Oct 28 — Lec 21

LINEAR PROGRAMMING

Definition of Linear Programs

list of given variables $x_1, \dots, x_n$ where $x_i \in \mathbb{R}$

list of linear conditions $\sum_{1 \le i \le n} a_{ij} x_i \quad \le \text{ or } \ge \text{ or } = b_j$

an objective function to maximize/minimize: $\sum_{1 \le i \le n} c_i x_i$

where c_i are real number constants

linear: $x_1 + 2x_2 + 5x_3$

nonlinear: $x_1^2 + x_2^2$, $x_1 x_2$, $\sin x$

Standard Form — linear programs can be presented in matrix form

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & \ddots & & \\ \vdots & & a_{ij} & \\ a_{m1} & & & a_{mn} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ \vdots \\ x_i \\ \vdots \\ x_n \end{bmatrix} \begin{matrix} \ge \\ \text{or} \\ = \\ \text{or} \\ \le \end{matrix} \begin{bmatrix} b_1 \\ \vdots \\ b_j \\ \vdots \\ b_m \end{bmatrix}$$

$A \cdot \vec{x} \ge \vec{b}$ — enforcing constraints

* vars ↑ ↑ * constraints

objective function

→ max/min $\vec{c}^T \vec{x} \iff \sum_{i=1,\dots,n} c_i x_i$

ex1. Political Advertising Problem

4 issues to place advertisements on: roads, guns, farms, gas

3 voting demographics to balance: urban, suburbs, rural

for each dollar spent on issue, votes change:

	road (x_1)	guns (x_2)	farms (x_3)	gas (x_4)
urban	-2	8	0	10
suburbs	5	2	0	0
rural	3	-5	10	-2

budget limit = $200000

need at least 50K urban, 100K suburb, 25K rural votes

WANT: minimize budget

LP — variables: x_1, x_2, x_3, x_4 = $ spent on each issue

constraints: $x_1, x_2, x_3, x_4 \ge 0$

$x_1 + x_2 + x_3 + x_4 \le 200000$

$-2x_1 + 8x_2 + 0x_3 + 10x_4 \ge 50000$

$5x_1 + 2x_2 + 0x_3 + 0x_4 \ge 100000$

$3x_1 - 5x_2 + 10x_3 - 2x_4 \ge 25000$

objective function: minimize $x_1 + x_2 + x_3 + x_4$

In standard form:

$$\begin{bmatrix} -2 & 8 & 0 & 10 \\ 5 & 2 & 0 & 0 \\ 3 & -5 & 10 & -2 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -1 & -1 & -1 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \ge \begin{bmatrix} 50 \\ 100 \\ 25 \\ 0 \\ 0 \\ 0 \\ 0 \\ -200000 \end{bmatrix}$$

ex2. A company produces 2 kinds of products A, B

can produce at most 400 units total

product A → $1 profit max 300 units

product B → $6 profit max 200 units

want: max profit

LP — variables x_A: # product A to produce

x_B: # product B to produce

constraints $x_A + x_B = 400$

$x_A, x_B \ge 0$

$-x_A \ge -300 \quad -x_B \ge -200$

objective function: max $x_A + 6x_B$

ex3 Network Flow in LP

Given network $G = (V, E)$ with $c(e)$ for $e \in E$

Construct a LP to produce optimal flow

variables: $f(e)$ for all $e \in E$

constraints: $f(e) \ge 0$ for all $e \in E$

$f(e) \le c(e)$ for all $e \in E$

$$\sum_{(u,v) \in E} f(u,v) - \sum_{(v,u) \in E} f(v,u) = 0$$

into v; out of v; for all $v \in V - \{s,t\}$

objective function: max $\sum_{(s,u) \in E} f(s,u)$

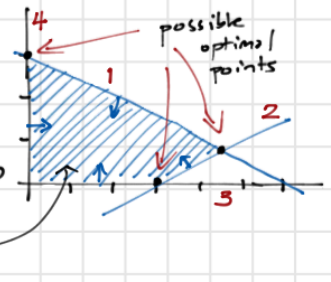
can be max flow in/out any vertex

~~CS 330 Lecture~~

Geometric representation of a LP

constraints: 1. $x_1 + 2x_2 \leq 6$
2. $2x_1 - 3x_2 \leq 6$
3. $x_1 \geq 0$ 4. $x_2 \geq 0$

$\max 2x_1 + 3x_2$



Feasible Region — set of values for variables that satisfy all constraints — types of feasible regions:

1. empty ⇒ no solution to given constraints
   - ex. $x_1 \geq 2$, $x_1 \leq 1$
2. unbounded ⇒ no solution for max optimization
   treated as bounded for min optimization
3. bounded ⇒ one or infinitely many solutions depending on objective function

Simplex — solve LP by moving from vertex to vertex

reason: optimality occurs on the boundary of a constraint with respect to one variable → intersecting point with respect to multiple

but not optimal to check all points since n variables ⇒ $O(2^n)$

- ex. $0 \leq x, y, z \leq 1$
  6 points with 3 variables

Simplex starts from a given point, each time moves to an adjacent point with a higher/lower obj function value
⇒ local search based

Worst-case running time ∈ exponential
in practice — very efficient (shown through smooth analysis)

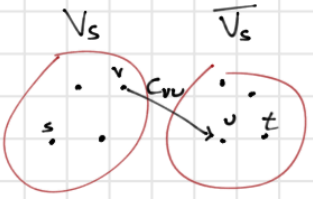
Express Mincut in LP

max flow problem ⟺ min cut problem
↳ $f(s,T) \leq cap(S,T)$ ↲ find the cut with minimum capacity
maximize this

Mincut — find a cut $X = (V_s, \overline{V_s})$ such that $s \in V_s$, $V_s \subseteq V$ and $cap(V_s, \overline{V_s})$ is min

introduce a new variable x

$0 \leq x_v \leq 1$
$x_v = 1$ if $v \in V_s$
$= 0$ if $v \notin V_s$
$x_s = 1$; $x_t = 0$



Consider an edge $e = (v, u)$. Want:

$$x_{vu} = \begin{cases} 1 & \text{if } v \in V_s \text{ and } u \notin V_s \\ 0 & \text{otherwise} \end{cases}$$

can be expressed as: $x_{vu} \leq x_v$ ← ensures $v \in V_s$
$x_u + x_{vu} \leq 1$ ← ensure $u \notin V_s$

Full constraints

$0 \leq x_v \leq 1$
$0 \leq x_{vu} \leq 1$
$x_s = 1$, $x_t = 0$
$x_{vu} - x_v \leq 0$
$x_{vu} + x_u \leq 1$

Objective Function:

$$\text{minimize} \sum_{vu \in E} x_{vu} c_{vu}$$

2 tricks: express conditions as inequality
introduce new variables if needed

for every LP that is maximizing, ∃ a LP that is minimizing

Nov 01 Lec 23

MinCut in LP

Vars: for $v \in V$: x_v Intuition: $x_v = 1$ iff. $v \in S$

for $vu \in E$: x_{vu} vu crosses the cut $v \in S, u \notin S$

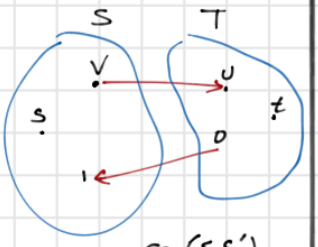
Constraints: $x_s = 1 \quad s \in S$

$x_t = 0 \quad t \notin S$

$0 \le x_v \le 1 \quad v \in V, v \neq s, t$

$0 \le x_{vu} \le 1 \quad vu \in E$

$x_{vu} \ge x_v - x_u$



$Cap(S, S')$

objective f: $\min \sum_{vu \in E} c_{vu} x_{vu}$

note:

1. indicator variables

$x_v \rightsquigarrow v \in S$

2. new variables can make expressing validity conditions & objective functions easier

3. multiple conditions

a. $cond_1$ AND $cond_2$ → express them seperately and put them both under LP constraints

b. $cond_1$ OR $cond_2$ → $x_1 \ge 0$ OR $x_2 \ge 0$

$\iff x_1 + x_2 \ge 0$

c. not cond → not $x_1 > 0$ $\iff 1 - x_1 \ge 1$

d. if $cond_1$ then $cond_2$ → $x_1 > 0 \Rightarrow x_2 > 0$

$\iff x_2 \ge x_1$

e. if $cond_1$ → need a new var x_1

then $cond_1$ iff $x_1 = 1$

$x_1 = 1$ iff. $x \ge 3$

$\iff x_1 - 1 \ge x - 3$

$x_1 < 1$ iff. $x < 3$

$\iff x_1 - 1 \le x - 3$

Dual LP:

$\vec{x} \ge 0 \quad A\vec{x} \ge \vec{b} \quad \vec{y}^T A\vec{x} \le \vec{y}^T \vec{b} \quad \vec{y} \ge 0 \quad \vec{y}^T A \ge \vec{c}^T \quad \vec{y}^T A\vec{x} \ge \vec{c}^T \vec{x}$

$\max \vec{c}^T \cdot \vec{x} = \max \sum_{1 \le i \le n} c_i x_i$ Primal LP — $\min \vec{y}^T \cdot \vec{b}$ Dual LP

Max Flow $\le$ Cap of Min Cut

$\vec{c}^T \le \vec{y}^T A\vec{x} \le \vec{y}^T \vec{b}$

Sometimes the dual LP is easier to solve than primal

Single Source Shortest Path

given graph $G = (V, E)$ and $w(e)$ for all $e \in$

find the shortest path $s \to v$ for all $v \in V - \{$

Variable: x_v = weight of the shortest path $s \to v$

constraint: $x_s = 0$

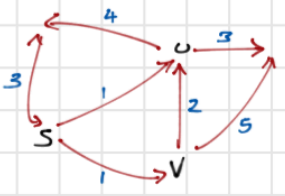
$x_v \ge 0$

$x_v \le x_u + w(u, v)$ for each $(u, v) \in E$

this will eventually find the correct shortest path weight for each $v \in V$ and set that as the upperbound limit for x_v

- ex. $x_u \le x_s + 1$

$x_u \le x_v + 2$



objective function $\max \sum_v x_v$

min doesn't work b/c allows algo to set x_v smaller than true distances

max works b/c constraints forces x_v to be no more than shortest distance and max forces x_v to be atleast the shortest distance

Oct 31 — Tutor

Linear Programming Problem: max/min $c \cdot x$ $\quad A \in \mathbb{Q}^{m \times n}$

given constraints $Ax \leq b$ $\quad x \in \mathbb{Q}^n$

$b \in \mathbb{Q}^m \quad c \in \mathbb{Q}^m$

ex. Simple Scheduling Problem

INPUT: n jobs with durations $d_1, d_2, \ldots, d_n \in \mathbb{Z}$ and prerequisities $P_{i,j} \in \{0,1\}$ such that

$$P_{ij} = \begin{cases} 1 & \text{if } j_i \text{ is a prerequisite of } j_j \\ 0 & \text{otherwise} \end{cases}$$

jobs cannot be interrupted

OUTPUT — T = total time to complete all jobs ← minimize

Variables: x_i = start time of job i

T = total time

Constraints: $x_i \geq 0$ for all $i = 1, \ldots, n$

$x_i \geq x_j + d_j$ if $P_{ji} = 1$ for all $i, j = 1, \ldots, n$

This constraint ensures each job only starts after all of its prerequisites has completed

$\iff x_i \geq P_{ji}(x_j + d_j)$

$T \geq x_i + d_i$ for $i = 1, \ldots, n$

using T ensures none of the values in the solution are greater than they need to

Obj function: min T

ex. Hitting Set Problem

INPUT: set of elements $E = \{x_1, x_2, \ldots, x_n\}$

set of subsets of E, $S = \{H_1, H_2, \ldots, H_m\}$

OUTPUT: smallest subset C of E such that for each set H_i, $C \cap H_i \neq \emptyset$

Variable: v_i for each $x_i \in E$

Constraints: $v_i \in \{0,1\}$ for each element, such that

$$v_i = \begin{cases} 1 & \text{if } v_i \in C \\ 0 & \text{otherwise} \end{cases}$$

$v_a + \cdots + v_k \geq 1$ for each $H_j = \{x_a, \ldots, x_k\}$

This constraint ensures atleast one element from each subset H is picked

objective function: minimize $\sum_{i=1 \text{ to } n} v_i$

note: this is Integer Programming b/c restrict v_i to $\{0,1\}$, not $0 \leq v_i \leq 1$

$x_4 \leq x_5 + 1$

$\Rightarrow x_4 \leq 1$

Test 2 up to here

Nov. 4, 2013

$\max \vec{c}^T \vec{x} = \min -\vec{c}^T \vec{x}$

Dual

OPT

Primal

• "Feasability":

$\begin{cases} x + y \leq 1 + d_1 \\ x \geq 0 \\ y \geq 1 \end{cases}$

$\Leftrightarrow -x \leq -1 + d_2$

$\Leftrightarrow -y \leq -1 + d_3$

Min d

First valid soln:

$x = y = 0$

$d_1 = \max\{-(-1), -1\} = 1$

*LP vars that are for mind?

No answer: Make feasible sets nonempty

IDEA for checking feasibility and finding first soln:

1) Add new vars to RHS of inequalities of constraints (weaker side)
2) Minimize them: $\sum d_i$
3) Check if all $d_i \leq 0$. Not $\Rightarrow$ original LP not feasible.

$\begin{cases} A\vec{x} \leq \vec{b} + \vec{d} \\ \text{Min } \vec{d} \end{cases}$

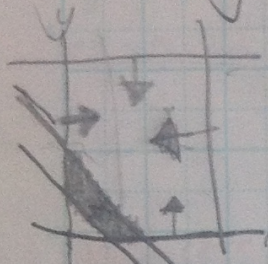
* Solve it, initial $\vec{x} = \vec{0} \wedge \vec{d} = \max -b_i$
* Check if Min $\vec{d} \leq 0$. Not $\Rightarrow$ original LP not feasible

- $x, y \geq 0$
  $x + y \geq 1$

~~Unbounded region~~

Add $x + y \leq d$ ← constant

max $2x + y$



(Don't want to cut bounded solns)
Can cut unbounded solns

Checking Unboundedness
1) Add new constraints that make sure new region is bounded.
2) If original region was bounded, we don't lose any solns.
3) Solve new LP. If optimal is on new constraints, then original was unbounded.

$x + y \leq 2$
$-x + y \leq 0$
$y \leq 1$

max $2x + y$

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

$$\begin{bmatrix} 1 & 1 \\ -1 & 0 \\ 0 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \leq \begin{bmatrix} 2 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$$\vec{y}^T A \geq \vec{c}$$

$$[a\ b\ c\ d] \begin{bmatrix} 1 & 1 \\ -1 & 0 \\ 0 & -1 \\ -1 & 1 \end{bmatrix} \geq \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

max $\vec{c}^T \vec{x}$

$$\begin{bmatrix} 2 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

min $\vec{y}^T \vec{b}$

$$[a\ b\ c\ d] \begin{bmatrix} 2 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

Primal LP

Dual LP

Problem: A[1...n]
Find a substring of A w/ max sum

-1	10	9	-1	3	4	-20

Q: What possibilities do we have, assuming what? Think recursively:

Assume we have a max sum substring.
If either contains the last number or not:
Case 1: Contains a
Then max substring

} Induction step

$x + y \leq 2$

$-y \leq 0$

$y \leq 1$

max $2x+y$

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

$$\begin{bmatrix} 1 & 1 \\ -1 & 0 \\ 0 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \leq \begin{bmatrix} 2 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$[a\ b\ c\ d] \begin{bmatrix} 1 & 1 \\ -1 & 0 \\ 0 & -1 \\ -1 & 1 \end{bmatrix} \geq \begin{bmatrix} 2 \\ 1 \end{bmatrix}$

max $\vec{c}^T \vec{x}$

$$[2\ 1] \begin{bmatrix} x \\ y \end{bmatrix}$$

min $\vec{y}^T \vec{b}$

$$[a\ b\ c\ d] \begin{bmatrix} 2 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

Primal LP

Dual LP

Problem: A[1...n]

Find a substring of A w/ max sum

($O(n^2)$ many)

-1	10	5	-1	3	4	-20

Q: What possibilities do we have, assuming what? Think recursively →

Assume we have a max sum substring. } Induction step:

It either contains the last number or not.

Case 1: Contains a_n

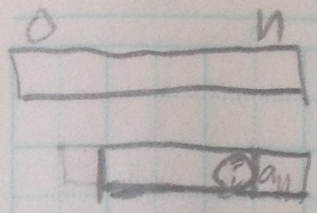
Then max substring $M[n] = M[n-1]$

Case 2: Contains a_n

Then max substring = Max substring ending w/ $a_{n-1} + a_n$

$M[n] = M[n-1] + a_n$

↳ Max substring ending w/o a_n.



↱ Max substring ending w/ a_n

Two cases for $N[n]$:
1) Contains a_{n-1} ⇒ $N[n] = N[n-1] + a_n$
2) ↛ Contains a_{n-1} ⇒ $N[n] = \max(a_n, 0)$ ↓ considers empty string

$N[n] = \max\{N[n-1] + a_n, a_n, 0\}$

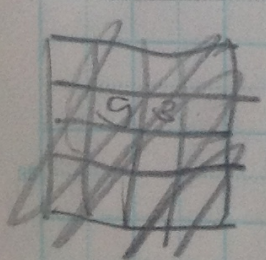
$M[n] = \max\{M[n-1], N[n-1] + a_n, 0\}$

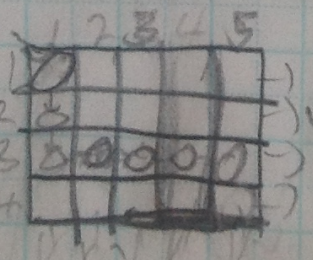
$M[0] = 0$; $N[0] = 0$
for i from 1 to n: } $\in O(n)$
→ $N[i] = \bullet$
→ $M[i] = \bullet$

Correctness: How to formulate (Case-by-case formulate).

Induction of value computed after i^{th} iteration are correct (LI).
($M[i] = \bullet$ $N[i] = \bullet$).

Assume true for $j < i$. Show true for i:
1) $M[n]$ corresp. a valid substring (cases)
2) $M[n]$ is optimal
Assume S is optimal substring of $A[1 \ldots n]$: $s_1, \ldots, s_k$.
Show $S \leq M[n]$
← Consider substring corresp to S.
1) Contains a_n:
$s_1 \ldots s_k \Rightarrow s_1 \ldots s_{k-1}$ is a substring in $A[1 \ldots n-1]$.
$\therefore \sum_{s_1 \ldots s_{k-1}} \leq N[n-1] \Rightarrow M[n] = N[n-1] + a_n \geq S$.
2) ↛ Contains a_n.
$S \leq N[n-1] \Rightarrow M[n] = M[n-1] \geq S$





↓ for loop, top-down.

Subproblem: Max sum from square (i, j) =: $M[i, j]$

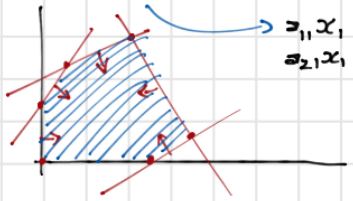
$M[i, j] = M[i+1, j] + a_{ij}$
$\max [M[i, j+1]] + a_{ij}$

Nov 08 — Lec 28

Sign up for take home —— 1 week before class ends

Algorithms for solving LP: Simplex or Interior Point

LP: $A\vec{x} \le \vec{b}$, min/max $\vec{c}^T\vec{x}$



$a_{11}x_1 + a_{12}x_2 = b_1$

$a_{21}x_1 + a_{22}x_2 = b_2$

if we have n variables, each vertex corresponds to the solution of n constraints. The given m constraints

total # vertex = $\binom{m}{n} \in O(m^n)$

Simplex —— find an initial feasible vertex

if there is none

return unfeasible

let v be the initial feasible vertex

while v is not locally optimal

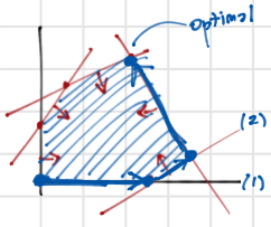
find a new feasible vertex u next to v which is better

v = u

if there are no u, return unbounded

return v

exponential / poly:



(1) $y \ge 0$

(2) $y \ge x - 5$

$\max(x+y)$

to find the next point:

have n constraints, peak next vertex and relax a constraint

given the region is not convex

Best running time: $O(nm) \times O(\#\text{ vertices})$