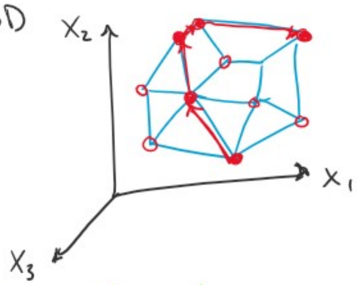
MAP Inference by MILP

Wednesday, November 16, 2022 11:39 AM

2:11 PM

Simplex Algo for LPs (in pictures)

Ex: 3D



☆ not a polytime algo, but common in practice

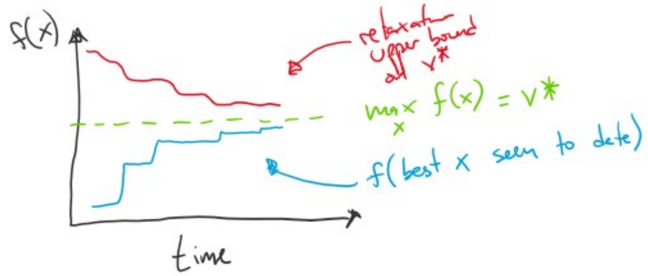
Key Idea:
- OPT is always at a corner of the polyhedron
- walk around the corners greedily

1. pick initial corner
2. inspect adjacent corners
3. pick the "best" adjacent corner ← w/ highest $c^T x$
4. if no improvement, stop

Interior Points

Key Idea: always move within feasible region

☆ polytime algorithms but slower in practice

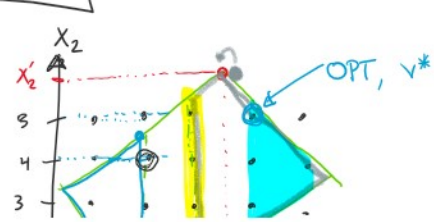


Ex: Branch and Bound for ILP

ILP:

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

s.t. $A\vec{x} \le \vec{b}$

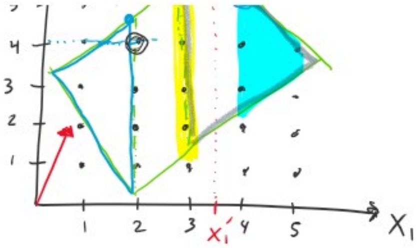


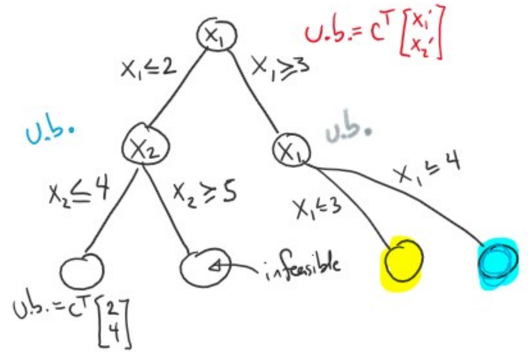
$\vec{x}$

s.t. $A\vec{x} \le \vec{b}$

$0 \le x_i \ \forall i$

$x_i \in \mathbb{Z}$





Lots of heuristics

1. which var to branch on
2. which explore node next
3. ...

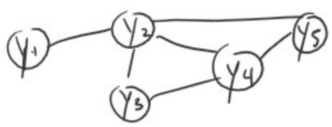
MAP Inference as ILP

Ex: Binary Pairwise MRF, $G = (V, E)$

$\vec{y} \in \{0,1\}^T$, $\quad p_\theta(\vec{y}) = \frac{1}{Z} \left[\prod_{i \in V} \psi_i(y_i)\right] \left[\prod_{(i,j) \in E} \psi_{ij}(y_i, y_j)\right]$

$\psi_i =$

y	$\psi_i(y)$
0	703
1	27



Assume $\psi_i(y_i) = \exp(s_i(y_i))$, $\quad \psi_{ij}(y_i, y_j) = \exp(s_{ij}(y_i, y_j))$

$$\Rightarrow \log p_\theta(\vec{y}) = \sum_{i \in V} s_i(y_i) + \sum_{(i,j) \in E} s_{ij}(y_i, y_j) - \log Z$$

$s_i =$

y	$s_i(y)$
0	$\log(703)$
1	$\log(27)$

Goal $\hat{y} = \arg\max_{y} \log p_\theta(\vec{y})$

Intermediate Step: Quadratic Program

Q1: fill in the blank

quadratic terms

$$\max_{\vec{y}} \underbrace{\left[\sum_{i \in V} \left(y_i\, s_i(1) + (1-y_i)\, s_i(0)\right)\right]}_{= \sum_{i \in V} s_i(y_i)} + \underbrace{\left[\sum_{(i,j) \in E} \left(y_i y_j\, s_{ij}(1,1) + (1-y_i) y_j\, s_{ij}(0,1) + y_i(1-y_j)\, s_{ij}(1,0) + (1-y_i)(1-y_j)\, s_{ij}(0,0)\right)\right]}_{= \sum_{(i,j) \in E} s_{ij}(y_i, y_j)}$$

s.t. $0 \le y_i \ \forall i$, $\quad I\vec{y} \le \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \Longleftrightarrow y_i \le 1 \ \forall i$ — Standard form version of $y_i \in \{0,1\}$

$y_i \in \mathbb{Z}$

ILP:

Need to encode $y_i y_j$ for $y_i \in \{0,1\}$ and $y_j \{0,1\}$

as a linear functio[n]

Need to encode $y_i y_j$ for $y_i \in \{0,1\}$ and $y_j \{0,1\}$ as a linear function

Solution: substitute a new variable w_{ij} for each $y_i y_j$ term to get back a linear objective.

not in our ILP

Let $w_{ij} \triangleq y_i y_j$ $\quad$ $w_{ij} \in \{0,1\}$

Three constraints:

(1) $w_{ij} \leq y_i$

(2) $w_{ij} \leq y_j$

(3) $w_{ij} \geq y_i + y_j - 1$ $\quad$ Q2: fill in the blank with a linear constraint

OR

$$w_{ij} \geq \frac{y_i + y_j}{2} - 0.5$$