Outline:

1. What problem it deals with: $\boxed{\text{weighted b-matching}}$ bipartite, weights (not restricted to R^+), degree b (≥ 1), max. See Fig 1. edges uniform capacity

3. Approach: formulate weighted b-matching as a prob distribution function (on GM). run BP (can be parallel) to find the MAP. Node state: Edge. why BP: parallel, early stop for approximate real-time

4. Significance ① theory: identifies another situation where BP on a loopy graphical model can provably and ① practical: efficiently converge to the MAP (max-product) (single loop BP). still bn^3, but much faster

2. Application: ① combinatorial: Resource allocation: n supplier n customer, ship b supplies btwn customer supplier. ② match bidders to sellers in auctions. ③ VLSI, Chinese Postman, shortest path in undirected graphs with negative cost (no negative cycles)

④ In machine learning: modified KNN, pruning weighted affinity graph & removing noisy edges

Not first attempt: $b=1$ using BP, 2005; In NIPS 2006 ~~describe~~ use GM for Combi Opt by blowing up state space (exponential), so standard BP is impractical.

Algo.

Notation: $\mathcal{U} = \{u_1 \cdots u_n\}$, $V = \{v_1 \cdots v_n\}$, $E = \mathcal{U} \times V$ (fully), A_{ij} for edge (u_i, v_j) can be negative, asymmetric

b-matching $M(u_i), M(v_j)$ return set of neighboring ~~the~~ vertices in b-matching. In Fig 1.

e.g. $M(u_1) = \{v_1, v_2\}, \{v_1 v_3\} \cdots \{v_3, v_4\}$, 6 possibilities $\binom{n}{b}$. Pick one only!

So $\forall i \in \{1 \cdots n\}$, $|M(u_i)| = b$, range $\forall j \in \{1 \cdots n\}$, $|M(v_j)| = b$

obj: $$\max_{M(u_i), M(v_j)} \sum_{i=1}^{n} \sum_{v_k \in M(u_i)} A_{ik} + \sum_{j=1}^{n} \sum_{u_k \in M(v_j)} A_{kj} \leftarrow W(M)$$

First Write out $M(u_i), M(v_j)$, and all candidate $M(u_i)$

Consistency: Eg. $M(u_1) = \{v_2, v_3\}$ $M(v_2) = \{u_2, u_3\}$ contradictory! need indicator

~~Denote: $x_i = M(u_i)$, $y_j = M(v_j)$~~

① $v_j \bar{\in} M(u_i) \wedge u_i \bar{\in} M(v_j)$ ① no link

② $v_j \in M(u_i) \wedge u_i \in M(v_j)$ ② link

Denote $x_i = M(u_i)$, $y_j = M(v_j)$.

then $\psi_{ij}(x_i, y_j) = \overline{XOR}(v_j \in x_i, u_i \in y_j)$

$v_j \in M(u_i)$ (x_i)	$u_i \in M(v_j)$ (y_j)	
② T	T	✓
T	F	✗
F	T	✗
① F	F	✓
		$\overline{XOR}$

Now define MRF, s.t. MAP = max b-match ~~and b-match~~ / still same graph, fully connected, but larger state space ($\binom{n}{b}$) x_i can be

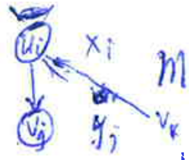
$$\phi_i(x_i) = \exp\left(\sum_{v_j \in x_i} A_{ij}\right) \quad \phi_j(y_j) = \exp\left(\sum_{u_i \in y_j} A_{ij}\right)$$

LOOPY. $\binom{n}{b}$ large space

$$P(\{x_i, y_i\}_{i=1}^{n}) \propto \prod_{i=1}^{n} \phi_i(x_i) \prod_{j=1}^{n} \phi_j(y_j) \prod_{i,j=1}^{n} \psi_{ij}(x_i, y_j) \propto \exp(W(M))$$

But edge pot has nice structure

consistency, not smoothness.

BP: 

$$m_{u_i \to v_j}(y_j) = \max_{x_i} \phi_i(x_i) \overbrace{\boxed{\psi_{ij}(x_i, y_j)}}^{\alpha_{ij}(x_i, y_j)} \prod_{k:k\neq j} m_{v_k \to u_i}(x_i)$$

$k = 1 \dots N$ since fully connected

Therefore diff y_j :

$$b_i(x_i) = \phi_i(x_i) \prod_k m_{v_k \to u_i}(x_i)$$

起手板 ②

x_i is an assignment of u_i, among the $\binom{n}{b}$ choices.
y_j ... v_j

Efficient / compact, exploit structure of ψ_{ij}

motivation: $y_j \left(\right) = \psi_{ij} \left(\psi_{ij} \right)$ ... x_i ... ~~nonlocal~~

same row in ψ_{ij}, then same msg. 这样才行 ① y_j, y_j' both contain u_i, then $\forall$ $\psi_{ij}(x_i, y_j) = \psi_{ij}(x_i, y_j')$ same row

For a particular y_j. ~~enumerate all x_i~~

~~If $u_i \in y_j$, then RHS of~~ $\alpha_{ij}(x_i, y_j) = 0$ ~~if~~

~~if $v_j \bar{\in} x_i$~~ sender is in receiver's matching list by y_j (receiver's assignment) ~~sender & receiver~~

① If $u_i \in y_j$ (~~sender~~ u_i, v_j connected) then ~~need~~ only need to consider x_i which also says $u_i - v_j$ connected

for all x_i.

If $v_j \bar{\in} x_i$ (disconnected) then $\psi_{ij}(x_i, y_j) = 0$. ~~$\alpha_{ij}(x_i, y_j) = 0$~~.

Else if $\psi_{ij}(x_i, y_j) = 1$

右转 ← $\therefore m_{u_i \to v_j}(y_j) = \max_{x_i : v_j \in x_i} \phi_i(x_i) \prod_{k:k\neq j} m_{v_k \to u_i}(x_i)$ ← Independent of y_j! 最后 ...

② If $u_i \bar{\in} y_j$

for all x_i

If $v_j \in x_i$, then $\psi_{ij}(x_i, y_j) = 0$.

Else if $v_j \bar{\in} x_i$, then $\psi_{ij} = 1$.

← $\therefore m_{u_i \to v_j}(y_j) = \max_{x_i : v_j \bar{\in} x_i} \phi_i(x_i) \prod_{k:k\neq j} m_{v_k \to u_i}(x_i)$ set to 1

指入

So the $\binom{n}{b}$ y_j are divided into two categories, and we only need to calculate two REAL numbers. Furthermore Normalize, so that $\boxed{\forall y_j : u_i \bar{\in} u_j,\ m_{u_i \to v_j}(y_j) = 1}$

~~In ① $\prod_{k:k\neq j} m_{v_k \to u_i}(x_i) = \prod_{k:k\neq j, v_k \in x_i} m_{v_k \to u_i}(x_i) \prod_{k:k\neq j, v_k \bar{\in} x_i} m_{v_k \to u_i}(x_i)$~~

~~sender~~ ~~receiver~~

~~In ②~~

~~$m_{u_i \to v_j}(y_j)$~~ m

$\prod_{\cancel{k:k\neq j}, v_k \in x_i} m_{v_k \to u_i}(x_i)$

sender isn't in receiver's matching ~~sender & receiver are NOT conn~~

But ~~then~~ what should $m_{u_i \to v_j}(y_j)$ be if $i \sim j$ by y_j?

Let's call the normalized message by $\tilde{m}$. So $\tilde{m}_{u_i \to v_j}(y_j) = 1$

now can actually drop (y_j).

But keeps for clarity

if $u_i \bar{\in} y_j$ (i.e. u_i, v_j ~~not connected~~

i.e. sender isn't in receiver's matching)

Otherwise, if connected then.

3

$$\frac{m_{u_i \to v_j}(y_j) \;\; j\sim i}{m_{u_i\to v_j}(y_j) \;\; j \nmid i} = \frac{\max_{x_i : v_j \in x_i} \phi_i(x_i) \prod_{k: k\neq j} \tilde{m}_{v_k \to u_i}(x_i)}{\max_{x_i : v_j \notin x_i} \ldots}$$

$\tilde{m}_{u_i \to v_j}(y_j)$ //

$$= \frac{\max_{x_i : v_j \in x_i} \cancel{\phi_i(x_i)} \prod_{k: v_k\in x_i} \exp(A_{ik}) \prod_{k:k\neq j,\, v_k \in x_i} \tilde{m}_{v_k\to u_i}(x_i) \cdot \overbrace{\prod_{k:k\neq j,\, v_k\notin x_i} \tilde{m}}^{1}}{\max_{x_i : v_j \notin x_i} \cancel{\phi_i(x_i)} \prod_{k: v_k \in x_i} e^{A_{ik}} \prod_{k: k\neq j,\, v_k\in x_i} \tilde{m}_{v_k \to u_i}(x_i) \cdot \prod 1}$$

$k \neq j$ redundant

but $\left.\begin{matrix} v_j \notin x_i \\ v_k \in x_i \end{matrix}\right\} \Rightarrow k\neq j$

product over the rest $b-1$ elements in x_i ⊛

$$= \frac{\max_{x_i : v_j \in x_i} e^{A_{ij}} \prod_{k:k\neq j,\, v_k \in x_i} e^{A_{ik}} \tilde{m}_{v_k \to u_i}(x_i)}{\max_{x_i : v_j \notin x_i} \prod_{k: v_k \in x_i} e^{A_{ik}} \tilde{m}_{v_k\to u_i}(x_i)} .$$

In numerator, ~~x_i to maximize~~ the feasible x_i ~~will be chosen among~~ are those which s.t.
① $v_j \in x_i$ ② $|x_i| = b$. To maximize, the rest $b-1$ v_k's will be the largest $\boxed{b-1}$ ~~candidates~~ values of $e^{A_{ik}} \tilde{m}_{v_k \to u_i}(x_i)$ over $k\neq j$

$$= \frac{e^{A_{ij}} \cdot \text{Product of } b-1 \text{ largest values of } e^{A_{ik}} \tilde{m}_{v_k \to u_i}(x_i) \text{ over } k\neq j}{\text{Product of } b \text{ largest values of } e^{A_{ik}} \tilde{m}_{v_k \to u_i}(x_i) \text{ over } k \neq j}$$

$$= \frac{e^{A_{ij}}}{\text{the } b^{th} \text{ largest } e^{A_{ik}} \tilde{m}_{v_k\to u_i}(x_i) \text{ over } k\neq j}$$

So only need to find the b^{th} largest. keep b largest $O(bn)$ each takes b.

Finally, we can't efficiently reconstruct $b_i(x_i)$, but can efficiently find it's argmax

$$b_i(x_i) = \prod_{k: v_k\in x_i} e^{A_{ik}} \prod_{k: v_k \in x_i} \tilde{m}_{v_k \to u_i}(x_i) \prod_{k: v_k \notin x_i} \tilde{m}_{\to}(\,)$$

$$= \prod_{k: v_k\in x_i} e^{A_{ik}} \tilde{m}_{v_k \to u_i}(x_i) .$$ So to maximize $b_i(x_i)$ over x_i, just pick the b largest values of $e^{A_{ik}} \tilde{m}_{v_k\to u_i}(x_i)$ over $k = 1 \cdots N$

Nash Equili — ∃MRF, s.t. MAP = best policy = LBP fixed point

msg:

direct motivation?
Like chinese resturaunt process