Math 294 Optimization (linear & convex, w/constraints)

- overview of Moodle page

"Knapsack" problem

Suppose an investment fund wants to invest all or part of capital \$c among n investment opportunities.

Cost for i^{th} investment is \$$w_i$

Profit " \$$p_i$

Available units " b_i

How many units x_i of each investment should be bought to maximize profit?

$\max_{x_1,\dots,x_n} \sum_{i=1}^{n} p_i x_i$ where $\sum_{i=1}^{n} w_i x_i \le C$ and $x_i \in \{0,1,2,\dots,b_i\}$ for $i=1,\dots,n$

(or $0 \le x_i \le b_i$)

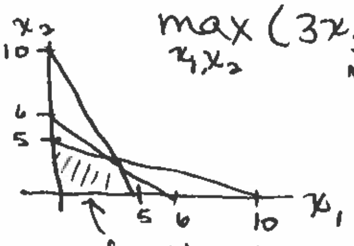
Production problem

Clothes manufacturer has 10 sq yards of cotton fabric, 10 of wool, 6 of silk

Pair of pants requires 1 2 1 Profit \$

skirt 2 1 1 \$

How many of each to maximize profit?



feasible region

$\max_{x_1,x_2} (3x_1 + 4x_2)$ subject to $x_1 + 2x_2 \le 10$, $2x_1 + x_2 \le 10$, $x_1 + x_2 \le 6$, $x_1, x_2 \ge 0$ (integers)

↖# pants ↖# skirts

Linear programming: find max or min of linear function with linear constraints

Convex optimization: convex fns rather than linear

* usually won't require integer sol'ns

Least squares problem

Fit regression line to pts $(0,1),(1,3),(2,4)$: $\min_{m,b}\left((b-1)^2 + (m+b-3)^2 + (2m+b-4)^2\right)$

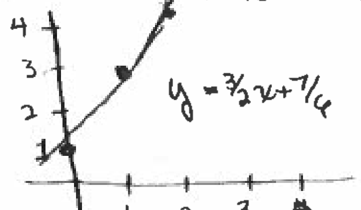
$(y = mx+b)$ $0 \cdot b \approx 1$ $m+b \approx 3$ $2m+b \approx 4$

$= \min_{m,b} (5m^2 + 6mb - 22m + 3b^2 - 16b + 26)$

Calculus method: $\frac{\partial f}{\partial m} = 10m + 6b - 22 = 0$

(crit pts) $\frac{\partial f}{\partial b} = 6m + 6b - 16 = 0$

$\Rightarrow m = 3/2$, $c = 7/4$



→ Want more general approach to constrained nonlinear (max/min) optimization problems

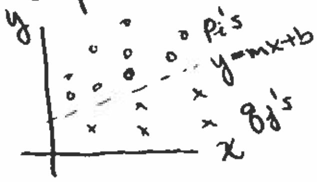
Bit of history: In 1947 George Dantzig developed the simplex method, an efficient way to optimize a linear fn constrained by linear inequalities (though more efficient methods have been developed since then).

Due to its effectiveness in industry, a variety of more general convex optimization algorithms have also been developed.

Read Chapter 1 intro.

Day 2 - first show production example solved in R

Separation of points example: linear criterion to distinguish two sets of points



Set 1: $p_1, \dots, p_m$

Set 2: $q_1, \dots, q_n$

Require $y(p_i) > a x(p_i) + b$ for $i=1,\dots,m$

$y(q_j) < a x(q_j) + b$ for $j=1,\dots,n$

y coord of pt   x coord of pt

There may be many such lines, and linear programs may not involve strict inequalities, so introduce gap variable $\delta > 0$

$\boxed{\text{R script}}$

Write out equations, then put into matrix format

* Doesn't have to be a line of separation — can use any desired fn since linear in coefficients.

Chapter 2

Section 2.1 Intro to linear programs

<u>Linear program</u> (LP) takes standard form

$$\begin{aligned} \underset{x_1,\dots,x_n}{\text{minimize}} \quad & c_1x_1+\dots+c_nx_n \quad \swarrow \text{ objective fn} \\ \text{subject to} \quad & a_{11}x_1+\dots+a_{1n}x_n = b_1 \\ & a_{21}x_1+\dots+a_{2n}x_n = b_2 \\ & \vdots \\ & a_{m1}x_1+\dots a_{mn}x_n = b_m \\ & x_1\geq 0,\dots,x_n\geq 0 \end{aligned}$$

assume $b_i \geq 0$ (multiply eq by -1 if needed)

Matrix-vector form: $\vec{c}=\begin{bmatrix} c_1\\ \vdots \\ c_n\end{bmatrix}$ $A=\underbrace{\begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{m1} & \cdots & a_{mn}\end{bmatrix}}_{m\times n \text{ matrix}}$ $\vec{x}=\begin{bmatrix} x_1\\ \vdots \\ x_n\end{bmatrix}$ $\vec{b}=\begin{bmatrix} b_1\\ \vdots \\ b_m\end{bmatrix}$

$$\underset{\vec{x}}{\text{minimize}}\ \vec{c}^T\vec{x} \quad \text{subject to } A\vec{x}=\vec{b} \text{ and } \vec{x}\geq\vec{0}$$

$\nwarrow$ vector of zeros

<u>Slack variables</u> are used to convert LPs in other forms to this standard form.

Example: $\underset{x_1,x_2}{\max}\ 3x_1+4x_2$ subject to $x_1+2x_2\leq 10$, $2x_1+x_2\leq 10$, $x_1+x_2\leq 6$, $x_1,x_2\geq 0$

Switch max to min by take negative of objective fn: $\underset{x_1,x_2}{\min}(-3x_1-4x_2)$

Convert constraint inequalities by adding slack variables:

$$\begin{aligned} x_1+2x_2+x_3 &= 10 \\ 2x_1+x_2+x_4 &= 10 \qquad x_1,\dots,x_5\geq 0 \\ x_1+x_2+x_5 &= 6 \end{aligned}$$

Idea: $x_1+2x_2=10-x_3\leq 10-0$, that is, x_3 takes up the slack between x_1+2x_2 and 10

If inequality is $\geq$, then subtract "surplus variable"

$$x_1+3x_2\geq 5 \ \rightarrow\ x_1+3x_2-x_3=5,\ x_3\geq 0$$

If some variable x_k ~~among $x_1, \ldots, x_n$ aren't~~ isn't constrained to be nonnegative, can convert to standard form by either replacing x_k with $u_k - v_k$, $u_k \geq 0$, $v_k \geq 0$, or trying to eliminate that variable.

Example: $\min_{x_1,x_2,x_3} 4x_1 + \cancel{3}5x_2 + x_3$ subject to

$$2x_1 + x_2 + x_3 = 4$$
$$x_1 + 3x_2 + x_3 = 5$$
$$x_1 \geq 0,\ x_2 \geq 0,\ x_3 \in \mathbb{R}$$

Solve a constraint for x_3: $x_3 = 5 - x_1 - 3x_2$

Substitute to generate equivalent LP:

$\min_{x_1,x_2} 4x_1 + 5x_2 + (5 - x_1 - 3x_2)$ s.to $2x_1 + x_2 + 5 - x_1 - 3x_2 = 4 \Leftrightarrow x_1 - 2x_2 = -1$

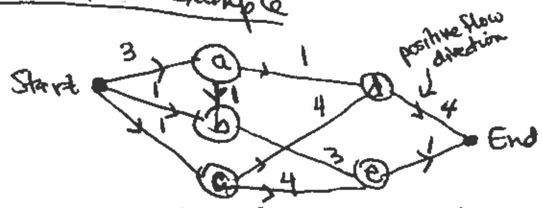
drop constant since doesn't affect sol'n

$\min_{x_1,x_2} 3x_1 + 2x_2$ subject to $-x_1 + 2x_2 = 1$

want b's ≥ 0

Read example application in Section 2.2

Network flow example



max flow rate on each segment

What flows on each segment maximize overall flow from start to end?

$\max x_{start,a} + x_{start,b} + x_{start,c}$

subject to

$x_{start,a} = x_{a,b} + x_{a,d}$

$x_{start,b} + x_{a,b} = x_{b,e}$

$x_{start,c} = x_{c,d} + x_{c,e}$

$x_{a,d} + x_{c,d} = x_{d,end}$

$x_{b,e} + x_{c,e} = x_{e,end}$

balance flow through each node (junction)

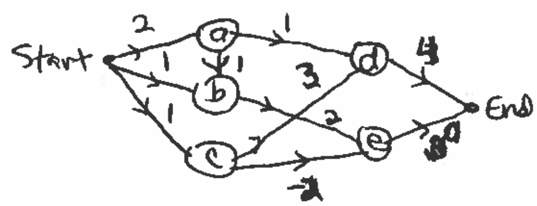
$-3 \leq x_{start,a} \leq 3$, etc

so replace $x_{start,a}$ with $u_{start,a} = x_{start,a} + 3 \geq 0$ and ≤ 6

Equalities:

$$\begin{array}{cccccccccc} x_{start,a} & x_{start,b} & x_{start,c} & x_{a,b} & x_{a,d} & x_{b,e} & x_{c,d} & x_{c,e} & x_{d,end} & x_{e,end} \\ \end{array}$$
$$\begin{bmatrix} 1 & 0 & 0 & -1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & -1 \end{bmatrix}$$

Optimal sol'n:



(not unique) due to loops

$\boxed{\text{R script}}$ don't need standard form, but does assume all nonnegative variables

Need a bit of linear algebra to study the theory of solving LPs.

A $m\times n$ matrix
x $n\times 1$ vector
b $m\times 1$ vector

$$\begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix}\begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix}$$

m eq's in n unknowns

Suppose $m \le n$ and A has m linearly independent columns $\rightarrow$ $\{\vec a_1, \ldots, \vec a_m\}$ lin ind if $x_1\vec a_1 + \cdots + x_m\vec a_m = \vec 0$ always implies $x_1 = \cdots = x_m = 0$

Can reduce system to yield particular sol'n plus linear combination of vectors involving free variables

Example: $A = \begin{bmatrix} 1 & 2 & 1 & 4 & 3 \\ 0 & 1 & 1 & 2 & 2 \\ 1 & 2 & 0 & 1 & 0 \end{bmatrix}$ $[A|b]$ row reduces to $\left[\begin{array}{ccccc|c} 1 & 0 & 0 & 3 & 2 & 5 \\ 0 & 1 & 0 & -1 & -1 & -2 \\ 0 & 0 & 1 & 3 & 3 & 0 \end{array}\right]$ $\begin{aligned} x_1 &= 5 - 3x_4 - 2x_5 \\ x_2 &= -2 + x_4 + x_5 \\ x_3 &= -3x_4 - 3x_5 \end{aligned}$

$m = 3$
$n = 5$

$b = \begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix}$ $\Rightarrow$ general sol'n is $\vec x = \begin{bmatrix} 5 \\ -2 \\ 0 \\ 0 \\ 0 \end{bmatrix} + x_4\begin{bmatrix} -3 \\ 1 \\ -3 \\ 1 \\ 0 \end{bmatrix} + x_5\begin{bmatrix} -2 \\ 1 \\ -3 \\ 0 \\ 1 \end{bmatrix}$ x_4, x_5 free var

Let B be the $m\times m$ matrix formed from the m lin ind columns of A (correspond to pivots)

In example, $B = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 1 & 1 \\ 1 & 2 & 0 \end{bmatrix}$

Solving $B\vec x_B = \vec b$ gives $\vec x_B = B^{-1}\vec b$ (in example, $\vec x_B = B^{-1}\begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix} = \begin{bmatrix} 2 & -2 & -1 \\ -1 & 1 & 1 \\ 1 & 0 & -1 \end{bmatrix}\begin{bmatrix} 1 \\ -2 \\ 1 \end{bmatrix} = \begin{bmatrix} 5 \\ -2 \\ 0 \end{bmatrix}$)

~~The general~~ A particular sol'n to $A\vec x = \vec b$ is then $\begin{bmatrix} \vec x_B \\ \vec 0 \end{bmatrix}$ $\leftarrow$ $n-m$ zeros (if pivots in first m columns, otherwise need to order accordingly)

which the text calls a <u>basic sol'n</u>

and the <u>basic variables</u> are those corresponding to the components of $\vec x_B$ ($x_1, x_2,$ in exam

Example: $A = \begin{bmatrix} 1 & 2 & 3 & -1 & -1 \\ 0 & 1 & 1 & 0 & -1 \\ 1 & 0 & 1 & 2 & 4 \end{bmatrix}$ $\vec b = \begin{bmatrix} 0 \\ 2 \\ 2 \end{bmatrix}$ $[A|b]$ row reduces to $\left[\begin{array}{ccccc|c} 1 & 0 & 1 & 0 & 2 & -2 \\ 0 & 1 & 1 & 0 & -1 & 2 \\ 0 & 0 & 0 & 1 & 1 & 2 \end{array}\right]$

basic variables are x_1, x_2, x_4

$B = \begin{bmatrix} 1 & 2 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & 2 \end{bmatrix}$ $\vec x_B = B^{-1}\vec b = \begin{bmatrix} -2 \\ 2 \\ 2 \end{bmatrix}$ basic sol'n is $\begin{bmatrix} -2 \\ 2 \\ 0 \\ 2 \\ 0 \end{bmatrix}$

Note: columns of B (square matrix) are lin ind, so form a <u>basis</u> for $\mathbb{R}^m$

(can express any $\vec b \in \mathbb{R}^m$ as a linear combination of columns of B, with coefficients given by $\vec x_B$) $\vec b = B\vec x_B = [\vec b_1 \cdots \vec b_m]\begin{bmatrix} x_{B,1} \\ \vdots \\ x_{B,m} \end{bmatrix} = x_{B,1}\vec b_1 + \cdots + x_{B,m}\vec b_m$

Full rank assumption: The $m \times n$ matrix A has $m < n$ and the m rows of A are lin ind

(typically assumed for LP's in textbook) (pivot in every row after row reducing) (no redundant equations)

This implies $A\vec{x} = \vec{b}$ always has ~~at least one sol'n~~ a basic sol'n.

Def'n If one or more components of basic sol'n equals zero, then that basic sol'n is said to be degenerate (like in 1st example) * ambiguity in particular sol'n between basic & free variables, since both have value 0

Now add nonnegativity constraint:

Def'n A vector $\vec{x}$ satisfying $A\vec{x} = \vec{b}$ and $\vec{x} \geq \vec{0}$ is said to be feasible.

A feasible sol'n that is also a basic sol'n is called a basic feasible sol'n.

Section 2.4 Fundamental Theorem of Linear Programming

Given LP in standard form, minimize $\vec{c}^T\vec{x}$ subject to $A\vec{x} = \vec{b}$ and $\vec{x} \geq \vec{0}$, where A is $m \times n$ matrix of rank m, then

(i) if there is a feasible sol'n, there is a basic feasible sol'n.
(ii) if there is an optimal feasible sol'n, there is an optimal basic feasible sol'n.

Importance of this thm: reduces search for optimal sol'ns to finding the basic feasible sol'ns. Still very inefficient, so will need an efficient algorithm for solving LPs.