VE485, Optimization in Machine Learning (Summer 2020) **Homework: 7. Duality**

7. Duality

Lecturer: Xiaolin Huang `xiaolinhuang@sjtu.edu.cn`

Student: Chongdan Pan `pandddddda@sjtu.edu.cn`

Problem 1

Consider the following *non-convex* problem,

$$\begin{array}{ll} \min & -3x_1^2 + x_2^2 + 2x_3^2 + 2(x_1 + x_2 + x_3) \\ \text{s.t.} & x_1^2 + x_2^2 + x_3^2 = 1. \end{array}$$

Please give the KKT conditions, find all x and ν that satisfy the KKT conditions, give the optimal solution, and verify the strong duality.

Answer.
First we can construct a Lagrange function:

$$L(x_1, x_2, x_3, v) = -3x_1^2 + x_2^2 + 2x_3^2 + 2(x_1 + x_2 + x_3) + v(x_1^2 + x_2^2 + x_3^2 - 1)$$

Then the KKT condition is:

$$\frac{\partial L}{\partial x_1} = (2v - 6)x_1 + 2 = 0 \rightarrow x_1 = -\frac{1}{v - 3}$$

$$\frac{\partial L}{\partial x_2} = (2v + 2)x_2 + 2 = 0 \rightarrow x_2 = -\frac{1}{v + 1}$$

$$\frac{\partial L}{\partial x_3} = (2v + 4)x_3 + 2 = 0 \rightarrow x_3 = -\frac{1}{v + 2}$$

$$\frac{\partial L}{\partial v} = x_1^2 + x_2^2 + x_3^2 - 1 = 0 \rightarrow \frac{1}{(v-3)^2} + \frac{1}{(v+1)^2} + \frac{1}{(v+2)^2} = 1$$

Through mathematica we get four sets of solutions:

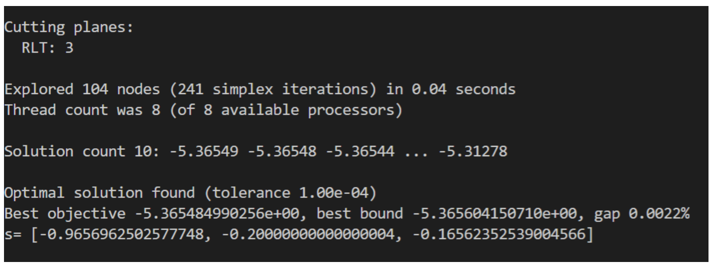
$$\begin{cases} v = 4.035 \\ x_1 = -0.966 \\ x_2 = -0.199 \\ x_3 = -0.166 \end{cases} \text{with } f(x) = -5.367$$

$$\begin{cases} v = -3.149 \\ x_1 = 0.163 \\ x_2 = 0.465 \\ x_3 = 0.87 \end{cases} \text{with } f(x) = 4.647$$

$$\begin{cases} v = 1.892 \\ x_1 = 0.903 \\ x_2 = -0.346 \\ x_3 = -0.257 \end{cases} \text{with } f(x) = -1.592$$

$$\begin{cases} v = 0.224 \\ x_1 = 0.36 \\ x_2 = -0.450 \\ x_3 = -0.817 \end{cases} \text{with } f(x) = -1.13$$

Hence the optimal solution is achieved through the first solution with value equals to -5.367, which is the optimal solution for L^*. If we plug the solution into $f(x_1, x_2, x_3)$ we can get the same value -5.367 as well.



```
Cutting planes:
  RLT: 3

Explored 104 nodes (241 simplex iterations) in 0.04 seconds
Thread count was 8 (of 8 available processors)

Solution count 10: -5.36549 -5.36548 -5.36544 ... -5.31278

Optimal solution found (tolerance 1.00e-04)
Best objective -5.365484990256e+00, best bound -5.365604150710e+00, gap 0.0022%
s= [-0.9656962502577748, -0.20000000000000004, -0.16562352539004566]
```

Figure 1: Solution from Gurobi, which is same to the dual problem

We can verify the solution through Gurobi package in python, and it gives the same solution hence it's strong duality.

Problem 2

Consider the following problem

$$\begin{aligned} \min_{x,z,\xi,\rho} \quad & \frac{1}{2} x^\top x - \nu\xi + \frac{1}{m} \sum\nolimits_{i=1}^{m} \rho_i \\ \text{s.t.} \quad & b_i(x^\top \phi(a_i) + z) \geq \xi - \rho_i, \forall i = 1, 2, \ldots, m \\ & \xi \geq 0, \rho_i \geq 0, \forall i = 1, 2, \ldots, m, \end{aligned}$$

and denote its optimal solution as x^*, z^*, ξ^*, ρ^*.

You are asked to prove that ν is an upper bound on the fraction of margin errors, i.e., the number of samples falling in the margin is less than νm:

$$\#\{i : b_i f^*(a_i) < \xi^*\} \leq \nu m,$$

where $f^*(a) = {x^*}^\top \phi(a) + z^*$

Answer. Aaccording to the process for SVM:

$$L(x, z, \rho, \xi, \lambda, v, u) = \frac{1}{2} x^T x - \nu\xi + \frac{1}{m} \sum_i^m \rho_i + \sum_i^m \lambda_i(\xi - \rho_i - b_i(x^\top \phi(a_i) + z)) - \sum_i^m v_i \rho_i - u\xi$$

Then for the original problem, our object becomes:

$$\min_{x,z,\rho,\xi} \max_{\lambda_i, v_i, u} \frac{1}{2} x^T x - \nu\xi + \frac{1}{m} \sum_i^m \rho_i + \sum_i^m \lambda_i(\xi - \rho_i - b_i(x^\top \phi(a_i) + z)) - \sum_i^m v_i \rho_i - u\xi$$

s.t.

$$\lambda_i, v_i, u \geq 0, \forall i$$

We can first calculate the minimization problem through gradient:

$$\frac{\partial L}{\partial x} = 0 \rightarrow x = \sum_i^m \lambda_i b_i \phi(a_i)$$

$$\frac{\partial L}{\partial z} = 0 \rightarrow \sum_i^m \lambda_i b_i = 0$$

$$\frac{\partial L}{\partial \rho} = 0 \rightarrow v_i = \frac{1}{m} - \lambda_i$$

$$\frac{\partial L}{\partial \xi} = 0 \rightarrow \sum_i^m \lambda_i = u + \nu$$

Plug it back we get:

$$\max_{\lambda_i} -\frac{1}{2} \sum_j^m \sum_i^m \lambda_i b_i \phi(a_i) \phi(a_j) b_j \lambda_j$$

s.t.

$$\frac{1}{m} \geq \lambda_i \geq 0, \forall i$$

$$\sum_i^m \lambda_i b_i = 0$$

$$\sum_i^m \lambda_i \geq \nu$$

For the KKT condition we have:

$$\begin{cases} \lambda_i, v_i, u_i, \rho_i \geq 0, \xi \geq 0, \forall i \\ b_i(x^\top \phi(a_i) + z) + \rho_i - \xi \geq 0 \\ \lambda_i(\xi - \rho_i - b_i(x^\top \phi(a_i) + z)) = 0 \\ u\xi = 0 \\ \sum_i^m v_i \rho_i = 0 \end{cases}$$

Assume $n = \#\{i : b_i f(a_i) < \rho\}$, **for them** $\xi = \rho - b_i f(a_i) > 0 \rightarrow u = 0 \rightarrow \nu = \sum_i^m \lambda_i \geq \frac{n}{m}$
Hence, ν **is the upper bound for** n**, which is the fraction of margin errors.**

Python Code for P1

```
from gurobipy import *
WW = Model()
WW.Params.NonConvex = 2
s = WW.addVars(3,lb=-GRB.INFINITY, ub=GRB.INFINITY, vtype='C', name="x") # m sets
WW.setObjective(-3*s[0]*s[0]+s[1]*s[1]+2*s[2]*s[2]+2*(s[0]+s[1]+s[2]), GRB.MINIMIZE)
WW.addQConstr(s[0]*s[0]+s[1]*s[1]+s[2]*s[2]==1)
WW.optimize()
print('s=',WW.getAttr('X',s).values())
```

Mathematica Code for P1

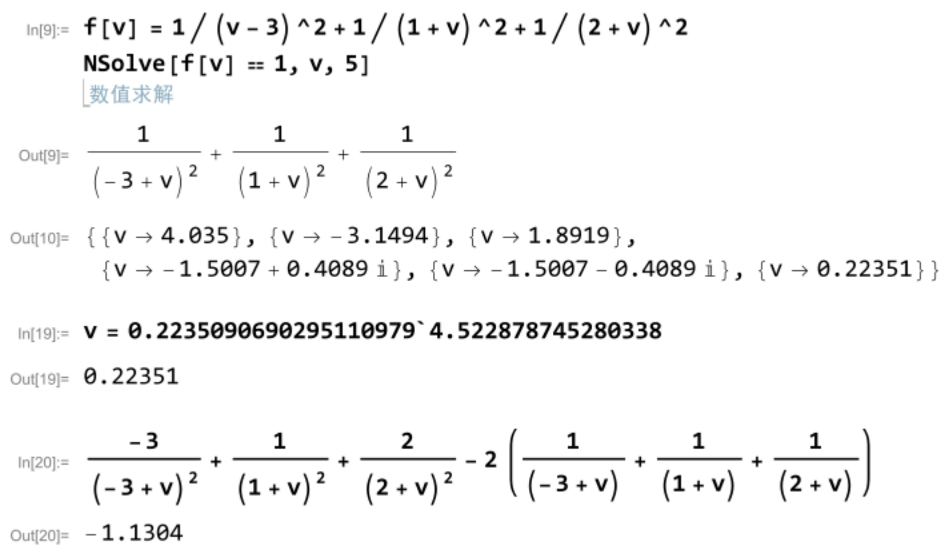


Figure 2: Mathematica Code to solve root for v in P1