.
Thus, at the point $(2,2)$, a point can attain optimum value.

1.2 Relation between Constraint Optimization problem and SVM (5 points)

Write down the formulation (i.e. mathematical equation) of the constraint optimization problem used in Support Vector Machine (SVM).

Answer

Constraint optimization problem in SVM is

$$\text{minimize} \quad \frac{1}{2}||w||^2$$

subject to

$$y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1$$

$\forall i = [1, n]$.

where the Data is $\{\mathbf{x}_i, y_i\}_{i=[1,n]}$.

2 Practice

2.1 Constraint Optimization in Python (10 points)

Perform constraint optimization for the objective function defined in Q1.1 using Python. (i) Choose the initial guess of $x = 0.0$, and $y = 0.001$, and report the value of x and y at which f achieves maximum or minimum; (ii) Choose the initial guess of $x = 0.01$, and $y = 0.01$, and report the value of x and y at which f achieves maximum or minimum.

Answer

(i) The value of x and y are -5.5626 and 5.7299 respectively.

(ii) The value of x and y are 2.0 and 2.0 respectively.

Code is below:

```
import numpy as np
from scipy.optimize import minimize

# Define the objective function
def objective(x):
    x, y = x
    return np.exp(x*y)
```

```python
# Define the constraint function
def constraint(x):
    x, y = x
    return ((x**3) + (y**3) - 16)

# Initial guess for x and y
x0 = np.array([0.0, 0.001])

# Set up the optimization problem
constraints = {'type': 'eq', 'fun': constraint}
result = minimize(objective, x0, constraints=constraints)

# Print the results
print("Optimal values are achieved at:")
print(f"x = {result.x[0]:.4f}")
print(f"y = {result.x[1]:.4f}")
print(f"Minimum value of f(x, y) = {result.fun:.4f}")


# %%
import numpy as np
from scipy.optimize import minimize

# Define the objective function
def objective(x):
    x, y = x
    return np.exp(x*y)

# Define the constraint function
def constraint(x):
    x, y = x
    return ((x**3) + (y**3) - 16)

# Initial guess for x and y
x0 = np.array([0.01, 0.01])

# Set up the optimization problem
constraints = {'type': 'eq', 'fun': constraint}
result = minimize(objective, x0, constraints=constraints)

# Print the results
print("Optimal values are achieved at:")
print(f"x = {result.x[0]:.4f}")
print(f"y = {result.x[1]:.4f}")
print(f"Optimal value of f(x, y) = {result.fun:.4f}")


# %%
import numpy as np
import plotly.graph_objects as go

# Define the objective function f(x, y)
def f(x, y):
    return np.exp(x*y)

# Define the constraint function g(x, y)
def g(x, y):
```

```
    return ((x**3) + (y**3) - 16)

# Create a meshgrid for x and y values
x_vals = np.linspace(-1.5, 1.5, 100)
y_vals = np.linspace(-3.5, 3.5, 100)
X, Y = np.meshgrid(x_vals, y_vals)
Z_f = f(X, Y)
Z_g = g(X, Y)

# Create the 3D surface plot
fig = go.Figure()

# Add the surface for f(x, y)
fig.add_trace(go.Surface(z=Z_f, x=X, y=Y, colorscale='Viridis', name='f(x, y)'))

# Add the constraint surface for g(x, y)
fig.add_trace(go.Surface(z=Z_g, x=X, y=Y,
                         colorscale='Reds',
                         opacity=0.5, showscale=False,
                         name='4x^2 + y^2 = 9'))

# Set layout
fig.update_layout(
    title='3D Surface Plot',
    scene=dict(
        xaxis_title='x',
        yaxis_title='y',
        zaxis_title='Value',
    ),
    margin=dict(l=0, r=0, b=0, t=40),
)

# Show the interactive plot
fig.show()
```