IE426 - Optimization models and application

Fall 2015 - Homework #5 Solution

1 Support Vector Machines (5 pts.)

The model file (svm.mod):

```
param n;  # Number of instances
param m;  # Number of features
set I := 1..n;
set J := 1..m;
param x {i in I, j in J};
param y {i in I};
param c;

var xi {i in I} >= 0;
var w {j in J};
var beta;

minimize obj: 1/2*sum{j in J}w[j]^2 + c*sum{i in I}xi[i];

subject to con{i in I}: y[i]*(sum{j in J} w[j]*x[i,j] + beta) >= 1 - xi[i];
```

The data file (svm.dat):

```
param n := 35;
param m := 2;
param c := 10000;

param x :  1 2:=
1 -0.0192 0.4565
2 -0.0302 -0.8531
3 -0.1170 -0.9854
4 0.4454 0.3952
5 -0.7989 -0.2569
6 0.0935 0.7398
7 0.2654 0.3098
8 0.6040 -0.0959
```

```
9 -0.6324 -0.9139
10 0.9770 -0.4862
11 0.9260 0.0075
12 0.8055 -0.0103
13 0.3007 0.9564
14 -0.2771 0.1357
15 0.3782 0.6800
16 -0.5911 -0.1808
17 -0.2501 0.4231
18 -0.1130 0.8032
19 0.9353 -0.2590
20 -0.1272 0.9856
21 -0.0244 0.7780
22 0.2476 0.7701
23 0.1555 -0.8341
24 -0.9507 -1.0000
25 -0.6986 -0.0473
26 -0.4293 -0.9466
27 -0.8917 0.2226
28 0.1545 0.4526
29 -0.8230 0.7856
30 -0.5885 0.5231
31 -0.6578 -0.7660
32 -0.2009 -0.7598
33 0.0965 -0.8755
34 0.3399 0.8383
35 -0.0519 0.9419;

param y :=
1 1
2 -1
3 -1
4 1
5 1
6 1
7 1
8 -1
9 -1
10 -1
11 -1
12 -1
```

```
13 1
14 1
15 1
16 1
17 1
18 1
19 -1
20 1
21 1
22 1
23 -1
24 -1
25 1
26 -1
27 1
28 1
29 1
30 1
31 -1
32 -1
33 -1
34 1
35 1;
```

The command file (svm.run):

```
model 1.mod;
data 1.dat;
option cplex_options 'dualopt';

# Solve with c=10000;
solve;
display xi;
display w;
display beta;

# Solve with c=100;
reset data c;
let c := 100;
solve;
display xi;
display w;
display beta;

# Solve with c=1;
reset data c;
let c := 1;
solve;
display xi;
display w;
display beta;
```

1. Result with $c = 10000$:

```
CPLEX 12.6.1.0: dualopt
No QP presolve or aggregator reductions.
CPLEX 12.6.1.0: optimal solution; objective 8.483095223
12 QP dual simplex iterations (0 in phase I)
xi [*] :=
 1 0    5 0    9 0   13 0   17 0   21 0   25 0   29 0   33 0
 2 0    6 0   10 0   14 0   18 0   22 0   26 0   30 0   34 0
 3 0    7 0   11 0   15 0   19 0   23 0   27 0   31 0   35 0
 4 0    8 0   12 0   16 0   20 0   24 0   28 0   32 0
;

w [*] :=
1  -1.93193
2   3.63783
;

beta = 0.515755
```

Remark: no classification constraint is violated and there are 3 support vectors: x_8, x_{16}, x_{31};

2. Result with $c = 100$:

```
CPLEX 12.6.1.0: dualopt
CPLEX 12.6.1.0: optimal solution; objective 8.483095223
0 QP dual simplex iterations (0 in phase I)
xi [*] :=
 1 0    5 0    9 0   13 0   17 0   21 0   25 0   29 0   33 0
 2 0    6 0   10 0   14 0   18 0   22 0   26 0   30 0   34 0
 3 0    7 0   11 0   15 0   19 0   23 0   27 0   31 0   35 0
 4 0    8 0   12 0   16 0   20 0   24 0   28 0   32 0
;

w [*] :=
1  -1.93193
2   3.63783
;

beta = 0.515755
```

Remark: no classification constraint is violated and there are 3 support vectors: x_8, x_{16}, x_{31};

3. Result with $c = 1$:

```
CPLEX 12.6.1.0: dualopt
CPLEX 12.6.1.0: optimal solution; objective 6.056335781
16 QP dual simplex iterations (16 in phase I)
xi [*] :=
 1 0           8 0.207215   15 0          22 0          29 0
 2 0           9 0          16 0.311718   23 0          30 0
 3 0          10 0          17 0          24 0.243217   31 0.388117
 4 0.399453   11 0          18 0          25 0          32 0
 5 0.200319   12 0.127322   19 0          26 0          33 0
 6 0          13 0          20 0          27 0          34 0
 7 0.349366   14 0          21 0          28 0          35 0
;

w [*] :=
1  -1.40852
2   2.38229
;

beta = 0.286423
```

Remark: violated classification constraints are 4, 5, 7, 8, 12, 16, 24, 31; there are 3 support vectors: x_9, x_{11}, x_{14}.

2 Quadratic Integer Programming - Support Vector Machines (5pts)

$$\begin{array}{rc}
\min_{\xi,\mathrm{w},\beta} & \frac{1}{2}\|w\|^2 + c(\sum_{i=1}^{n}(\xi)) \\
\text{s.t} & y_i(w^\top x_i + \beta) \geq 1 - \xi_i, \quad i = 1, \ldots, n \\
& \text{NumOfNonzeros}(w) \leq K \\
& \xi \geq 0
\end{array}$$

1. Formulate this problem as a problem with integer variables, quadratic objective and linear constraints.

Answer: We employ the logic that $z_i = 0 \Rightarrow w_i = 0$, the problem can be reformulated as

$$\begin{aligned} \min_{\xi,\mathrm{w},\beta} \quad & \tfrac{1}{2}w^T w + c(\textstyle\sum_{\mathrm{i=1}}^{\mathrm{n}} \xi_{\mathrm{i}}) \\ \text{s.t} \quad & y_i(w^\top x_i + \beta) \geq 1 - \xi_i, \quad i = 1, \ldots, n \\ & \xi \geq 0 \\ & w_i \leq M z_i \\ & -w_i \leq M z_i \\ & \textstyle\sum_{i=1}^{n} z_i \leq k \\ & z \in \{0,1\}^n \end{aligned}$$

where M is a sufficiently large number.

2. Write down a convex relaxation of this problem
Answer: this can be done by relaxing the binary variables

$$\begin{aligned} \min_{\xi,\mathrm{w},\beta} \quad & \tfrac{1}{2}w^T w + c(\textstyle\sum_{\mathrm{i=1}}^{\mathrm{n}} \xi_{\mathrm{i}}) \\ \text{s.t} \quad & y_i(w^\top x_i + \beta) \geq 1 - \xi_i, \quad i = 1, \ldots, n \\ & \xi \geq 0 \\ & w_i \leq M z_i \\ & -w_i \leq M z_i \\ & \textstyle\sum_{i=1}^{n} z_i \leq k \\ & 0 \leq z_i \leq 1 \quad i = 1, \ldots, n \end{aligned}$$

3 Modeling

Let $x, b \in \mathbb{R}^n$. A is n x n matrix.

The first norm is defined as follows: $||x||_1 = \sum_{i=1}^{n} |x_i| = |x_1| + ... + |x_n|$

Let $t_i = |x_i|$. Hence $t_i = max\{x_i, -x_i\}$. Hence, we have $x_i \leq t_i$ and $-x_i \leq t_i$ where we will minimize t_i in the objective.

Now, we consider the linerazation of second constraint.

Let $y = Ax - b$. We know $y \in \mathbb{R}^n$

The infinite norm is defined as follows: $||y||_\infty = max_i|y_i|$. Hence, for the constraint, we have $max_i|y_i| \leq \epsilon$. It can written as: $y_i \leq \epsilon$ and $-y_i \leq \epsilon$ for every $i \in \{1, 2, ..., n\}$

$$
\begin{array}{lll}
\min & \sum_{i=1}^n t_i & \\
\text{s.t.} & x_i \leq t_i, & i = 1, \ldots, n \\
& -x_i \leq t_i, & i = 1, \ldots, n \\
& y = Ax - b & \\
& y_i \leq \epsilon, & i = 1, \ldots, n \\
& y_i \geq -\epsilon, & i = 1, \ldots, n
\end{array}
$$

4 Stochastic Programming

1. Define these decision variables:

   x: The Dinners Ordered today
   e: Excessive Dinners to Original Order
   In this part, we know what would happen in the future. Hence, it is logical to order as many dinners as the number of attendees. That is:

Scenario	d	x	Profit
1	450	450	58500
2	900	900	117000
3	1250	1250	162500

   Thus the weighted average is:

   $$58500(0.25) + 117000(0.55) + 162500(0.2) = 111475 \tag{4.1}$$

2. Let t denote the minimum profit. The first stage variable is x and the second-stage variables are $e1, e2, e3$.

   Then, the robust model would be as:

```
var x >= 0 integer;
var e1>= 0 integer;
var e2>= 0 integer;
var e3 >= 0 integer;
var t;

maximize obj: t;

c1: t <= 450*250-120*x -550*e1;
c2: t <= 900*250-120*x -550*e2;
c3: t <= 1250*250-120*x -550*e3;
c4: 450<=x+e1;
c5:  900<=x+e2;
c6: 1250<=x+e3;
c7: x<=1400;

option solver cplexamp;
solve;
display x;
display t;
display e1,e2,e3;
```

The corresponding solution would be:

```
CPLEX 12.6.1.0: optimal integer solution; objective 6060
4 MIP simplex iterations
0 branch-and-bound nodes
x = 887

t = 6060

e1 = 0
e2 = 13
e3 = 363
```

Now we can find the corresponding profit for each scenerio: $x = 887$ and $e1 = 0, e2 = 13, e3 = 363$

Scenario	d	x	e	Profit $(250 * d - 120 * x - 550 * e)$
1	450	887	0	6060
2	900	887	13	111410
3	1250	887	363	6410

With this solution, the expected profit would be:

$$6060(0.25) + 111410(0.55) + 6410(0.2) = 64072.5 \tag{4.2}$$

3. The average number of attending people is $450(0.25) + 900(0.55) + 1250(0.2) = 857.5$ Solving the problem with this parameter would result in x=857.5≈858. The profit/loss of each scenario would be:

Scen.	Attendance	Ticket Revenue(Attendance x 250)	Dinner Cost(858x120)	Excessive Cost	TOTAL
1	450	112500	-102960	0	9540
2	900	225000	-102960	-23100	98940
3	1250	312500	-102960	-215600	-6060

With this solution, the expected profit would be:

$$9540(0.25) + 98940(0.55) - 6060(0.2) = 55590 \tag{4.3}$$

Thus, the expected profit would shrink to 55590$.

4. The x variables are the ones that we should decide NOW, thus they are first stage. However the e variables are the ones that depend on which scenario is happening, thus they belong to second stage. Here, d_i are the parameters related to each scenario

$$\begin{aligned} \max \quad & -120x + 0.25(250d_1 - 550e_1) + 0.55(250d_2 - 550e_2) + 0.2(250d_3 - 550e_3) \\ \text{s.t} \quad & d_1 \leq x + e_1 \\ & d_2 \leq x + e_2 \\ & d_3 \leq x + e_3 \\ & x \geq 0 \\ & e_i \geq 0, i = 1, 2, 3 \end{aligned}$$

5. Here is ampl file:

```
var x >= 0 <=1400;
var e1 >= 0;
var e2 >= 0;
var e3 >= 0;
param d1;
param d2;
param d3;
maximize obj: -120*x + 0.25*(250*d1-550*e1) + 0.55*(250*d2-550*e2) +
 0.2*(250*d3-550*e3);
c1: d1 <= x+e1;
c2: d2 <= x+e2;
c3: d3 <= x+e3;
data;
param d1=450;
param d2=900;
param d3=1250;
```

The result is as follows:

```
CPLEX 12.6.1.0: optimal solution; objective 67875
0 dual simplex iterations (0 in phase I)
x = 900

e1 = 0

e2 = 0

e3 = 350
```

6. The expected profit for SP(Stochastic Programming) is 67875. This is higher than the robust case and the average case value. This solution means that on average the planner would make this much profit over time. This is significantly lower than what we had in the first part, the difference is expected value we get from having perfect information.