Lecture #33: Linear Programming

Today: New class of convex optimization problems that can express all sorts of things you'll want to solve

"Hypothetical" Example: MIT wants to create a new major: 6.4, artificial intelligence

Students should have foundations in

Algorithms

Model-Based Learning

Data-Based Learning

Problem: Most courses cover multiple points of view

Solution: Credit System

e.g. 6.036 $\rightarrow \begin{bmatrix} 1 \\ 1 \\ 4 \end{bmatrix}$ Algorithms / Models / Data

$$6.046 \rightarrow \begin{bmatrix} 6 \\ 0 \\ 0 \end{bmatrix}, \text{etc}$$

Q: How do you find the easiest classes to satisfy the major?

$$\min \quad c^T x \quad \leftarrow \text{vector of difficulties}$$

$$\text{s.t.} \quad A x \geq b \quad \leftarrow \text{matrix of points per class}$$

$$0 \leq x \leq 1 \qquad \text{really } x \in \{0,1\}^n$$

non convex

This is called a linear program

Linear objective + Linear Inequality Constraints

Many sorts of problems can be modeled as LPs

def: We say that an LP is written in canonical form if it is of the form:

$$\max \quad c^T x$$

$$\text{s.t.} \quad Ax \leq b$$

$$x \geq 0$$

Q2: How can we write the following optimization problem in canonical form?

$$\begin{aligned} \min \ & w^T x \\ \text{s.t.} \ & Ux \leq v \\ & x \geq 0 \end{aligned}$$

set $c = -w$, $A = U$ and $v = b$

Because the objective is linear it's easy to switch from max to min and vice-versa

Q3: How can we write the following optimization problem in canonical form?

(*) $$\begin{aligned} \max \ & w^T z \\ \text{s.t.} \ & Uz = v \\ & z \geq 0 \end{aligned}$$

called standard form

The key idea is:

$$Uz = v \iff \begin{matrix} Uz \leq v \\ Uz \geq v \end{matrix}$$

Thus (*) $=$ $\max w^T z$

$$\text{s.t.} \begin{bmatrix} U \\ -U \end{bmatrix} z \le \begin{bmatrix} v \\ -v \end{bmatrix}$$

$$z \ge 0$$

So we can set $c = w$, $A = \begin{bmatrix} U \\ -U \end{bmatrix}$, $b = \begin{bmatrix} v \\ -v \end{bmatrix}$

It's easy to see, using canonical form, can even mix and match and get things like

$$\begin{aligned} \max\ & w^T x \\ \text{s.t.}\ & Ax \le b \\ & Ux = v \\ & x \ge 0 \end{aligned}$$

Finally, let's say an LP is in non-standard form if it is

$$\begin{aligned} \max\ & c^T x \\ \text{s.t.}\ & Ax = b \end{aligned}$$

Hmm...

Q4: Can you express an LP written in canonical form instead using non-standard form?

No because there are no inequalities in non-standard form

Now what about actually solving an LP?

There are many methods that work in theory, in practice, or both, e.g.

Simplex **Ellipsoid** **Interior Point**

We will give you a simple (but not the best) way to think about how tools you already know can be used

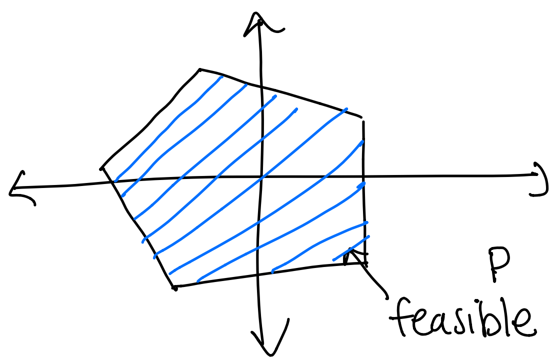
Idea: Set up an unconstrained convex optimization problem and use gradient descent

Main Question: How do we enforce the constraints?

$$Ax \le b$$

(we'll omit $x \geq 0$ from what we do next; can be handled analogously)

Let's visualize what $Ax \leq b$ looks like:



We'll use a barrier – a function B that is

(1) convex

(2) blows up to $+\infty$ as we approach the boundary of P from inside

Let's see an important example called the logarithmic barrier:

$$B(x) \triangleq \sum_{i=1}^{n} -\log\left(b_i - \underset{\substack{\uparrow \\ i^{\text{th}} \text{ row of } A}}{a_i^T} x\right)$$

$Ax \leq b$

This has the desired properties

Now how can we use it? Suppose we want to solve:

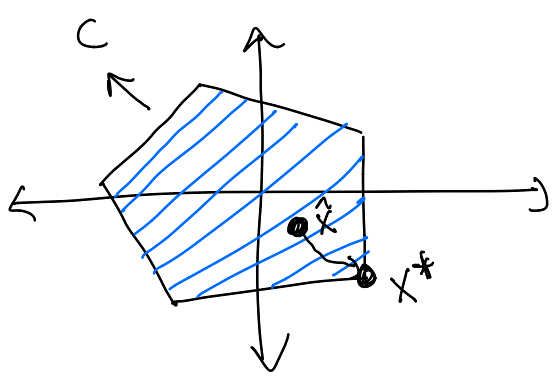
$$(\Delta) \qquad \min \; c^T x \quad \text{s.t. } Ax \leq b$$

Attempt #1: Solve the problem

$$(0) \qquad \min \; c^T x + B(x)$$

Q5: will this find the optimal solution to (Δ)?

It will find something inside P:



The optimal solution to (Δ) will always be on the boundary of P, but the problem $(\circ)$ will find something strictly inside

Instead, let's make the barrier count less!

<u>Attempt #2</u>: Solve the problem

$$(\square) \qquad \min\ c^T x + \alpha B(x)$$

and consider the solution as $\alpha \to 0$

<u>Theorem</u>: As $\alpha \to 0$, then solution $(\square) \to x^*$

Now we can just use gradient descent on the unconstrained problem

Many details to be worked out to check, beyond the sope of this class

Let's do one more example with LPs

<u>Example:</u> Arbitrage in Exchange Rates

Stock exchanges post rates at which they will exchange currency, i.e.g.

$$\begin{tikzcd} \text{USD} \arrow[r, bend left, "20.05"] & \text{Pesos} \arrow[l, bend left, "0.049"] \end{tikzcd}$$

Let c_{ij} = amount in currency j you get from
$c_{i \to j}$ one unit in currency i

Notice $20.05 \times 0.049 = 0.982\ldots < 1$

<u>Q6</u>: what would happen if $c_{ij}\, c_{ji} > 1$?

You can arbitrage (turn $\$1$ into $> \$2$)

It is easy to recognize when it involves just one pair of countries

Q7: How can you find an arbitrage opportunity with LPs?

In particular let's write an LP to turn \$1 into > \$1, if possible

Let's start with the constraints:

$$\forall_{i \neq \underset{\text{USD}}{1}} \quad \sum_j \underbrace{x_{ij}}_{\substack{\text{amount of}\\ \text{currency } i \text{ that}\\ \text{we're turning into } j}} = \sum_j c_{ji} x_{ji}$$

$$x_{ij} \geq 0$$

Informally, we're net zero in everything but USD

Now we want to increase the amount of dollars after all these trades

$$\max \sum_j c_{j1} x_{j1} - \sum_j x_{1j}$$

"net gain in dollars"

Corollary: Optimization can make you rich

Side Note: If you can make the objective positive, you can make it arbitrarily large