ML 540 Tutorial

Reza Babanezhad

rezababa@cs.ubc.ca

UBC

Outline

- Linear Programming
  - Some example
- MAP estimation with different distributions
- Fun with ML

Linear Programming

- Linear programming has two main component:
  - Linear Objective function
  - Linear constraints could including both equalities and inequalities

let $a_{ji}, c_i, x_i, b_j \in \mathbb{R}$, $x^T = (x_1, \dots, x_n)$ $i \in [1..n]$

$j \in [1..m]$

objective function: $\min_{x} \sum_{i=1}^{n} c_i x_i$

$$\text{s.t.} \sum_{i=1}^{n} a_{ji} x_i \le b_j \quad \forall j \in [1..m]$$

$$x_i \ge 0$$

- In matrix form

Let $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $x, c \in \mathbb{R}^n$

then

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & Ax \le b \\ & x \ge 0 \end{aligned}$$

This is extra constraint on x!

- To deal with extra constraint on x
  - Divide x into positive and negative parts

$$x = x^+ - x^-$$

$$x_i^+ = \max(x_i, 0) \geq 0, \quad x_i^- = \max(-x_i, 0) \geq 0$$

objective : min $c^T x^+ - c^T x^-$

s.t. $A x^+ - A x^- \leq b$

$$x^+ \geq 0$$

$$x^- \geq 0$$

- A little geometry
  - Hyperplane : C'x=b is a hyperplane $C, x \in R^n$
    - For example 2x+3y+5z=4 is a hyperplane in 3D space.
    - Hyperplane is a convex set- if we connect two points of the set, the entire line is still in the set
  - Each hyperplane divides the space into 2 half spaces: C'x <= b or C'x > b
    - Half space is a convex set
  - Intersection of two convex set is still convex. (Why?)
  - Polytope: intersection of some half spaces and hyperplane: $Ax \leq b$
    - Polytope is a convex set

$A \in R^{m \times n}$

$b \in R^m$

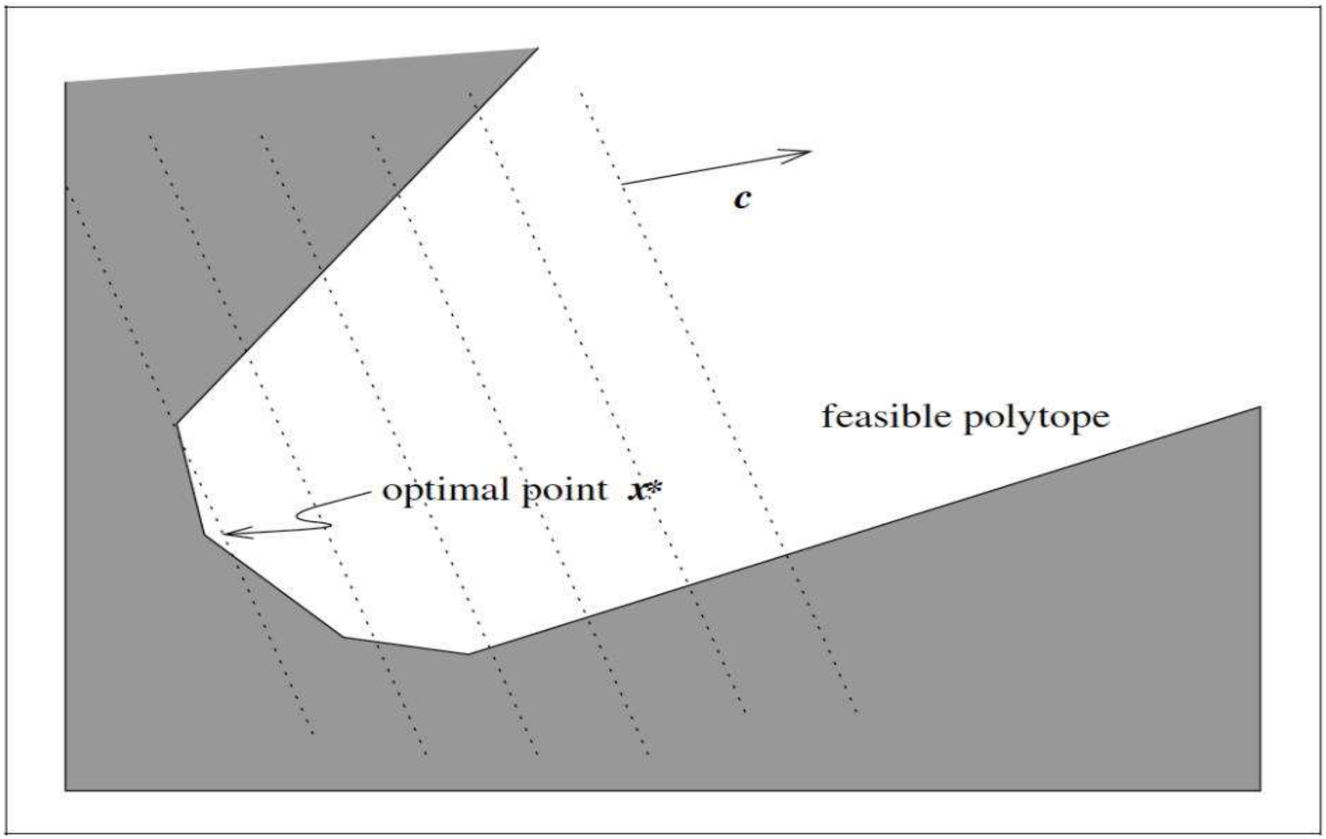
- How can we interpret LP with geometrical objects?
- We can set C’x=z in our LP so
- We want to find a hyperplane which its intersection with a polytope is minimum among all other hyperplanes

$$\min \quad Z$$

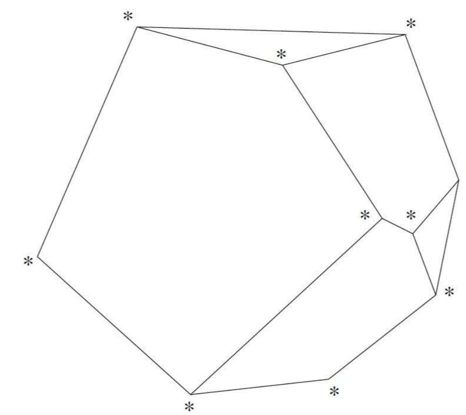
$$\text{s.t.} \quad \underbrace{\begin{cases} Ax \le b \\ x \ge 0 \end{cases}}_{\text{Polytope}}$$

$$x \ge 0 \longrightarrow \text{half-space}$$

$$\underbrace{c^T x = Z}_{\text{hyperplane}}$$


c
feasible polytope
optimal point x*

- How to solve LP?
  - Simplex Method
    - All feasible solutions are vertices of the feasible polytope
    - Cost in worst case: exponential
  - Interior point methods
    - formulate as non-linear problem
    - Polynomial time in worst case e.g. O(n^4L) for ellipsoid method
  - MATLAB uses Interior-Point-Legacy Algorithm



- MATLAB command for LP

$$\min_{x} c^T x$$

$$\text{s.t. } Ax \le b$$

$$A_{eq} x = b_{eq}$$

$$lb \le x \le ub$$

$$x = \mathrm{linprog}(c, A, b)$$

$$x = \mathrm{linprog}(c, A, b, Aeq, beq)$$

$$x = \mathrm{linprog}(c, A, b, Aeq, beq, lb, ub)$$

e.g. 1- $\arg\min_{x} |a_i^T x + b|$ $a_i \in \mathbb{R}^d$, $x \in \mathbb{R}^d$, $b \in \mathbb{R}$

1- Introduce new variable $r \in \mathbb{R}$

$$r \geq \{ax + b, -ax - b\}$$

2- rewrite the problem

$$\min_{r} \; r$$

$$ax - r \leq -b$$

$$-r - ax \leq b$$

3- find appropriate parameters for linprog : C, A, b?

$\min_{r,x} \; r \implies r = \vec{0} \cdot x + 1 \times r \implies C^T = [\underbrace{0 \cdots 0}_{d} \; 1]_{1\times(d+1)}$

$ax - r \le -b$

$-r - ax \le b$

$$\implies \underbrace{\begin{bmatrix} a & -1 \\ -a & -1 \end{bmatrix}}_{A}{}_{2\times(d+1)} \begin{bmatrix} x \\ r \end{bmatrix}_{d+1\times 1} \le \underbrace{\begin{bmatrix} -b \\ b \end{bmatrix}_{2\times 1}}_{b}$$

e.g. 2 :

$$\min_{x} |a_1 x + b| + |a_2 x + b| + |a_3 x + b|$$

$$a_i, x \in \mathbb{R}^d , \; b \in \mathbb{R}$$

1: Introduce r_i corresponding to each $|a_i x + b|$ s.t. $r_i \in \mathbb{R}$

and $r_i \geq \max\{a_i x + b, -a_i x - b\}$

2_ rewrite LP

$$\min_{r_i, x} \; r_1 + r_2 + r_3$$

$$-r_i - a_i x \leq b_i \quad \forall i$$

$$-r_i + a_i x \leq -b_i$$

3 - find c, A, b

$$r_1+r_2+r_3 = \vec{0}^T x + \vec{1}^T \cdot \vec{r} \Rightarrow c^T = [\vec{0}^T_{1\times d} \quad \vec{1}^T_{1\times 3}]$$

$$-r_i - a_i^T x \le b_i \quad \forall i$$

$$-r_i + a_i^T x \le -b_i \Rightarrow \underbrace{\begin{bmatrix} a_1^T & -1 & 0 & 0 \\ a_2^T & 0 & -1 & 0 \\ a_3^T & 0 & 0 & -1 \\ -a_1^T & -1 & 0 & 0 \\ -a_2^T & 0 & -1 & 0 \\ -a_3^T & 0 & 0 & -1 \end{bmatrix}_{6\times(d+3)}}_{A} \begin{bmatrix} x \\ r_1 \\ r_2 \\ r_3 \end{bmatrix}_{(d+3)\times 1} \le \underbrace{\begin{bmatrix} -b_1 \\ -b_2 \\ -b_3 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix}_{6\times 1}}_{b}$$

- let I_n to be $n \times n$ identity matrix

let $\begin{bmatrix} a_1^T \\ a_2^T \\ a_3^T \end{bmatrix} = \alpha \quad \beta = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$

then $A = \begin{bmatrix} \alpha & -I \\ -\alpha & -I \end{bmatrix}$, $b = \begin{bmatrix} -\beta \\ \beta \end{bmatrix}$

In Matlab: A = [α -eye(3); -α -eye(3)], b = [-β; β] e.g.: $\operatorname{argmin}_x \max_i \{ |a_i x + b_i| \}$ $\quad i \in [1..n]$

Let r be the minimum of the LP so for all i we have: $\quad r \geq |a_i x + b_i|$ since $r = \max_i \{ |a_i x + b_i| \}$

so the LP would be

$$\begin{aligned} &\min \; r \\ &\text{s.t.} \quad r \geq a_i x + b_i \qquad \forall i \\ &\qquad\;\; r \geq -a_i x - b_i \end{aligned}$$

and from previous examples we can solve this one ~~too~~!

MAP Estimation

Let $y_i \sim N(\mu_i, 1)$ & $\mu_i \sim N(0, \eta_i)$ $i \in [1..n]$

MAP for μ?

For MAP we need likelihood $\times$ prior

So $\prod_{i=1}^{n} N(\mu_i, 1) \cdot N(0, \eta_i) = \mathcal{L}(\mu)$

now taking the Log ($\log \prod = \sum \log$)

$$\ell(\mu) = \sum_{i=1}^{n} \log N(\mu_i, 1) + \log N(0, \eta_i)$$

$$\log \mathcal{N}(\mu_i, 1) = -\frac{1}{2}(y_i - \mu_i)^2$$

$$\log \mathcal{N}(0, \eta_i) = -\frac{1}{2}\eta_i^{-1}\mu_i^2$$

so $\ell(\mu) = \sum_{i=1}^{n} -\frac{1}{2}(y_i - \mu_i)^2 - \frac{1}{2}\eta_i^{-1}\mu_i^2$

if $y^T = (y_1, \dots, y_n)$, $\mu^T = (\mu_1, \dots, \mu_n)$ and $\eta^{-1} = \begin{bmatrix} \eta_1^{-1} & & 0 \\ & \ddots & \\ 0 & & \eta_n^{-1} \end{bmatrix}_{n \times n}$

$$\ell(\mu) = -\frac{1}{2}\|y - \mu\|^2 - \frac{1}{2}\mu^T \eta^{-1} \mu$$

Let $y_i \sim \mathcal{L}(\mu_i, 1) = \frac{1}{2} \exp(-|y_i - \mu_i|)$, $\mu_i \sim \mathcal{N}(0, \eta)$

MAP for μ?

likelihood $\times$ prior $= \prod_{i=1}^{n} \mathcal{L}(\mu_i, 1) \times \mathcal{N}(0, \eta) = L(\mu)$

$- \log L(\mu) = \sum_{i=1}^{n} |y_i - \mu_i| + \frac{\eta^{-1}}{2} \mu_i^2 + C \longrightarrow$ constant

$\Rightarrow$ MAP :: $\underset{\mu}{\operatorname{argmin}} \; \|y - \mu\| + \frac{\eta^{-1}}{2} \|\mu\|_2^2$

Consider y_i comes from a Student t with ν degree of freedom. And the mean of r.v. y_i is $w^T x_i$ & w is the parameter of the model and $w \sim N(0, \eta I)$. Student t distribution function with 0 mean i.e. $E[\theta]=0$ is:

$$P(\theta|\nu) = c\left(1+\frac{\theta^2}{\nu}\right)^{-\frac{\nu+1}{2}}$$

Find MAP for w?

$E[y_i] = w^T x_i \neq 0$ so to use Student t PDF we set $\theta_i = y_i - w^T x_i$ and let scale parameter to be 1

So $P(\theta_i \mid \nu) = c\left(1 + \frac{|y_i - w^T x_i|^2}{\nu}\right)^{-\frac{\nu+1}{2}}$

likelihood $\times$ prior $= \prod_{i=1}^{n} c\left(1 + \frac{|y_i - w^T x_i|^2}{\nu}\right)^{-\frac{\nu+1}{2}} \times N(0, \eta) = L(w)$

$$-\log L(w) = \sum \frac{\nu+1}{2} \log\left(1 + \frac{|y_i - w^T x_i|^2}{\nu}\right) + \frac{\eta^{-1}}{2} \|w\|_2^2$$

Fun Time

- Assume a start up company hired you as a data scientist to design and implement a recommendation system for them. This company like Youtube allows its users to upload videos. So your job is to build a simple recommendation system considering the list of watched videos and given number of views by each user, recommend new videos to users.

Set-based approach

- Ignore watching number!
- For each video build a set of users
- Find the intersections of these user sets for all videos
- Recommend the video to the users who watched another video whose intersection with this one is big enough.
- What do you think about this model? (population effect?)

Cosine based

- Assume each video as a vector in user space.
- Each dimension shows the number times that video is watched by corresponding user.
- Find the cosine between vectors.
- Rank related videos based on cosine for each videos.
- What is the problem?
  - Ignoring the overall activity of each user- more selective users versus listening to everything

TF-IDF based

- Treat each user as term
- Treat each video as document
- TF: how many times each user watched a video
- IDF: how many times a user watched videos on the site
- Now build a vector of TF-IDF weight for each video
- Find the cosine ☺