Large-Scale Optimization

L14. OPTIMIZATION SYSTEMS
Today

Optimization Systems
CVXPY
Examples
Optimization Systems

“Best plan for buying stocks...?”

Standard form

Modeling language

Solver

LP, QP, MIP, SDP, SOCP, QCQP, ...
Modeling Languages

AMPL
- “A Mathematical Programming Language”
- 1985 ~ current

GAMS
- “General Algebraic Modeling System”
- 1976 ~ current

MPS, ...

Most are for commercial licenses
set Plants;
set Markets;

param Capacity{p in Plants};
param Demand{m in Markets};
param Distance{Plants, Markets};
param Freight;
param TransportCost{p in Plants, m in Markets} := Freight * Distance[p, m] / 1000;

var shipment{Plants, Markets} >= 0;

minimize cost:
    sum{p in Plants, m in Markets} TransportCost[p, m] * shipment[p, m];

# Observe supply limit at plant p
s.t. supply{p in Plants}: sum{m in Markets} shipment[p, m] <= Capacity[p];

# Satisfy demand at market m
s.t. demand{m in Markets}: sum{p in Plants} shipment[p, m] >= Demand[m];

TU Dortmund, Dr. Sangkyun Lee
Sets
   i   canning plants   / seattle, san-diego /
   j   markets          / new-york, Chicago, topeka / ;

Parameters
   a(i)  capacity of plant i in cases
         /    seattle     350
             san-diego   600  / 
   b(j)  demand at market j in cases
         /    new-york    325
             Chicago     300
             topeka      275  / ;

Table d(i,j)  distance in thousands of miles

   new-york       Chicago      topeka
seattle             2.5              1.7             1.8
san-diego        2.5              1.8            1.4  ;

Scalar f  freight in dollars per case per thousand miles /90/ ;

Parameter c(i,j)  transport cost in thousands of dollars per case ;
   c(i,j) = f * d(i,j) / 1000 ;

Variables
   x(i,j)  shipment quantities in cases
   z       total transportation costs in thousands of dollars ;

Positive Variable x ;

Equations
   cost        define objective function
   supply(i)   observe supply limit at plant i
   demand(j)   satisfy demand at market j ;

   cost ..        z  =e=  sum((i,j), c(i,j)*x(i,j)) ;
   supply(i) ..   sum(j, x(i,j))  =l=  a(i) ;
   demand(j) ..   sum(i, x(i,j))  =g=  b(j) ;

Model transport /all/ ;
Solve transport using lp minimizing z ;
Display x.l, x.m ;
NEOS Optimization Server

http://www.neos-server.org

- List of standard forms and available state-of-the-art solvers
- Runs optimization on server
- Accepts optimization problems in several forms

Python-based optimization modeling system

- [http://www.cvxpy.org](http://www.cvxpy.org)
- Spinoff of CVX, a Matlab-based modeling system
- Convex optimization
- Relies on open-source conic problem solvers
  - ECOS (default)
  - CVXOPT
  - SCS
Linear Program (LP)

\[
\min_{x \in \mathbb{R}^n} \quad c^T x \\
\text{s.t.} \quad Ax = b \\
x \geq 0
\]

Data:
\[
c \in \mathbb{R}^n \\
A \in \mathbb{R}^{k \times n} \\
b \in \mathbb{R}^k
\]

Equivalent forms

\[
\min_{x \in \mathbb{R}^n} \quad c^T x \\
\text{s.t.} \quad Ax \geq b \\
x \geq 0
\]
Conversions

Inequalities to equalities:

\[ Ax \geq b \iff Ax - y = 0, \ y \geq 0 \]

Free variables to nonnegative variables:

\[ x \in \mathbb{R}^n \iff x = u - v, \ u \geq 0, \ v \geq 0 \]
LP: Toy Example (CVXPY)

```python
from cvxpy import *
import numpy

numpy.random.seed(1)

x = Variable(2)

c = vec([0.5, 2.4])
A = numpy.random.rand(3,2)
b = numpy.random.rand(3,1)

const = [0 <= x, A*x >= b]

obj = Minimize(c.T*x)
prob = Problem(obj, const)

prob.solve(solver=ECOS, verbose=True)
print "status", prob.status
print "solution\n", x.value
# print const[0].dual_value
print "optimal value\n", prob.value
print "dual variables\n", const[0].dual_value, "\n", const[1].dual_value
```

TU Dortmund, Dr. Sangkyun Lee
Exercises

Transport
Sudoku
LASSO
LASSO Path
Sudoku

7  8  1  2
2       3  8
1  9  8  7  3

3  2
9  7  3  2  4
8  7  6  1  9

6  7  3  2
9  3  1  4
2  4  3  6  7
Sudoku - Answer

```
7 3 8  | 9 6 1  | 2 4 5  
2 4 6  | 5 7 3  | 1 9 8  
1 5 9  | 2 4 8  | 7 6 3  
3 1 2  | 6 9 4  | 5 8 7  
9 6 5  | 1 8 7  | 3 2 4  
4 8 7  | 3 2 5  | 6 1 9  
6 7 1  | 4 5 9  | 8 3 2  
8 9 3  | 7 1 2  | 4 5 6  
5 2 4  | 8 3 6  | 9 7 1  
```