Beyond Junction Tree: High Tree-Width Models

Tony Jebara
Columbia University

February 25, 2015

Joint work with A. Weller, N. Ruozzi and K. Tang

Data as graphs

Want to perform inference on large networks...
Junction tree algorithm becomes inefficient...

Figure: Social network

Outline

- Goals: perform inference on large networks
- Approach: set up tasks as finding maxima and marginals of probability distribution $p(x_1, \ldots, x_n)$
- Limitation: for cyclic $p(x_1, \ldots, x_n)$ these are intractable
- Methodology: graphical modeling and efficient solvers
- Verification: perfect graph theory and bounds

Graphical models

- We depict a graphical model G as a bipartite factor graph with round *variable* vertices $X = \{x_1, \ldots, x_n\}$ and square *factor* vertices $\{\psi_1, \ldots, \psi_l\}$. Assume x_i are discrete variables.
- This represents $p(x_1, \ldots, x_n) = \frac{1}{Z} \exp\left(\sum_{c \in W} \psi_c(X_c)\right)$ where X_c are variables that neighbor factor c

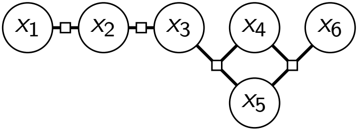
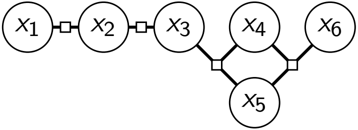


Figure: $p(X) = \frac{1}{Z} e^{\psi_{1,2}(x_1,x_2)} e^{\psi_{2,3}(x_2,x_3)} e^{\psi_{3,4,5}(x_3,x_4,x_5)} e^{\psi_{4,5,6}(x_4,x_5,x_6)}$

Graphical models



- Use marginal or maximum a posteriori (MAP) inference
  - Marginal inference: $p(x_i) = \sum_{X \setminus x_i} p(X)$
  - MAP inference: x_i^* where $p(X^*) \geq p(X)$
- In general:
  - Both are NP-hard [Cooper 1990, Shimony 1994]
  - Both are hard to approximate [Dagum 1993, Abdelbar 1998]
- On acyclic graphical models both are easy [Pearl 1988]
- But most models (e.g. Medical Diagnostics) are *not* acyclic

Belief propagation for tree inference

- Acyclic models are efficiently solvable by belief propagation
- Marginal inference via the sum-product:
  - Send messages from variable v to factor u

$$\mu_{v \to u}(x_v) = \prod_{u^* \in \mathrm{N}(v) \setminus \{u\}} \mu_{u^* \to v}(x_v)$$

  - Send messages from factor u to variable v

$$\mu_{u \to v}(x_v) = \sum_{X'_u : x'_v = x_v} e^{\psi_u(X'_u)} \prod_{v^* \in \mathrm{N}(u) \setminus \{v\}} \mu_{v^* \to u}(x'_{v*})$$

  - Efficiently converges to $p(X_u) \propto e^{\psi_u(X_u)} \prod_{v \in \mathrm{N}(u)} \mu_{v \to u}(x_u)$
- MAP inference via max-product: swap $\sum_{X'_u}$ with $\max_{X'_u}$

How to handle cyclic (loopy) graphical models?

- To make loopy models non-loopy, we *triangulate* into a junction tree. This can make big cliques...
- Messages are exponential in the size of the clique
- **Tree-width** of a graph: size of the largest clique after triangulation

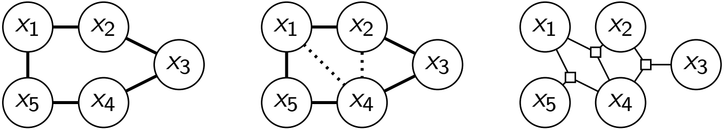
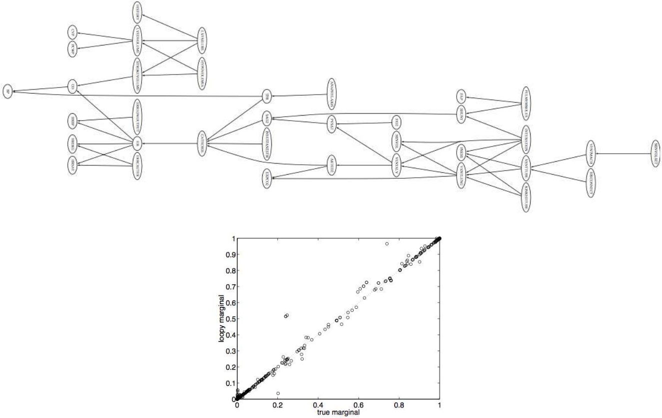


Figure: Triangulating cyclic model $p(X) \propto \phi_{12}\phi_{23}\phi_{34}\phi_{45}\phi_{51}$ makes a less efficient acyclic model $p(X) \propto \phi_{145}\phi_{124}\phi_{234}$.

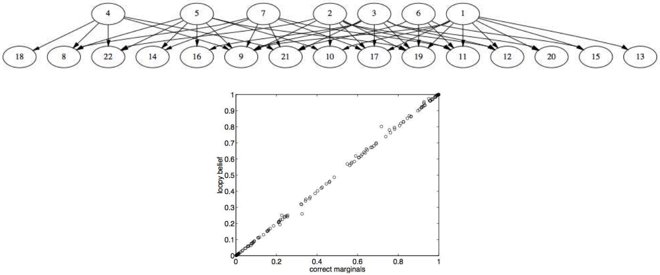
- So... what if we skip triangulation?
- JTA messages may not converge and may give wrong answers

Loopy sum-product belief propagation



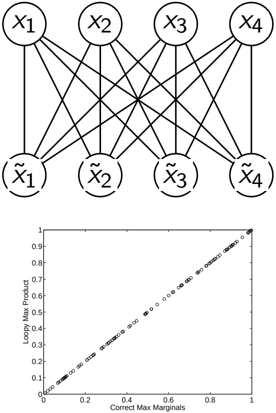
Alarm Network and Results

Loopy sum-product belief propagation



Medical Diagnostics Network and Results

Loopy max-product belief propagation



Bipartite Matching Network and Results

Bipartite matching

	Motorola	Apple	Dell
"laptop"	0\$	2\$	2\$
"server"	0\$	2\$	3\$
"phone"	2\$	3\$	0\$

$$\rightarrow C = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

- Given W, $\max_{C \in \mathbb{B}^{n \times n}} \sum_{ij} W_{ij} C_{ij}$ such that $\sum_i C_{ij} = \sum_j C_{ij} = 1$
- Can be written as a very loopy graphical model
- But... max-product finds MAP solution in $O(n^3)$ [HJ 2007]

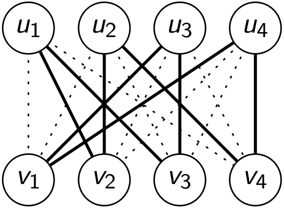
Bipartite b-matching

	Motorola	Apple	Dell
"laptop"	0$	2$	2$
"server"	0$	2$	3$
"phone"	2$	3$	0$

$$\rightarrow C = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

- Given W, $\max_{C \in \mathbb{B}^{n \times n}} \sum_{ij} W_{ij} C_{ij}$ such that $\sum_i C_{ij} = \sum_j C_{ij} = b$
- Also creates a very loopy graphical model
- Max-product also finds MAP solution in $O(n^3)$ [HJ 2007]

Bipartite generalized matching



- Graph $G = (U, V, E)$ with $U = \{u_1, \ldots, u_n\}$ and $V = \{v_1, \ldots, v_n\}$ and $M(.)$, a set of neighbors of node u_i or v_j
- Define $x_i \in X$ and $y_i \in Y$ where $x_i = M(u_i)$ and $y_i = M(v_j)$
- Then $p(X, Y) = \frac{1}{Z} \prod_i \prod_j \varphi(x_i, y_j) \prod_k \phi(x_k)\phi(y_k)$ where $\phi(y_j) = \exp(\sum_{u_i \in y_j} W_{ij})$ and $\varphi(x_i, y_j) = \neg(v_j \in x_i \oplus u_i \in y_j)$

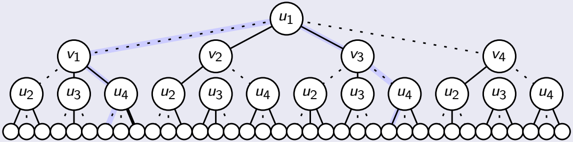
So... why does loopy max-product work for matching?

Theorem (HJ 2007)

Max product finds generalized bipartite matching MAP in $O(n^3)$.

Proof.

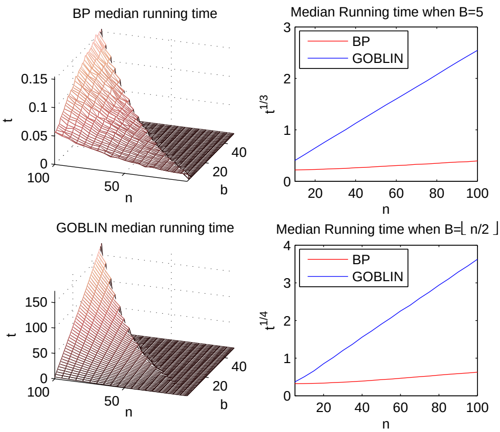
Using unwrapped tree T of depth $\Omega(n)$, we show that maximizing belief at root of T is equivalent to maximizing belief at corresponding node in original graphical model.



$\square$

So some loopy graphical models are tractable...

Generalized matching



Empirically, max product belief propagation needs $O(|E|)$ messages
Code at http://www.cs.columbia.edu/∼jebara/code

Generalized matching

Applications:

- alternative to k-nearest neighbors [JWC 2009]
- clustering [JS 2006]
- classification [HJ 2007]
- collaborative filtering [HJ 2008]
- semisupervised learning [JWC 2009]
- visualization [SJ 2009]
- metric learning [SHJ 2012]
- privacy-preservation [CJT 2013]

Generalized matching vs. k-nearest neighbors

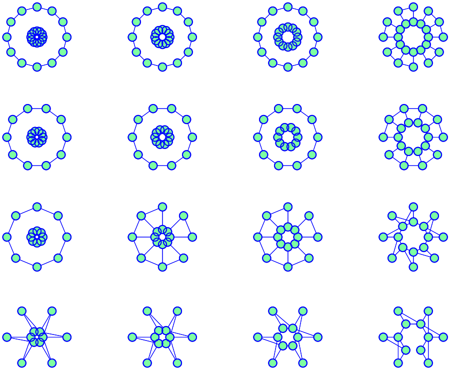


Figure: k-nearest neighbors with $k = 2$ (a.k.a. kissing number)

Generalized matching vs. k-nearest neighbors

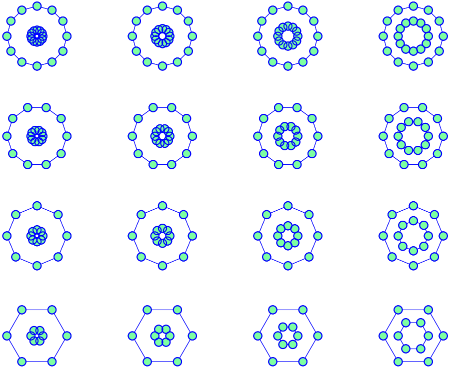
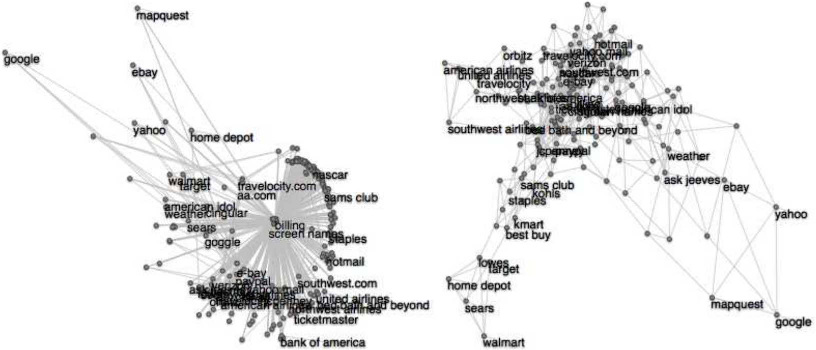


Figure: b-matching with $b = 2$

Generalized matching for link prediction



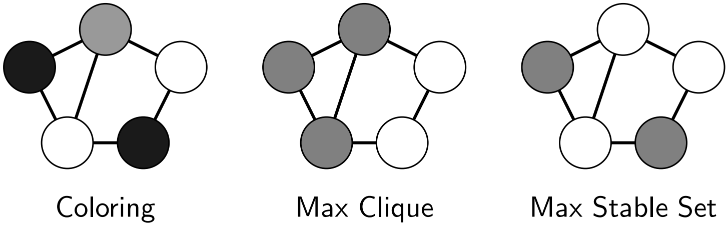
- Linking websites according to traffic similarity
- Left is k-nearest neighbors, right is b-matching

What is a perfect graph?

Figure: Claude Berge

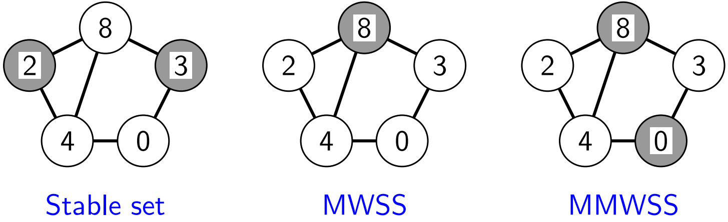
- In 1960, Berge introduced perfect graphs as
  - G perfect iff $\forall$induced subgraphs H, the coloring number of H equals the clique number of H.
- Stated *Strong Perfect Graph Conjecture*, open for 50 years
- Many NP-hard problems become polynomial time for perfect graphs [Grötschel Lovász Schrijver 1984]
  - Graph coloring
  - Maximum clique
  - Maximum stable set

Efficient problems on perfect graphs



- Coloring: color nodes with fewest colors such that no adjacent nodes have the same color
- Max Clique: largest set of nodes, all pairwise adjacent
- Max Stable Set: largest set of nodes, none pairwise adjacent

Efficient problems on weighted perfect graphs

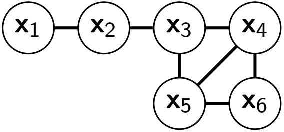


- Stable set: no two vertices adjacent
- Max Weight Stable Set (MWSS): stable set with max weight
- Maximal MWSS (MMWSS): MWSS with max cardinality
  (includes as many 0 weight nodes as possible)

MWSS solvable in polynomial time via linear programming, semidefinite programming or message passing ($\tilde{\mathcal{O}}(n^5)$ and faster).

MWSS via linear programming

$$\max_{\mathbf{x}\in\mathbb{R}^n, \mathbf{x}\geq\mathbf{0}} \mathbf{f}^\top \mathbf{x} \text{ s.t. } \mathbf{A}\mathbf{x} \leq \mathbf{1}$$



$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}$$

- $\mathbf{A} \in \mathbb{R}^{m\times n}$ is vertex versus maximal cliques incidence matrix
- $\mathbf{f} \in \mathbb{R}^n$ is vector of weights
- For perfect graphs, LP is binary and finds MWSS in $\mathcal{O}(\sqrt{m}n^3)$
- Note m is number of cliques in graph (may be exponential)

MWSS via message-passing

Input: $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, cliques $\mathcal{C} = \{\mathbf{c}_1, \ldots, \mathbf{c}_m\}$ and weights f_i for $i \in \mathcal{V}$

Initialize $z_j = \max_{i \in \mathbf{c}_j} \frac{f_i}{\sum_{\mathbf{c} \in \mathcal{C}} [i \in \mathbf{c}]}$ for $j \in \{1, \ldots, m\}$

Until converged do

- Randomly choose $a \neq b \in \{1, \ldots, m\}$
- Compute $h_i = \max\left(0, \left(f_i - \sum_{j: i \in \mathbf{c}_j, j \neq a,b} z_j\right)\right)$ for $i \in \mathbf{c}_a \cup \mathbf{c}_b$
- Compute $s_a = \max_{i \in \mathbf{c}_a \setminus \mathbf{c}_b} h_i$
- Compute $s_b = \max_{i \in \mathbf{c}_b \setminus \mathbf{c}_a} h_i$
- Compute $s_{ab} = \max_{i \in \mathbf{c}_a \cap \mathbf{c}_b} h_i$
- Update $z_a = \max\left[s_a, \frac{1}{2}(s_a - s_b + s_{ab})\right]$
- Update $z_b = \max\left[s_b, \frac{1}{2}(s_b - s_a + s_{ab})\right]$

Output: $\mathbf{z}^* = [z_1, \ldots, z_m]^\top$

MWSS via semi-definite programming

$$\vartheta = \max_{\mathbf{M}\succeq \mathbf{0}} \sum_{ij} \sqrt{\mathbf{f}_i \mathbf{f}_j} \mathbf{M}_{ij} \text{ s.t. } \sum_i \mathbf{M}_{ii} = 1, \; \mathbf{M}_{ij} = 0 \, \forall (i,j) \in E$$

- This is known as the Lovász theta-function
- Let $\mathbf{M} \in \mathbb{R}^{n \times n}$ be the maximizer of $\vartheta_{\mathcal{F}}(\mathcal{G})$
- Let ϑ be the recovered total weight of the MWSS.
- Under mild assumptions, get $\mathbf{x}^* = \mathrm{round}(\vartheta \mathbf{M1})$
- For perfect graphs, find MWSS in $\tilde{\mathcal{O}}(n^5)$

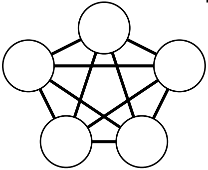
Perfect graph theory

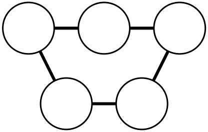
Theorem (Strong Perfect Graph Theorem, Chudnovsky et al 2006)

G perfect $\Leftrightarrow$ G contains no odd hole or antihole

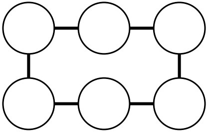
- Hole: an induced subgraph which is a (chordless) cycle of length at least 4. An odd hole has odd cycle length.
- Antihole: the complement of a hole



Perfect



Not Perfect

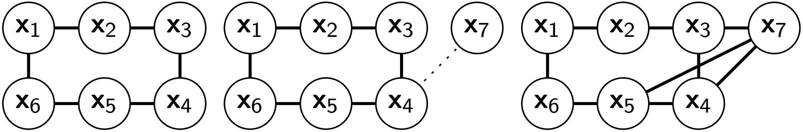


Perfect

Other perfect graph theorems

Lemma (Replication, Lovász 1972)

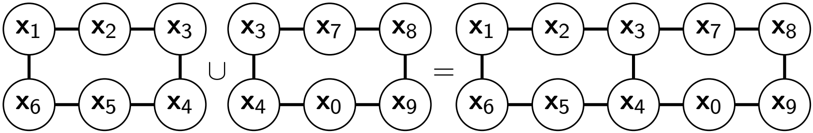
Let $\mathcal{G}$ be a perfect graph and let $v \in V(\mathcal{G})$. Define a graph $\mathcal{G}'$ by adding a new vertex v' and joining it to v and all the neighbors of v. Then $\mathcal{G}'$ is perfect.



Other perfect graph theorems

Lemma (Pasting on a Clique, Gallai 1962)

Let $\mathcal{G}$ be a perfect graph and let $\mathcal{G}'$ be a perfect graph. If $\mathcal{G} \cap \mathcal{G}'$ is a clique (clique cutset), then $\mathcal{G} \cup \mathcal{G}'$ is a perfect graph.



Our plan: reduce NP-hard inference to MWSS

- Reduce MAP to MWSS on weighted graph
- If reduction produces a perfect graph, inference is efficient
- Proves efficiency of MAP on
  - Acyclic models
  - Bipartite matching models
  - Attractive models
  - Slightly frustrated models (new)
- Reduce Bethe marginal inference to MWSS on weighted graph
- Proves efficiency of Bethe marginals on
  - Acyclic models
  - Attractive models (new)
  - Frustrated models (new)

Reduction: graphical model $M \to$ NMRF N

Given an graphical model M, construct a *nand Markov random field (NMRF) N*:

- Weighted graph $N(V_N, E_N, w)$ with vertices V_N, edges E_N and weight function $w : V_N \to \mathbb{R}_{\geq 0}$
- Each $c \in C$ from M maps to a *clique group* of N with one node for each configuration x_c, all pairwise adjacent
- Nodes are adjacent iff inconsistent settings for any variable X_i
- Weights of each node in N set as $\psi_c(x_c) - \min_{x_c} \psi_c(x_c)$

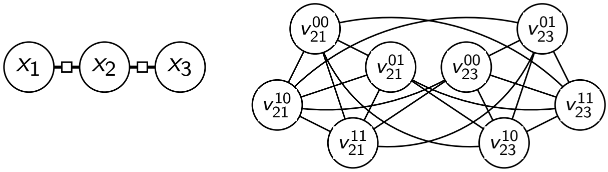
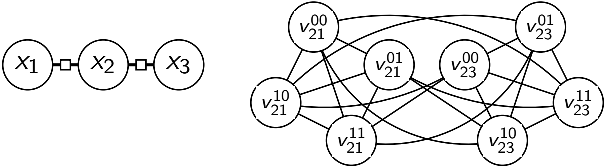


Figure: MRF M with binary variables (left) and NMRF N (right).

Reduction: graphical model $M \longrightarrow$ NMRF N



MAP inference: identify $x^* = \arg\max_x \sum_{c \in C} \psi_c(x_c)$

Lemma (J 2009)

A MMWSS of the NMRF finds a MAP solution

Proof.

Sketch: *MAP selects, for each ψ_c, one configuration of x_c which must be globally consistent with all other choices, so as to max the total weight. This is exactly what MMWSS does.* □

Reparameterization and pruning

Lemma (WJ 2013)

To find a MMWSS, it is sufficient to prune any 0 weight nodes, solve MWSS on the remaining graph, then greedily reintroduce 0 weight nodes while maintaining stability.

A *reparameterization* is a transformation

$$\{\psi_c\} \rightarrow \{\psi'_c\} \text{ s.t. } \forall x, \sum_{c \in C} \psi_c(x_c) = \sum_{c \in C} \psi'_c(x_c) + \text{constant}.$$

Does not change the MAP solution but can simplify the NMRF

Lemma (WJ 2013)

MAP inference is tractable provided $\exists$ an efficient reparameterization s.t. we obtain a perfect pruned *NMRF*

Reparameterization and pruning

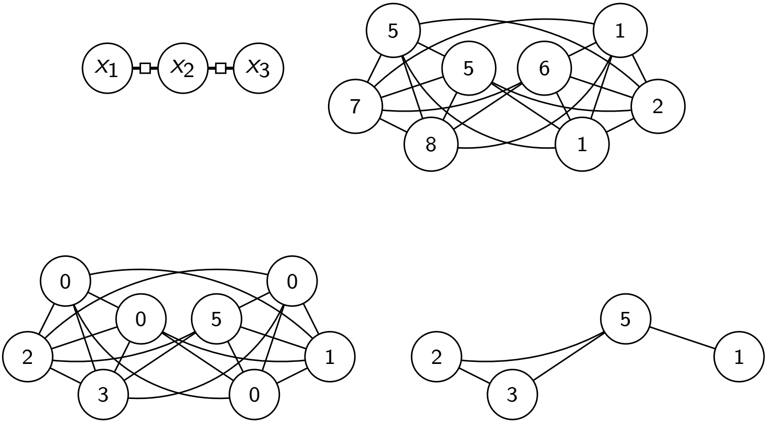
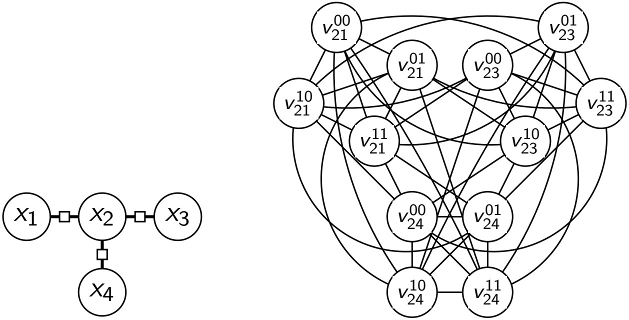


Figure: Graphical model's ψ values and final weights in pruned NMRF

NMRF for tree models is perfect



(a) Graphical model (b) NMRF

Figure: Reducing a tree model

NMRF for tree models is perfect

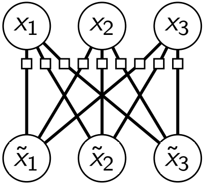
Theorem (J 2009)

Let G be a tree, the NMRF $\mathcal{G}$ obtained from G is a perfect graph.

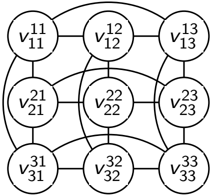
Proof.

First prove perfection for a star graph with internal node v with $|v|$ configurations. First obtain $\mathcal{G}$ for the star graph by only creating one configuration for non internal nodes. The resulting graph is a complete $|v|$-partite graph which is perfect. Introduce additional configurations for non-internal nodes one at a time using the replication lemma. The resulting $\mathcal{G}_{star}$ is perfect. Obtain a tree by induction. Add two stars $\mathcal{G}_{star}$ and $\mathcal{G}_{star'}$. The intersection is a fully connected clique (clique cutset) so by [Gallai 1962], the resulting graph is perfect. Continue gluing stars to form full tree G. □

NMRF for matching models is perfect



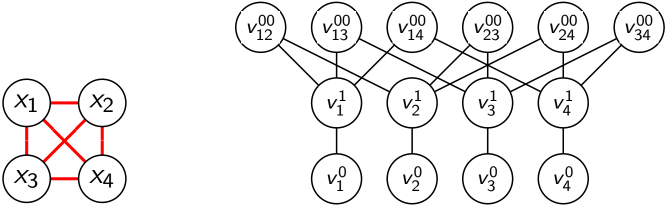
(a) Graphical model



(b) pruned NMRF

Figure: Reducing a matching model

NMRF for attractive models is perfect



(a) Graphical model (b) pruned NMRF

Figure: Reducing an attractive binary pairwise model

- Attractive edges (solid red) have potential functions which satisfy $\psi_c(0,0) + \psi_c(1,1) \geq \psi_c(0,1) + \psi_c(1,0)$
- In fact, since this makes a bipartite graph, we can use an ultra-fast max-flow linear programming solver for MWSS

NMRF for attractive models is perfect

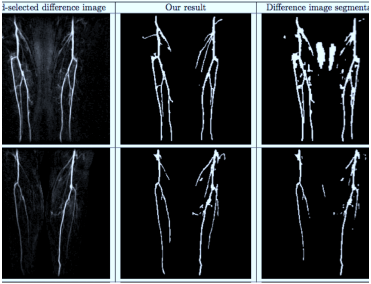


Image segmentation via Kolomogorov's Graph-Cuts code

$$p(x) = \frac{1}{Z} \prod_{ij \in E(G)} \exp(\psi(x_i, x_j)) \prod_{i \in V(G)} \exp(\psi_i(x_i))$$

Here, all $\psi(x_i, x_j) = [\alpha \;\; \beta; \beta \;\; \alpha]$ where $\alpha > \beta$

Each $\psi_i(x_i) = [(1 - z_i) \;\; (z_i)]$ where z_i is the grayscale of pixel i

Signed graphical models

- More generally, a binary model can have edges with either attractive or repulsive signs

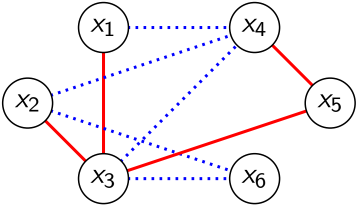


Figure: A signed graph, solid (dashed) edges are attractive (repulsive)

- Attractive edges (red) $\psi(0,0) + \psi(1,1) \geq \psi(0,1) + \psi(1,0)$
- Repulsive edges (blue) $\psi(0,0) + \psi(1,1) \leq \psi(0,1) + \psi(1,0)$

Which signed models give perfect NMRFs?

Definition

A *frustrated* cycle contains an *odd* number of repulsive edges.

Consider the cycles in the graphical model:

- Non-frustrated cycle: what we call a B_R structure, no odd holes
- Frustrated cycle with > 3 edges: creates odd holes
- Frustrated cycle with exactly 3 edges
  - 1 repulsive edge: to avoid odd holes must have U_n structure
  - 3 repulsive edges: to avoid odd holes must have $T_{m,n}$ structure

Theorem (WJ 2013)

A graphical model maps to a perfect pruned NMRF for all valid ψ_c *iff it decomposes into blocks of the form* B_R*,* $T_{m,n}$ *or* U_n*.*

Example of a B_R structure

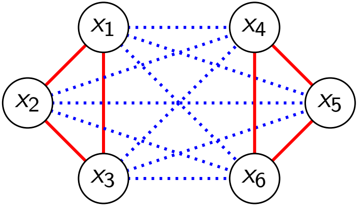


Figure: A B_R structure is 2-connected and contains no frustrated cycle. Solid (dashed) edges are attractive (repulsive). Deleting any edges maintains the B_R property

Examples of $T_{m,n}$ and U_n structures

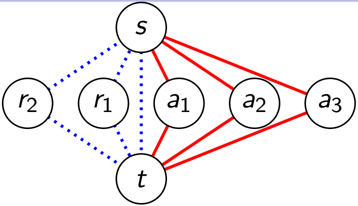


Figure: A $T_{m,n}$ structure with $m = 2$ and $n = 3$. Note triangle with 3 repulsive edges. Solid (dashed) edges are attractive (repulsive).

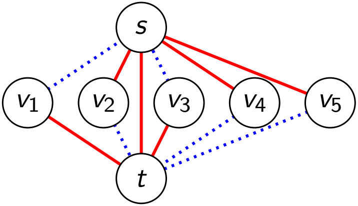


Figure: A U_n structure with $n = 5$. Note triangle with 1 repulsive edge. Solid (dashed) edges are attractive (repulsive).

NMRF for slightly frustrated models is perfect

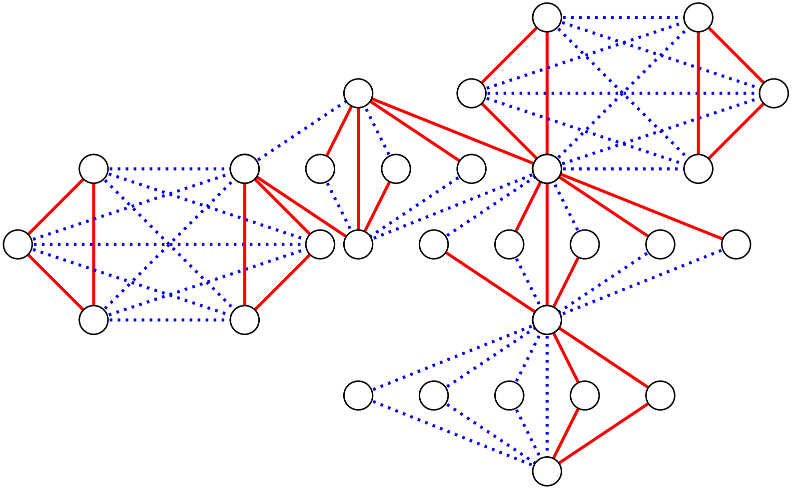


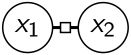
Figure: Binary pairwise graphical model, provably tractable with perfect pruned NMRF due to decomposition into B_r, $T_{m,n}$ and U_n structures.

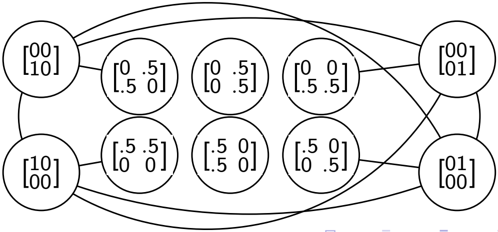
Our plan: reduce NP-hard inference to MWSS

- Reduce MAP to MWSS on weighted graph
- If reduction produces a perfect graph, inference is efficient
- Proves efficiency of MAP on
  - Acyclic models
  - Bipartite matching models
  - Attractive models
  - Slightly frustrated models (new)
- Reduce Bethe marginal inference to MWSS on weighted graph
- Proves efficiency of Bethe marginals on
  - Acyclic models
  - Attractive models (new)
  - Frustrated models (new)

Reduce marginal inference $p(x_i) = \sum_{X \setminus x_i} p(X)$ to MWSS

- Our plan to solve marginal inference
  - 1) reduce summation to a continuous minimization problem
  - 2) discretize the continuous minimization on a mesh
  - 3) find the optimal discrete solution using MWSS
- Loosely speaking, given graphical model M, construct *nand Markov random field* N where each node is a setting of a marginal. Rather than 1 node per configuration of ψ_c, enumerate all possible marginals on ψ_c that are within ϵ away from each other. Then connect pairwise inconsistent nodes.





Reduce marginal inference $p(x_i) = \sum_{X \backslash x_i} p(X)$ to MWSS

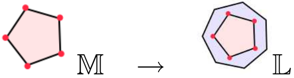
- Marginal inference involves large summation problems like

$$p(x_i) = \sum_{X \backslash x_i} \frac{1}{Z} \exp\left(\sum_{c \in C} \psi_c(x_c) \right)$$

- Finding $p(x_i)$ is equivalent to computing partition function Z
- Minimize the Gibbs free energy over all possible distributions q

$$\log Z = -\min_{q \in \mathbb{M}} \mathcal{F}_G = \max_{q \in \mathbb{M}} \mathbb{E}_q \sum_{c \in C} \psi_c(x_c) + S(q(x))$$

Approximating marginals with the Bethe free energy



Bethe (1935) gave alternative to minimizing Gibbs free energy by finding the partition function as the minimum of Bethe free energy[1] over local polytope $\mathbb{L}$ rather than marginal polytope $\mathbb{M}$

$$
\begin{aligned}
\log Z &= -\min_{q\in\mathbb{M}} \mathcal{F}_G = \max_{q\in\mathbb{M}} \mathbb{E}_q \sum_{c\in C} \psi_c(x_c) + S(q(x)) \\
&\approx \log Z_B = -\min_{q\in\mathbb{L}} \mathcal{F} = \max_{q\in\mathbb{L}} \mathbb{E}_q \sum_{c\in C} \psi_c(x_c) + S_B(q(x))
\end{aligned}
$$

In many cases, the Bethe partition function Z_B bounds the true Z.

[1] The Bethe entropy is $S_B = \sum_{(i,j)\in\mathcal{E}} S_{ij} + \sum_{i\in\mathcal{V}}(1-d_i)S_i$.

Approximating marginals with the Bethe free energy

- Remarkable result: [YFW01] showed that any fixed point of loopy belief propagation (LBP) corresponds to a stationary point of the Bethe free energy $\mathcal{F}$
- But LBP can fail to converge or may converge to bad stationary points
- No previous method could find the global Bethe solution
- We will derive the first polynomial time approximation scheme (PTAS) that finds the global optimum of the Bethe free energy $\mathcal{F}$ to within ϵ accuracy [WJ 2013, WJ 2014] for attractive models
- The PTAS recovers the Bethe partition Z_B and the corresponding optimal marginal probabilities $q(x)$

Approximating marginals with the Bethe free energy

We will recover the distribution $q(x)$ that minimizes $\mathcal{F}$.
It is defined by the following

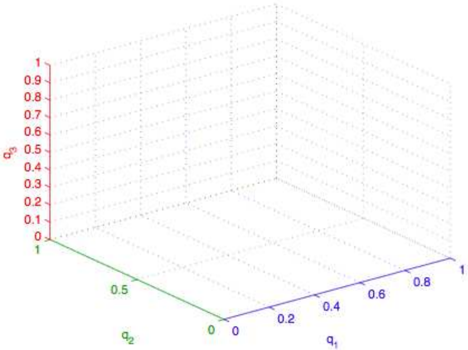
- Singleton marginals q_i for all vertices $i \in \mathcal{V}(G)$
- Pairwise marginals μ_{ij} for all edges $(i,j) \in \mathcal{E}(G)$

$$
\begin{aligned}
q_i &= p(X_i = 1) \\
\mu_{ij} &= \begin{bmatrix} p(X_i = 0, X_j = 0) & p(X_i = 0, X_j = 1) \\ p(X_i = 1, X_j = 0) & p(X_i = 1, X_j = 1) \end{bmatrix} \\
&= \begin{bmatrix} 1 + \xi_{ij} - q_i - q_j & q_j - \xi_{ij} \\ q_i - \xi_{ij} & \xi_{ij} \end{bmatrix}
\end{aligned}
$$

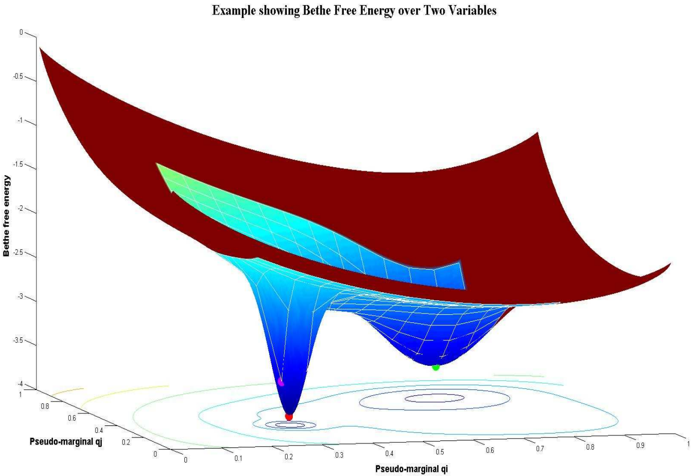
Fortunately minimizing $\mathcal{F}$ over ξ_{ij} is analytic via [WT01]
Only numerical optimization over $(q_1, \ldots, q_n) \in [0,1]^n$ remains

A mesh over Bethe pseudo-marginals

We discretize the space $(q_1, \ldots, q_n) \in [0,1]^n$ with a mesh $\mathcal{M}(\epsilon)$ that is sufficiently fine that the discrete solution $\hat{q}$ we obtain has $\mathcal{F}(\hat{q}) \leq \min_q \mathcal{F}(q) + \epsilon$



A mesh over Bethe pseudo-marginals



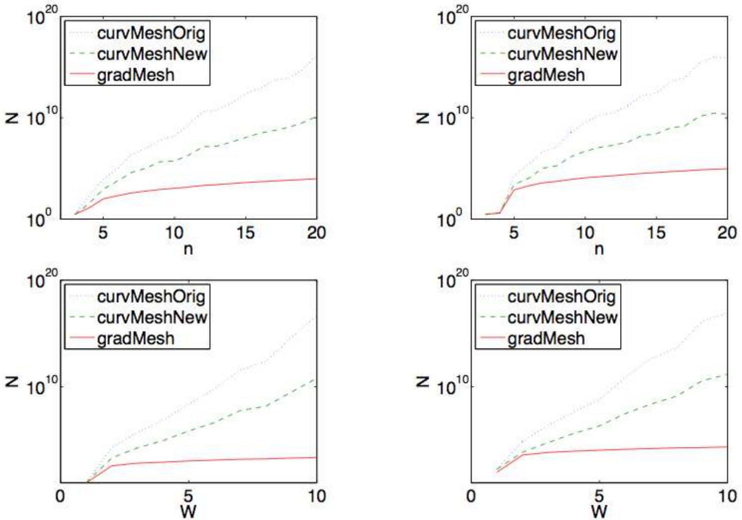
A mesh over Bethe pseudo-marginals

Given a model with n vertices, m edges, and max edge weight W

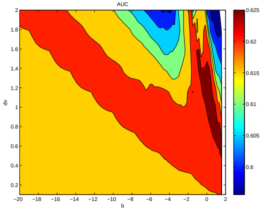
- If original model is attractive (submodular costs), then the discretized minimization problem is a perfect graph MWSS
- Solve via graph cuts [SF06] in $\mathcal{O}((\sum_{i \in \mathcal{V}} N_i)^3)$ where N_i is the number of discretized values in dimension i
- Two ways to make the mesh $\mathcal{M}(\epsilon)$ sufficiently fine:
  - Bounding curvature of $\mathcal{F}$ [WJ13] achieves slow polynomial
  - Bounding gradients of $\mathcal{F}$ [WJ14] achieves $O(\frac{n^3 m^3 W^3}{\epsilon^3})$
- Both algorithms find ϵ-close global solution for Z_B

Bethe pseudo-marginals

Left figures $\epsilon = 1$, right $\epsilon = 0.1$, when fixed $W = 5$, $n = 10$

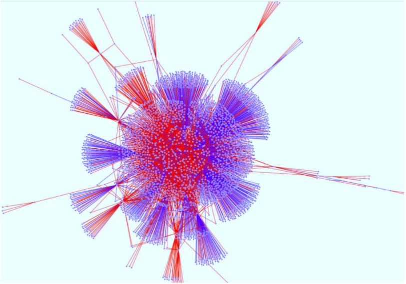


Marginal inference for attractive ranking



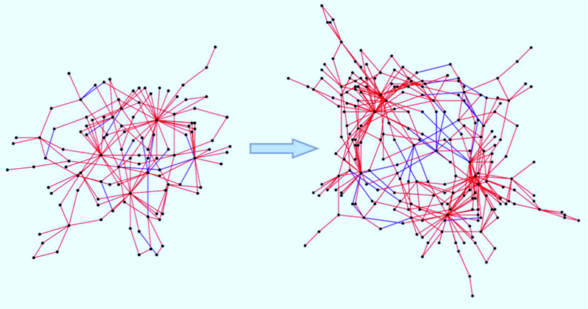
- Electric transformers network $x_1, \ldots, x_n$ where $x_i \in \{\mathit{fail}, \mathit{stable}\}$
- Rank transformers by marginal probability of failure $p(x_k)$ via $p(x_1, \ldots, x_n) = \frac{1}{Z} \exp\left(\sum_{ij \in E} \psi_{ij}(x_i, x_j) + \sum_{k=1}^{n} \psi(x_k)\right)$
- Each has known probability $\exp \psi(x_k)$ of failing in isolation
- Attractive edges between transformers couple their failures $\psi(x_i, x_j) = [\alpha \;\; \beta; \beta \;\; \gamma]$ with $\alpha + \gamma \geq 2\beta$
- PTAS improves AUC to 0.625 from independent ranking 0.59

Marginal inference for frustrated ranking



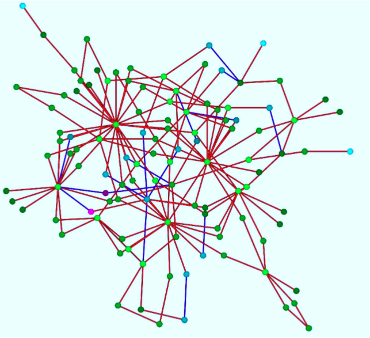
- Epinions users network $x_1, \ldots, x_n$ where $x_i \in \{suspect, trusted\}$
- Rank users trustworthiness using marginal $p(x_k)$ from $p(x_1, \ldots, x_n) = \frac{1}{Z} \exp\left(\sum_{e \in E} \psi_e(x_i, x_j)\right)$
- Attractive edges (red) $\psi(0,0) + \psi(1,1) \geq \psi(0,1) + \psi(1,0)$
- Repulsive edges (blue) $\psi(0,0) + \psi(1,1) \leq \psi(0,1) + \psi(1,0)$
- Can we use the PTAS on this frustrated graphical model??

Marginal inference for frustrated ranking



Given frustrated graph G, we form attractive *double-cover* $\mathcal{G}$:
FOR each $i \in V(G)$, create two copies denoted i_1 and i_2 in $V(\mathcal{G})$
FOR each edge $(i, j) \in E(G)$
 IF ψ_{ij} is log-supermodular: add edges (i_1, j_1) and (i_2, j_2) to $E(\mathcal{G})$
 ELSE: add edges (i_1, j_2) and (i_2, j_1) to $E(\mathcal{G})$
Flip nodes on one side of the double-cover

Marginal inference for frustrated ranking



We prove that our PTAS on this gives $\hat{Z}_B \geq Z_B$

Nodes shaded with $p(x_i = 1)$ to reflect trustworthiness

Loopy belief propagation is convergent on double-cover

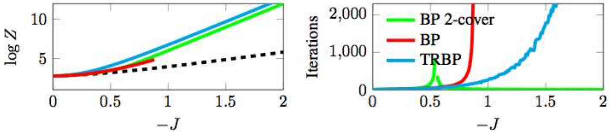


Figure 3: Plots of the log partition function and the number of iterations for the different algorithms to converge for a complete graph on four nodes with no external field as the strength of the negative edges goes from 0 to -2. For TRBP, $\rho_{ij} = .5$ for all $(i, j) \in E$. The dashed black line is the ground truth.

	a	BP	TRBP	BP 2-cover	BP Iter.	TRBP Iter	BP 2-cover Iter.
Grid	1	**100%**	**100%**	95%	**44.62**	110.41	222.99
	2	15%	30%	**100%**	210	815.3	**44.14**
	4	1%	0%	**100%**	219	-	**29.59**
EPIN1	1	47%	0%	**100%**	63.53	-	**21.12**
	2	37%	0%	**100%**	90.1	-	**16.19**
	4	38%	0%	**100%**	93.63	-	**15.9**
EPIN1	1	41%	0%	**100%**	51.8	-	**15.12**
	2	50%	0%	**99%**	42.46	-	**14.84**
	4	53%	0%	**100%**	86.66	-	**14.93**
Deep Networks	1	61%	0%	**100%**	89.2	-	**16.67**
	2	61%	0%	**100%**	30.66	-	**16.82**
	4	60%	0%	**100%**	24.88	-	**18.17**

Figure 4: Percent of samples on which each algorithm converged within 1000 iterations and the average number of iterations for convergence for 100 samples of edges weights in $[-a, a]$ for the designated graphs. For TRBP, performance was poor independent of the spanning trees selected.

Conclusions

- Goal: perform inference on large networks
- Approach: set up tasks as finding maxima and marginals of probability distribution $p(x_1, \dots, x_n)$
- Limitation: for big $p(x_1, \dots, x_n)$ these are intractable
- Methodology: graphical modeling and efficient solvers
- Verification: perfect graph theory and bounds
- Efficient MAP on
  - Bipartite matching models
  - Attractive models
  - Slightly frustrated models (new)
- Efficient Bethe marginals on
  - Attractive models (new)
  - Frustrated models (new)

References

- B. Huang and T. Jebara. "Loopy Belief Propagation for Bipartite Maximum Weight b-Matching". AISTATs, 2007.
- T. Jebara. "MAP Estimation, Message Passing, and Perfect Graphs". UAI, 2009.
- T. Jebara. "Perfect Graphs and Graphical Modeling". Tractability: Practical Approaches to Hard Problems, 2013.
- A. Weller and T. Jebara. "Bethe Bounds and Approximating the Global Optimum". AISTATs, 2013.
- A. Weller and T. Jebara. "On MAP Inference by MWSS on Perfect Graphs". UAI, 2013.
- K. Tang, A. Weller and T. Jebara. "Network ranking with Bethe pseudomarginals", NIPS Workshop 2013.
- A. Weller and T. Jebara. "Approximating the Bethe Partition Function". UAI, 2014.
- N. Ruozzi and T. Jebara. "Making Pairwise Binary Graphical Models Attractive". NIPS, 2014.