TEXAS
The University of Texas at Austin

TOPIC 4 NONLINEAR PROGRAMMING

Python Packages for NLP

- There are many good-ish packages for NLP in python
  - scipy.optimize – some built in solvers to scipy
  - pyomo – part of COIN-OR
  - pysolnp – just one algorithm
- scipy.optimize is tricky because different algorithms require different syntax of objective and constraints
- pyomo is a modeling suite that has many open-source algorithms, but requires learning their language
- pysolnp is based on PhD dissertation of former student of Dantzig
- I usually just stick with scipy

Python Packages for NLP

- For scipy.optimize you must tell it which solver to use
  - BFGS: just for unconstrained problems
  - L-BFGS-B: just for box constraints (a <= x <= b)
  - SLSQP: for general constraints
- These only work well/ok for convex programs
- It's not uncommon for scipy.optimize to give garbage solutions…

Minimization or Maximization

- Almost all NLP packages only solve min problems!!
  - What if you want to maximize something?!?!
- If you want to maximize, just minimize the negative of your objective function
- Trying to make a function as big as possible is the same as trying to make the negative of that function as small as possible

Example

- A company manufactures and sells a product
  - They get to set the price
  - It costs $50/unit to manufacture
- The price drives demand through a demand function
  - $D = \alpha P^{\beta}$, where $\alpha > 0,\ \beta < -1$
  - This is called a constant elasticity of demand function

- Let's use calculus to find the price that maximizes profit

Example

- Assume $\alpha = 3777178,\ \beta = -2.154$
- Let's plug this example into python
- What if I also get to pick the number of items manufactured?

Another Example

- A company wants to locate a warehouse from which it will ship products to four customers.
- The locations and number of shipments are given.
- A single warehouse must be used to service all the customers.
- The company wants to determine the location of the warehouse that minimizes the total distance traveled from the warehouse to the customers.

Customer data				
	X-coordinate	Y-coordinate		Annual shipments
Customer 1	5	10		200
Customer 2	10	5		150
Customer 3	0	12		200
Customer 4	12	0		300

Customer data				
	X-coordinate	Y-coordinate		Annual shipments
Customer 1	5	10		200
Customer 2	10	5		150
Customer 3	0	12		200
Customer 4	12	0		300

Class Participation

- Formulate an NLP and solve for the optimal location of the warehouse

Constraints

- Now suppose there is a river that runs diagonally through town (Southwest to Northeast)
- The warehouse needs to be on the north side of the river for permitting
  - The y coordinate must be larger than x coordinate
- Where should they put their warehouse now?
  - Don’t worry about bridges…they deliver using a helicopter…

Quirks of Non-Linear Solvers

- If you have an inequality constraint, it's usually best for your initial guess to be at a point that does not satisfy the inequality constraint at equality
  - If you have the constraint $x_2 \geq x_1$, it's best not to start at $x_2 = x_1$
  - Start somewhere that $x_2 > x_1$

- But if you have an equality constraint, you should start at a point that satisfies that constraint
  - If you have a constraint $x_1 + x_2 = 1$, then you should start somewhere like $x_1 = 0.4$, $x_2 = 0.6$