OPTIMIZATION

MIDTERM 1

Lulu & Cici's Flower Shop

REVEIW

PERCENTAGES

- what a throw back!
- we all know...

$$x\% = .01 \cdot x$$

↳ ex. $46\% = .46$

- if we want... $y\%$ of x: $\frac{x}{100} \cdot y$ — drop from z to x, what % drop? $\frac{x-z}{z} = y\%$

SUMMATIONS

- $\sum_{i=k}^{n} a_i$ → we sum terms a_k to a_n $= a_k + a_{k+1} + \cdots + a_n$
- summations in matrices...

$$A = \begin{bmatrix} A_{11} & \cdots & A_{1n} \\ \vdots & & \vdots \\ A_{n1} & \cdots & A_{nn} \end{bmatrix}$$ and lets sum elements in row i...

$$\sum_{j=1}^{n} A_{ij} = A_{i1} + A_{i2} + \cdots + A_{in}$$

- if we want to sum the diagnol of a matrix... $\sum_{i=1}^{n} A_{ii}$
- if we want the opposite diagnol... $\sum_{i=1}^{n} A_{i(n+1-i)}$
- if we want one line below normal diagnol... $\sum_{i=1}^{n-1} A_{(i+1)i}$

LINEAR ALGEBRA

- We have matrices and vectors, now lets mult. them...

$$A\vec{x} = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1n} \\ \vdots & & & \vdots \\ A_{n1} & A_{n2} & \cdots & A_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

$$= \begin{bmatrix} A_{11}x_1 + \cdots + A_{1n}x_n \\ \vdots \\ A_{n1}x_1 + \cdots + A_{nn}x_n \end{bmatrix}$$ ← makes a vector

$$= \sum_{j=1}^{n} A_{ij} x_j$$ ← for i^{th} row

- We can also look at transposes

$$A^T = \begin{bmatrix} A_{11} & A_{21} & \cdots & A_{m1} \\ A_{12} & & & \vdots \\ \vdots & & & \vdots \\ A_{1n} & \cdots & & A_{mn} \end{bmatrix}$$ ← diagnol stays same

$$A^T\vec{y} = \sum_{j=1}^{m} A_{j,i}\, y_j$$

- for matrix by matrix mult. $A \cdot B = C$ the $C_{i,j}$ term is...

$$C_{i,j} = \sum_{k=1}^{n} A_{i,k} B_{k,j}$$

- We can solve $A\vec{x} = \vec{b}$ by saying... $\vec{x} = A^{-1}\vec{b}$
- an important concept to understand is Linear Independence
  - if a collection of vectors $\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_k$ has a linear combo...

$$\lambda_1\vec{v}_1 + \lambda_2\vec{v}_2 + \cdots + \lambda\vec{v}_k$$

  then these vectors are linearly independent if the combo only $= \vec{0}$ if all $\lambda_i = 0$.
- The following facts are equivalent...
  1. A^{-1} exists
  2. the columns of A are lin. indp.
  3. $(A^T)^{-1}$ exists
  4. rows of A are lin. indp.
  5. there is a unique solution to $A\vec{x} = \vec{b}$ for all $\vec{b}$

LINEAR PROGRAMING

> Basic Example > How Many Flowers
> Lulu + Cici's flower shop will grow 2 types of flowers Roses and Daisys. It takes 2 feet² to grow a bushel of Roses and 1 foot² for a bushell of Daisys. Their garden has 17 ft² of space. There is a demand of 30 bushells of Daisys and +∞ for Roses. Roses bring in \$5 of profit and Daisys bring in \$3. How should Lulu and Cici max profit?

>① draw a table

	Roses	Daisys	Avai.
land	2	1	17
prof.	5	3	
dem.	+∞	30	

>② choose variables... x_1 = roses x_2 = daisys

>③ set objective...

max $5x_1 + 3x_2$

>④ define constraints

land... dem... nonn...

$17 \geq 2x_1 + x_2$

$30 \geq x_2$

$x_1 \geq 0$

$x_2 \geq 0$

- aka linear optimization, lets start w/ some definitions...

DEFINITIONS

- Variables → what's gonna change in our problem. What decisions are we making?
- objective → what are we maximizing or minimizing?
- Constraints → what are the restrictions
- Linear → our programs can only use $+, -, \times, \div$, no mult. variables by each other, no other functions.
- Linear Program → tool to maximize/minimize a linear function subject to linear constraints. Unlike Linear Algebra deals w/ inequalities, more general and more powerful.

BASIC EXAMPLE → set up LP.

SOLUTIONS

- What does it mean to solve an LP? Well we want an Optimal Solution → values of our variables that meets the constraints and objectives. The value of the objective function when the optimal solution is plugged in is the Optimal Vale.
- We have 3 types of solutions to an LP...
- infeasible → There is no possible solution to the LP. No feasible region
- Optimal solution → When there is a solution to the LP. Doesn't have to be unique.
- Unbounded → The feasible region is not bounded, so the optimal value can keep going to ∞ or −∞.

SOLVING GRAPHICALLY

- Typically we'll solve LPs using AMPL, but if there's only 2 variables we can solve it graphically by drawing the Feasible Region.

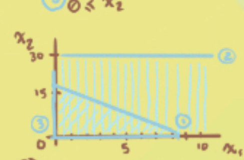
> solving graphically example > how many flowers
> lets solve the above example graphically...

>① graph constraints.

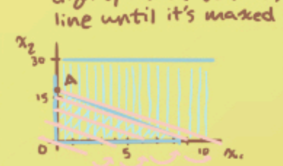
① $17 \geq 2x_1 + x_2$ ⇒ $x_2 \leq 17 - 2x_1$

② $30 \geq x_2$

③ $0 \leq x_1$, $0 \leq x_2$



>② graph the objective value with dif. optimal values, push this isoprofit line until it's maxed or min-ed.



The optimal solution is at point A when $x_1 = 0$ and $x_2 = 17$. And the optimal value is \$51

BLENDING PROBLEM EXAMPLE → lets see a more complicated LP.

> blending example > Flower soil
> Lulu and Cici buy gardening soil then mix it to make their special flower soil. They buy 3 dif. types of gardening soil with dif. % of fertilizer and silt...

soil	fertilizer %	silt %	cost \$	available
type 1	.5	30	25	50
type 2	2	10	35	50
type 3	3	20	45	50

in bags

and use it to make 3 dif types of flower soil...

soil	fert. min	silt max	flowers grown	flowers \$
type 1	2	15	2500	4
type 2	3	15	5000	4
type 3	1	25	900	3

> lets max Lulu + Cici's profits...

>① variables

g_i = amount of garden soil to buy

f_j = amount of flower soil to make in bags, ie $\{1,2,3\}$ $j \in \{1,2,3\}$

$x_{i,j}$ = amount of soil i used for flower soil j. $i,j \in \{1,2,3\}$

>② objective

$4 \cdot 2500 \cdot f_1 + 4 \cdot 5000 \cdot f_2 + 3 \cdot 900 \cdot f_3 - 25g_1 - 35g_2 - 45g_3$

>③ constraints — we're gonna have a lot of constraints

$0 \leq g_1, g_2, g_3 \leq 50$

$0 \leq f_1, f_2, f_3$

$g_i = x_{i1} + x_{i2} + x_{i3}$

$f_j = x_{1j} + x_{2j} + x_{3j}$

$$\left.\begin{aligned} .005x_{11} + .02x_{21} + .03x_{31} &\leq .02f_1 \\ .005x_{12} + .02x_{22} + .03x_{32} &\leq .03f_2 \\ .005x_{13} + .02x_{23} + .03x_{33} &\leq .01f_3 \end{aligned}\right\} \text{fertilizer}$$

$$\left.\begin{aligned} .3x_{11} + .1x_{21} + .2x_{31} &\geq .15f_1 \\ .3x_{12} + .1x_{22} + .2x_{32} &\geq .15f_2 \\ .3x_{13} + .1x_{23} + .2x_{33} &\geq .25f_3 \end{aligned}\right\} \text{silt}$$

LP'S W/ MATRICES

- you probably noticed from the blending problem that big LP's quickly become tedious to write out. We simplify things by writing problems w/ matrices...

BASICS

- rember linear combinations from linear algebra...

$$A\vec{x} = \vec{b}$$

- We're gonna use this form to write our LP's...

$\vec{x}$ = variables A = properties of variables $\vec{b}$ = requirements for var. properties

- and to write the objective we'll define...

$$\vec{c} = \text{prices}$$

- now lets look at an example...

> LP's w/ Matrices > How many Flowers?
> lets rewrite our first example w/ matrices

>① variables $\vec{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$

>② objective max $\vec{c}\vec{x}$

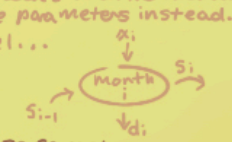
>③ constraints $A\vec{x} \leq \vec{b}$, $\vec{x} \geq \vec{0}$

with... $\vec{c} = \begin{bmatrix} 5 \\ 3 \end{bmatrix}$ $A = \begin{bmatrix} 2 & 1 \\ 0 & 1 \end{bmatrix}$ $\vec{b} = \begin{bmatrix} 17 \\ 30 \end{bmatrix}$

MULTIPERIOD PROBLEMS

- We might also choose to write the constants in our problem as parameters to make our work even more concise, lets see an example...

> Multiperiod Problem Example
> Flower demand
> Lulu and Cici want to grow flowers in good growing seasons and meet their customers demand. They can hold 40 flower bushells in their fridge and the flowers will not rot. How many flowers should Lulu and Cici grow per month to min costs?...
> lets say we have the actual #s for the constants in a computer, but we don't want to write them all out. We'll define parameters instead.
> model...



>① parameters

d_i = demand in month i

p_i = price to grow flowers in month i

>② variables

x_i = flowers grown in month i

s_i = flower leftover in storage after demand satisfied in month i.

>③ objective

$$\min \sum_{i=1}^{n} p_i x_i$$

>④ constraints

$x_i + s_{i-1} = d_i + s_i$ $x_i \geq 0$, $0 \leq s_i \leq 40$

PORTFOLIO OPTIMIZATION

DEFINITIONS

- a really common use of linear programs is to maximize profit from stock portfolios.
- these type of problems have a lot of special vocabulary, lets go over some of it so we can try out an example...
  - we start with a Return matrix → which gives us returns for \$1 invested in stock j if scenario i happens... $R = [R_{i,j}]$
- the probability of scenario i → P_i
  - in general we can say the return in scenario i for x_j invested in stock j...

$$r_i = \text{return in scenario } i = \sum_{j=1}^{n} R_{i,j}\, x_j$$ → n stocks

  - if we want our expected return across all scenarios...

$$r_p = \sum_{i=1}^{m} P_i r_i$$ → m scenarios

  - We may also care about quantifying risk...

$$d_i = \text{downside risk in s.}i = \begin{cases} 0, & \text{if } r_i \geq 0 \\ -r_i, & \text{if } r_i \leq 0 \end{cases}$$

↑ 0 if no loss; the amount of the loss if there is loss

ADR = average downside risk

$$= \sum_{i=1}^{m} P_i d_i$$

- in our LP we could want to max/min dif things...

EXAMPLE → stocks!

> Portfolio Optimization Example
> flower shop stock
> Lulu and Cici's flower shop is a publically traded business and their friend Dom wants to invest, but Lulu and Cici want to make sure Dom doesn't loose all his \$ so they help him invest in other stocks too. help Dom max his returns...

> $R = \begin{bmatrix} 2 & 0 \\ -1 & .1 \end{bmatrix}$ → other stock, flower shop stock; scenario 1, scenario 2

$\vec{P} = [.7\ \ .3]$

>① variables →

$\vec{x}$ = amount invested in each stock

$\vec{r}$ = return in each scenario

r_p = expected return

>② objective →

max r_p

>③ constraints →

$r_p = .7r_1 + .3r_2$

$r_1 = 2x_1 + 0x_2$, $r_2 = -x_1 + .1x_2$

$x_1, x_2 \geq 0$, $x_1 + x_2 \leq 1$

> what if instead Dom wants to min his risk?

>① variables →

same +

d_i = downside risk of each stock

ADR = average d_i

>② objective →

min ADR

>③ constraints →

same +

ADR $= .7d_1 + .3d_2$

$d_1 = \max\{0, -r_1\}$

$d_2 = \max\{0, -r_2\}$

but these aren't linear! let's look at how we can add these to an LP.

> Maximizing example
> flower shop stock cont.
> lets finish the stock question by redifining d_i...

> $d_1 \geq 0$

$d_1 \geq -r_1$

$d_2 \geq 0$

$d_2 \geq -r_2$

> this works cause the obj. is minimizing ADR which d_1 and d_2 are part of.

LP'S IN DISGUISE

> we can't put nonlinear functions in a linear program but we can rewrite nonlinear programs to be linear in an LP!

MIN & MAX

- min and max are obviously not linear but we can rewrite them.
- we can think of a max between two variables... $\max\{x_1, x_2\}$

and define a new variable z.

> We will say... $z \geq x_1$, $z \geq x_2$, so z will take the larger of the 2 values.

> but don't we have to worry about $z > x_1, x_2$? not if z is being minimized in the objective!

- Basically → objective → $\min \max\{x_1, x_2\}$

≡ object → $\min z$

↳ $z \geq x_1$, $z \geq x_2$

- We can look at an example...

ABSOLUTE VALUE

- absolute values are also nonlinear...
- lets start by breaking the function down...

$$|x_1| = \max\{x_1, -x_1\}$$

- and we know how make that linear!
- So... objective → $\min |x_1|$

≡ objective → $\min z$

↳ $z \geq x_1$, $z \geq -x_1$

- and we can apply this to sums of absolute values too!
- We can use these ideas to find lines of best fit in Linear Algebra.

good luck