flow_options

February 15, 2016

1 Data format

- m nodes, n edges
- $A \in \mathbf{R}^{m \times n}$ is an edge incidence matrix such that

$$A_{ij} = \begin{cases} +1 & \text{if edge } j \text{ leaves node } i \\ -1 & \text{if edge } j \text{ enters node } i \\ 0 & \text{otherwise} \end{cases}$$

- $f \in \mathbf{R}^m$ are the flows in and out of nodes (injectors and producers)
  - $f_i > 0$ indicates node i as a injector (flow in)
  - $f_i < 0$ indicates node i as a producer (flow out)
- $e \in \mathbf{R}^n_+$ will denote (nonnegative) flows across edges (to be found)

In [1]:
```
import numpy as np

A = np.loadtxt('data/edge_incidence.txt')
f = np.loadtxt('data/node_flows.txt')
m, n = A.shape # number of nodes and edges
```

In [2]:
```
f
```

Out[2]:
```
array([ 2.0762282 ,  2.3904465 ,  2.4151407 ,  3.1181847 ,  1.1348361 ,
        0.48064576,  0.50024172, -0.11572359, -1.4960089 , -1.2015186 ,
       -1.3024725 , -0.16193653, -0.2022861 , -0.13577737, -1.899456  ,
       -1.3538742 , -1.2410729 , -1.4416442 , -1.5639527 ])
```

2 Feasibility

For data A and f, there exists some valid flow e across the edges if the following convex problem is feasible.

- Data: $A \in \mathbf{R}^{m \times n}$, $f \in \mathbf{R}^m$
- Variables: $e \in \mathbf{R}^n$

$$\begin{array}{ll} \text{minimize} & 0 \\ \text{subject to} & Ae = f \\ & e \geq 0 \end{array}$$

```
In [3]: from cvxpy import Variable, Minimize, Problem

        e = Variable(n) # define decision variable
        obj = Minimize(0) # constant objective makes it a feasibility problem
        constr = [e >= 0, A*e == f] # constraints
        prob = Problem(obj, constr) # form the convex opt problem
        prob.solve(verbose=True) # solve the problem with verbose output

        # problem is feasible (flow exists) if return status is 'optimal
        print('Return status:', prob.status)

        # convert solution to a regular numpy array
        e = np.array(e.value).flatten()
```

```
ECOS 2.0.4 - (C) embotech GmbH, Zurich Switzerland, 2012-15. Web: www.embotech.com/ECOS

It     pcost       dcost      gap   pres   dres    k/t    mu     step   sigma     IR    |   BT
 0  +0.000e+00  -0.000e+00  +4e+01  5e-01  8e-01  1e+00  1e+00    ---    ---    1  0  - |  -  -
 1  +0.000e+00  +6.683e-01  +9e+00  4e-02  2e-01  8e-01  5e-01  0.8532  2e-01   1  0  0 |  0  0
 2  +0.000e+00  +1.557e-02  +6e-01  2e-03  1e-02  2e-02  3e-02  0.9448  6e-03   1  0  0 |  0  0
 3  +0.000e+00  +1.734e-04  +7e-03  2e-05  1e-04  2e-04  3e-04  0.9890  1e-04   1  1  1 |  0  0
 4  +0.000e+00  +1.907e-06  +8e-05  2e-07  9e-07  3e-06  3e-06  0.9890  1e-04   1  1  1 |  0  0
 5  +0.000e+00  +2.780e-08  +1e-06  2e-09  8e-09  4e-08  5e-08  0.9890  1e-04   2  2  2 |  0  0
 6  +0.000e+00  +7.995e-10  +2e-08  1e-09  2e-09  9e-10  1e-09  0.9890  1e-04   2  2  2 |  0  0
 7  +0.000e+00  +2.987e-11  +8e-10  1e-09  2e-09  3e-11  3e-11  0.9730  1e-04   2  3  2 |  0  0

OPTIMAL (within feastol=2.3e-09, reltol=2.7e+01, abstol=8.0e-10).
Runtime: 0.001881 seconds.

Return status: optimal
```

We can see the flow that the solver found by inspecting the variable e. Note that this is only one of possibly many solutions. We'll investigate finding more solutions in the next section.

```
In [4]: e.round(2)

Out[4]: array([ 0.12,  0.26,  0.28,  0.22,  0.63,  0.57,  1.  ,  0.77,  0.08,
                0.1 ,  0.45,  0.28,  0.3 ,  0.23,  0.81,  0.78,  0.5 ,  0.43,
                1.3 ,  0.09,  0.11,  0.14,  0.32,  0.24,  0.72,  0.41,  0.48,
                0.15,  0.14,  0.2 ])
```

3 Feasibility: largest and smallest feasible value

Among the feasible solutions to the flow problem, we can maximize and minimize single elements of e to see how much the flow could potentially vary.

- For some edge i, find a flow with the **smallest** possible value of e_i

$$\begin{array}{ll} \text{minimize} & e_i \\ \text{subject to} & Ae = f \\ & e \geq 0 \end{array}$$

- Find a flow with the **largest** possible value of e_i

$$\begin{array}{ll} \text{minimize} & -e_i \\ \text{subject to} & Ae = f \\ & e \geq 0 \end{array}$$

In [5]:
```
# this code finds a flow with the smallest possible value for e_3 (with zero indexing)

e = Variable(n)
obj = Minimize(e[2]) # note 0-based indexing
constr = [e >= 0, A*e == f]
prob = Problem(obj, constr)
prob.solve(verbose=False)

print('Return status:', prob.status)
e = np.array(e.value).flatten()
```

```
Return status: optimal
```

We see that a valid flow exists with e_3 as low as 0. In the previous (pure) feasibility problem, we found a solution with $e_3 = .28$.

In [6]:
```
e.round(2)
```

Out[6]:
```
array([ 0.12,  0.28,  0.  ,  0.22,  0.76,  0.7 ,  0.98,  0.76,  0.08,
        0.1 ,  0.48,  0.29,  0.56,  0.23,  0.68,  0.65,  0.52,  0.44,
        1.3 ,  0.08,  0.1 ,  0.14,  0.32,  0.21,  0.7 ,  0.43,  0.48,
        0.15,  0.14,  0.21])
```

The next code loops through every index of e, and solves an optimization problem to maximize and minimize that element. We keep track of the maximum and minimum occurances for each element.

In [7]:
```
e_max = np.zeros(n)
e_min = np.zeros(n)
e_max[:] = -np.inf
e_min[:] = np.inf

for i in range(n): # for each index of e
    for sign in [+1, -1]: # minimize and maximize the element
        e = Variable(n)
        prob = Problem(Minimize(sign*e[i]), [e >= 0, A*e == f])
        prob.solve(verbose=False)
        e = np.array(e.value).flatten()

        #e_max and e_min track the maximum and minimum values seen for each index of e
        e_max = np.maximum(e_max, e)
        e_min = np.minimum(e_min, e)
```

We can look at the difference between the maximum and minimum values for each index to get an idea of how much they vary. We note that a few elements are fixed, having zero difference between the extremes.

In [8]:
```
(e_max - e_min).round(2)
```

Out[8]:
```
array([ 0.  ,  1.42,  1.35,  1.24,  1.44,  1.56,  1.5 ,  1.2 ,  0.16,
        0.2 ,  1.01,  1.42,  1.35,  1.24,  1.44,  1.56,  1.5 ,  1.2 ,
        0.  ,  0.16,  0.2 ,  0.  ,  1.01,  1.01,  1.13,  1.13,  0.  ,
        0.5 ,  0.5 ,  0.5 ])
```

4 Other options

These following sections outline some other modeling options. I can fill these in with actual code next if everything so far looks like its on the right track.

5 Extra edges

Consider r "extra" or potential edges, encoded by edge incidence matrix B, in addition to the "free" or existing edges given by A. Attach some penalty to sending flow across edges given by B.

- Data: $A \in \mathbf{R}^{m \times n}$, $B \in \mathbf{R}^{m \times r}$, $f \in \mathbf{R}^m$
- Variables: $e \in \mathbf{R}^n$, $c \in \mathbf{R}^r$

$$\begin{array}{ll} \text{minimize} & \|c\| \\ \text{subject to} & Ae + Bc = f \\ & e \geq 0 \\ & c \geq 0 \end{array}$$

- Variations:
  - L1 norm (will depend on weighting, but will encourage sparsity, or fewest edges)
  - L2 norm (will encourage many edges, but all with "small" values)
  - weight terms in objective differently, according to some likelihood of a potential edge being nonzero.

6 Flow measurement errors: fixed range

Assume there are some errors in the injector and producer measurements f. Add a flow variable g, which is allowed to deviate from f by at most some number ϵ. Find a feasible flow for g, if it exists.

- Data: $A \in \mathbf{R}^{m \times n}$, $f \in \mathbf{R}^m$, $\epsilon \in \mathbf{R}$
- Variables: $e \in \mathbf{R}^n$, $g \in \mathbf{R}^m$

$$\begin{array}{ll} \text{minimize} & 0 \\ \text{subject to} & Ae = g \\ & f - \epsilon \leq g \leq f + \epsilon \\ & e \geq 0 \end{array}$$

- Variations:
  - lower and upper bound <u>vectors</u> $f_u, f_l \in \mathbf{R}^m$ with constraint $f_l \leq g \leq f_u$

7 Flow measurement errors: penalized deviation

Instead of fixing a hard constraint on how much we expect the flows to vary, we can penalize deviation from the observed f. In the problem below, the 2-norm would correspond to a model assuming Gaussian errors with identical variance in the measurements. Weighting the elements in the 2-norm corresponds to assigning different variance parameters to each observation (based on, say, confidence in the accuracy of that measurement).

- Data: $A \in \mathbf{R}^{m \times n}$, $f \in \mathbf{R}^m$
- Variables: $e \in \mathbf{R}^n$, $g \in \mathbf{R}^m$

$$\begin{array}{ll} \text{minimize} & \|g - f\| \\ \text{subject to} & Ae = g \\ & e \geq 0 \end{array}$$

- Variations:
  - L1, L2 norms
  - weighted penalty term, corresponding to certainty of measurement

8 Model combinations

We can pick and choose various elements of the models above and combine them into a single model, if that would make sense.

```
In [ ]:
```