Linear Programming Assignment Report

CS164 - Professor Shekhar, R.

Ash Nguyen

2018/12/06

1 KKT Conditions For Linear Programs

A linear program can be defined as:

$$\min_x c^T x$$

subject to:

$$Ax \leq b$$

The KKT conditions for the above linear program is:

Stationarity:

$$\nabla_x c^T x + \sum_{i=1}^{m} \lambda_i \nabla_x (Ax - b) = 0$$

for each and all x

Complementary slackness:

$$\lambda_i \geq 0 \quad i = 1, 2, ..., m$$

and

$$\lambda_i (A_i x - b_i) \geq 0 \quad i = 1, 2, ..., m$$

If the Lagrange multiplier for a linear constraint is positive then the constraint is considered active, otherwise it is inactive. If a constraint is active, the optimal solution is in the hyperplane describe by the equality of that constraint.

Geometrically, the sub-level set for a linear objective function will be a hyperplane, and the linear constraints will specified a simplex form by the union of the halfspaces defined by the constraints. Therefore, the optimal in which the stationarity condition is satisfied should be where the hyperplane objective function "touches" the simplex, because that is where the gradient of the halfspaces be parallel. Intuitively, if an optimal solution is found inside the simplex (inside the feasible region) then by definition we can follow the negative direction of the gradient of the objective function to decrease the objective function further, therefore rendering it not the actual optimal solution. And since a simplex is a convex set, any direction that is the negative direction of the gradient that we choose to move in will also be contained inside the set (or the feasible region), which makes the solution after the move be a solution that satisfied the original constraints. Each constraint in the case of linear programming will defines a facet of the simplex, and as we reasoned the optimal solution should only be found in these facets.

2 Defining Norm Problems As Linear Programs

The l_1-norm cost function is:

$$\min_{\Theta} = ||Y - X\Theta||_1 = \sum_{i=1}^{N} |Y_i - X_i\Theta|$$

To write this as a linear program, we define N slack variables t_i such that

$$t_i \geq |Y_i - X_i\Theta|$$

Rewriting all t_i in terms of a vector t. Therefore we have the problem of

$$\min_{\Theta,t} 1^T t$$

such that

$$Y_i - X_i\Theta \leq t_i \quad i = 1, 2, ..., N$$

and

$$Y_i - X_i\Theta \geq -t_i \quad i = 1, 2, ..., N$$

which is a linear program.

The l_∞-norm cost function is

$$\min_{\Theta} = ||Y - X\Theta||_\infty = max(|Y_i - X_i\Theta|) \quad i = 1, 2, ..., N$$

Because the infinity norm is bounded above by the max function, we can add one slack variable t such that

$$t \geq max(|Y_i - X_i\Theta|) \geq |Y_i - X_i\Theta| \quad i = 1, 2, ..., N$$

Therefore we can rewrite the problem in a linear program form as

$$\min_{\Theta,t} t$$

such that

$$Y_i - X_i\Theta \leq t \quad i = 1, 2, ..., N$$

and

$$Y_i - X_i\Theta \geq -t \quad i = 1, 2, ..., N$$

3 Optimization With CVXPY

Using CVXPY for the synthetic dataset (with random seed 251 for reproducibility), we have the following results. The best fit lines are plotted in Figure 1. The code is included in the appendix.

For l_1 norm, $[\theta_1, \theta_2] = [2.20622802, -16.83757968]$.

For l_∞ norm, $[\theta_1, \theta_2] = [2.03632014, 1.10850232]$.

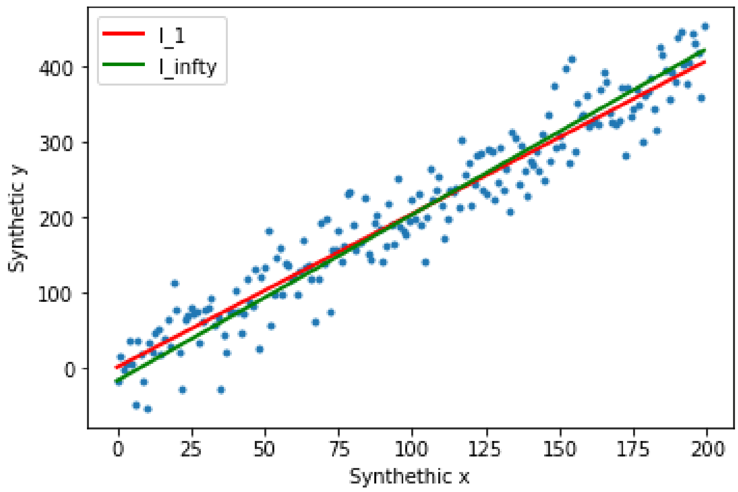

Figure 1: Best Fit Lines For l_1 and l_∞ Norms Of The Synthetic Dataset

APPENDIX

```python
import numpy as np
from cvxpy import *
import matplotlib.pyplot as plt

np.random.seed(251)
# generate a synthetic dataset

# actual parameter values
theta1_act = 2
theta2_act = 5

# Number of points in dataset
N = 200

# Noise magnitude
mag = 30

# datapoints
x_org = np.arange(0,N)
y_org = theta1_act * x_org + theta2_act *np.ones([1,N]) + np.random.normal(0,mag,N)



y=y_org.transpose()
x=x_org.transpose()
ones=np.ones(N)
x=np.column_stack((x,ones))


#l-infty norm
theta_inf = Variable(2,1)
t=Variable()
objective_inf = Minimize(t)
constraints_inf = [y-x*theta_inf<=t*ones, y-x*theta_inf>=-t*ones]
prob_inf = Problem(objective_inf, constraints_inf)
prob_inf.solve()
print( "status:", prob_inf.status)
print( "optimal value", prob_inf.value)
print ("optimal var", theta_inf.value)

#l-1 norm
theta_1 = Variable(2,1)
t_1=Variable(N,1)
objective_1=Minimize(ones.transpose()*t_1)
constraints_1= [y-x*theta_1<=t_1, y-x*theta_1>=-t_1]

prob_1 = Problem(objective_1, constraints_1)
prob_1.solve()
print( "status:", prob_1.status)
print( "optimal value", prob_1.value)
```

```
print ("optimal var", theta_1.value)

plt.figure()
# Scatter plot of data

plt.scatter(x_org,y_org,marker='.')

# Plot of l-1 norm
y_1=x*theta_1.value
print(y_1.shape)
ash1=plt.plot(x_org,y_1,color='red',label='l_1',linewidth=2.0)

# Plot of l-infty norm
y_inf=x*theta_inf.value
ash2=plt.plot(x_org,y_inf,color='green', label='l_infty',linewidth=2.0)

plt.legend()
plt.xlabel('Synthethic x')
plt.ylabel('Synthetic y')
plt.show()
```