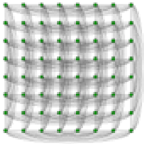
Graphical Modeling and Inference with Perfect Graphs

Tony Jebara, Columbia University

April 7, 2010

Background on Perfect Graphs

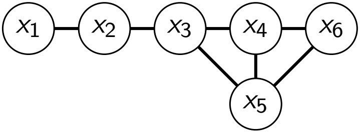
- In 1960, Berge introduces perfect graphs where every induced subgraph has clique# = coloring#
- Berge also poses two conjectures:
  - Weak: a graph is perfect iff its complement is perfect
  - Strong: a graph is perfect iff it is Berge
- Weak perfect graph theorem (Lovász 1972)
- Link between perfection and integral LPs (Lovász 1972)
- Strong perfect graph theorem (SPGT) open for 4+ decades

Background on Perfect Graphs

- SPGT Proof (Chudnovsky, Robertson, Seymour, Thomas 2003)
- Berge passes away shortly after hearing of the proof
- Many NP-hard and hard to approximate problems are P for perfect graphs
  - Graph coloring
  - Maximum clique
  - Maximum independent set
- Recognizing perfect graphs is $O(n^9)$ (Chudnovsky *et al.* 2006)

Graphical Models



- Perfect graphs for maximum a posteriori (MAP) (J 2009)
- Graphical model: an undirected graph G representing a distribution $p(X)$ where $X = \{x_1, \ldots, x_n\}$ and $x_i \in \mathbb{Z}$
- Distribution factorizes as product of cliques or functions over subsets of variables

$$p(x_1, \ldots, x_n) \quad = \quad \frac{1}{Z} \prod_{c \in C} \psi_c(X_c)$$

- E.g. $p(x_1, \ldots, x_6) = \psi(x_1, x_2)\psi(x_2, x_3)\psi(x_3, x_4, x_5)\psi(x_4, x_5, x_6)$

MAP Estimation

- A canonical problem, find most probable configuration

$$X^* = \text{argmax}\, p(x_1, \ldots, x_n)$$

- Useful for image processing, protein folding, coding, etc.
- Brute force requires $\prod_{i=1}^{n} |x_i|$
- Efficient for trees and singly linked graphs (Pearl 1988)
- NP-hard for general graphs (Shimony 1994)
- Approach A: relaxations and variational methods
  - First order LP relaxations (Wainwright *et al.* 2002)
  - TRW max-product (Kolmogorov & Wainwright 2006)
  - Higher order LP relaxations (Sontag *et al.* 2008)
  - Fractional and integral LP rounding (Ravikumar *et al.* 2008)
  - Open problem: when are LPs tight?
- Approach B: max product and message passing

Max Product Message Passing

1. For each x_i to each X_c: $m_{i \to c}^{t+1} = \prod_{d \in \mathrm{Ne}(i) \setminus c} m_{d \to i}^{t}$
2. For each X_c to each x_i: $m_{c \to i}^{t+1} = \max_{X_c \setminus x_i} \psi_c(X_c) \prod_{j \in c \setminus i} m_{j \to c}^{t}$
3. Set $t = t + 1$ and goto 1 until convergence
4. Output $x_i^* = \mathrm{argmax}_{x_i} \prod_{d \in \mathrm{Ne}(i)} m_{d \to i}^{t}$

- A simple and fast algorithm that performs well in practice
- Exact for trees (Pearl 1988)
- Converges for single-loop graphs (Weiss & Freeman 2001)
- Convergence and Gibbs measure (Tatikonda & Jordan 2002)
- Local optimality guarantees (Wainwright *et al.* 2003)
- Similar to first order LP relaxation
- Recent progress
  - Exact for bipartite matchings (Bayati *et al.* 2005)
  - Exact for bipartite *b*-matchings (Huang and J 2007)

Bipartite Matching

	Motorola	Apple	IBM
"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 $\sum_i C_{ij} = \sum_j C_{ij} = 1$
- Classical Hungarian marriage problem $O(n^3)$
- Creates a very loopy graphical model
- Max product takes $O(n^3)$ for matching (Bayati *et al.* 2005)

Bipartite Generalized Matching

	Motorola	Apple	IBM
"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 $\sum_i C_{ij} = \sum_j C_{ij} = b$
- Combinatorial b-matching problem $O(bn^3)$, (Google Adwords)
- Creates a very loopy graphical model
- Max product takes $O(bn^3)$ for exact MAP (Huang & J 2007)

Unipartite Generalized Matching

	p_1	p_2	p_3	p_4
p_1	0	2	1	2
p_2	2	0	2	1
p_3	1	2	0	2
p_4	2	1	2	0

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

- $\max_{C \in \mathbb{B}^{n \times n}, C_{ii}=0} \sum_{ij} W_{ij} C_{ij}$ such that $\sum_i C_{ij} = b, C_{ij} = C_{ji}$
- Combinatorial unipartite matching is efficient (Edmonds 1965)
- Makes an LP with exponentially many blossom inequalities
- Max product exact if blossomless LP integral (Sanghavi 2008)
- Same b-matching code as bipartite case (Huang and J 2007)

Back to Perfect Graphs

- Max product and exact MAP depend on the LP's integrality
- Matchings have special integral LPs (Edmonds 1965)
- How to generalize beyond matchings?
- Perfect graphs imply LP integrality (Lovász 1972)

Lemma (Lovász 1972)

For every non-negative vector $\vec{w} \in \mathbb{R}^N$, the linear program

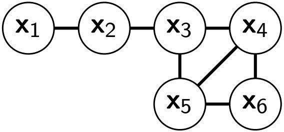
$$\beta = \max_{\vec{x} \in \mathbb{R}^N} \vec{w}^\top \vec{x} \text{ subject to } \vec{x} \geq 0 \text{ and } A\vec{x} \leq \vec{1}$$

recovers a vector $\vec{x}$ which is integral if and only if the (undominated) rows of A form the vertex versus maximal cliques incidence matrix of some perfect graph.

Back to Perfect Graphs

Lemma (Lovász 1972)

$$\beta = \max_{\vec{x} \in \mathbb{R}^N} \vec{w}^\top \vec{x} \text{ subject to } \vec{x} \geq 0 \text{ and } A\vec{x} \leq \vec{1}$$



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

nand Markov Random Fields

- Lovász's lemma is not solving $\max p(X)$ on G
- How to apply the lemma to any model G and space X?
- We have $p(x_1, \ldots, x_n) = \frac{1}{Z} \prod_{c \in C} \psi_c(X_c)$
- Without loss of generality assume $\psi_c(X_c) \leftarrow \frac{\psi_c(X_c)}{\min_{X_c} \psi_c(X_c)} + \epsilon$
- Consider procedure to G to $\mathcal{G}$ in NMRF form
- NMRF is a nand Markov random field over space $\mathbf{X}$
  - all variables are binary $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$
  - all potential functions are pairwise nand gates $\Phi(\mathbf{x}_i, \mathbf{x}_j) = \delta[\mathbf{x}_i + \mathbf{x}_j \leq 1]$

nand Markov Random Fields

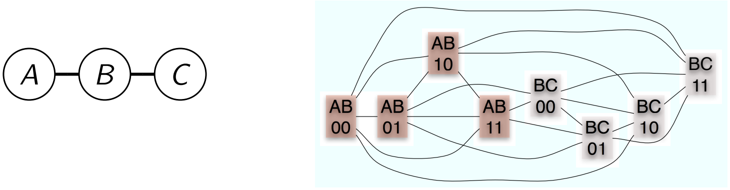
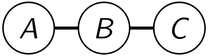


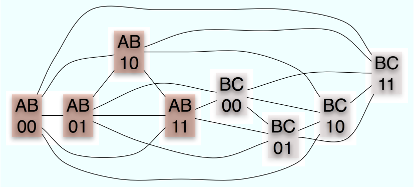
Figure: Binary graphical model G (left) and nand MRF $\mathcal{G}$ (right).

Initialize $\mathcal{G}$ as the empty graph
For each clique c in graph G do
    For each configuration $k \in X_c$ do
        add a corresponding binary node $\mathbf{x}_{c,k}$ to $\mathcal{G}$
        for each $\mathbf{x}_{d,l} \in \mathcal{G}$ which is incompatible with $\mathbf{x}_{c,k}$
            connect $\mathbf{x}_{c,k}$ and $\mathbf{x}_{d,l}$ with an edge

Figure: Algorithm to convert G into a NMRF $\mathcal{G}$

nand Markov Random Fields





$p(X) \propto \prod_{c \in C} \psi_c(X_c)$ $\rho(\mathbf{X}) \propto \prod_{i \in \mathsf{V}(\mathcal{G})} e^{\mathbf{w}_i \mathbf{x}_i} \prod_{(i,j) \in \mathsf{E}(\mathcal{G})} (1 - \mathbf{x}_i \mathbf{x}_j)$

- $\mathcal{G}$ has N binary variables with weights $\mathbf{w}_{c,k} = \log \psi_c(X_c = k)$
- If node $\mathbf{x}_{c,k} = 1$ then clique c is in configuration $k \in X_c$

Lemma (J 2009)

The MAP estimate for $\rho(\mathbf{X})$ on $\mathcal{G}$ recovers MAP for $p(X)$ on G

Packing Linear Programs

- MAP on $\rho(\mathbf{X})$ is just Maximum Weight Stable Set (MWSS)
- MWSS is NP-hard in general but P if graph $\mathcal{G}$ is perfect
- Relaxed MAP on $\log \rho(\mathbf{X}) \equiv$ set packing linear program
- If graph $\mathcal{G}$ is perfect, LP is integral

Lemma (Lovász 1972)

For every non-negative vector $\vec{w} \in \mathbb{R}^N$, the linear program

$$\beta = \max_{\vec{x} \in \mathbb{R}^N} \vec{w}^\top \vec{x} \text{ subject to } \vec{x} \geq 0 \text{ and } A\vec{x} \leq \vec{1}$$

recovers a vector $\vec{x}$ which is integral if and only if the (undominated) rows of A form the vertex versus maximal cliques incidence matrix of some perfect graph.

Packing Linear Programs

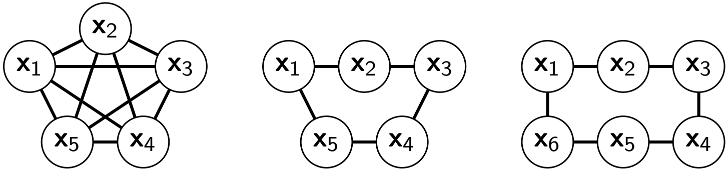
- For general graph G, MAP is NP-hard (Shimony 1994) but...
- Convert G to $\mathcal{G}$ (polynomial time)
- If graph $\mathcal{G}$ is perfect (polynomial time)
  - Solve MAP via MWSS (polynomial time)
    - ...by finding cliques C and solving LP in $O(\sqrt{|C|}N^3)$
    - ...by Lovász theta function semidefinite program in $O(N^5)$

Theorem (J 2009)

MAP estimation of any graphical model G with cliques $c \in C$ over variables $\{x_1, \ldots, x_n\}$ producing a nand Markov random with a perfect graph $\mathcal{G}$ is in P and requires no more than $O\left(\left(\sum_{c\in C}\left(\prod_{i\in c}|x_i|\right)\right)^5\right)$.

Perfect Graphs

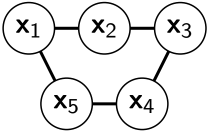
- To determine if $\mathcal{G}$ is perfect
  - Run algorithm on $\mathcal{G}$ in $O(N^9)$ (Chudnovsky *et al.* 2005)
  - or use tools from perfect graph theory to prove perfection
- Clique number of a graph $\omega(\mathcal{G})$: size of its maximum clique
- Chromatic number of a graph $\chi(\mathcal{G})$: minimum number of colors such that no two adjacent vertices have the same color
- A perfect graph $\mathcal{G}$ is a graph where every induced subgraph $\mathcal{H} \subseteq \mathcal{G}$ has $\omega(\mathcal{H}) = \chi(\mathcal{H})$



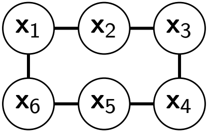
Perfect — Not Perfect — Perfect

Strong Perfect Graph Theorem

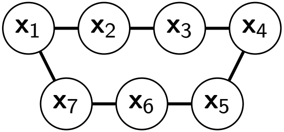
- A graph is perfect iff it is Berge (Chudnovsky *et al.* 2003)
- Berge graph: a graph that contains no odd hole and whose complement also contains no odd hole
- Hole: an induced subgraph of $\mathcal{G}$ which is a chordless cycle of length at least 5. An odd hole has odd cycle length.
- Complement: a graph $\bar{\mathcal{G}}$ with the same vertex set $V(\mathcal{G})$ as $\mathcal{G}$, where distinct vertices $\mathbf{u}, \mathbf{v} \in V(\mathcal{G})$ are adjacent in $\bar{\mathcal{G}}$ just when they are not adjacent in $\mathcal{G}$



odd hole



even hole



odd hole

Recognition using Strong Perfect Graph Theorem

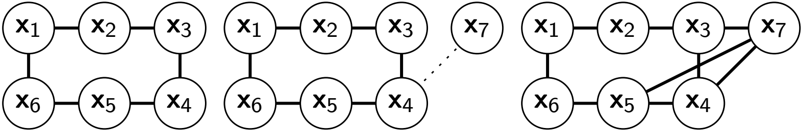
- SPGT implies that a Berge graph is one of these primitives
  - bipartite graphs
  - complements of bipartite graphs
  - line graphs of bipartite graphs
  - complements of line graphs of bipartite graphs
  - double split graphs
- or decomposes structurally (into graph primitives)
  - via a 2-join
  - via a 2-join in the complement
  - via an M-join
  - via a balanced skew partition
- Line graph: $\mathrm{L}(\mathcal{G})$ a graph which contains a vertex for each edge of $\mathcal{G}$ and where two vertices of $\mathrm{L}(\mathcal{G})$ are adjacent iff they correspond to two edges of $\mathcal{G}$ with a common end vertex

Recognition using Strong Perfect Graph Theorem

- SPGT and theory give tools to analyze graph
- Decompose using replication, 2-join, M-joins, skew partition...
- May help diagnose perfection when algorithm is too slow

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.

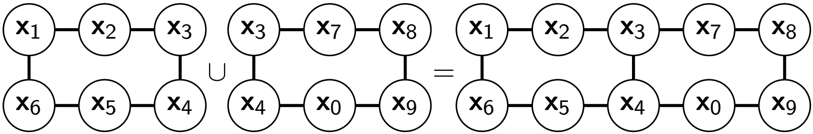


Recognition using Strong Perfect Graph Theorem

- SPGT and theory give tools to analyze graph
- Decompose using replication, 2-join, M-joins, skew partition...
- May help diagnose perfection when algorithm is too slow

Lemma (Gluing on Cliques, Skew Partition, Berge & Chvátal 1984)

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.



Proving Exact MAP for Tree Graphs

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 (Berge & Chvátal 1984), the resulting graph is perfect. Continue gluing stars until full tree G is formed. □

Proving Exact MAP for Bipartite Matchings

Theorem (J 2009)

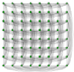
The maximum weight bipartite matching graphical model

$$p(X) = \prod_{i=1}^{n} \delta\left[\sum_{j=1}^{n} x_{ij} \leq 1\right] \delta\left[\sum_{j=1}^{n} x_{ji} \leq 1\right] \prod_{k=1}^{n} e^{f_{ik} x_{ik}}$$

with $f_{ij} \geq 0$ *has integral LP and yields exact MAP estimates.*

Proof.

The graphical model is in NMRF form so G and $\mathcal{G}$ are equivalent. $\mathcal{G}$ is the line graph of a (complete) bipartite graph (Rook's graph). Therefore, $\mathcal{G}$ is perfect, the LP is integral and recovers MAP. □



Proving Exact MAP for Unipartite Matchings

Theorem (J 2009)

The unipartite matching graphical model $G = (V, E)$ with $f_{ij} \geq 0$

$$p(X) = \prod_{i \in V} \delta \left[\sum_{j \in Ne(i)}^{n} x_{ij} \leq 1 \right] \prod_{ij \in E} e^{f_{ij} x_{ij}}$$

has integral LP and produces the exact MAP estimate if G is a perfect graph.

Proof.

The graphical model is in NMRF form and graphs G and $\mathcal{G}$ are equivalent. The set packing LP relaxation is integral and recovers the MAP estimate if $\mathcal{G}$ is a perfect graph. □

Proving Exact MAP for Associative MRFs

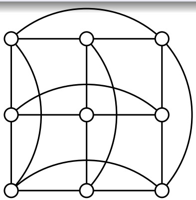
Theorem (Greig, Porteous, and Seheult (1989))

MAP for associate Markov random fields is in P via mincut

$$p(X) \propto \prod_{i=1}^{n}\prod_{j=1}^{n}\psi_{i,j,1}(x_{i,j}, x_{i+1 \bmod n,j})\psi_{i,j,2}(x_{i,j}x_{i,j+1 \bmod n}).$$

Proof via Perfect Graphs.

The resulting NMRF $\mathcal{G}$ from G is bipartite and hence perfect. □



Convergent Message Passing (Globerson & Jaakkola 2009)

- Can perform convergent message passing on $\mathcal{G}$
- If all variables are binary, it recovers same fixed points as LP

Input: NMRF $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and weights w_i for $i \in \mathcal{V}$.

1. Find all cliques C in $\mathcal{E}$.
2. Initialize all messages to any value.
3. For each node $i \in \mathcal{V}$ and clique $c \in C$

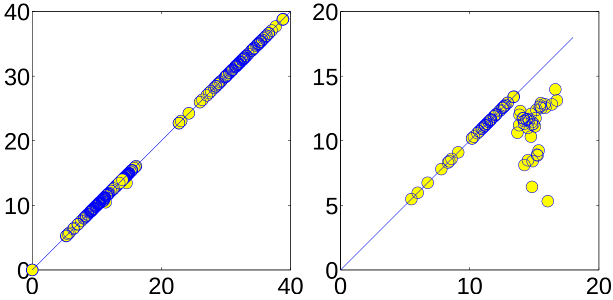
$$\lambda_{i,\mathbf{c}} = \frac{1-|\mathbf{c}|}{|\mathbf{c}|} \sum_{\mathbf{c}' \in C \setminus \mathbf{c} : i \in \mathbf{c}'} \lambda_{i,\mathbf{c}'} + \frac{1}{|\mathbf{c}|} \frac{w_i}{\sum_{\mathbf{c} \in C} [i \in \mathbf{c}]}$$
$$- \frac{1}{|\mathbf{c}|} \max\left[0, \max_{i' \in \mathbf{c} \setminus i} \left[\frac{\theta_{i'}}{\sum_{\mathbf{c} \in C} [i' \in \mathbf{c}]} + \sum_{\mathbf{c}' \in C \setminus \mathbf{c} : i' \in \mathbf{c}'} \lambda_{i',\mathbf{c}'} \right]\right]$$

4. Repeat 3 until convergence.

Theorem (J 2009)

Convergent message passing on perfect NMRFs solves MAP.

MAP Experiments for Unipartite Matching



(a) Perfect Graphs (b) Random Graphs

Figure: Scores for the exact MAP estimate (horizontal axis) and message passing estimate (vertical axis) for random graphs and weights. Figure (a) shows scores for four types of basic Berge graphs while (b) shows scores for arbitrary graphs. Minor score discrepancies on Berge graphs arose due to numerical issues and early stopping.

Conclusions

- Perfect graph theory is fascinating, many recent breakthroughs
- A crucial tool for exploring LP integrality and MAP estimation
- Solve MAP on NMRF $\mathcal{G}$ as Max Weight Stable Set problem
- If graph $\mathcal{G}$ is perfect, MWSS is polynomial
    - ...via Lovász theta function semidefinite program
    - ...via linear programming or message passing
- Exact MAP and message passing applies to
    - Trees and singly-linked graphs
    - Single loop graphs
    - Matchings
    - Generalized matchings
    - *and now* Perfect graphs

Further Reading and Thanks

- MAP Estimation, Message Passing, and Perfect Graphs, T. Jebara. *Uncertainty in Artificial Intelligence*, June 2009.
- Graphical Models, Exponential Families and Variational Inference, M.J. Wainwright and M.I. Jordan. *Foundations and Trends in Machine Learning*, Vol 1, Nos 1-2, 2008.
- Loopy Belief Propagation for Bipartite Maximum Weight b-Matching, B. Huang and T. Jebara. *Artificial Intelligence and Statistics*, March 2007.
- Thanks to Maria Chudnovsky, Delbert Dueck and Bert Huang.