Assignment 2: Dynamic Programming project

Francis Vo, Soo-Hyun Yoo

November 5, 2012

1 Problem 1: mmmm ... pork

1.1 Mathematical

1.1.1 Objective function

$$\begin{array}{rccccccc} & 8*ham\_f & + & 12*ham\_r & + & 11*ham\_o & + \\ \max & 4*bellies\_f & + & 12*bellies\_r & + & 7*bellies\_o & + \\ & 4*picnics\_f & + & 13*picnics\_r & + & 9*picnics\_o & \end{array}$$

1.1.2 Constraints

$$\begin{array}{rcl} ham\_f + ham\_r + ham\_o & \leq & 480 \\ bellies\_f + bellies\_r + bellies\_o & \leq & 400 \\ picnics\_f + picnics\_r + picnics\_o & \leq & 230 \\ ham\_r + bellies\_r + picnics\_r & \leq & 420 \\ ham\_o + bellies\_o + picnics\_o & \leq & 250 \end{array}$$

1.2 Standard

1.2.1 Objective function

$$\begin{array}{rccccccc} & 8*ham\_f & + & 12*ham\_r & + & 11*ham\_o & + \\ \max & 4*bellies\_f & + & 12*bellies\_r & + & 7*bellies\_o & + \\ & 4*picnics\_f & + & 13*picnics\_r & + & 9*picnics\_o & \end{array}$$

1.2.2 Constraints

$$\begin{array}{rcl} ham\_f + ham\_r + ham\_o + ham\_remain & = & 480 \\ bellies\_f + bellies\_r + bellies\_o + bellies\_remain & = & 400 \\ picnics\_f + picnics\_r + picnics\_o + picnics\_remain & = & 230 \\ ham\_r + bellies\_r + picnics\_r + smoke\_reg & = & 420 \\ ham\_o + bellies\_o + picnics\_o + smoke\_over & = & 250 \\ ham\_remain, bellies\_remain, picnics\_remain, smoke\_reg, smoke\_over & \geq & 0 \end{array}$$

1.3 Matrix

$\text{Max}(f' * x)$

$$f' = \begin{pmatrix} 8 & 14 & 11 & 4 & 12 & 7 & 4 & 13 & 9 \end{pmatrix}$$

$$a = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

$$b = \begin{pmatrix} 480 \\ 400 \\ 230 \\ 420 \\ 250 \end{pmatrix}$$

$$x = \begin{pmatrix} ham\_f \\ ham\_r \\ ham\_o \\ bellies\_f \\ bellies\_r \\ bellies\_o \\ picnics\_f \\ picnics\_r \\ picnics\_o \end{pmatrix}$$

1.4 Code

```
/* Decision variables */

var ham_f >=0;       /* ham */
var ham_r >=0;       /* ham */
var ham_o >=0;       /* ham */

var bellies_f >=0;  /* bellies */
var bellies_r >=0;  /* bellies */
var bellies_o >=0;  /* bellies */

var picnics_f >=0;  /* picnics */
var picnics_r >=0;  /* picnics */
var picnics_o >=0;  /* picnics */


/* Objective function */
maximize z: 8 * ham_f + 12 * ham_r + 11 * ham_o + 4 * bellies_f + 12 * bellies_r + 7 *
    bellies_o + 4 * picnics_f + 13 * picnics_r + 9 * picnics_o;


/* Constraints */

s.t. Ham             : ham_f + ham_r + ham_o <= 480;
s.t. Bellies         : bellies_f + bellies_r + bellies_o <= 400;
s.t. Picnics         : picnics_f + picnics_r + picnics_o <= 230;
s.t. Smoke_Regular   : ham_r + bellies_r + picnics_r <= 420;
s.t. Smoke_Overtime  : ham_o + bellies_o + picnics_o <= 250;

end;
```

pork.mod

1.5 Solution

Total net profit: $10,910

	fresh	smoked on regular time	smoked on overtime
hams	440	0	40
bellies	0	400	0
picnics	0	20	210

```
Problem:    pork
Rows:       6
Columns:    9
Non-zeros:  24
Status:     OPTIMAL
Objective:  z = 10910 (MAXimum)

   No.   Row name   St   Activity     Lower bound   Upper bound    Marginal
------ ------------ -- ------------- ------------- ------------- -------------
     1  z            B         10910
     2  Ham          NU          480                       480             8
     3  Bellies      NU          400                       400             5
     4  Picnics      NU          230                       230             6
     5  Smoke_Regular
                     NU          420                       420             7
     6  Smoke_Overtime
                     NU          250                       250             3

   No. Column name  St   Activity     Lower bound   Upper bound    Marginal
------ ------------ -- ------------- ------------- ------------- -------------
     1  ham_f        B           440             0
     2  ham_r        NL            0             0                        -3
     3  ham_o        B            40             0
     4  bellies_f    NL            0             0                        -1
     5  bellies_r    B           400             0
     6  bellies_o    NL            0             0                        -1
     7  picnics_f    NL            0             0                        -2
     8  picnics_r    B            20             0
     9  picnics_o    B           210             0

Karush-Kuhn-Tucker optimality conditions:

KKT.PE: max.abs.err = 0.00e+00 on row 0
        max.rel.err = 0.00e+00 on row 0
        High quality

KKT.PB: max.abs.err = 0.00e+00 on row 0
        max.rel.err = 0.00e+00 on row 0
        High quality

KKT.DE: max.abs.err = 0.00e+00 on column 0
        max.rel.err = 0.00e+00 on column 0
        High quality

KKT.DB: max.abs.err = 0.00e+00 on row 0
        max.rel.err = 0.00e+00 on row 0
        High quality

End of output
```

pork.sol

1.6 GNU Linear Programming Kit

We used a glpsol inputing a model file, pork.mod, and then it outputs a solution file, pork.sol. The command we used is "glpsol -m pork.mod -o pork.sol"

2 Problem 2: least squares isnt good enough for me

2.1 Standard

2.2 Code

```
/* Decision variables */

var a;
var b;
var c;
var t;

/* Objective function */
minimize z: t;


/* Constraints */

s.t. point_x_high_1 : 1 - b <= t;
s.t. point_x_low_1 : 1 - b >= -t;

s.t. point_x_high_2 : 2 - b <= t;
s.t. point_x_low_2 : 2 - b >= -t;

s.t. point_x_high_3 : 3 - b <= t;
s.t. point_x_low_3 : 3 - b >= -t;

s.t. point_x_high_4 : 5 - b <= t;
s.t. point_x_low_4 : 5 - b >= -t;

s.t. point_x_high_5 : 7 - b <= t;
s.t. point_x_low_5 : 7 - b >= -t;

s.t. point_x_high_6 : 8 - b <= t;
s.t. point_x_low_6 : 8 - b >= -t;

s.t. point_x_high_7 : 10- b <= t;
s.t. point_x_low_7 : 10- b >= -t;

s.t. point_high_1 : a*(1)+b*(3)-c <= t;
s.t. point_low_1 : a*(1)+b*(3)-c >= -t;

s.t. point_high_2 : a*(2)+b*(5)-c <= t;
s.t. point_low_2 : a*(2)+b*(5)-c >= -t;

s.t. point_high_3 : a*(3)+b*(7)-c <= t;
s.t. point_low_3 : a*(3)+b*(7)-c >= -t;

s.t. point_high_4 : a*(5)+b*(11)-c <= t;
s.t. point_low_4 : a*(5)+b*(11)-c >= -t;

s.t. point_high_5 : a*(7)+b*(14)-c <= t;
s.t. point_low_5 : a*(7)+b*(14)-c >= -t;

s.t. point_high_6 : a*(8)+b*(15)-c <= t;
s.t. point_low_6 : a*(8)+b*(15)-c >= -t;

s.t. point_high_7 : a*(10)+b*(19)-c <= t;
s.t. point_low_7 : a*(10)+b*(19)-c >= -t;

end;
```

bestFit2.mod

2.3 Solution

$a = -8.8$
$b = 5.5$
$c = 12$

```
Problem:    bestFit2
Rows:       29
Columns:    4
Non-zeros:  85
Status:     OPTIMAL
Objective:  z = 4.5 (MINimum)

   No.   Row name    St   Activity     Lower bound   Upper bound    Marginal
------ ------------ -- ------------- ------------- ------------- -------------
     1 z            B            4.5
     2 point_x_high_1
                    B            -10                         -1
     3 point_x_low_1
                    NL            -1            -1                         0.5
     4 point_x_high_2
                    B            -10                         -2
     5 point_x_low_2
                    B             -1            -2
     6 point_x_high_3
                    B            -10                         -3
     7 point_x_low_3
                    B             -1            -3
     8 point_x_high_4
                    B            -10                         -5
     9 point_x_low_4
                    B             -1            -5
    10 point_x_high_5
                    B            -10                         -7
    11 point_x_low_5
                    B             -1            -7
    12 point_x_high_6
                    B            -10                         -8
    13 point_x_low_6
                    B             -1            -8
    14 point_x_high_7
                    NU           -10                        -10          -0.5
    15 point_x_low_7
                    B             -1           -10
    16 point_high_1 B           -8.8                         -0
    17 point_low_1  B            0.2            -0
    18 point_high_2 B           -6.6                         -0
    19 point_low_2  B            2.4            -0
    20 point_high_3 B           -4.4                         -0
    21 point_low_3  B            4.6            -0
    22 point_high_4 NU             0                         -0        < eps
    23 point_low_4  B              9            -0
    24 point_high_5 B           -1.1                         -0
    25 point_low_5  B            7.9            -0
    26 point_high_6 B           -4.4                         -0
    27 point_low_6  B            4.6            -0
    28 point_high_7 NU             0                         -0        < eps
    29 point_low_7  B              9            -0

   No. Column name  St   Activity     Lower bound   Upper bound    Marginal
------ ------------ -- ------------- ------------- ------------- -------------
     1 a            B           -8.8
     2 b            B            5.5
     3 c            B             12
     4 t            B            4.5

Karush-Kuhn-Tucker optimality conditions:

KKT.PE: max.abs.err = 1.42e-14 on row 28
        max.rel.err = 1.48e-16 on row 3
        High quality
```

```
KKT.PB:  max.abs.err = 0.00e+00 on row 0
         max.rel.err = 0.00e+00 on row 0
         High quality

KKT.DE:  max.abs.err = 0.00e+00 on column 0
         max.rel.err = 0.00e+00 on column 0
         High quality

KKT.DB:  max.abs.err = 0.00e+00 on row 0
         max.rel.err = 0.00e+00 on row 0
         High quality

End of output
```

bestFit2.sol

2.4 Plot

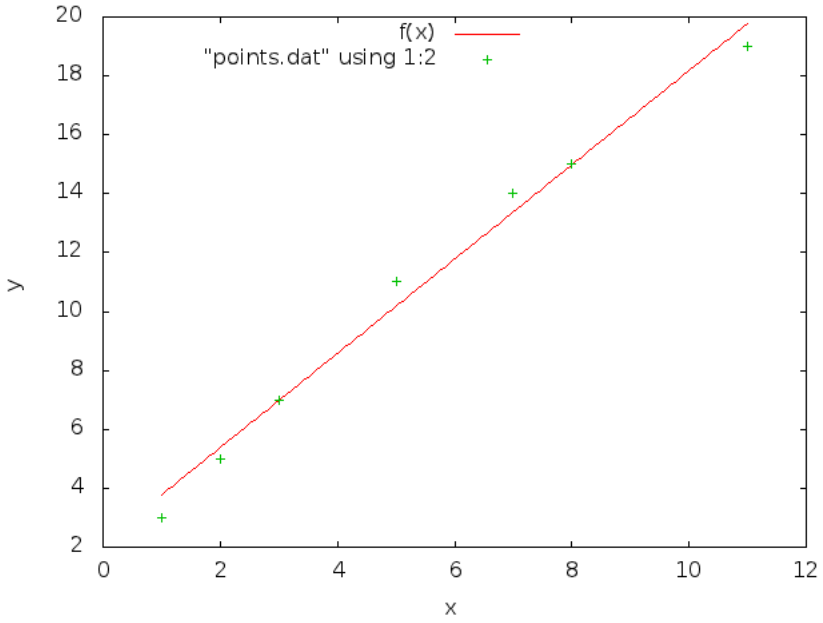

Figure 1: points and best fit line