4/10/14

MAP LINEAR PROGRAM (LP)

1. Recall Integer Program (IP) from last time

$$MAP(\theta) = \max_{\vec{\mu}} \vec{\theta}^T \vec{\mu}$$

s.t. $\mu_i(x_i) \in \{0,1\}$ $\mu_c(x_c) \in \{0,1\}$

$$\left[\begin{array}{l} \sum_{x_i} \mu_i(x_i) = 1 \\ \sum_{x_c} \mu_c(x_c) = 1 \\ \sum_{x_c \setminus x_i} \mu_c(x_c) = \mu_i(x_i) \end{array}\right]$$

large $A\mu = b$ system

$$\equiv \quad \max \ \theta^T \mu$$

s.t. $A\mu = b$

$\mu(\cdot) \in \{0,1\}$

Note1: # feasible μ's $\equiv$ size of output space

$|X_1| \times |X_2| \times \ldots |X_n|$

Note2: Convex Hull of feasible μ's is called Marginal Polytope

$$Marg(G) \equiv \text{Conv-Hull}\{\mu_{\underline{x}}\}$$

$$= \{\vec{\mu} \mid \vec{\mu} = \sum \lambda_i \mu_{\underline{x}_i}\}$$

$\lambda_i \geq 0$

$\sum \lambda_i = 1$

Linear Programming (LP) Relaxation

$$\max_{\vec{\mu}} \ \vec{\theta}^T \vec{\mu}$$

$$\text{s.t} \quad A\mu = b$$

$$\mu(\cdot) \in \{0,1\} \Rightarrow \mu(\cdot) \in [0,1]$$

LP's are polytime solvable but this LP is too large

$$|\mu| = \sum_i |X_i| + \sum_c |X_c|$$

$$= \#\text{cols in } A$$

$$\#\text{rows in } A = \#\text{vars } (n) + \#\text{factors } |\vec{C}| + \sum_{c \in \vec{C}} |c| \cdot \prod_{i \in c} |X_i|$$

(2)
How to solve this LP?
Lagrangian Dual / Relaxation

Note: LPs are always convex optimization problems
so duality gap is always 0.
So nothing is lost in relaxation

Primal [Assume Pairwise MRF for notational simplicity]

$$\max_{\{\mu_i, \mu_{ij}\}} \quad \sum_i \theta_i^T \mu_i + \sum_{ij} \theta_{ij}^T \mu_{ij}$$

$$\sum_{x_i} \mu_i(x_i) = 1$$

$$\sum_{x_i, x_j} \mu_{ij}(x_i, x_j) = 1$$

$$\text{Dualize} \rightarrow \begin{cases} \sum_{x_j} \mu_{ij}(x_i, x_j) = \mu_i(x_i) \; : \; \lambda_{j\to i}(x_i) & \forall i, j, x_i \\ \sum_{x_i} \mu_{ij}(x_i, x_j) = \mu_j(x_j) \; : \; \lambda_{i\to j}(x_j) & \forall i, j, x_j \end{cases}$$

$$L(\{\mu_i, \mu_{ij}\}, \{\lambda_{i\to j}, \lambda_{j\to i}\}) = \sum_i \theta_i^T \mu_i + \sum_{ij} \theta_{ij}^T \mu_{ij}$$

$$- \underbrace{\sum_{ij}}_{\forall i,j} \Big[\underbrace{\sum_{x_i}}_{\forall x_i} \lambda_{j\to i}(x_i) \Big\{ \sum_{x_j} \mu_{ij}(x_i, x_j) - \mu_i(x_i) \Big\}$$

$$+ \overbrace{\sum_{x_j}}^{\forall x_j} \lambda_{i\to j}(x_j) \Big\{ \sum_{x_i} \mu_{ij}(x_i, x_j) - \mu_j(x_j) \Big\} \Big]$$

[Moving terms around]

$$L(\cdot, \cdot) = \sum_i \sum_{x_i} \Big[\theta_i(x_i) \cdot \mu_i(x_i) + \sum_{j \in N(i)} \lambda_{j\to i}(x_i) \cdot \mu_i(x_i) \Big]$$

$$\underbrace{+ \sum_{ij} \sum_{x_i} \sum_{x_j} \Big[\theta_{ij}(x_i, x_j) \cdot \mu_{ij}(x_i, x_j) - \lambda_{i\to j}(x_i, x_j) \cdot \mu_{ij}(x_i, x_j) - \lambda_{j\to i}(x_i, x_j) \cdot \mu_{ij}(x_i, x_j) \Big]}_{\text{Modified Node \& Edge Potentials}}$$

Why did we do this?
$\therefore$ μ_i & μ_{ij} are now completely decompose variables in the dual!

Specifically dual

$$g(\{\lambda_{i\to j}, \lambda_{j\to i}\}) = \underbrace{\max_{\{\mu_i, \mu_{ij}\}} \quad}_{} L(\{\mu_i, \mu_{ij}\}, \{\lambda_{i\to j}, \lambda_{j\to i}\})$$

Can compute this max separately at each node & edge

Dual is a convex non-differentiable function

$\Rightarrow$ Sub-gradient Descent on Dual

$$\lambda_{i\to j}^{(t+1)}(x_i) = \lambda_{i\to j}^{(t)}(x_i) - \alpha_t \frac{\partial g}{\partial \lambda_{i\to j}(x_i)}$$

After simplification Update Rule:
→ While Not converged
→ Solve modified node & edge potentials independently

$$x_i^* = \arg\max_{x_i} \; \theta_i(x_i) + \sum_{j\in N(i)} \lambda_{j\to i}(x_i)$$

$$x_{ij,i}^* = \arg\max_{x_i, x_j} \; \theta_{ij}(x_i, x_j) - \lambda_{i\to j}(x_j) - \lambda_{j\to i}(x_i)$$

→ If $x_i^* \neq x_{ij,i}^*$
→ $\lambda_{i\to j}^{(t+1)}(x_i^*) = \lambda_{i\to j}^{(t)}(x_i^*) + \alpha_t$
→ $\lambda_{i\to j}^{(t+1)}(x_{ij,i}^*) = \lambda_{i\to j}^{(t)}(x_{ij,i}^*) - \alpha_t$