Graphical models and message-passing
Part I: Basics and MAP computation

Martin Wainwright

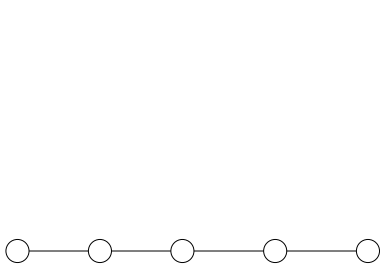
UC Berkeley
Departments of Statistics, and EECS

Tutorial materials (slides, monograph, lecture notes) available at:
www.eecs.berkeley.edu/~wainwrig/kyoto12

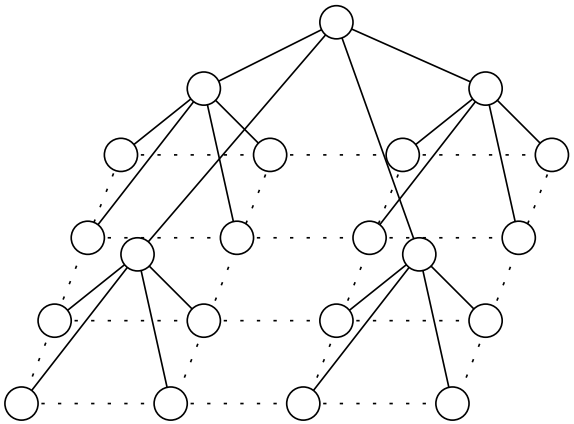
September 2, 2012

Introduction

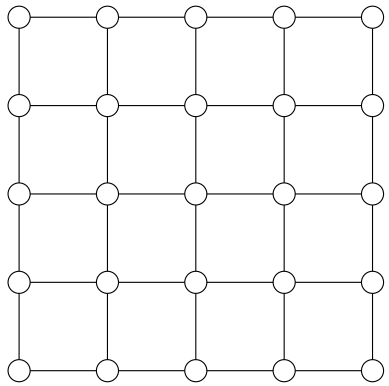
- graphical model:
  - graph $G = (V, E)$ with N vertices
  - random vector: $(X_1, X_2, \ldots, X_N)$



(a) Markov chain



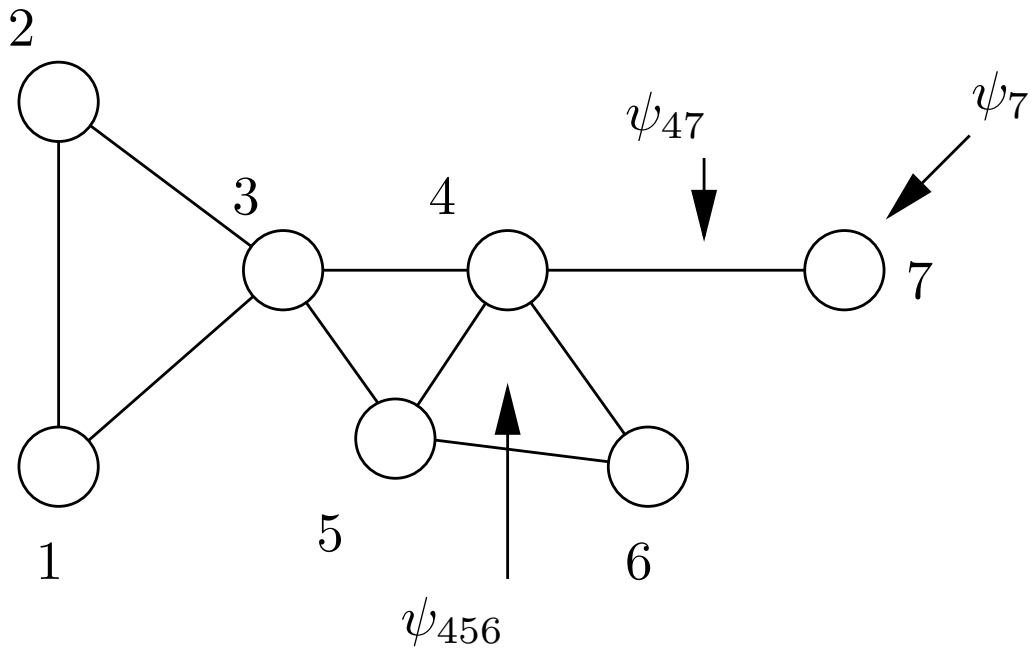
(b) Multiscale quadtree



(c) Two-dimensional grid

- useful in many statistical and computational fields:
  - machine learning, artificial intelligence
  - computational biology, bioinformatics
  - statistical signal/image processing, spatial statistics
  - statistical physics
  - communication and information theory

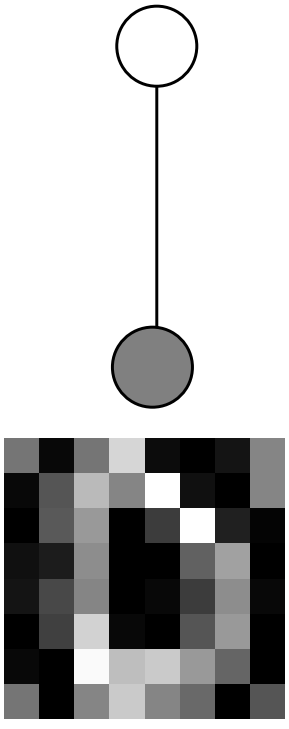
Graphs and factorization



- clique C is a fully connected subset of vertices
- compatibility function ψ_C defined on variables $x_C = \{x_s, s \in C\}$
- factorization over all cliques

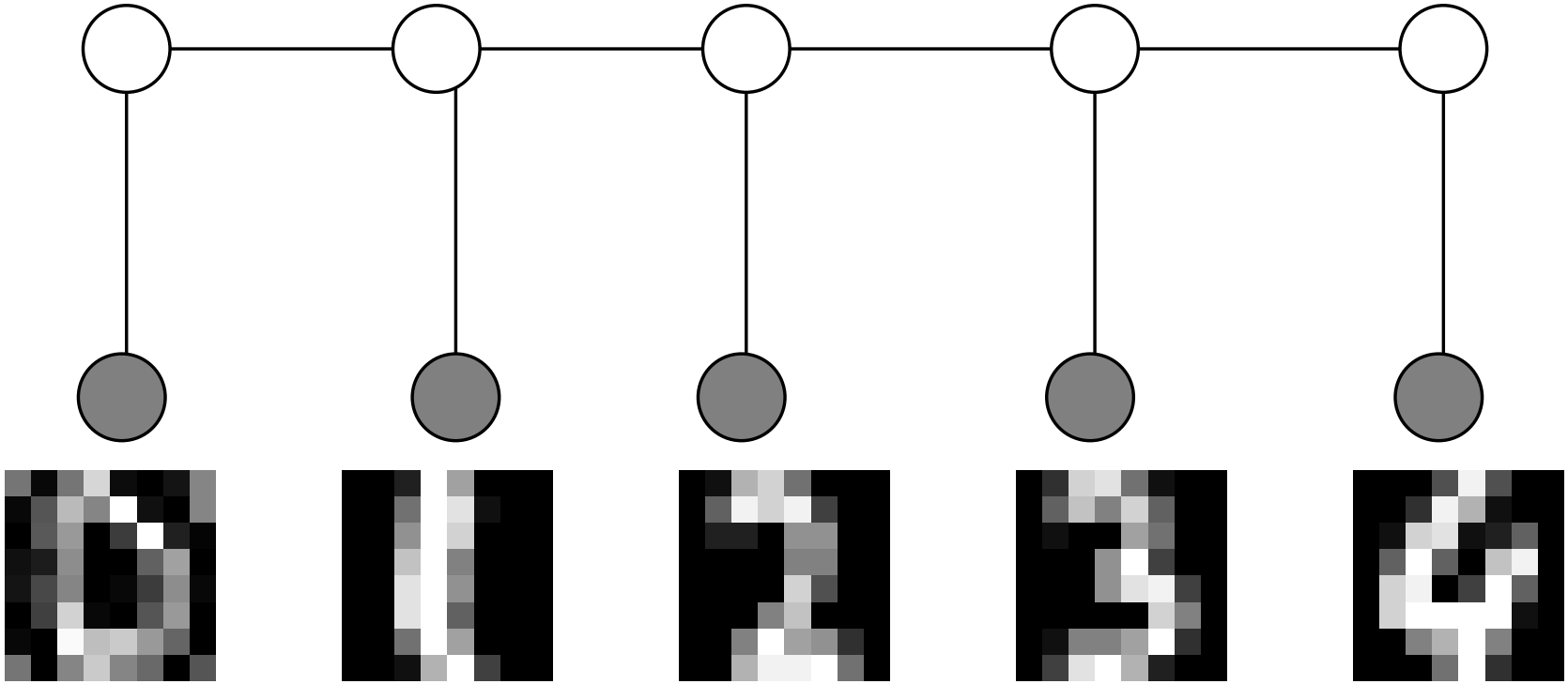
$$p(x_1, \ldots, x_N) = \frac{1}{Z} \prod_{C \in \mathfrak{C}} \psi_C(x_C).$$

Example: Optical digit/character recognition



- **Goal:** correctly label digits/characters based on "noisy" versions
- E.g., mail sorting; document scanning; handwriting recognition systems

Example: Optical digit/character recognition



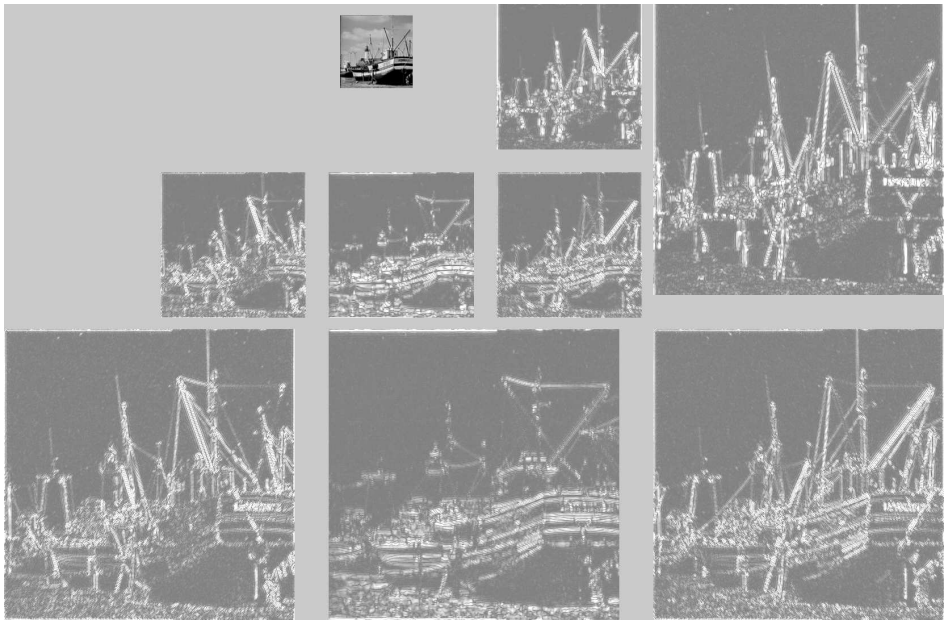
- **Goal:** correctly label digits/characters based on "noisy" versions
- strong sequential dependencies captured by (hidden) Markov chain
- "message-passing" spreads information along chain

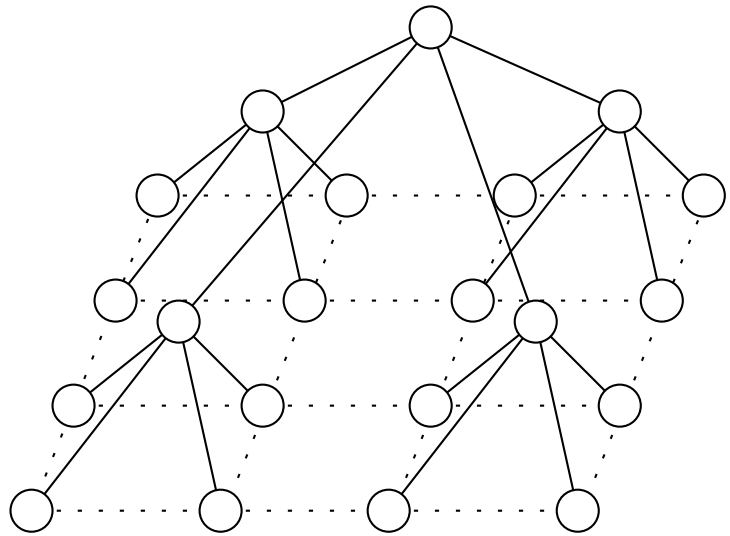
(Baum & Petrie, 1966; Viterbi, 1967, and many others)

Example: Image processing and denoising

- 8-bit digital image: matrix of intensity values $\{0, 1, \ldots 255\}$
- enormous redundancy in "typical" images (useful for denoising, compression, etc.)

Example: Image processing and denoising





- 8-bit digital image: matrix of intensity values $\{0, 1, \ldots 255\}$
- enormous redundancy in "typical" images (useful for denoising, compression, etc.)
- multiscale tree used to represent coefficients of a multiscale transform (e.g., wavelets, Gabor filters etc.)

(e.g., Willsky, 2002)

Example: Depth estimation in computer vision

Stereo pairs: two images taken from horizontally-offset cameras

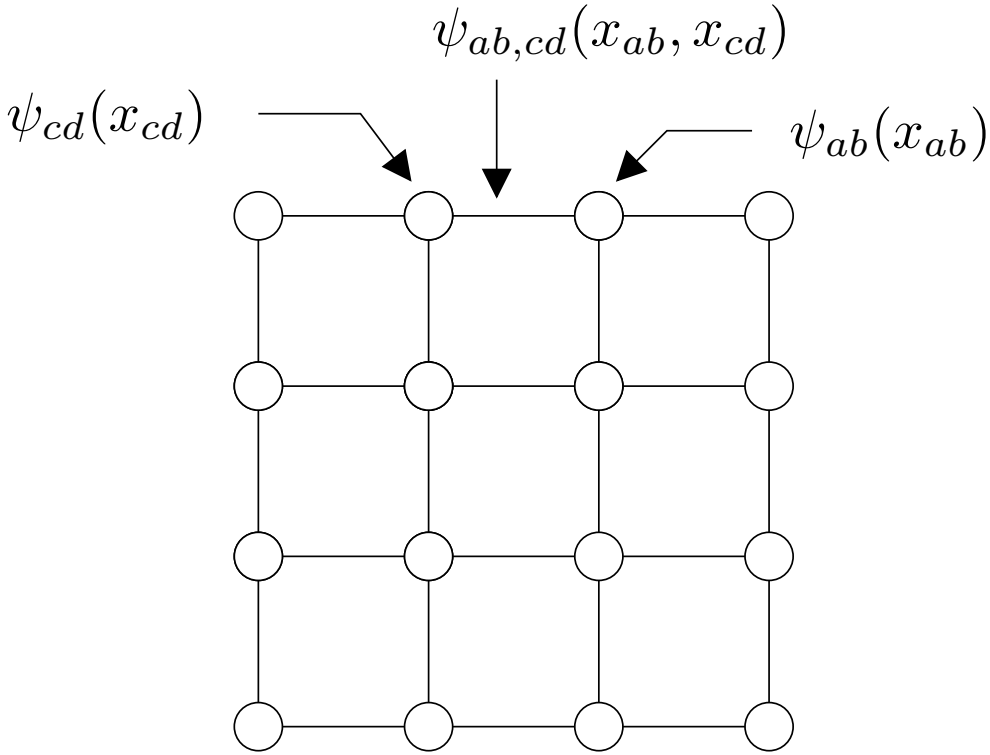
Modeling depth with a graphical model

Introduce variable at pixel location (a, b):

$$x_{ab} \equiv \text{Offset between images in position } (a, b)$$

Left image

Right image



Use message-passing algorithms to estimate most likely offset/depth map.
(Szeliski et al., 2005)

Many other examples

- natural language processing (e.g., parsing, translation)
- computational biology (gene sequences, protein folding, phylogenetic reconstruction)
- social network analysis (e.g., politics, Facebook, terrorism.)
- communication theory and error-control decoding (e.g., turbo codes, LDPC codes)
- satisfiability problems (3-SAT, MAX-XORSAT, graph colouring)
- robotics (path planning, tracking, navigation)
- sensor network deployments (e.g., distributed detection, estimation, fault monitoring)
- ...

Core computational challenges

Given an undirected graphical model (Markov random field):

$$p(x_1, x_2, \ldots, x_N) \quad = \quad \frac{1}{Z} \prod_{C \in \mathcal{C}} \psi_C(x_C)$$

How to efficiently compute?

- most probable configuration (MAP estimate):

$$\textbf{Maximize}: \qquad \widehat{x} = \arg \max_{\mathbf{x} \in \mathcal{X}^N} p(x_1, \ldots, x_N) \; = \; \arg \max_{\mathbf{x} \in \mathcal{X}^N} \prod_{C \in \mathcal{C}} \psi_C(x_C).$$

- the data likelihood or normalization constant

$$\textbf{Sum/integrate}: \qquad Z \; = \; \sum_{x \in \mathcal{X}^N} \prod_{C \in \mathcal{C}} \psi_C(x_C)$$

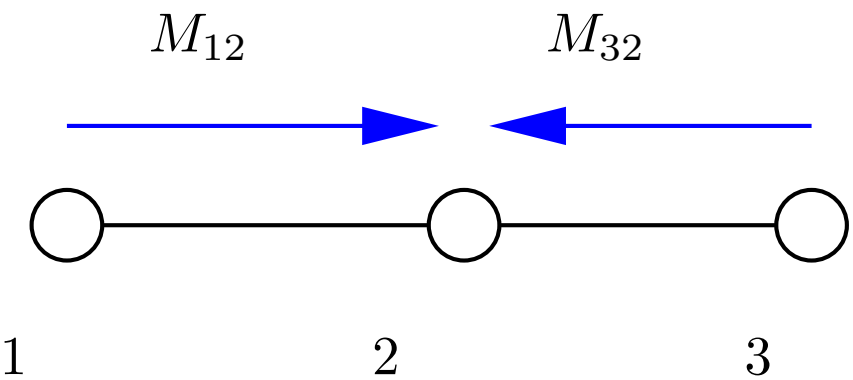
- marginal distributions at single sites, or subsets:

$$\textbf{Sum/integrate}: \qquad p(X_s = x_s) = \frac{1}{Z} \sum_{x_t, \, t \neq s} \prod_{C \in \mathcal{C}} \psi_C(x_C)$$

§1. Max-product message-passing on trees

Goal: Compute most probable configuration (MAP estimate) on a tree:

$$\widehat{x} \;=\; \arg\max_{\mathbf{x}\in\mathcal{X}^N} \left\{ \prod_{s\in V} \exp(\theta_s(x_s) \prod_{(s,t)\in E} \exp(\theta_{st}(x_s, x_t)) \right\}.$$

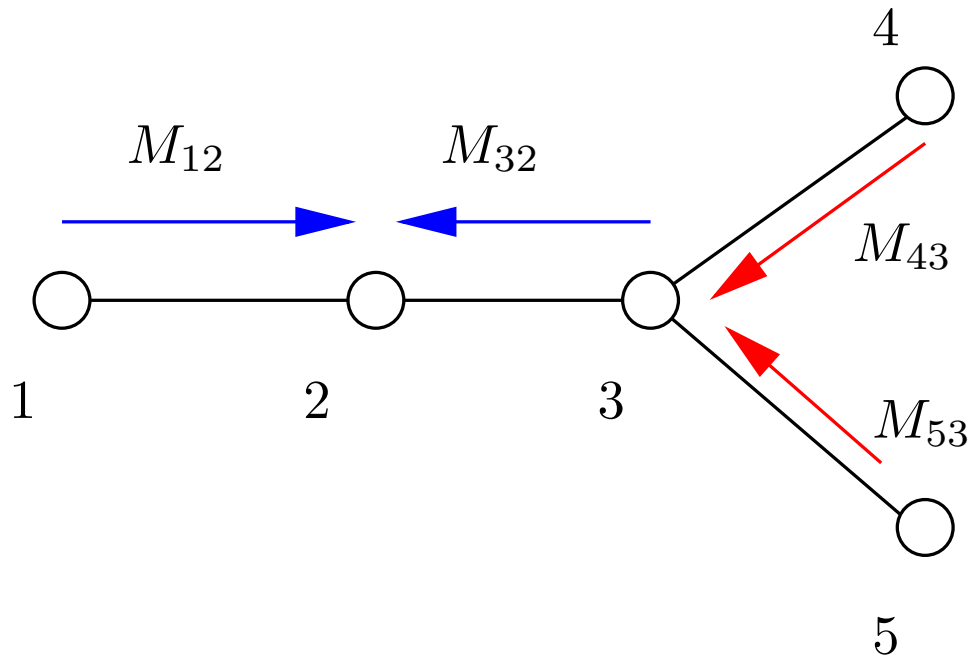


$$\max_{x_1,x_2,x_3} p(\mathbf{x}) \;=\; \max_{x_2} \left[\exp(\theta_2(x_2)) \prod_{t\in 1,3} \left\{ \max_{x_t} \exp[\theta_t(x_t) + \theta_{2t}(x_2, x_t)] \right\} \right]$$

Max-product strategy: "Divide and conquer": break global maximization into simpler sub-problems. (Lauritzen & Spiegelhalter, 1988)

Max-product on trees

Decompose: $\max_{x_1,x_2,x_3,x_4,x_5} p(\mathbf{x}) = \max_{x_2} \left[\exp(\theta_1(x_1)) \prod_{t \in N(2)} M_{t2}(x_2) \right].$

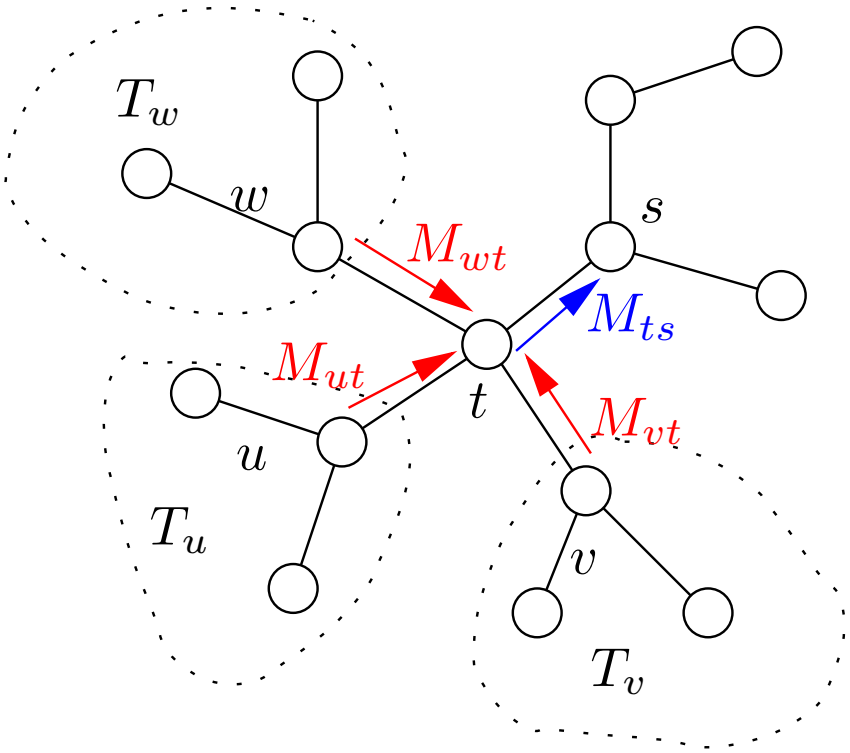


Update messages:

$$M_{32}(x_2) = \max_{x_3} \left[\exp(\theta_3(x_3) + \theta_{23}(x_2, x_3) \prod_{v \in N(3) \setminus 2} M_{v3}(x_3) \right]$$

Putting together the pieces

Max-product is an exact algorithm for any tree.



$M_{ts} \equiv$ message from node t to s

$\mathcal{N}(t) \equiv$ neighbors of node t

Update: $\mathbf{M_{ts}(x_s)} \leftarrow \max_{x'_t \in \mathcal{X}_t} \left\{ \exp\left[\theta_{st}(x_s, x'_t) + \theta_t(x'_t)\right] \prod_{v \in \mathcal{N}(t) \setminus s} \mathbf{M_{vt}(x_t)} \right\}$

Max-marginals: $\widetilde{p}_s(x_s; \theta) \propto \exp\{\theta_s(x_s)\} \prod_{t \in \mathcal{N}(s)} M_{ts}(x_s).$

Summary: max-product on trees

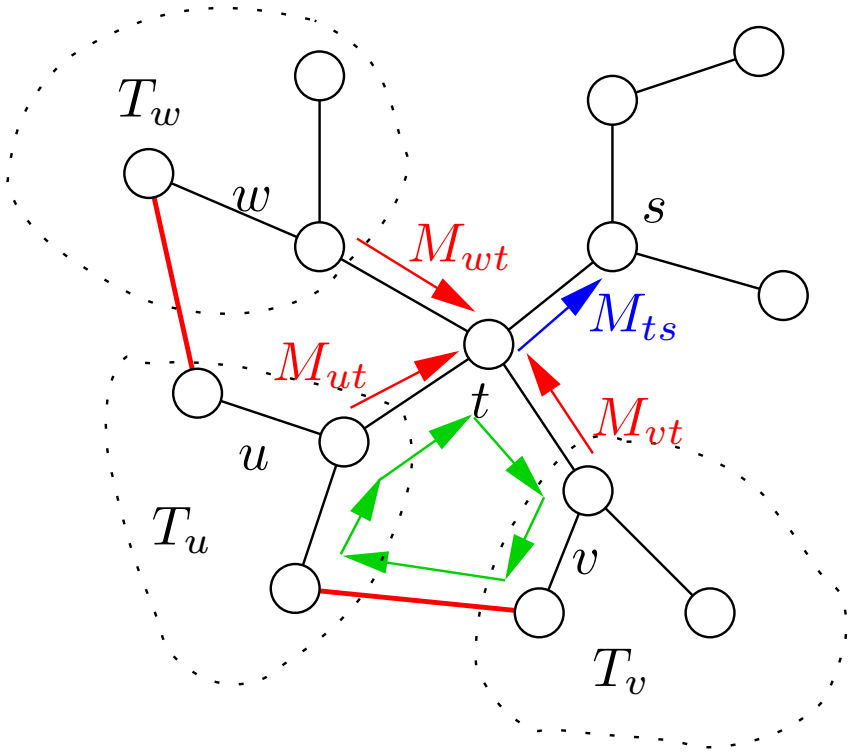
- converges in at most graph diameter # of iterations
- updating a single message is an $\mathcal{O}(m^2)$ operation
- overall algorithm requires $\mathcal{O}(Nm^2)$ operations

- upon convergence, yields the exact *max-marginals*:

$$\widetilde{p}_s(x_s) \propto \exp\{\theta_s(x_s)\} \prod_{t \in \mathcal{N}(s)} M_{ts}(x_s).$$

- when $\arg max_{x_s} \widetilde{p}_s(x_s) = \{x^s\}$ for all $s \in V$, then $x^* = (x_1^*, \ldots, x_N^*)$ is the *unique MAP solution*
- otherwise, there are multiple MAP solutions and one can be obtained by back-tracking

§2. Max-product on graph with cycles?



$M_{ts} \equiv$ message from node t to s

$\mathcal{N}(t) \equiv$ neighbors of node t

- max-product can be applied to graphs with cycles (no longer exact)
- empirical performance is often very good

Partial guarantees for max-product

- single-cycle graphs and Gaussian models
  (Aji & McEliece, 1998; Horn, 1999; Weiss, 1998, Weiss & Freeman, 2001)

- local optimality guarantees:
  - "tree-plus-loop" neighborhoods (Weiss & Freeman, 2001)
  - optimality on more general sub-graphs (Wainwright et al., 2003)

- existence of fixed points for general graphs (Wainwright et al., 2003)

- exactness for certain matching problems (Bayati et al., 2005, 2008, Jebara & Huang, 2007, Sanghavi, 2008)

- no general optimality results

Partial guarantees for max-product

- single-cycle graphs and Gaussian models
  (Aji & McEliece, 1998; Horn, 1999; Weiss, 1998, Weiss & Freeman, 2001)

- local optimality guarantees:
  - "tree-plus-loop" neighborhoods (Weiss & Freeman, 2001)
  - optimality on more general sub-graphs (Wainwright et al., 2003)

- existence of fixed points for general graphs (Wainwright et al., 2003)

- exactness for certain matching problems (Bayati et al., 2005, 2008, Jebara & Huang, 2007, Sanghavi, 2008)

- no general optimality results

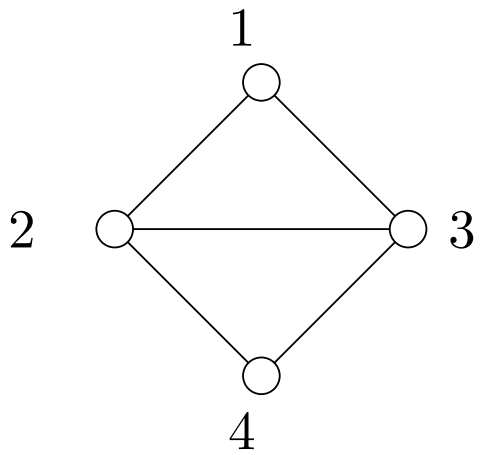
Questions:

- Can max-product return an incorrect answer with high confidence?
- Any connection to classical approaches to integer programs?

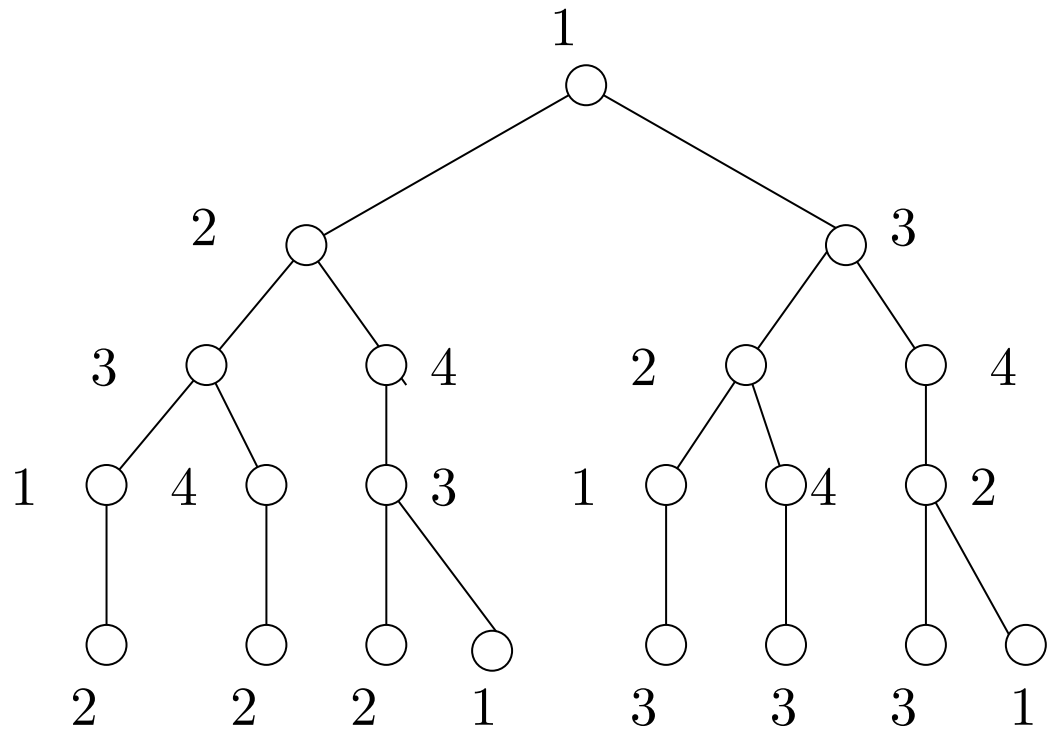
Standard analysis via computation tree

- standard tool: computation tree of message-passing updates

(Gallager, 1963; Weiss, 2001; Richardson & Urbanke, 2001)



(a) Original graph



(b) Computation tree (4 iterations)

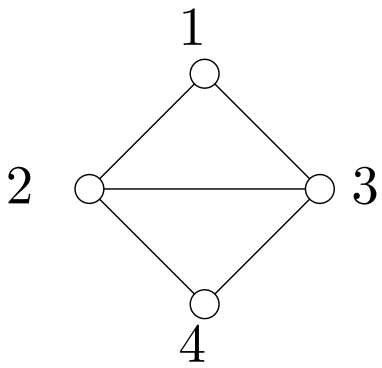
- level t of tree: all nodes whose messages reach the root (node 1) after t iterations of message-passing

Example: Inexactness of standard max-product

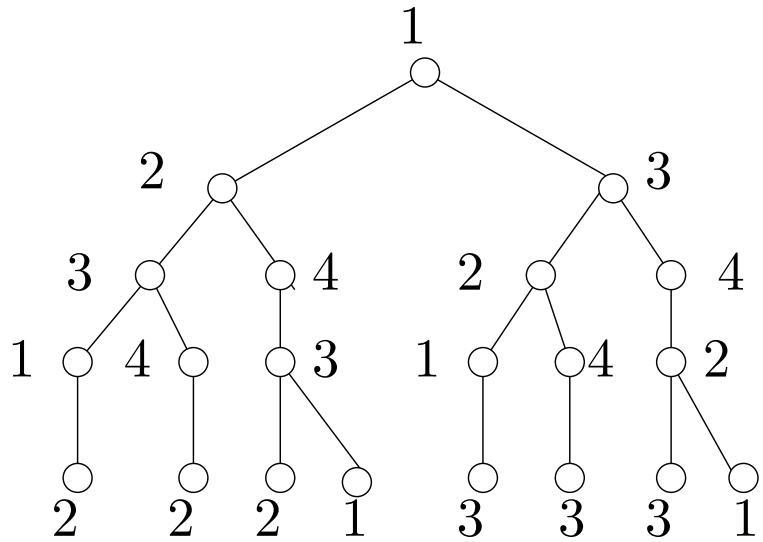
(Wainwright et al., 2005)

Intuition:

- max-product solves (exactly) a modified problem on computation tree
- nodes *not equally weighted* in computation tree $\Rightarrow$ max-product can output an incorrect configuration



(a) Diamond graph G_{dia}

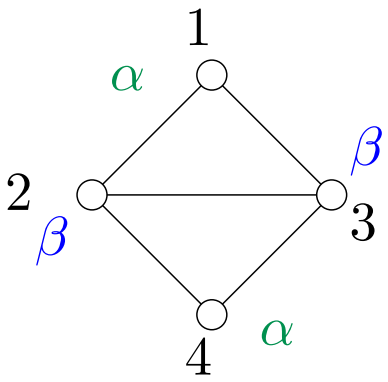


(b) Computation tree (4 iterations)

- for example: asymptotic node fractions ω in this computation tree:

$$\begin{bmatrix}\omega(1) & \omega(2) & \omega(3) & \omega(4)\end{bmatrix} = \begin{bmatrix}0.2393 & 0.2607 & 0.2607 & 0.2393\end{bmatrix}$$

A whole family of non-exact examples



$$\theta_s(x_s) \qquad \begin{cases} \alpha x_s & \text{if } s = 1 \text{ or } s = 4 \\ \beta x_s & \text{if } s = 2 \text{ or } s = 3 \end{cases}$$

$$\theta_{st}(x_s, x_t) \quad = \quad \begin{cases} -\gamma & \text{if } x_s \neq x_t \\ 0 & \text{otherwise} \end{cases}$$

- for γ sufficiently large, optimal solution is always either $1^4 = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}$ or $(-1)^4 = \begin{bmatrix} (-1) & (-1) & (-1) & (-1) \end{bmatrix}$
- first-order LP relaxation always exact for this problem
- max-product and LP relaxation give *different* decision boundaries:

Optimal/LP boundary: $\widehat{\mathbf{x}} = \begin{cases} 1^4 & \text{if } 0.25\alpha + 0.25\beta \geq 0 \\ (-1)^4 & \text{otherwise} \end{cases}$

Max-product boundary: $\widehat{\mathbf{x}} = \begin{cases} 1^4 & \text{if } 0.2393\alpha + 0.2607\beta \geq 0 \\ (-1)^4 & \text{otherwise} \end{cases}$

§3. A more general class of algorithms

- by introducing weights on edges, obtain a more general family of *reweighted max-product algorithms*
- with suitable edge weights, connected to linear programming relaxations
- many variants of these algorithms:
  - tree-reweighted max-product (W., Jaakkola & Willsky, 2002, 2005)
  - sequential TRMP (Kolmogorov, 2005)
  - convex message-passing (Weiss et al., 2007)
  - dual updating schemes (e.g., Globerson & Jaakkola, 2007)

Tree-reweighted max-product algorithms

(Wainwright, Jaakkola & Willsky, 2002)

Message update from node t to node s:

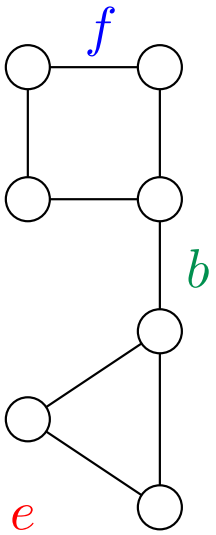
$$M_{ts}(x_s) \quad \leftarrow \quad \kappa \max_{x'_t \in \mathcal{X}_t} \left\{ \exp\left[\underbrace{\frac{\theta_{st}(x_s, x'_t)}{\rho_{st}}}_{\text{reweighted edge}} + \theta_t(x'_t) \right] \frac{\prod_{v \in \mathcal{N}(t) \setminus s} \overbrace{\left[M_{vt}(x_t)\right]^{\rho_{vt}}}^{\text{reweighted messages}}}{\underbrace{\left[M_{st}(x_t)\right]^{(1-\rho_{ts})}}_{\text{opposite message}}} \right\}.$$

Properties:

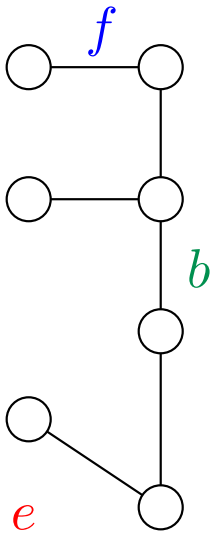
1. Modified updates remain *distributed* and *purely local* over the graph.
2. Key differences:
   - Messages are reweighted with $\rho_{st} \in [0, 1]$.
   - Potential on edge (s, t) is rescaled by $\rho_{st} \in [0, 1]$.
   - Update involves the reverse direction edge.
3. The choice $\rho_{st} = 1$ for all edges (s, t) recovers standard update.

Edge appearance probabilities

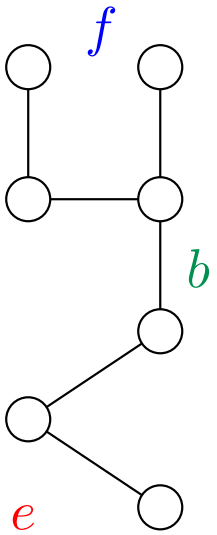
Experiment: What is the probability ρ_e that a given edge $e \in E$ belongs to a tree T drawn randomly under $\boldsymbol{\rho}$?



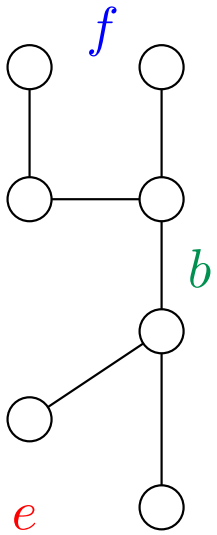
(a) Original



(b) $\rho(T^1) = \frac{1}{3}$



(c) $\rho(T^2) = \frac{1}{3}$



(d) $\rho(T^3) = \frac{1}{3}$

In this example: $\rho_b = 1$; $\rho_e = \frac{2}{3}$; $\rho_f = \frac{1}{3}$.

The vector $\boldsymbol{\rho_e} = \{ \rho_e \mid e \in E \}$ must belong to the *spanning tree polytope*.
(Edmonds, 1971)

§4. Reweighted max-product and linear programming

- MAP as integer program: $f^* = \max_{\mathbf{x} \in \mathcal{X}^N} \{ \sum_{s \in V} \theta_s(x_s) + \sum_{(s,t) \in E} \theta_{st}(x_s, x_t) \}$

- define local marginal distributions (e.g., for $m = 3$ states):

$$\mu_s(x_s) = \begin{bmatrix} \mu_s(0) \\ \mu_s(1) \\ \mu_s(2) \end{bmatrix} \qquad \mu_{st}(x_s, x_t) = \begin{bmatrix} \mu_{st}(0,0) & \mu_{st}(0,1) & \mu_{st}(0,2) \\ \mu_{st}(1,0) & \mu_{st}(1,1) & \mu_{st}(1,2) \\ \mu_{st}(2,0) & \mu_{st}(2,1) & \mu_{st}(2,2) \end{bmatrix}$$

§4. Reweighted max-product and linear programming

- MAP as integer program: $f^* = \max_{\mathbf{x} \in \mathcal{X}^N} \{ \sum_{s \in V} \theta_s(x_s) + \sum_{(s,t) \in E} \theta_{st}(x_s, x_t) \}$

- define local marginal distributions (e.g., for $m = 3$ states):

$$\mu_s(x_s) = \begin{bmatrix} \mu_s(0) \\ \mu_s(1) \\ \mu_s(2) \end{bmatrix} \qquad \mu_{st}(x_s, x_t) = \begin{bmatrix} \mu_{st}(0,0) & \mu_{st}(0,1) & \mu_{st}(0,2) \\ \mu_{st}(1,0) & \mu_{st}(1,1) & \mu_{st}(1,2) \\ \mu_{st}(2,0) & \mu_{st}(2,1) & \mu_{st}(2,2) \end{bmatrix}$$

- alternative formulation of MAP as linear program?

$$g^* = \max_{(\mu_s, \mu_{st}) \in \mathbb{M}(G)} \{ \sum_{s \in V} \mathbb{E}_{\mu_s}[\theta_s(x_s)] + \sum_{(s,t) \in E} \mathbb{E}_{\mu_{st}}[\theta_{st}(x_s, x_t)] \}$$

Local expectations: $\mathbb{E}_{\mu_s}[\theta_s(x_s)] := \sum_{x_s} \mu_s(x_s) \theta_s(x_s).$

§4. Reweighted max-product and linear programming

- MAP as integer program: $f^* = \max_{\mathbf{x} \in \mathcal{X}^N} \{ \sum_{s \in V} \theta_s(x_s) + \sum_{(s,t) \in E} \theta_{st}(x_s, x_t) \}$

- define local marginal distributions (e.g., for $m = 3$ states):

$$\mu_s(x_s) = \begin{bmatrix} \mu_s(0) \\ \mu_s(1) \\ \mu_s(2) \end{bmatrix} \qquad \mu_{st}(x_s, x_t) = \begin{bmatrix} \mu_{st}(0,0) & \mu_{st}(0,1) & \mu_{st}(0,2) \\ \mu_{st}(1,0) & \mu_{st}(1,1) & \mu_{st}(1,2) \\ \mu_{st}(2,0) & \mu_{st}(2,1) & \mu_{st}(2,2) \end{bmatrix}$$

- alternative formulation of MAP as linear program?

$$g^* = \max_{(\mu_s, \mu_{st}) \in \mathbb{M}(G)} \{ \sum_{s \in V} \mathbb{E}_{\mu_s}[\theta_s(x_s)] + \sum_{(s,t) \in E} \mathbb{E}_{\mu_{st}}[\theta_{st}(x_s, x_t)] \}$$

Local expectations: $\mathbb{E}_{\mu_s}[\theta_s(x_s)] := \sum_{x_s} \mu_s(x_s) \theta_s(x_s).$

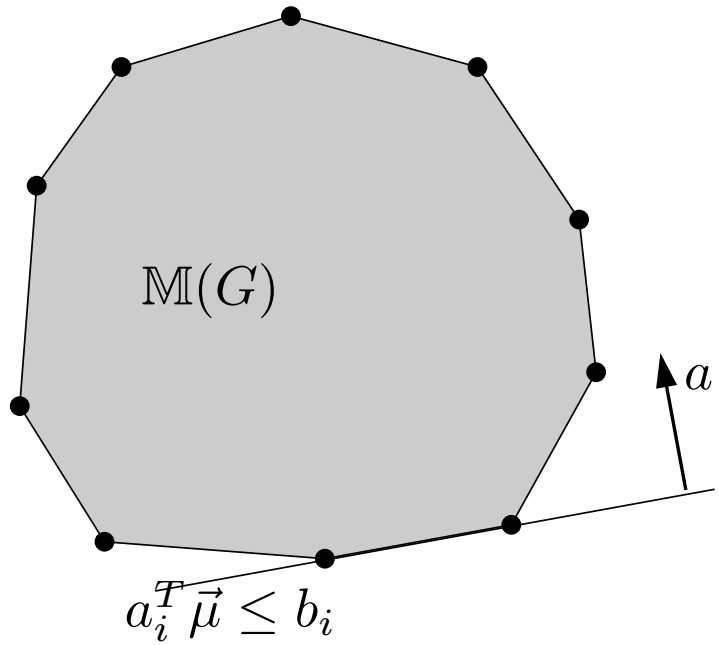
Key question: What constraints must local marginals $\{\mu_s, \mu_{st}\}$ satisfy?

Marginal polytopes for general undirected models

- $\mathbb{M}(G) \equiv$ set of all *globally realizable* marginals $\{\mu_s, \mu_{st}\}$:

$$\left\{ \vec{\mu} \in \mathbb{R}^d \;\middle|\; \mu_s(x_s) \;=\; \sum_{x_t, t \neq s} p_\mu(\mathbf{x}), \text{ and } \mu_{st}(x_s, x_t) \;=\; \sum_{x_u, u \neq s,t} p_\mu(\mathbf{x}) \right\}$$

  for some $p_\mu(\cdot)$ over $(X_1, \ldots, X_N) \in \{0, 1, \ldots, m-1\}^N$.



- polytope in $d = m|V| + m^2|E|$ dimensions (m per vertex, m^2 per edge)
- with m^N vertices
- number of facets?

Marginal polytope for trees

- $\mathbb{M}(T) \equiv$ special case of marginal polytope for tree T
- local marginal distributions on nodes/edges (e.g., $m = 3$)

$$\mu_s(x_s) = \begin{bmatrix} \mu_s(0) \\ \mu_s(1) \\ \mu_s(2) \end{bmatrix} \qquad \mu_{st}(x_s, x_t) = \begin{bmatrix} \mu_{st}(0,0) & \mu_{st}(0,1) & \mu_{st}(0,2) \\ \mu_{st}(1,0) & \mu_{st}(1,1) & \mu_{st}(1,2) \\ \mu_{st}(2,0) & \mu_{st}(2,1) & \mu_{st}(2,2) \end{bmatrix}$$

Deep fact about tree-structured models: If $\{\mu_s, \mu_{st}\}$ are non-negative and *locally consistent*:

$$\textbf{Normalization}: \quad \sum_{x_s} \mu_s(x_s) = 1$$

$$\textbf{Marginalization}: \quad \sum_{x'_t} \mu_{st}(x_s, x'_t) = \mu_s(x_s),$$

then on any tree-structured graph T, they are *globally consistent*.

Follows from junction tree theorem (Lauritzen & Spiegelhalter, 1988).

Max-product on trees: Linear program solver

- MAP problem as a simple linear program:

$$f(\widehat{x}) \quad = \quad \arg \max_{\vec{\mu} \in \mathbb{M}(T)} \left\{ \sum_{s \in V} \mathbb{E}_{\mu_s}[\theta_s(x_s)] + \sum_{(s,t) \in E} \mathbb{E}_{\mu_{st}}[\theta_{st}(x_s, x_t)] \right\}$$

subject to $\vec{\mu}$ in tree marginal polytope:

$$\mathbb{M}(T) \; = \; \left\{ \vec{\mu} \geq 0, \quad \sum_{x_s} \mu_s(x_s) \; = \; 1, \qquad \sum_{x'_t} \mu_{st}(x_s, x'_t) \; = \; \mu_s(x_s) \right\}.$$

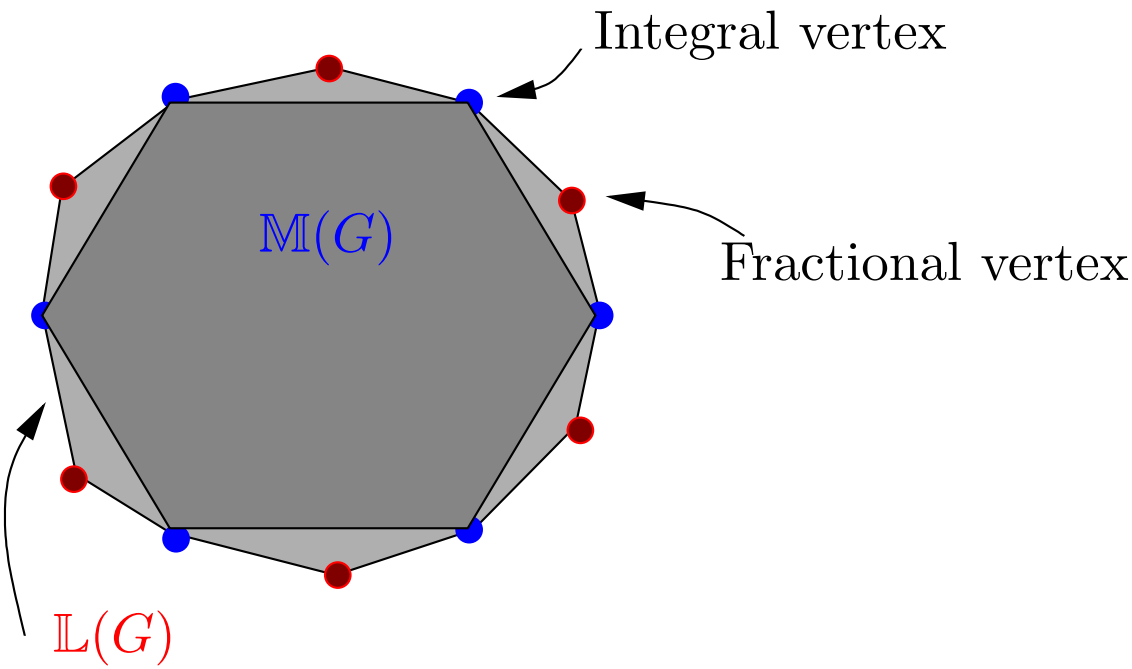
Max-product and LP solving:

- on tree-structured graphs, max-product is a dual algorithm for solving the tree LP. (Wai. & Jordan, 2003)
- max-product message $M_{ts}(x_s) \equiv$ Lagrange multiplier for enforcing the constraint $\sum_{x'_t} \mu_{st}(x_s, x'_t) \; = \; \mu_s(x_s)$.

Tree-based relaxation for graphs with cycles

Set of *locally consistent pseudomarginals* for general graph G:

$$\mathbb{L}(G) \;=\; \left\{ \vec{\tau} \in \mathbb{R}^d \;\mid\; \vec{\tau} \geq 0, \sum_{x_s} \tau_s(x_s) = 1, \quad \sum_{x_t} \tau_{st}(x_s, x_t') = \tau_s(x_s) \right\}.$$

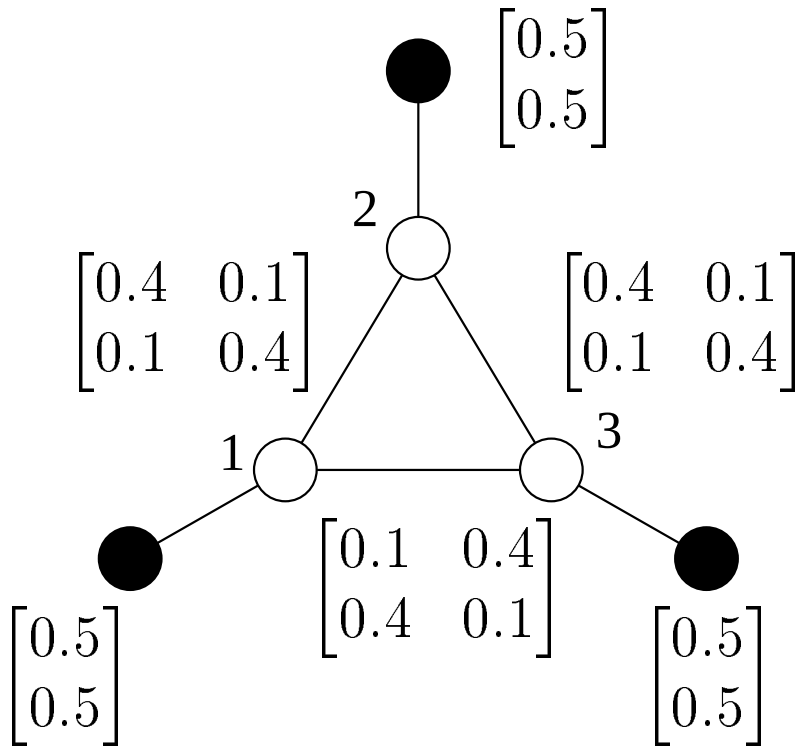


Key: For a general graph, $\mathbb{L}(G)$ is an outer bound on $\mathbb{M}(G)$, and yields a *linear-programming relaxation* of the MAP problem:

$$f(\widehat{x}) \;=\; \max_{\vec{\mu} \in \mathbb{M}(G)} \theta^T \vec{\mu} \;\leq\; \max_{\vec{\tau} \in \mathbb{L}(G)} \theta^T \vec{\tau}.$$

Looseness of $\mathbb{L}(G)$ with graphs with cycles

Locally consistent (pseudo)marginals



Pseudomarginals satisfy the "obvious" local constraints:

Normalization: $\sum_{x'_s} \tau_s(x'_s) = 1$ for all $s \in V$.

Marginalization: $\sum_{x'_s} \tau_s(x'_s, x_t) = \tau_t(x_t)$ for all edges (s, t).

TRW max-product and LP relaxation

First-order (tree-based) LP relaxation:

$$f(\widehat{x}) \leq \max_{\vec{\tau} \in \mathbb{L}(G)} \left\{ \sum_{s \in V} \mathbb{E}_{\tau_s}[\theta_s(x_s)] + \sum_{(s,t) \in E} \mathbb{E}_{\tau_{st}}[\theta_{st}(x_s, x_t)] \right\}$$

Results: (Wainwright et al., 2005; Kolmogorov & Wainwright, 2005):

(a) **Strong tree agreement** Any TRW fixed-point that satisfies the strong tree agreement condition specifies an optimal LP solution.

(b) **LP solving:** For any binary pairwise problem, TRW max-product solves the first-order LP relaxation.

(c) **Persistence for binary problems:** Let $S \subseteq V$ be the subset of vertices for which there exists a single point $x_s^* \in \arg\max_{x_s} \nu_s^*(x_s)$. Then for *any optimal solution*, it holds that $y_s = x_s^*$.

On-going work on LPs and conic relaxations

- tree-reweighted max-product solves first-order LP for any binary pairwise problem (Kolmogorov & Wainwright, 2005)
- convergent dual ascent scheme; LP-optimal for binary pairwise problems (Globerson & Jaakkola, 2007)
- convex free energies and zero-temperature limits (Wainwright et al., 2005, Weiss et al., 2006; Johnson et al., 2007)
- coding problems: adaptive cutting-plane methods (Taghavi & Siegel, 2006; Dimakis et al., 2006)
- dual decomposition and sub-gradient methods: (Feldman et al., 2003; Komodakis et al., 2007, Duchi et al., 2007)
- solving higher-order relaxations; rounding schemes (e.g., Sontag et al., 2008; Ravikumar et al., 2008)

Hierarchies of conic programming relaxations

- tree-based LP relaxation using $\mathbb{L}(G)$: first in a hierarchy of hypertree-based relaxations (Wainwright & Jordan, 2004)
- hierarchies of SDP relaxations for polynomial programming (Lasserre, 2001; Parrilo, 2002)
- intermediate between LP and SDP: second-order cone programming (SOCP) relaxations (Ravikumar & Lafferty, 2006; Kumar et al., 2008)
- all relaxations: particular outer bounds on the marginal polyope

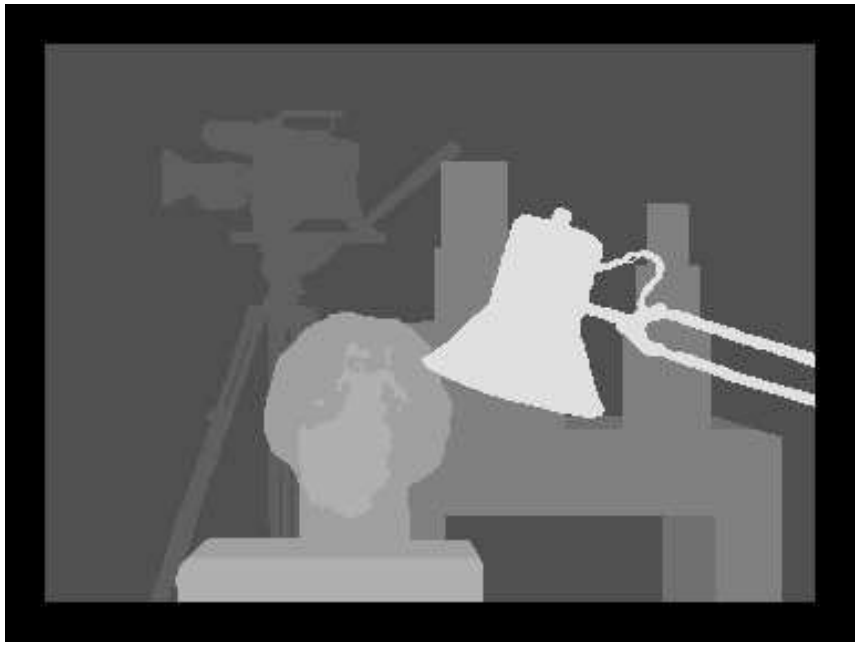
Key questions:

- when are particular relaxations tight?
- when does more computation (e.g., LP $\rightarrow$ SOCP $\rightarrow$ SDP) yield performance gains?

Stereo computation: Middlebury stereo benchmark set

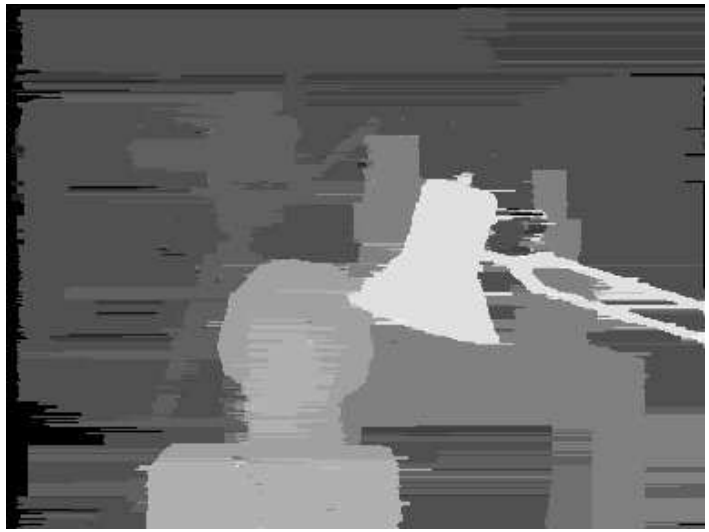
- standard set of benchmarked examples for stereo algorithms (Scharstein & Szeliski, 2002)
- Tsukuba data set: Image sizes $384 \times 288 \times 16$ $(W \times H \times D)$

(a) Original image



(b) Ground truth disparity

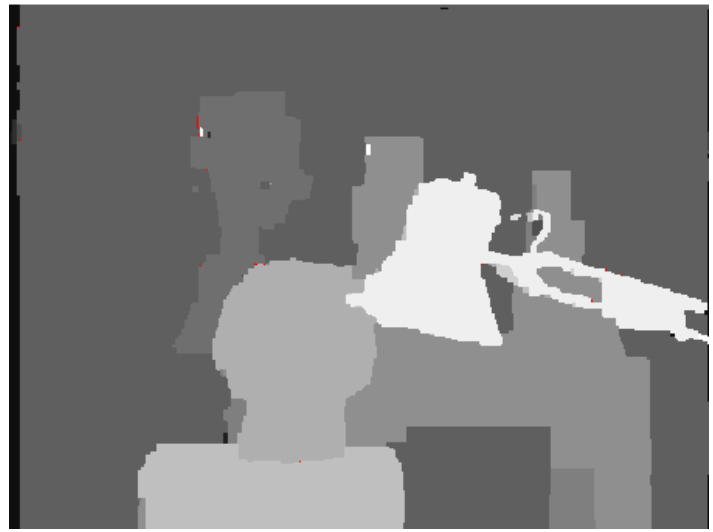
Comparison of different methods



(a) Scanline dynamic programming

(b) Graph cuts

(c) Ordinary belief propagation

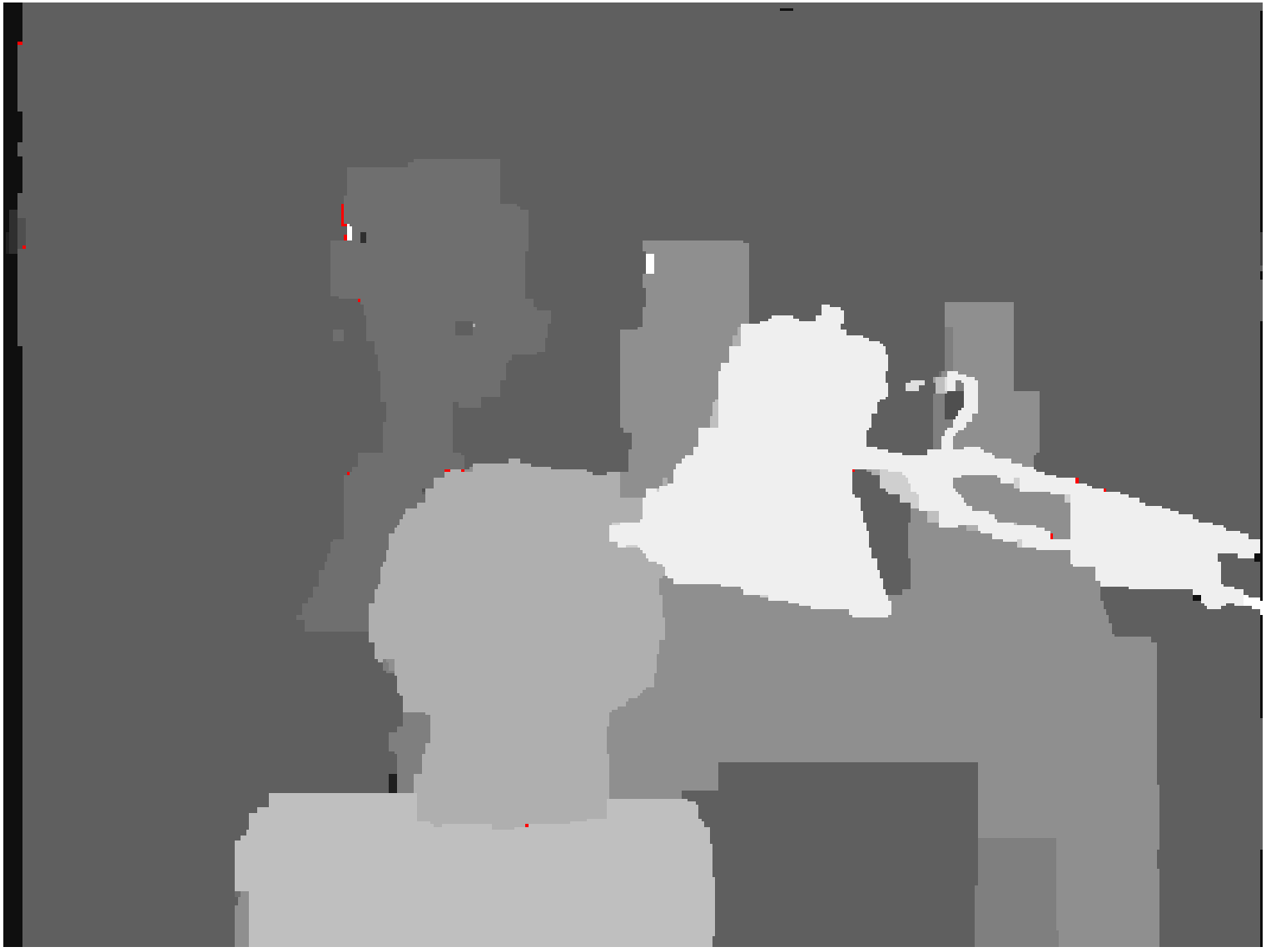


(d) Tree-reweighted max-product

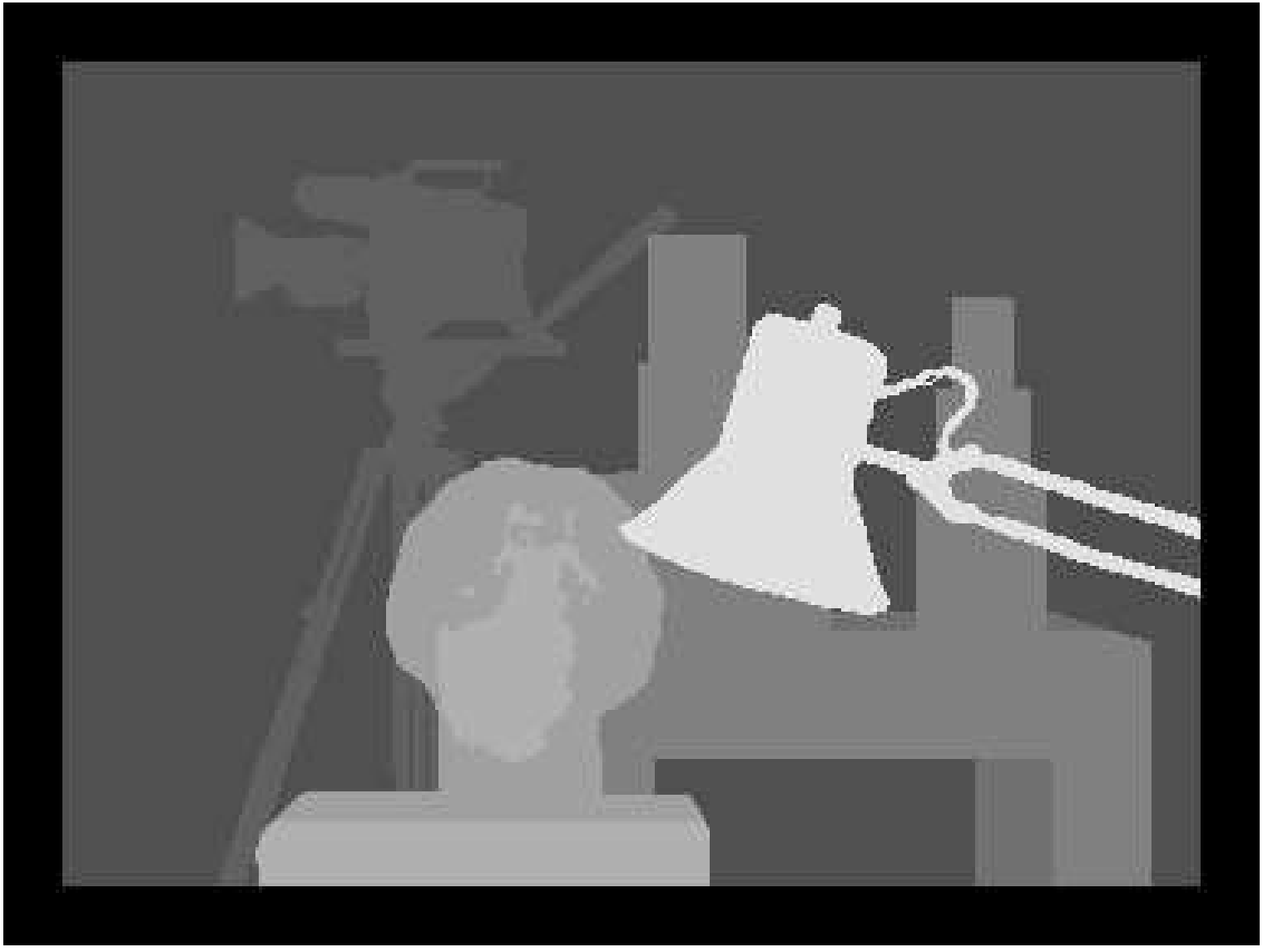
(a), (b): Scharstein & Szeliski, 2002; (c): Sun et al., 2002 (d): Weiss, et al., 2005;

Ordinary belief propagation

Tree-reweighted max-product



Ground truth



Graphical models and message-passing
Part II: Marginals and likelihoods

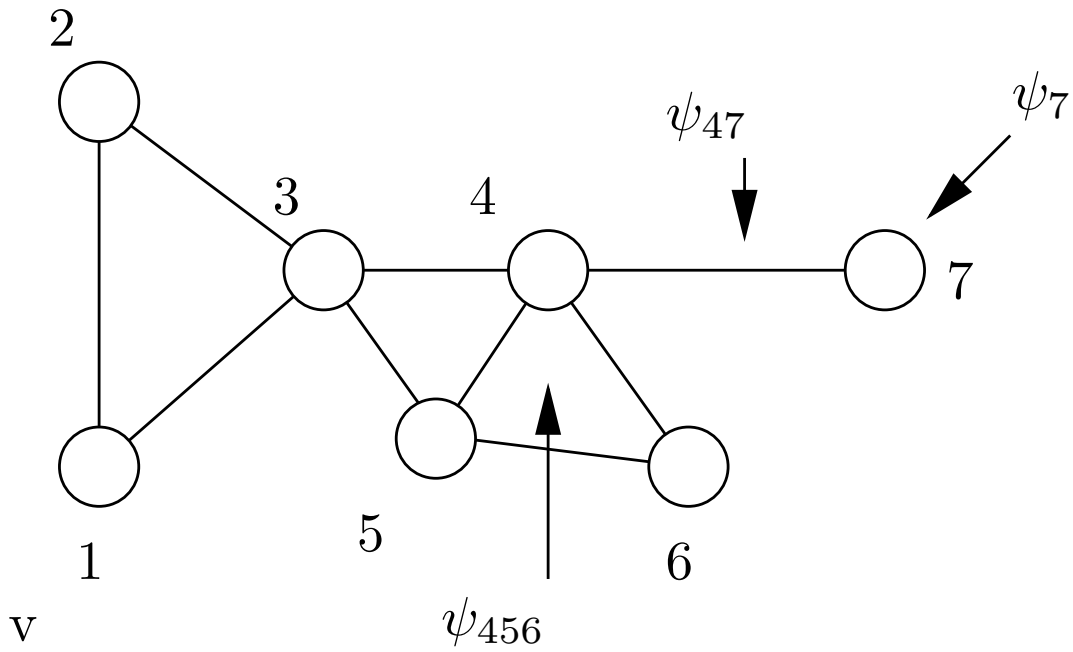
Martin Wainwright

UC Berkeley
Departments of Statistics, and EECS

Tutorial materials (slides, monograph, lecture notes) available at:
www.eecs.berkeley.edu/~wainwrig/kyoto12

September 3, 2012

Graphs and factorization



- clique C is a fully connected subset of vertices
- compatibility function ψ_C defined on variables $x_C = \{x_s, s \in C\}$
- factorization over all cliques

$$p(x_1, \ldots, x_N) = \frac{1}{Z} \prod_{C \in \mathfrak{C}} \psi_C(x_C).$$

Core computational challenges

Given an undirected graphical model (Markov random field):

$$p(x_1, x_2, \ldots, x_N) \quad = \quad \frac{1}{Z} \prod_{C \in \mathcal{C}} \psi_C(x_C)$$

How to efficiently compute?

- most probable configuration (MAP estimate):

$$\textbf{Maximize}: \qquad \widehat{x} = \arg \max_{\mathbf{x} \in \mathcal{X}^N} p(x_1, \ldots, x_N) \; = \; \arg \max_{\mathbf{x} \in \mathcal{X}^N} \prod_{C \in \mathcal{C}} \psi_C(x_C).$$

- the data likelihood or normalization constant

$$\textbf{Sum/integrate}: \qquad Z \; = \; \sum_{x \in \mathcal{X}^N} \prod_{C \in \mathcal{C}} \psi_C(x_C)$$

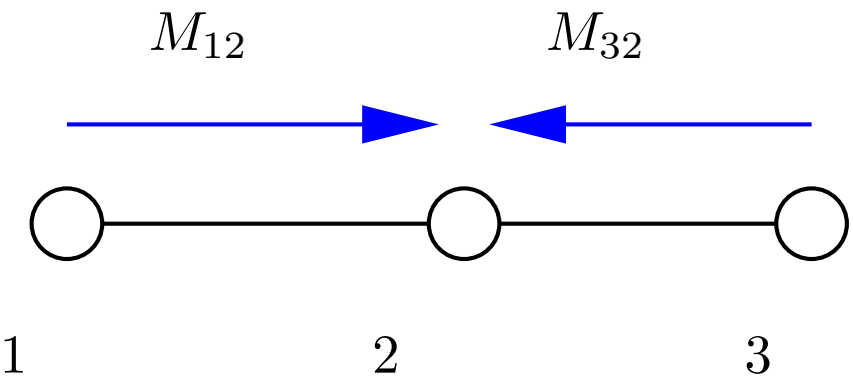
- marginal distributions at single sites, or subsets:

$$\textbf{Sum/integrate}: \qquad p(X_s = x_s) = \frac{1}{Z} \sum_{x_t, \, t \neq s} \prod_{C \in \mathcal{C}} \psi_C(x_C)$$

§1. Sum-product message-passing on trees

Goal: Compute marginal distribution at node u on a tree:

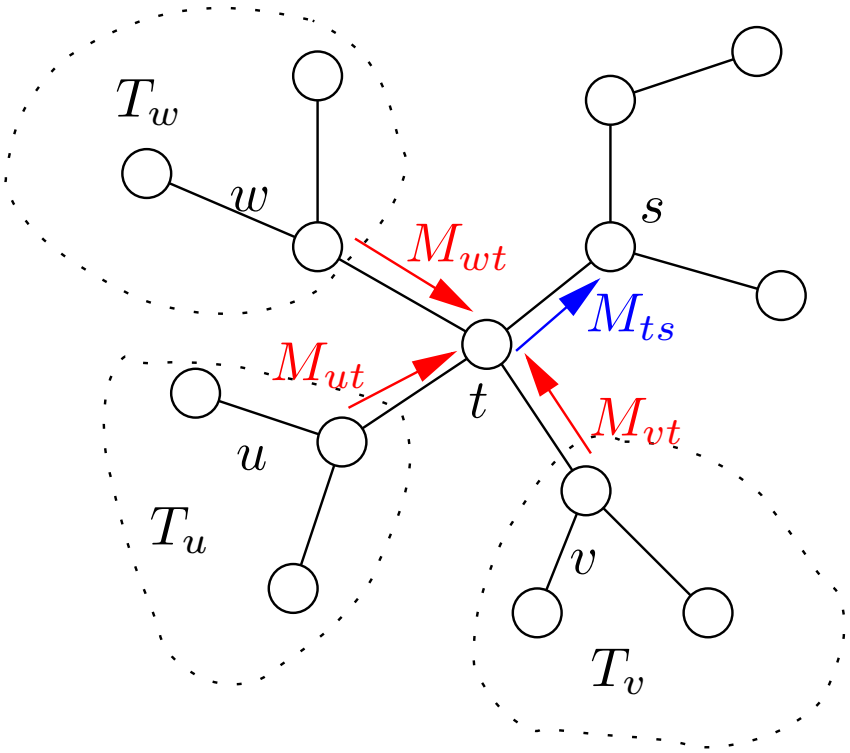
$$\widehat{x} \;=\; \arg\max_{\mathbf{x}\in\mathcal{X}^N} \left\{ \prod_{s\in V} \exp(\theta_s(x_s) \prod_{(s,t)\in E} \exp(\theta_{st}(x_s, x_t)) \right\}.$$



$$\sum_{x_1,x_2,x_3} p(\mathbf{x}) \;=\; \sum_{x_2} \left[\exp(\theta_1(x_1)) \prod_{t\in 1,3} \left\{ \sum_{x_t} \exp[\theta_t(x_t) + \theta_{2t}(x_2, x_t)] \right\} \right]$$

Putting together the pieces

Sum-product is an exact algorithm for any tree.



$M_{ts} \equiv$ message from node t to s

$\mathcal{N}(t) \equiv$ neighbors of node t

Update: $\mathbf{M_{ts}(x_s)} \leftarrow \sum_{x'_t \in \mathcal{X}_t} \left\{ \exp\left[\theta_{st}(x_s, x'_t) + \theta_t(x'_t)\right] \prod_{v \in \mathcal{N}(t) \setminus s} \mathbf{M_{vt}(x_t)} \right\}$

Sum-marginals: $p_s(x_s; \theta) \propto \exp\{\theta_s(x_s)\} \prod_{t \in \mathcal{N}(s)} M_{ts}(x_s)$.

Summary: sum-product on trees

- converges in at most graph diameter # of iterations
- updating a single message is an $\mathcal{O}(m^2)$ operation
- overall algorithm requires $\mathcal{O}(Nm^2)$ operations

- upon convergence, yields the exact node and edge marginals:

$$p_s(x_s) \propto e^{\theta_s(x_s)} \prod_{u \in \mathcal{N}(s)} M_{us}(x_s)$$

$$p_{st}(x_s, x_t) \propto e^{\theta_s(x_s)+\theta_t(x_t)+\theta_{st}(x_s,x_t)} \prod_{u \in \mathcal{N}(s)} M_{us}(x_s) \prod_{u \in \mathcal{N}(t)} M_{ut}(x_t)$$

- messages can also be used to compute the partition function

$$Z = \sum_{x_1,\ldots,x_N} \prod_{s \in V} e^{\theta_s(x_s)} \prod_{(s,t) \in E} e^{\theta_{st}(x_s,x_t)}.$$

§2. Sum-product on graph with cycles

- as with max-product, a widely used heuristic with a long history:
  - error-control coding: Gallager, 1963
  - artificial intelligence: Pearl, 1988
  - turbo decoding: Berroux et al., 1993
  - etc..

§2. Sum-product on graph with cycles

- as with max-product, a widely used heuristic with a long history:
  - error-control coding: Gallager, 1963
  - artificial intelligence: Pearl, 1988
  - turbo decoding: Berroux et al., 1993
  - etc..

- some concerns with sum-product with cycles:
  - no convergence guarantees
  - can have multiple fixed points
  - final estimate of Z is not a lower/upper bound

§2. Sum-product on graph with cycles

- as with max-product, a widely used heuristic with a long history:
  - error-control coding: Gallager, 1963
  - artificial intelligence: Pearl, 1988
  - turbo decoding: Berroux et al., 1993
  - etc..

- some concerns with sum-product with cycles:
  - no convergence guarantees
  - can have multiple fixed points
  - final estimate of Z is not a lower/upper bound

- as before, can consider a broader class of reweighted sum-product algorithms

Tree-reweighted sum-product algorithms

Message update from node t to node s:

$$M_{ts}(x_s) \leftarrow \kappa \sum_{x'_t \in \mathcal{X}_t} \left\{ \exp\left[\underbrace{\frac{\theta_{st}(x_s, x'_t)}{\rho_{st}}}_{\text{reweighted edge}} + \theta_t(x'_t) \right] \frac{\prod_{v \in \mathcal{N}(t) \setminus s} \overbrace{[M_{vt}(x_t)]^{\rho_{vt}}}^{\text{reweighted messages}}}{\underbrace{[M_{st}(x_t)]^{(1-\rho_{ts})}}_{\text{opposite message}}} \right\}.$$

Properties:

1. Modified updates remain *distributed* and *purely local* over the graph.
2. Key differences:
   - Messages are reweighted with $\rho_{st} \in [0, 1]$.
   - Potential on edge (s, t) is rescaled by $\rho_{st} \in [0, 1]$.
   - Update involves the reverse direction edge.
3. The choice $\rho_{st} = 1$ for all edges (s, t) recovers standard update.

Bethe entropy approximation

- define local marginal distributions (e.g., for $m = 3$ states):

$$\mu_s(x_s) = \begin{bmatrix} \mu_s(0) \\ \mu_s(1) \\ \mu_s(2) \end{bmatrix} \qquad \mu_{st}(x_s, x_t) = \begin{bmatrix} \mu_{st}(0,0) & \mu_{st}(0,1) & \mu_{st}(0,2) \\ \mu_{st}(1,0) & \mu_{st}(1,1) & \mu_{st}(1,2) \\ \mu_{st}(2,0) & \mu_{st}(2,1) & \mu_{st}(2,2) \end{bmatrix}$$

- define node-based entropy and edge-based mutual information:

$$\text{Node-based entropy:} H_s(\mu_s) = -\sum_{x_s} \mu_s(x_s) \log \mu_s(x_s)$$

$$\text{Mutual information:} I_{st}(\mu_{st}) = \sum_{x_s, x_t} \mu_{st}(x_s, x_t) \log \frac{\mu_{st}(x_s, x_t)}{\mu_s(x_s)\mu_t(x_t)}.$$

- ρ-reweighted Bethe entropy

$$H_{\text{Bethe}}(\mu) = \sum_{s \in V} H_s(\mu_s) - \sum_{(s,t) \in E} \rho_{st} \, I_{st}(\mu_{st}),$$

Bethe entropy is exact for trees

- exact for trees, using the factorization:

$$p(\mathbf{x};\theta) \quad = \quad \prod_{s \in V} \mu_s(x_s) \quad \prod_{(s,t) \in E} \frac{\mu_{st}(x_s, x_t)}{\mu_s(x_s)\mu_t(x_t)}$$

Reweighted sum-product and Bethe variational principle

Define the local constraint set

$$\mathbb{L}(G) = \left\{ \tau_s, \tau_{st} \mid \tau \geq 0, \quad \sum_{x_s} \tau_s(x_s) = 1, \quad \sum_{x_t} \tau_{st}(x_s, x_t) = \tau_s(x_s) \right\}$$

Reweighted sum-product and Bethe variational principle

Define the local constraint set

$$\mathbb{L}(G) = \Big\{\tau_s, \tau_{st} \mid \tau \geq 0, \quad \sum_{x_s} \tau_s(x_s) = 1, \quad \sum_{x_t} \tau_{st}(x_s, x_t) = \tau_s(x_s)\Big\}$$

Theorem

For any choice of positive edge weights $\rho_{st} > 0$:

(a) *Fixed points of reweighted sum-product are stationary points of the Lagrangian associated with*

$$A_{Bethe}(\theta; \rho) := \max_{\tau \in \mathbb{L}(G)} \Big\{ \sum_{s \in V} \langle \tau_s, \theta_s \rangle + \sum_{(s,t) \in E} \langle \tau_{st}, \theta_{st} \rangle + H_{Bethe}(\tau; \rho) \Big\}.$$

Reweighted sum-product and Bethe variational principle

Define the local constraint set

$$\mathbb{L}(G) = \{\tau_s, \tau_{st} \mid \tau \geq 0, \quad \sum_{x_s} \tau_s(x_s) = 1, \quad \sum_{x_t} \tau_{st}(x_s, x_t) = \tau_s(x_s)\}$$

Theorem

For any choice of positive edge weights $\rho_{st} > 0$:

(a) *Fixed points of reweighted sum-product are stationary points of the Lagrangian associated with*

$$A_{Bethe}(\theta; \rho) := \max_{\tau \in \mathbb{L}(G)} \Big\{ \sum_{s \in V} \langle \tau_s, \theta_s \rangle + \sum_{(s,t) \in E} \langle \tau_{st}, \theta_{st} \rangle + H_{Bethe}(\tau; \rho) \Big\}.$$

(b) *For valid choices of edge weights $\{\rho_{st}\}$, the fixed points are unique and moreover $\log Z(\theta) \leq A_{Bethe}(\theta; \rho)$. In addition, reweighted sum-product converges with appropriate scheduling.*

Lagrangian derivation of ordinary sum-product

- let's try to solve this problem by a (partial) Lagrangian formulation
- assign a Lagrange multiplier $\lambda_{ts}(x_s)$ for each constraint $C_{ts}(x_s) := \tau_s(x_s) - \sum_{x_t} \tau_{st}(x_s, x_t) = 0$
- will enforce the normalization ($\sum_{x_s} \tau_s(x_s) = 1$) and non-negativity constraints explicitly
- the Lagrangian takes the form:

$$\mathcal{L}(\tau; \lambda) = \langle \theta, \tau \rangle + \sum_{s \in V} H_s(\tau_s) - \sum_{(s,t) \in E(G)} I_{st}(\tau_{st}) + \sum_{(s,t) \in E} \Big[\sum_{x_t} \lambda_{st}(x_t) C_{st}(x_t) + \sum_{x_s} \lambda_{ts}(x_s) C_{ts}(x_s) \Big]$$

Lagrangian derivation (part II)

- taking derivatives of the Lagrangian w.r.t τ_s and τ_{st} yields

$$\frac{\partial \mathcal{L}}{\partial \tau_s(x_s)} = \theta_s(x_s) - \log \tau_s(x_s) + \sum_{t \in \mathcal{N}(s)} \lambda_{ts}(x_s) + C$$

$$\frac{\partial \mathcal{L}}{\partial \tau_{st}(x_s, x_t)} = \theta_{st}(x_s, x_t) - \log \frac{\tau_{st}(x_s, x_t)}{\tau_s(x_s)\tau_t(x_t)} - \lambda_{ts}(x_s) - \lambda_{st}(x_t) + C'$$

- setting these partial derivatives to zero and simplifying:

$$\tau_s(x_s) \propto \exp\{\theta_s(x_s)\} \prod_{t \in \mathcal{N}(s)} \exp\{\lambda_{ts}(x_s)\}$$

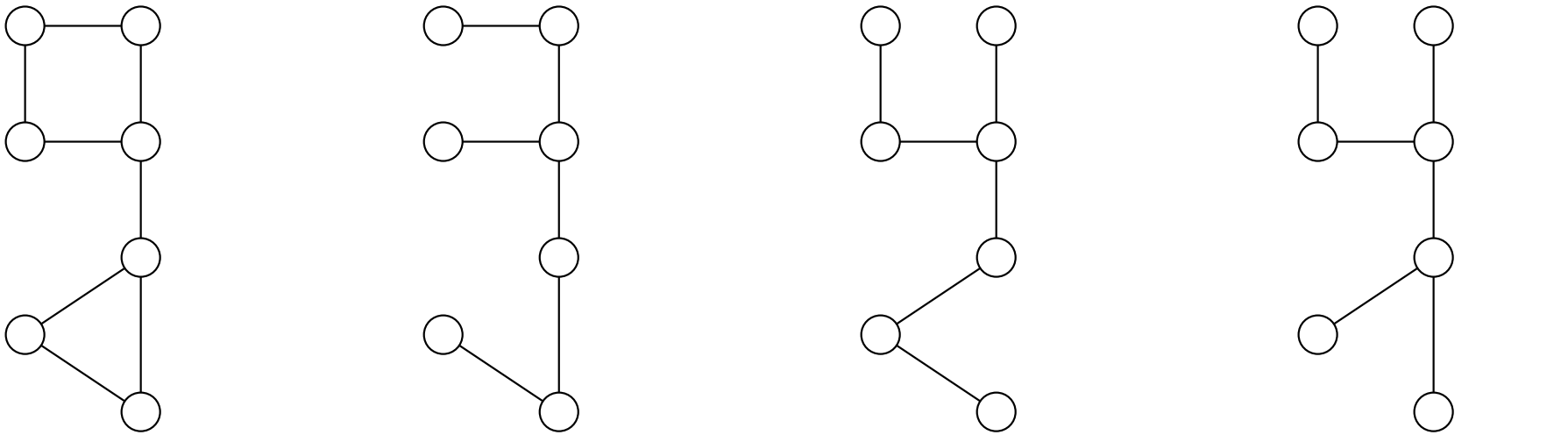
$$\tau_s(x_s, x_t) \propto \exp\{\theta_s(x_s) + \theta_t(x_t) + \theta_{st}(x_s, x_t)\} \times \prod_{u \in \mathcal{N}(s)\setminus t} \exp\{\lambda_{us}(x_s)\} \prod_{v \in \mathcal{N}(t)\setminus s} \exp\{\lambda_{vt}(x_t)\}$$

- enforcing the constraint $C_{ts}(x_s) = 0$ on these representations yields the familiar update rule for the *messages* $M_{ts}(x_s) = \exp(\lambda_{ts}(x_s))$:

$$M_{ts}(x_s) \leftarrow \sum_{x_t} \exp\{\theta_t(x_t) + \theta_{st}(x_s, x_t)\} \prod_{u \in \mathcal{N}(t)\setminus s} M_{ut}(x_t)$$

Convex combinations of trees

Idea: Upper bound $A(\theta) := \log Z(\theta)$ with a convex combination of tree-structured problems.



$$\begin{array}{ccccccc} \theta & = & \rho(T^1)\theta(T^1) & + & \rho(T^2)\theta(T^2) & + & \rho(T^3)\theta(T^3) \\ A(\theta) & \leq & \rho(T^1)A(\theta(T^1)) & + & \rho(T^2)A(\theta(T^2)) & + & \rho(T^3)A(\theta(T^3)) \end{array}$$

$$\begin{array}{lcl} \boldsymbol{\rho} = \{\rho(T)\} & \equiv & \text{probability distribution over spanning trees} \\ \theta(T) & \equiv & \text{tree-structured parameter vector} \end{array}$$

Finding the tightest upper bound

Observation: For each fixed distribution $\boldsymbol{\rho}$ over spanning trees, there are many such upper bounds.

Goal: Find the tightest such upper bound over all trees.

Challenge: Number of spanning trees grows rapidly in graph size.

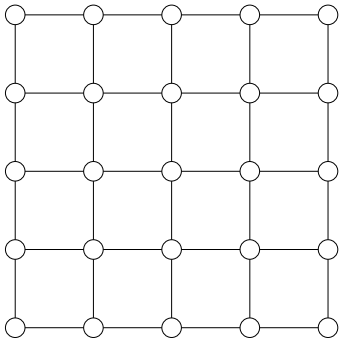
Finding the tightest upper bound

Observation: For each fixed distribution $\boldsymbol{\rho}$ over spanning trees, there are many such upper bounds.

Goal: Find the tightest such upper bound over all trees.

Challenge: Number of spanning trees grows rapidly in graph size.

Example:
On the 2-D lattice:



Grid size	# trees
9	192
16	100352
36	3.26×10^{13}
100	5.69×10^{42}

Finding the tightest upper bound

Observation: For each fixed distribution $\boldsymbol{\rho}$ over spanning trees, there are many such upper bounds.

Goal: Find the tightest such upper bound over all trees.

Challenge: Number of spanning trees grows rapidly in graph size.

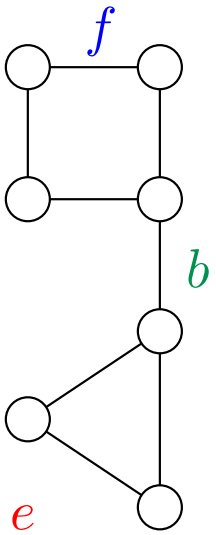
By a suitable dual reformulation, problem can be avoided:

Key duality relation:

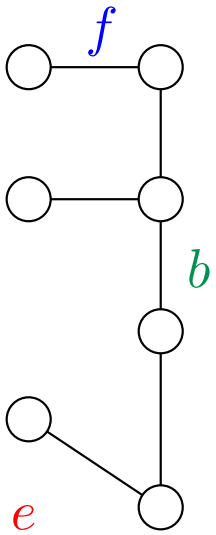
$$\min_{\sum_T \rho(T)\theta(T)=\theta} \rho(T)A(\theta(T)) = \max_{\mu \in \mathbb{L}(G)} \left\{ \langle \mu, \theta \rangle + H_{\text{Bethe}}(\mu; \rho_{st}) \right\}.$$

Edge appearance probabilities

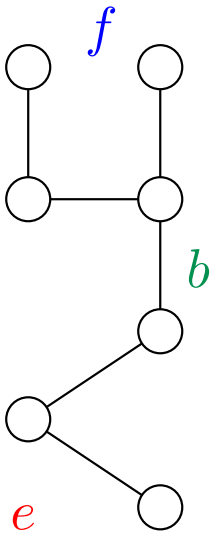
Experiment: What is the probability ρ_e that a given edge $e \in E$ belongs to a tree T drawn randomly under $\boldsymbol{\rho}$?



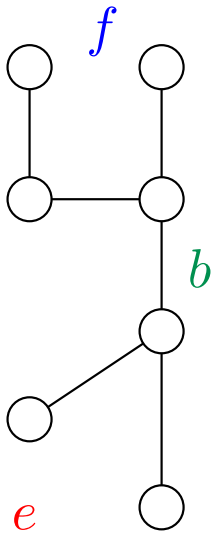
(a) Original



(b) $\rho(T^1) = \frac{1}{3}$



(c) $\rho(T^2) = \frac{1}{3}$



(d) $\rho(T^3) = \frac{1}{3}$

In this example: $\rho_b = 1$; $\rho_e = \frac{2}{3}$; $\rho_f = \frac{1}{3}$.

The vector $\boldsymbol{\rho_e} = \{ \rho_e \mid e \in E \}$ must belong to the *spanning tree polytope*.
(Edmonds, 1971)

Why does entropy arise in the duality?

Due to a deep correspondence between two problems:

Maximum entropy density estimation

$$\text{Maximize entropy} \qquad H(p) = -\sum_{\mathbf{x}} p(x_1, \ldots, x_N) \log p(x_1, \ldots, x_N)$$

subject to expectation constraints of the form

$$\sum_{\mathbf{x}} p(\mathbf{x}) \phi_\alpha(\mathbf{x}) = \widehat{\mu}_\alpha.$$

Why does entropy arise in the duality?

Due to a deep correspondence between two problems:

Maximum entropy density estimation

$$\text{Maximize entropy} \qquad H(p) = -\sum_{\mathbf{x}} p(x_1, \ldots, x_N) \log p(x_1, \ldots, x_N)$$

subject to expectation constraints of the form

$$\sum_{\mathbf{x}} p(\mathbf{x}) \phi_\alpha(\mathbf{x}) = \widehat{\mu}_\alpha.$$

Maximum likelihood in exponential family

Maximize likelihood of parameterized densities

$$p(x_1, \ldots, x_N; \theta) = \exp\Big\{ \sum_{\alpha} \theta_\alpha \phi_\alpha(x) - A(\theta) \Big\}.$$

Conjugate dual functions

- conjugate duality is a fertile source of variational representations
- any function f can be used to define another function f^* as follows:

$$f^*(v) \quad := \quad \sup_{u \in \mathbb{R}^n} \left\{ \langle v, u \rangle - f(u) \right\}.$$

- easy to show that f^* is always a convex function
- how about taking the "dual of the dual"? I.e., what is $(f^*)^*$?
- when f is well-behaved (convex and lower semi-continuous), we have $(f^*)^* = f$, or alternatively stated:

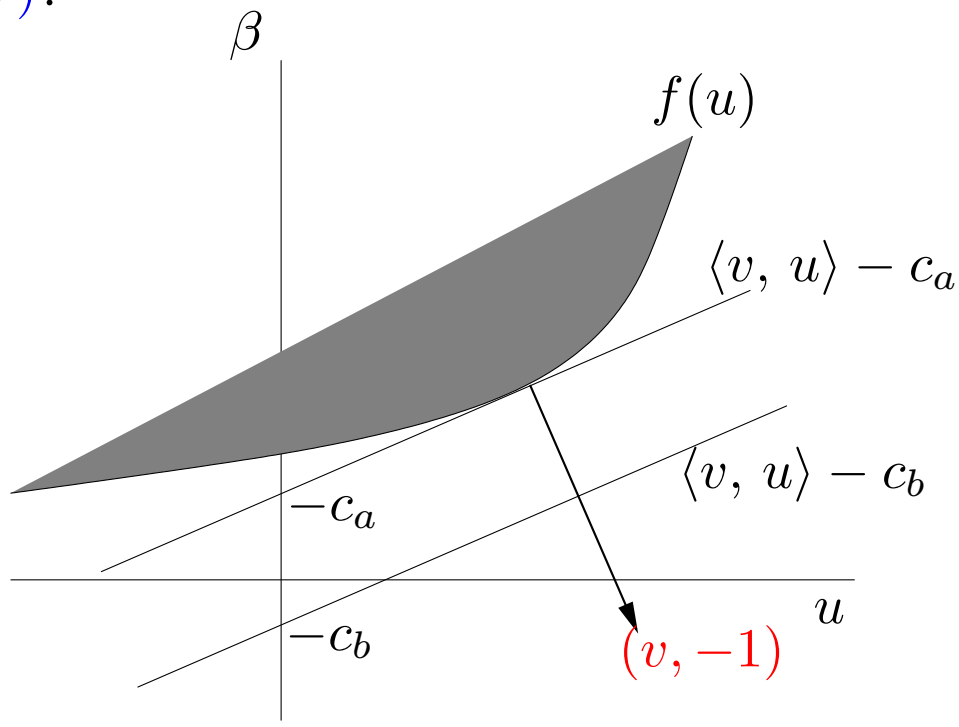
$$f(u) \quad = \quad \sup_{v \in \mathbb{R}^n} \left\{ \langle u, v \rangle - f^*(v) \right\}$$

Geometric view: Supporting hyperplanes

Question: Given all hyperplanes in $\mathbb{R}^n \times \mathbb{R}$ with normal $(v, -1)$, what is the intercept of the one that supports $\text{epi}(f)$?

Epigraph of f:

$\text{epi}(f) := \{(u, \beta) \in \mathbb{R}^{n+1} \mid f(u) \leq \beta\}$.



Analytically, we require the smallest $c \in \mathbb{R}$ such that:

$$\langle v,\, u\rangle - c \quad \leq \quad f(u) \quad \text{for all} \quad u \in \mathbb{R}^n$$

By re-arranging, we find that this optimal c^* is the dual value:

$$c^* \quad = \quad \sup_{u \in \mathbb{R}^n} \left\{\langle v,\, u\rangle - f(u)\right\}.$$

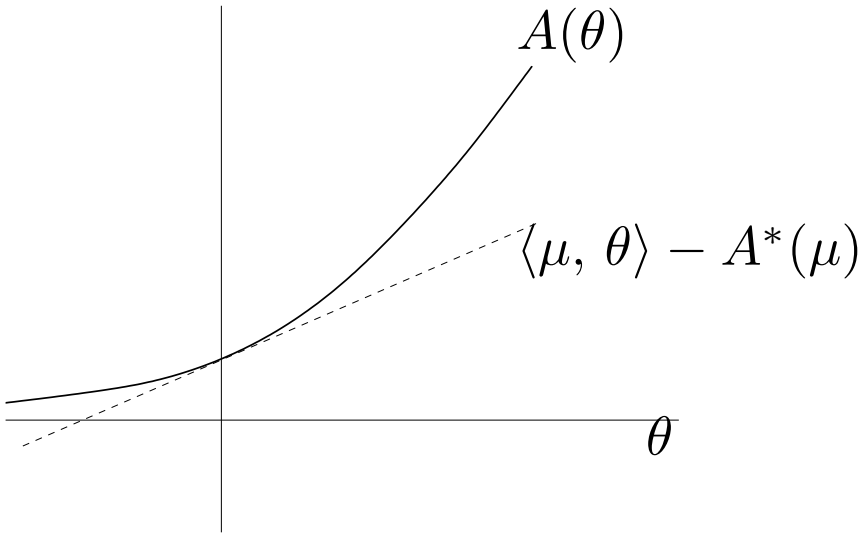
Example: Single Bernoulli

Random variable $X \in \{0, 1\}$ yields exponential family of the form:

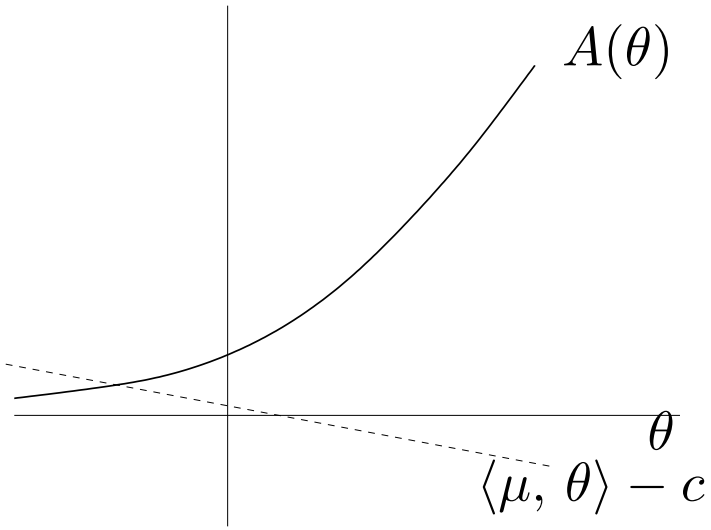
$$p(x; \theta) \propto \exp\{\theta\, x\} \quad \text{with} \quad A(\theta) = \log\left[1 + \exp(\theta)\right].$$

Let's compute the dual $A^*(\mu) := \sup_{\theta \in \mathbb{R}} \{\mu\theta - \log[1 + \exp(\theta)]\}$.

(Possible) stationary point: $\mu = \exp(\theta)/[1 + \exp(\theta)]$.



(a) Epigraph supported

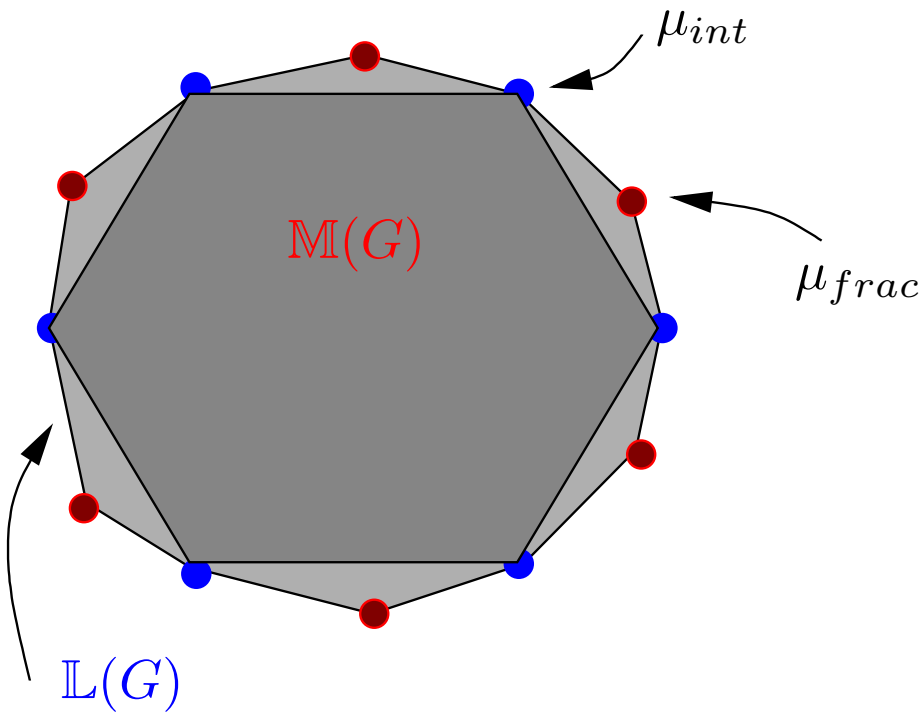


(b) Epigraph *cannot* be supported

We find that:
$$A^*(\mu) = \begin{cases} \mu \log \mu + (1 - \mu) \log(1 - \mu) & \text{if } \mu \in [0, 1] \\ +\infty & \text{otherwise.} \end{cases}.$$

Leads to the variational representation: $A(\theta) = \max_{\mu \in [0,1]} \{\mu \cdot \theta - A^*(\mu)\}$.

Geometry of Bethe variational problem



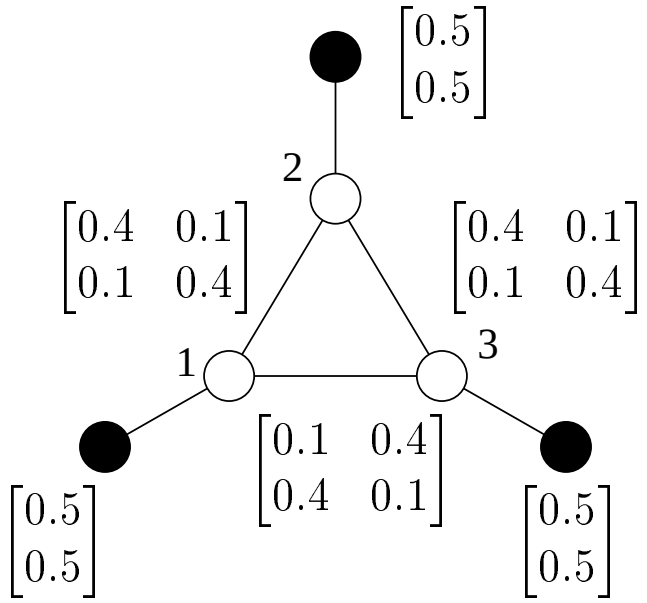
- belief propagation uses a *polyhedral outer approximation* to $\mathbb{M}(G)$:
  - for any graph, $\mathbb{L}(G) \supseteq \mathbb{M}(G)$.
  - equality holds $\iff$ G is a tree.

Natural question: Do BP fixed points ever fall outside of the marginal polytope $\mathbb{M}(G)$?

Illustration: Globally inconsistent BP fixed points

Consider the following assignment of pseudomarginals τ_s, τ_{st}:

Locally consistent (pseudo)marginals



- can verify that $\tau \in \mathbb{L}(G)$, and that τ is a fixed point of belief propagation (with all constant messages)
- however, τ is globally inconsistent

Note: More generally: for any τ in the interior of $\mathbb{L}(G)$, can construct a distribution with τ as a BP fixed point.

High-level perspective: A broad class of methods

- message-passing algorithms (e.g., mean field, belief propagation) are solving approximate versions of exact variational principle in exponential families
- there are two *distinct* components to approximations:
  - **(a)** can use either inner or outer bounds to $\mathbb{M}$
  - **(b)** various approximations to entropy function $-A^*(\mu)$

Refining one or both components yields better approximations:

- BP: polyhedral outer bound and non-convex Bethe approximation
- Kikuchi and variants: tighter polyhedral outer bounds and better entropy approximations (e.g.,Yedidia et al., 2002)
- Expectation-propagation: better outer bounds and Bethe-like entropy approximations (Minka, 2002)

Graphical models and message-passing: Part III: Learning graphs from data

Martin Wainwright

UC Berkeley
Departments of Statistics, and EECS

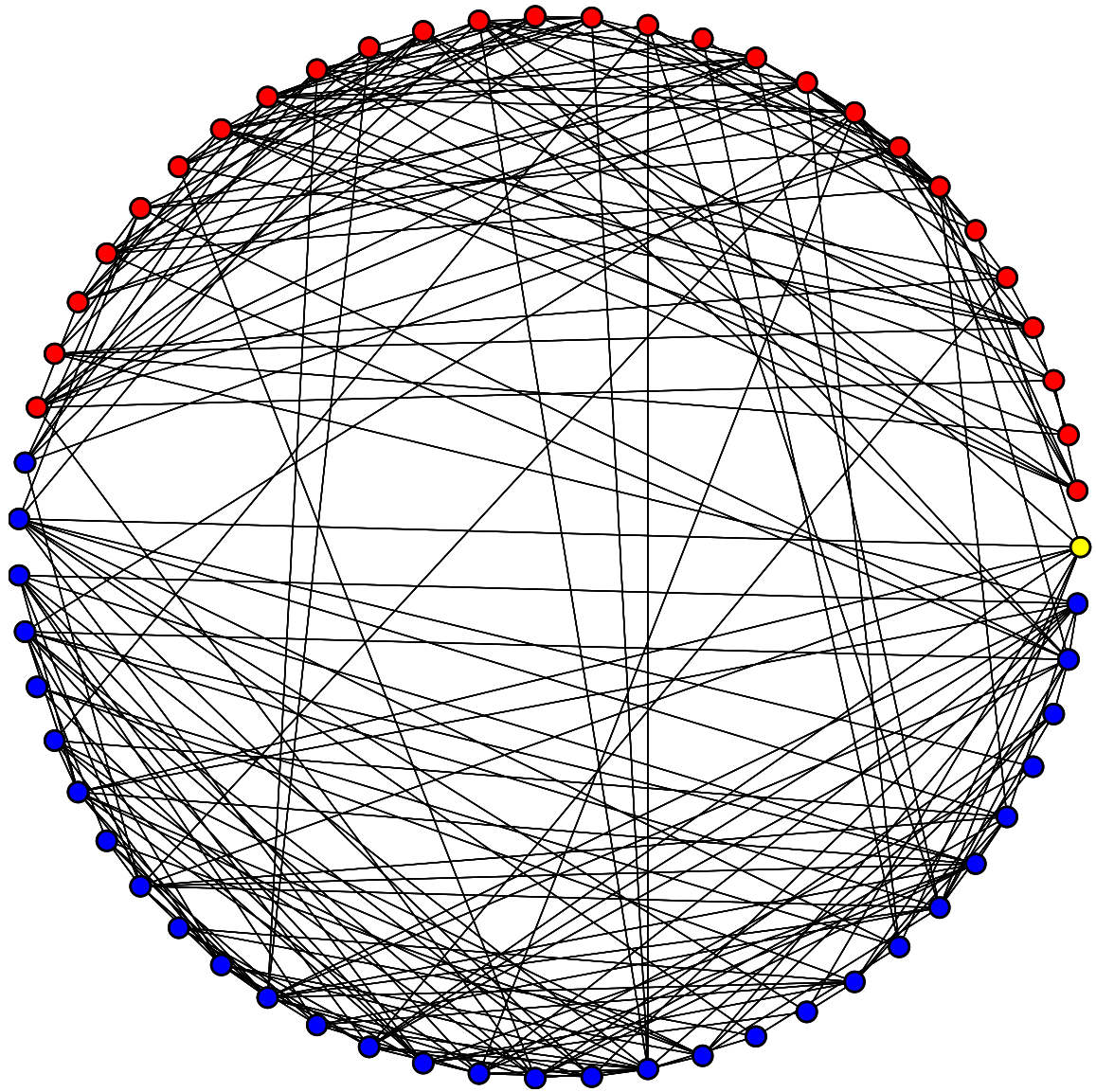
Introduction

- previous lectures on "forward problems": given a graphical model, perform some type of computation
  - Part I: compute most probable (MAP) assignment
  - Part II: compute marginals and likelihoods

- inverse problems concern learning the parameters and structure of graphs from data

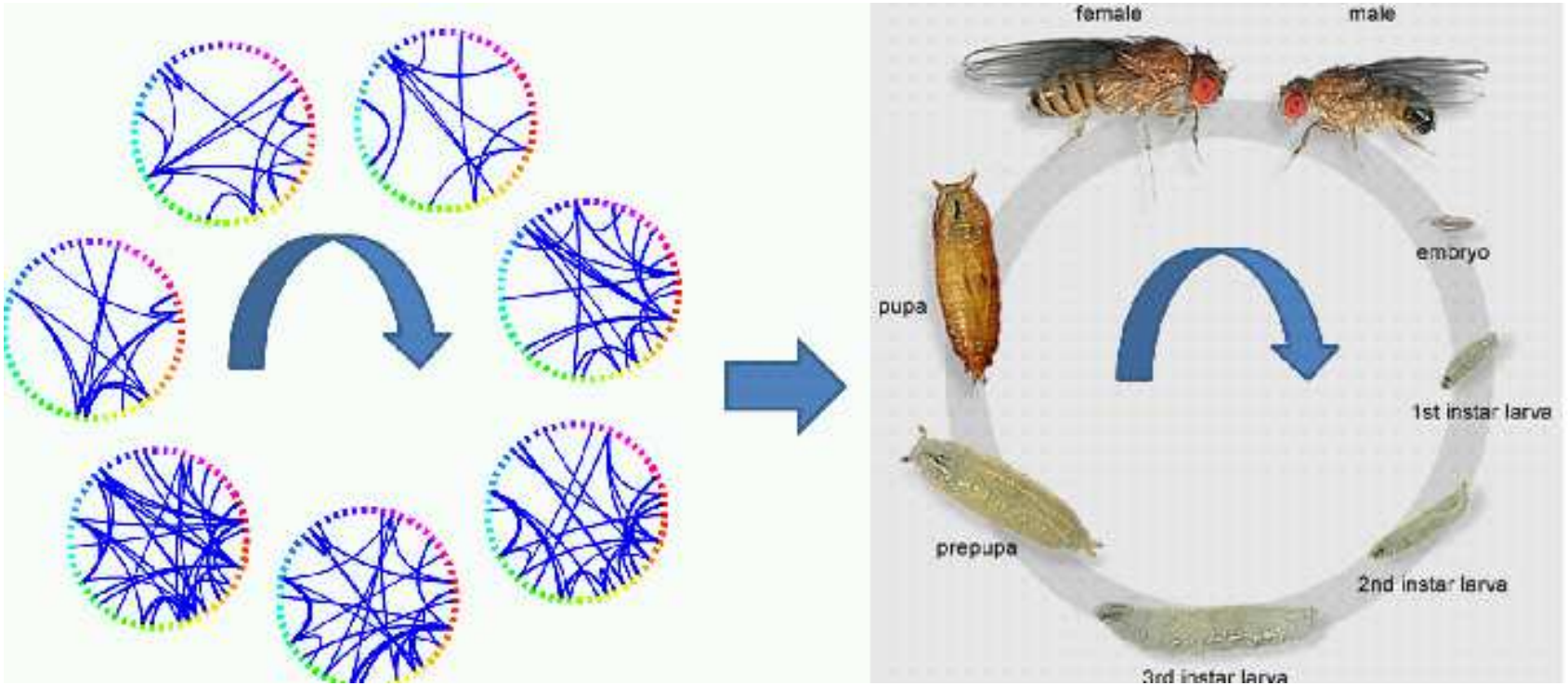
- many instances of such graph learning problems:
  - fitting graphs to politicians' voting behavior
  - modeling diseases with epidemiological networks
  - traffic flow modeling
  - interactions between different genes
  - and so on....

Example: US Senate network (2004–2006 voting)



(Banerjee et al., 2008; Ravikumar, W. & Lafferty, 2010)

Example: Biological networks



- gene networks during Drosophila life cycle (Ahmed & Xing, PNAS, 2009)
- many other examples:
  - protein networks
  - phylogenetic trees

Learning for pairwise models

- drawn n samples from

$$\mathbb{Q}(x_1, \ldots, x_p; \Theta) \quad = \quad \frac{1}{Z(\Theta)} \exp\Big\{ \sum_{s \in V} \theta_s x_s^2 + \sum_{(s,t) \in E} \theta_{st} x_s x_t \Big\}$$

- graph G and matrix $[\Theta]_{st} = \theta_{st}$ of edge weights are unknown

Learning for pairwise models

- drawn n samples from

$$\mathbb{Q}(x_1, \ldots, x_p; \Theta) \quad = \quad \frac{1}{Z(\Theta)} \exp\Big\{ \sum_{s \in V} \theta_s x_s^2 + \sum_{(s,t) \in E} \theta_{st} x_s x_t \Big\}$$

- graph G and matrix $[\Theta]_{st} = \theta_{st}$ of edge weights are unknown

- data matrix:
  - Ising model (binary variables): $\mathbf{X}_1^n \in \{0,1\}^{n \times p}$
  - Gaussian model: $\mathbf{X}_1^n \in \mathbb{R}^{n \times p}$

- estimator $\mathbf{X}_1^n \mapsto \widehat{\Theta}$

Learning for pairwise models

- drawn n samples from

$$\mathbb{Q}(x_1, \ldots, x_p; \Theta) \quad = \quad \frac{1}{Z(\Theta)} \exp\Big\{ \sum_{s \in V} \theta_s x_s^2 + \sum_{(s,t) \in E} \theta_{st} x_s x_t \Big\}$$

- graph G and matrix $[\Theta]_{st} = \theta_{st}$ of edge weights are unknown

- data matrix:
  - Ising model (binary variables): $\mathbf{X}_1^n \in \{0,1\}^{n \times p}$
  - Gaussian model: $\mathbf{X}_1^n \in \mathbb{R}^{n \times p}$

- estimator $\mathbf{X}_1^n \mapsto \widehat{\Theta}$

- various loss functions are possible:
  - graph selection: $\mathrm{supp}[\widehat{\Theta}] = \mathrm{supp}[\Theta]$?
  - bounds on Kullback-Leibler divergence $D(\mathbb{Q}_{\widehat{\Theta}} \,\|\, \mathbb{Q}_{\Theta})$
  - bounds on $\|\widehat{\Theta} - \Theta\|_{\mathrm{op}}$.

Challenges in graph selection

For pairwise models, negative log-likelihood takes form:

$$\begin{aligned}\ell(\Theta; \mathbf{X}_1^n) &:= -\frac{1}{n}\sum_{i=1}^{n} \log \mathbb{Q}(x_{i1}, \ldots, x_{ip}; \Theta) \\ &= \log Z(\Theta) - \sum_{s \in V} \theta_s \widehat{\mu}_s - \sum_{(s,t)} \theta_{st} \widehat{\mu}_{st}\end{aligned}$$

Challenges in graph selection

For pairwise models, negative log-likelihood takes form:

$$\ell(\Theta; \mathbf{X}_1^n) := -\frac{1}{n}\sum_{i=1}^{n} \log \mathbb{Q}(x_{i1}, \ldots, x_{ip}; \Theta)$$
$$= \log Z(\Theta) - \sum_{s \in V} \theta_s \widehat{\mu}_s - \sum_{(s,t)} \theta_{st} \widehat{\mu}_{st}$$

- maximizing likelihood involves computing $\log Z(\Theta)$ or its derivatives (marginals)
- for Gaussian graphical models, this is a log-determinant program
- for discrete graphical models, various work-arounds are possible:
  - Markov chain Monte Carlo and stochastic gradient
  - variational approximations to likelihood
  - pseudo-likelihoods

Methods for graph selection

- for Gaussian graphical models:
  - ℓ_1-regularized neighborhood regression for Gaussian MRFs
    (e.g., Meinshausen & Buhlmann, 2005; Wainwright, 2006, Zhao & Yu, 2006)
  - ℓ_1-regularized log-determinant (e.g., Yuan & Lin, 2006; d'Asprémont et al., 2007; Friedman, 2008; Rothman et al., 2008; Ravikumar et al., 2008)

Methods for graph selection

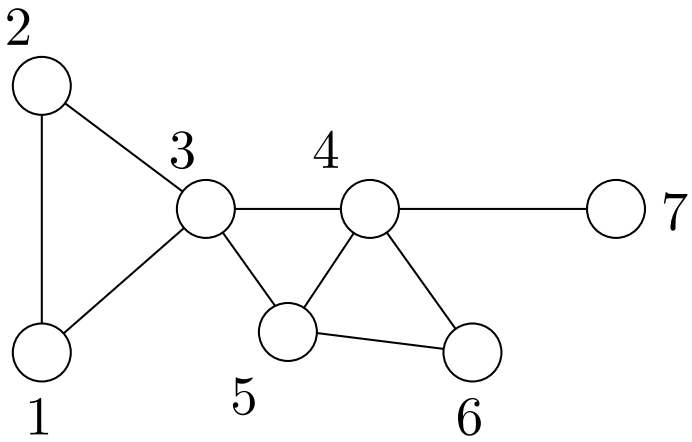
- for Gaussian graphical models:
  - ℓ_1-regularized neighborhood regression for Gaussian MRFs (e.g., Meinshausen & Buhlmann, 2005; Wainwright, 2006, Zhao & Yu, 2006)
  - ℓ_1-regularized log-determinant (e.g., Yuan & Lin, 2006; d'Asprémont et al., 2007; Friedman, 2008; Rothman et al., 2008; Ravikumar et al., 2008)
- methods for discrete MRFs
  - exact solution for trees (Chow & Liu, 1967)
  - local testing (e.g., Spirtes et al, 2000; Kalisch & Buhlmann, 2008)
  - various other methods
    - ⋆ distribution fits by KL-divergence (Abeel et al., 2005)
    - ⋆ ℓ_1-regularized log. regression (Ravikumar, W. & Lafferty et al., 2008, 2010)
    - ⋆ approximate max. entropy approach and thinned graphical models (Johnson et al., 2007)
    - ⋆ neighborhood-based thresholding method (Bresler, Mossel & Sly, 2008)

Methods for graph selection

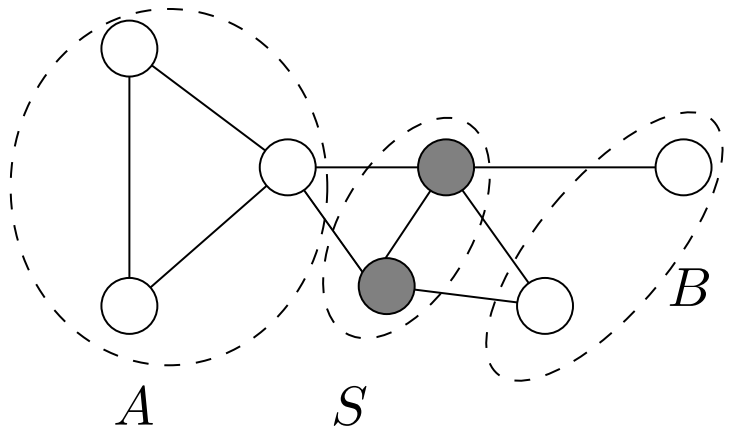
- for Gaussian graphical models:
  - ℓ_1-regularized neighborhood regression for Gaussian MRFs (e.g., Meinshausen & Buhlmann, 2005; Wainwright, 2006, Zhao & Yu, 2006)
  - ℓ_1-regularized log-determinant (e.g., Yuan & Lin, 2006; d'Asprémont et al., 2007; Friedman, 2008; Rothman et al., 2008; Ravikumar et al., 2008)
- methods for discrete MRFs
  - exact solution for trees (Chow & Liu, 1967)
  - local testing (e.g., Spirtes et al, 2000; Kalisch & Buhlmann, 2008)
  - various other methods
    - ⋆ distribution fits by KL-divergence (Abeel et al., 2005)
    - ⋆ ℓ_1-regularized log. regression (Ravikumar, W. & Lafferty et al., 2008, 2010)
    - ⋆ approximate max. entropy approach and thinned graphical models (Johnson et al., 2007)
    - ⋆ neighborhood-based thresholding method (Bresler, Mossel & Sly, 2008)
- information-theoretic analysis
  - pseudolikelihood and BIC criterion (Csiszar & Talata, 2006)
  - information-theoretic limitations (Santhanam & W., 2008, 2012)

Graphs and random variables

- associate to each node $s \in V$ a random variable X_s
- for each subset $A \subseteq V$, random vector $X_A := \{X_s, s \in A\}$.



Maximal cliques $(123), (345), (456), (47)$



Vertex cutset S

- a *clique* $C \subseteq V$ is a subset of vertices all joined by edges
- a *vertex cutset* is a subset $S \subset V$ whose removal breaks the graph into two or more pieces

Factorization and Markov properties

The graph G can be used to impose constraints on the random vector $X = X_V$ (or on the distribution $\mathbb{Q}$) in different ways.

Markov property: X is *Markov w.r.t G* if X_A and X_B are conditionally indpt. given X_S whenever S separates A and B.

Factorization: The distribution $\mathbb{Q}$ *factorizes according to G* if it can be expressed as a product over cliques:

$$\mathbb{Q}(x_1, x_2, \ldots, x_p) = \underbrace{\frac{1}{Z}}_{\text{Normalization}} \prod_{C \in \mathcal{C}} \underbrace{\psi_C(x_C)}_{\text{compatibility function on clique } C}$$

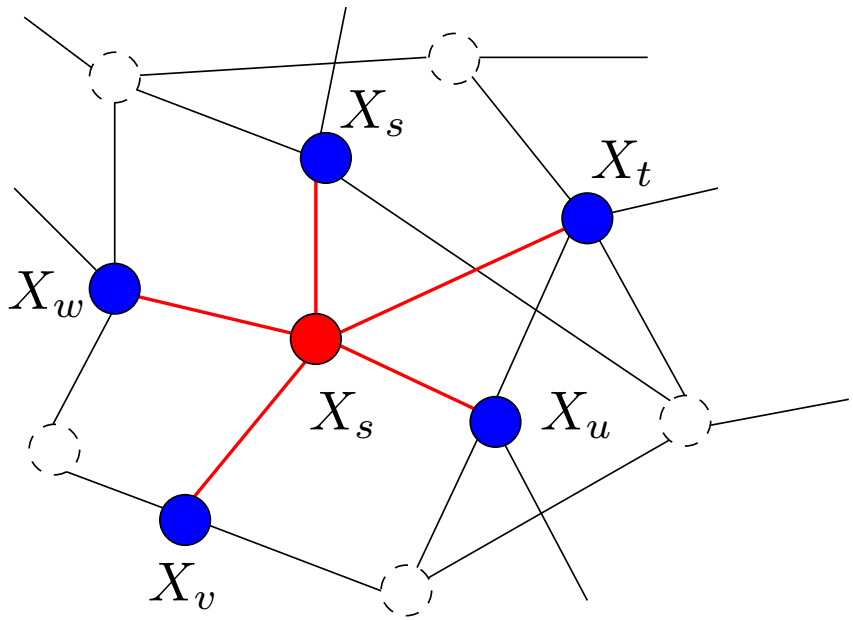
Theorem: (Hammersley & Clifford, 1973) For strictly positive $\mathbb{Q}(\cdot)$, the Markov property and the Factorization property are equivalent.

Markov property and neighborhood structure

- Markov properties encode neighborhood structure:

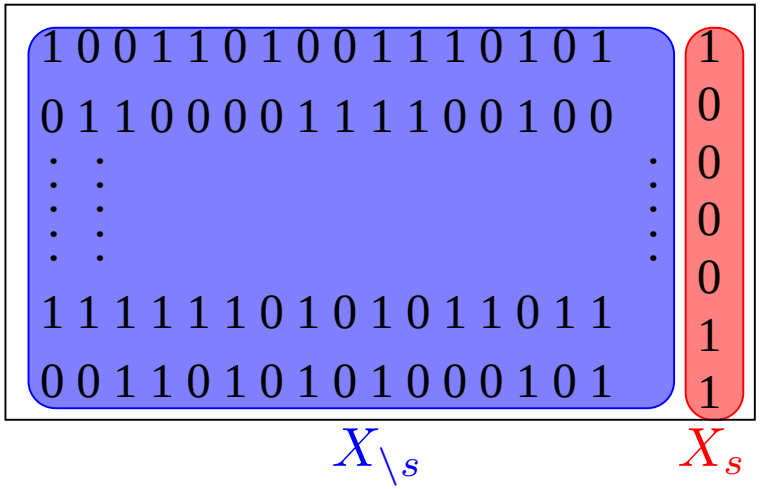
$$\underbrace{(X_s \quad | \quad X_{V \setminus s})}_{\text{Condition on full graph}} \stackrel{d}{=} \underbrace{(X_s \quad | \quad X_{N(s)})}_{\text{Condition on Markov blanket}}$$

$$N(s) = \{s, t, u, v, w\}$$



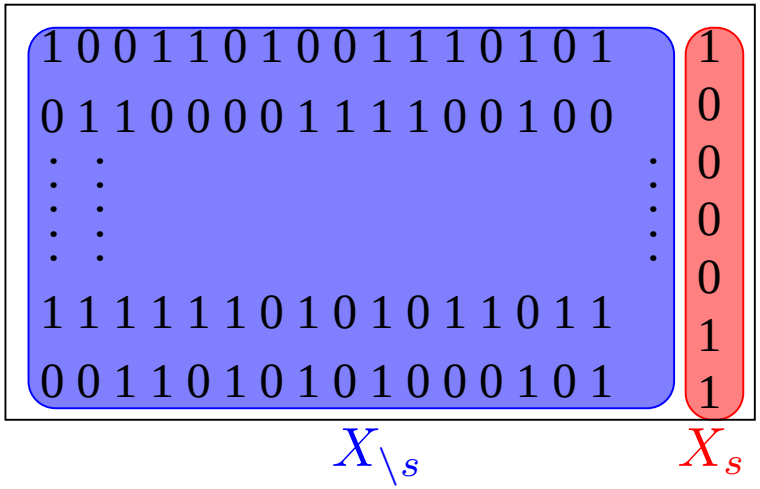
- basis of pseudolikelihood method (Besag, 1974)
- basis of many graph learning algorithms (Friedman et al., 1999; Csiszar & Talata, 2005; Abeel et al., 2006; Meinshausen & Buhlmann, 2006)

Graph selection via neighborhood regression



Predict X_s based on $X_{\setminus s} := \{X_s,\ t \neq s\}$.

Graph selection via neighborhood regression

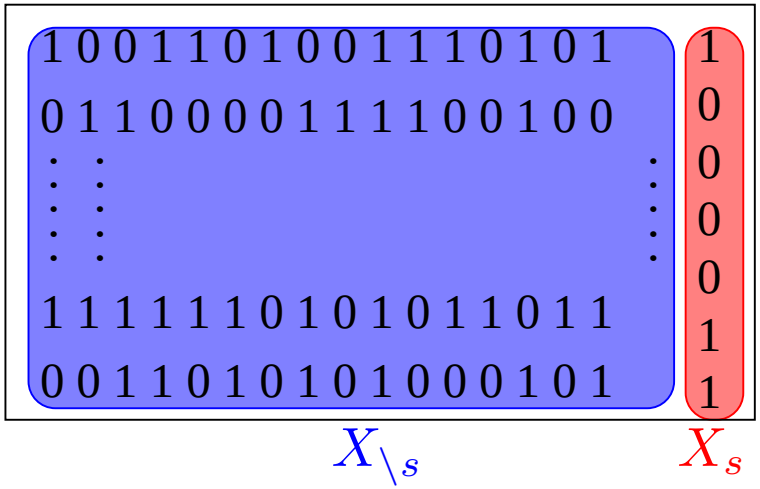


Predict X_s based on $X_{\setminus s} := \{X_s,\ t \neq s\}$.

1. For each node $s \in V$, compute (regularized) max. likelihood estimate:

$$\widehat{\theta}[s] \quad := \quad \arg \min_{\theta \in \mathbb{R}^{p-1}} \left\{ \quad \underbrace{-\frac{1}{n} \sum_{i=1}^{n} \mathcal{L}(\theta; X_{i,\setminus s})}_{\text{local log. likelihood}} \quad + \quad \underbrace{\lambda_n \|\theta\|_1}_{\text{regularization}} \right\}$$

Graph selection via neighborhood regression



Predict X_s based on $X_{\setminus s} := \{X_s,\ t \neq s\}$.

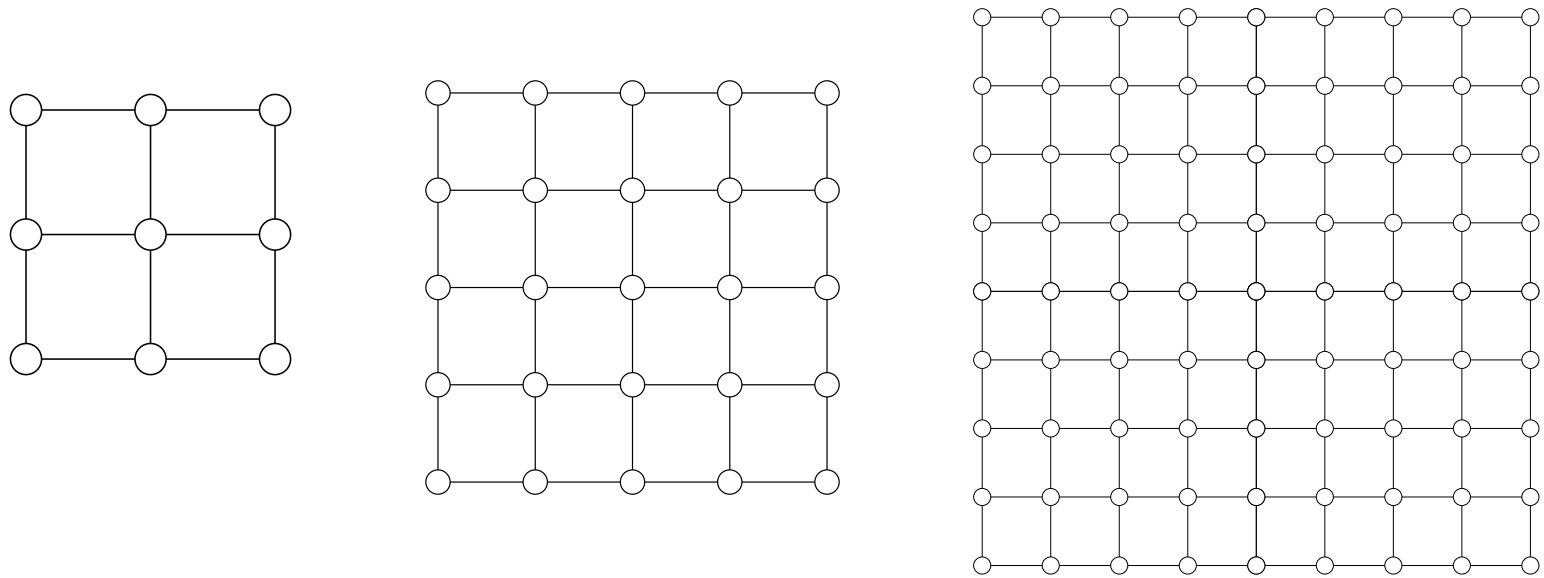
1. For each node $s \in V$, compute (regularized) max. likelihood estimate:

$$\widehat{\theta}[s] \quad := \quad \arg\min_{\theta \in \mathbb{R}^{p-1}} \left\{ \quad -\frac{1}{n} \sum_{i=1}^{n} \underbrace{\mathcal{L}(\theta; X_{i,\setminus s})}_{\text{local log. likelihood}} \quad + \quad \lambda_n \underbrace{\|\theta\|_1}_{\text{regularization}} \right\}$$

2. Estimate the local neighborhood $\widehat{N}(s)$ as support of regression vector $\widehat{\theta}[s] \in \mathbb{R}^{p-1}$.

High-dimensional analysis

- classical analysis: graph size p fixed, sample size $n \to +\infty$
- high-dimensional analysis: allow both dimension p, sample size n, and maximum degree d to increase at arbitrary rates

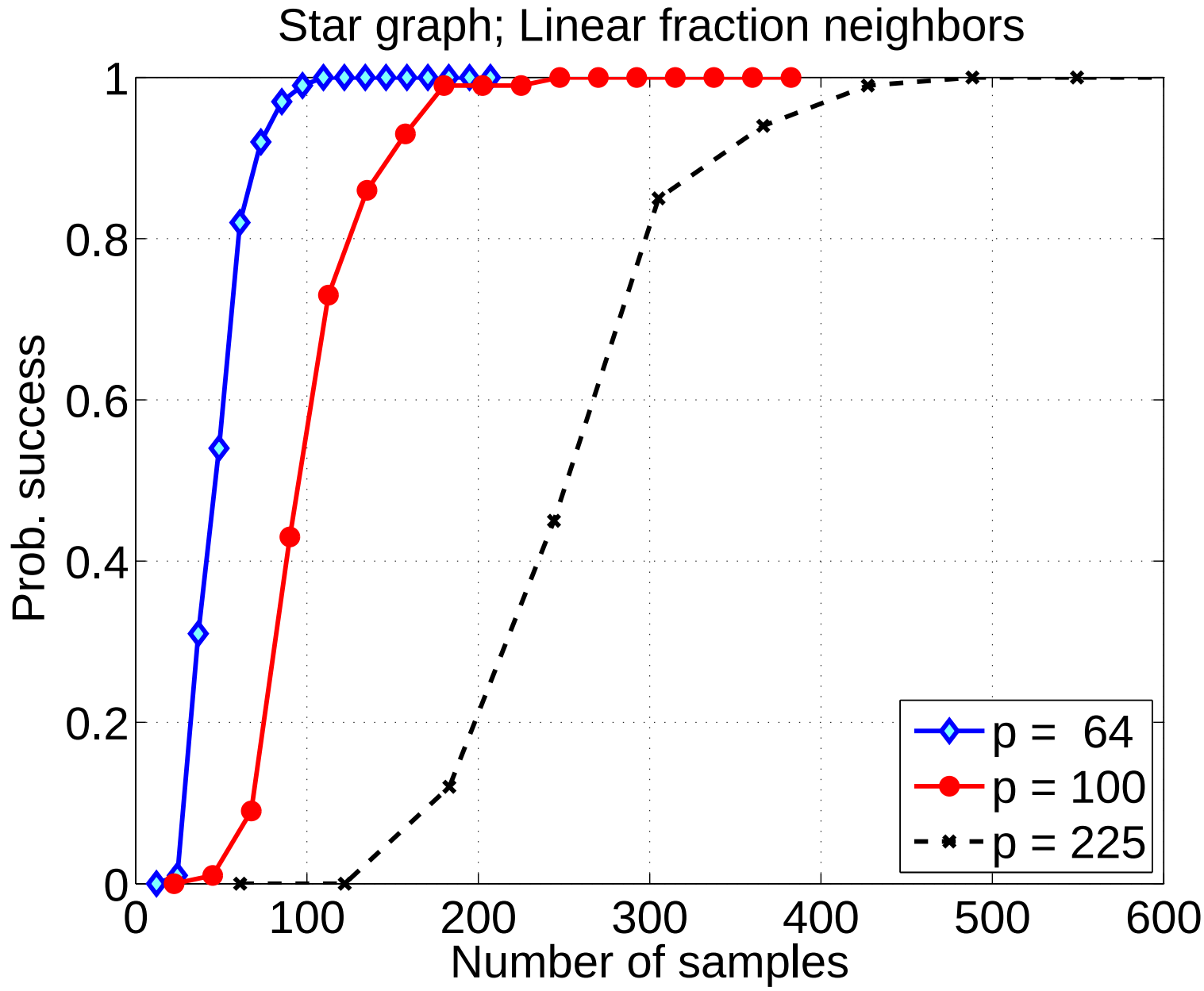


- take n i.i.d. samples from MRF defined by $G_{p,d}$
- study probability of success as a function of three parameters:

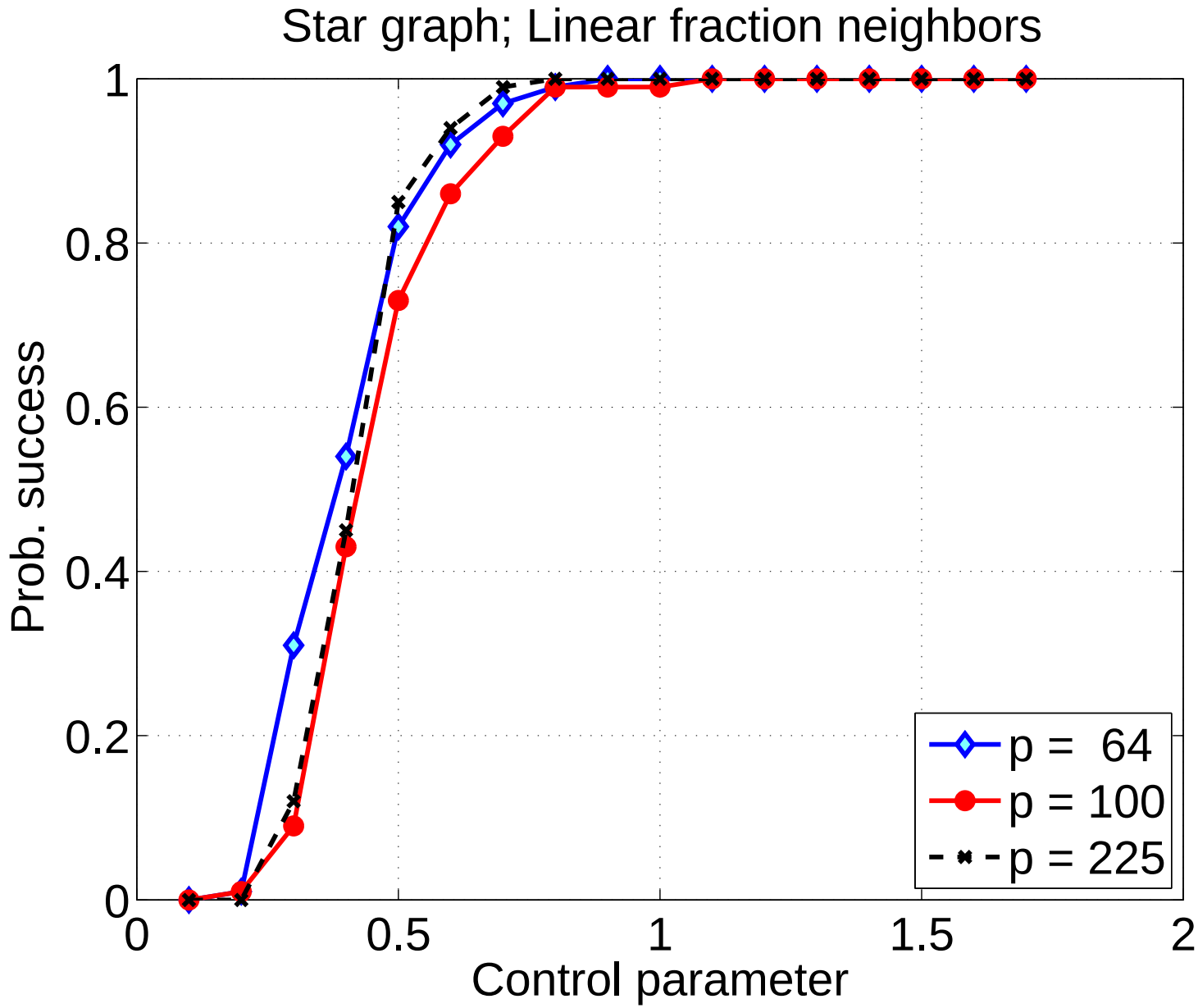
$$\text{Success}(n, p, d) \quad = \quad \mathbb{Q}[\text{Method recovers graph } G_{p,d} \text{ from } n \text{ samples}]$$

- theory is non-asymptotic: explicit probabilities for finite (n, p, d)

Empirical behavior: Unrescaled plots

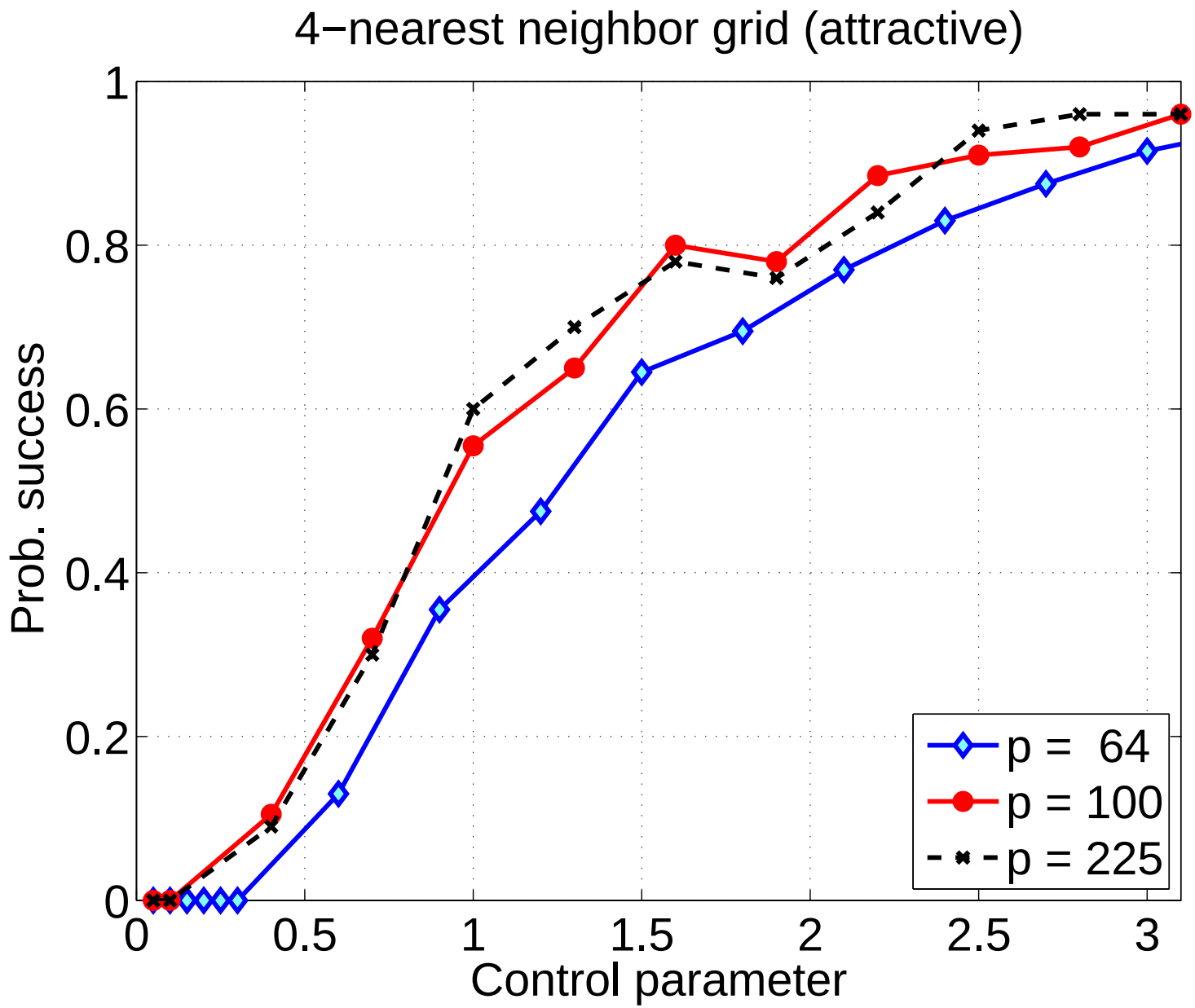


Empirical behavior: Appropriately rescaled



Plots of $\frac{\text{number}}{\text{probability}}$ versus control parameter $\frac{\alpha}{\beta}(n,p,k)$

Rescaled plots (2-D lattice graphs)



Sufficient conditions for consistent Ising selection

- graph sequences $G_{p,d} = (V, E)$ with p vertices, and maximum degree d.
- edge weights $|\theta_{st}| \geq \theta_{\min}$ for all $(s,t) \in E$
- draw n i.i.d, samples, and analyze prob. success indexed by (n, p, d)

Theorem (Ravikumar, W. & Lafferty, 2006, 2010)

Sufficient conditions for consistent Ising selection

- graph sequences $G_{p,d} = (V, E)$ with p vertices, and maximum degree d.
- edge weights $|\theta_{st}| \geq \theta_{\min}$ for all $(s,t) \in E$
- draw n i.i.d, samples, and analyze prob. success indexed by (n, p, d)

Theorem (Ravikumar, W. & Lafferty, 2006, 2010)

Under incoherence conditions, for a rescaled sample

$$\gamma_{LR}(n,p,d) \quad := \quad \frac{n}{d^3 \log p} \; > \; \gamma_{\text{crit}}$$

and regularization parameter $\lambda_n \geq c_1 \sqrt{\frac{\log p}{n}}$, *then with probability greater than* $1 - 2\exp\big(-c_2 \lambda_n^2 n\big)$*:*

(a) Correct exclusion: *The estimated sign neighborhood* $\widehat{N}(s)$ *correctly excludes all edges not in the true neighborhood.*

Sufficient conditions for consistent Ising selection

- graph sequences $G_{p,d} = (V, E)$ with p vertices, and maximum degree d.
- edge weights $|\theta_{st}| \geq \theta_{\min}$ for all $(s,t) \in E$
- draw n i.i.d, samples, and analyze prob. success indexed by (n, p, d)

Theorem (Ravikumar, W. & Lafferty, 2006, 2010)

Under incoherence conditions, for a rescaled sample

$$\gamma_{LR}(n,p,d) \quad := \quad \frac{n}{d^3 \log p} \; > \; \gamma_{\text{crit}}$$

and regularization parameter $\lambda_n \geq c_1 \sqrt{\frac{\log p}{n}}$, *then with probability greater than* $1 - 2\exp\big(-c_2 \lambda_n^2 n\big)$*:*

(a) Correct exclusion: *The estimated sign neighborhood* $\widehat{N}(s)$ *correctly excludes all edges not in the true neighborhood.*

(b) Correct inclusion: *For* $\theta_{\min} \geq c_3 \lambda_n$, *the method selects the correct signed neighborhood.*

Some related work

- thresholding estimator (poly-time for bounded degree) works with $n \gtrsim 2^d \log p$ samples (Bresler et al., 2008)

Some related work

- thresholding estimator (poly-time for bounded degree) works with $n \gtrsim 2^d \log p$ samples (Bresler et al., 2008)

- information-theoretic lower bound over family $\mathcal{G}_{p,d}$: any method requires at least $n = \Omega(d^2 \log p)$ samples (Santhanam & W., 2008)

Some related work

- thresholding estimator (poly-time for bounded degree) works with $n \gtrsim 2^d \log p$ samples (Bresler et al., 2008)

- information-theoretic lower bound over family $\mathcal{G}_{p,d}$: any method requires at least $n = \Omega(d^2 \log p)$ samples (Santhanam & W., 2008)

- ℓ_1-based method: sharper achievable rates, also failure for θ large enough to violate incoherence (Bento & Montanari, 2009)

Some related work

- thresholding estimator (poly-time for bounded degree) works with $n \gtrsim 2^d \log p$ samples (Bresler et al., 2008)

- information-theoretic lower bound over family $\mathcal{G}_{p,d}$: any method requires at least $n = \Omega(d^2 \log p)$ samples (Santhanam & W., 2008)

- ℓ_1-based method: sharper achievable rates, also failure for θ large enough to violate incoherence (Bento & Montanari, 2009)

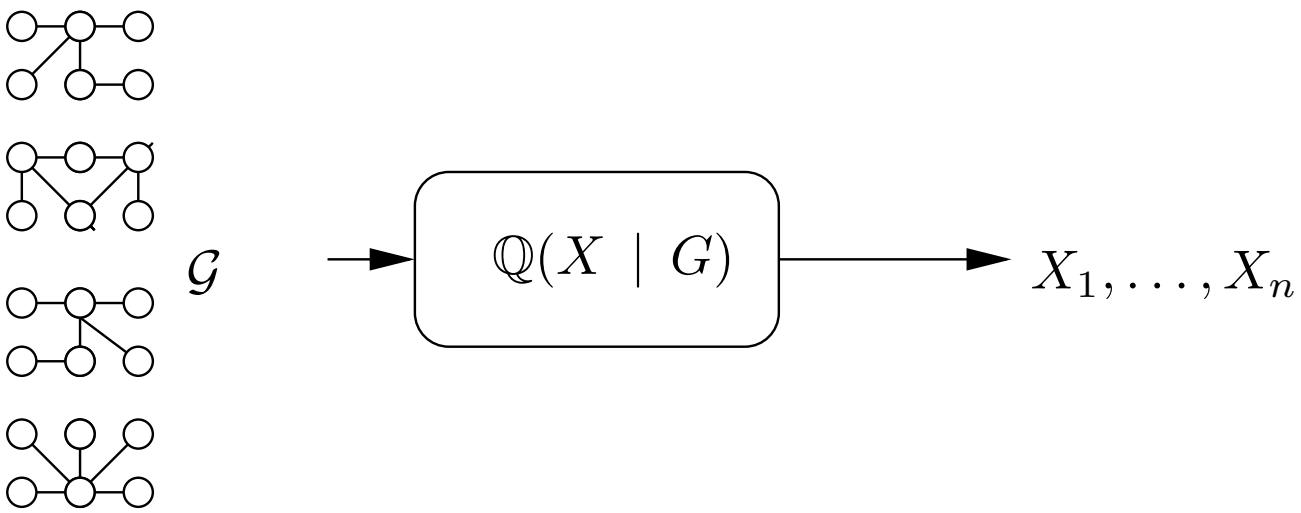
- empirical study: ℓ_1-based method can succeed beyond phase transition on Ising model (Aurell & Ekeberg, 2011)

§3. Info. theory: Graph selection as channel coding

- graphical model selection is an *unorthodox* channel coding problem:

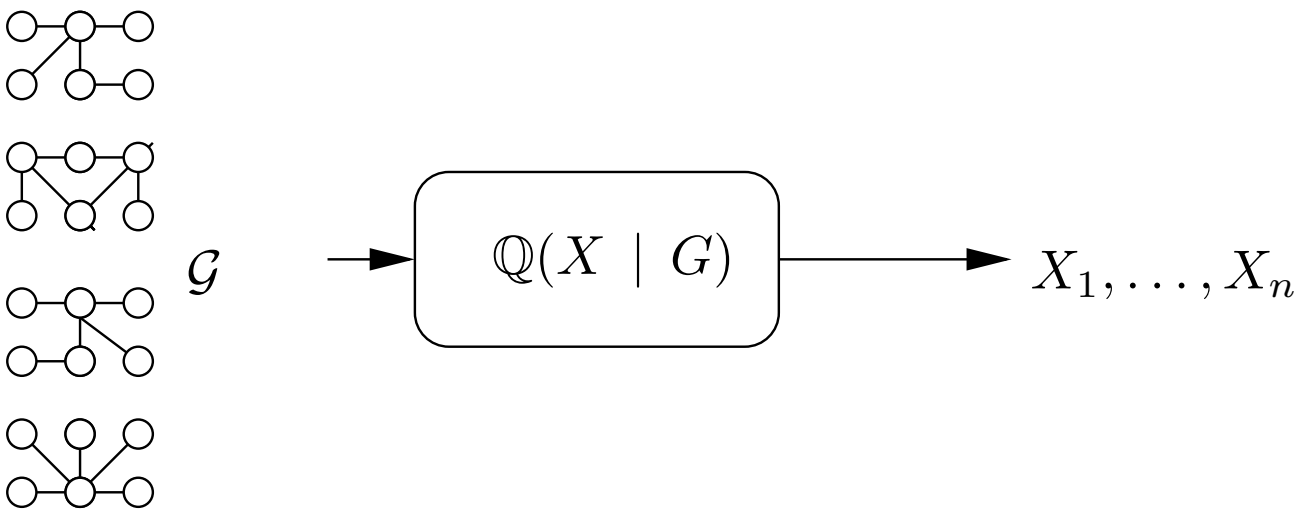
§3. Info. theory: Graph selection as channel coding

- graphical model selection is an *unorthodox* channel coding problem:
  - codewords/codebook: graph G in some graph class $\mathcal{G}$
  - channel use: draw sample $X_i = (X_{i1}, \ldots, X_{ip}$ from Markov random field $\mathbb{Q}_{\theta(G)}$
  - decoding problem: use n samples $\{X_1, \ldots, X_n\}$ to correctly distinguish the "codeword"



§3. Info. theory: Graph selection as channel coding

- graphical model selection is an *unorthodox* channel coding problem:
  - codewords/codebook: graph G in some graph class $\mathcal{G}$
  - channel use: draw sample $X_i = (X_{i1}, \ldots, X_{ip}$ from Markov random field $\mathbb{Q}_{\theta(G)}$
  - decoding problem: use n samples $\{X_1, \ldots, X_n\}$ to correctly distinguish the "codeword"



Channel capacity for graph decoding determined by balance between

- log number of models
- relative distinguishability of different models

Necessary conditions for $\mathcal{G}_{d,p}$

- $G \in \mathcal{G}_{d,p}$: graphs with p nodes and max. degree d
- Ising models with:
  - *Minimum edge weight:* $|\theta^*_{st}| \geq \theta_{\min}$ for all edges
  - *Maximum neighborhood weight:* $\omega(\theta) := \max_{s \in V} \sum_{t \in N(s)} |\theta^*_{st}|$

Necessary conditions for $\mathcal{G}_{d,p}$

- $G \in \mathcal{G}_{d,p}$: graphs with p nodes and max. degree d
- Ising models with:
  - *Minimum edge weight:* $|\theta^*_{st}| \geq \theta_{\min}$ for all edges
  - *Maximum neighborhood weight:* $\omega(\theta) := \max_{s \in V} \sum_{t \in N(s)} |\theta^*_{st}|$

Theorem

If the sample size n is upper bounded by *(Santhanam & W, 2008)*

$$n \quad < \quad \max \left\{ \frac{d}{8} \log \frac{p}{8d}, \; \frac{\exp(\frac{\omega(\theta)}{4}) \, d\theta_{\min} \log(pd/8)}{128 \exp(\frac{3\theta_{\min}}{2})}, \; \frac{\log p}{2\theta_{\min} \tanh(\theta_{\min})} \right\}$$

then the probability of error of any algorithm over $\mathcal{G}_{d,p}$ is at least $1/2$.

Necessary conditions for $\mathcal{G}_{d,p}$

- $G \in \mathcal{G}_{d,p}$: graphs with p nodes and max. degree d
- Ising models with:
  - *Minimum edge weight:* $|\theta^*_{st}| \geq \theta_{\min}$ for all edges
  - *Maximum neighborhood weight:* $\omega(\theta) := \max_{s \in V} \sum_{t \in N(s)} |\theta^*_{st}|$

Theorem

If the sample size n is upper bounded by *(Santhanam & W, 2008)*

$$n \quad < \quad \max \left\{ \frac{d}{8} \log \frac{p}{8d}, \; \frac{\exp(\frac{\omega(\theta)}{4}) \, d\theta_{\min} \log(pd/8)}{128 \exp(\frac{3\theta_{\min}}{2})}, \; \frac{\log p}{2\theta_{\min} \tanh(\theta_{\min})} \right\}$$

then the probability of error of any algorithm over $\mathcal{G}_{d,p}$ is at least $1/2$.

Interpretation:

- Naive bulk effect: Arises from log cardinality $\log |\mathcal{G}_{d,p}|$
- d-clique effect: Difficulty of separating models that contain a near d-clique
- Small weight effect: Difficult to detect edges with small weights.

Some consequences

Corollary

For asymptotically reliable recovery over $\mathcal{G}_{d,p}$, any algorithm requires at least $n = \Omega(d^2 \log p)$ samples.

Some consequences

Corollary

For asymptotically reliable recovery over $\mathcal{G}_{d,p}$, any algorithm requires at least $n = \Omega(d^2 \log p)$ samples.

- note that maximum neighborhood weight $\omega(\theta^*) \geq d\,\theta_{\min} \Longrightarrow$ require $\theta_{\min} = \mathcal{O}(1/d)$

Some consequences

Corollary

For asymptotically reliable recovery over $\mathcal{G}_{d,p}$, any algorithm requires at least $n = \Omega(d^2 \log p)$ samples.

- note that maximum neighborhood weight $\omega(\theta^*) \geq d\,\theta_{\min} \Longrightarrow$ require $\theta_{\min} = \mathcal{O}(1/d)$
- from small weight effect

$$n = \Omega\Big(\frac{\log p}{\theta_{\min} \tanh(\theta_{\min})}\Big) \;=\; \Omega\Big(\frac{\log p}{\theta^2_{\min}}\Big)$$

Some consequences

Corollary

For asymptotically reliable recovery over $\mathcal{G}_{d,p}$, any algorithm requires at least $n = \Omega(d^2 \log p)$ samples.

- note that maximum neighborhood weight $\omega(\theta^*) \geq d\,\theta_{\min} \Longrightarrow$ require $\theta_{\min} = \mathcal{O}(1/d)$
- from small weight effect

$$n = \Omega\Big(\frac{\log p}{\theta_{\min} \tanh(\theta_{\min})}\Big) \;=\; \Omega\Big(\frac{\log p}{\theta^2_{\min}}\Big)$$

- conclude that ℓ_1-regularized logistic regression (LR) is optimal up to a factor $\mathcal{O}(d)$ (Ravikumar., W. & Lafferty, 2010)

Proof sketch: Main ideas for necessary conditions

- based on assessing difficulty of graph selection over various sub-ensembles $\mathcal{G} \subseteq \mathcal{G}_{p,d}$

Proof sketch: Main ideas for necessary conditions

- based on assessing difficulty of graph selection over various sub-ensembles $\mathcal{G} \subseteq \mathcal{G}_{p,d}$

- choose $G \in \mathcal{G}$ u.a.r., and consider multi-way hypothesis testing problem based on the data $\mathbf{X}_1^n = \{X_1, \ldots, X_n\}$

Proof sketch: Main ideas for necessary conditions

- based on assessing difficulty of graph selection over various sub-ensembles $\mathcal{G} \subseteq \mathcal{G}_{p,d}$

- choose $G \in \mathcal{G}$ u.a.r., and consider multi-way hypothesis testing problem based on the data $\mathbf{X}_1^n = \{X_1, \ldots, X_n\}$

- for any graph estimator $\psi : \mathcal{X}^n \to \mathcal{G}$, Fano's inequality implies that

$$\mathbb{Q}[\psi(\mathbf{X}_1^n) \neq G] \geq 1 - \frac{I(\mathbf{X}_1^n; G) + \log 2}{\log |\mathcal{G}|}$$

  where $I(\mathbf{X}_1^n; G)$ is mutual information between observations $\mathbf{X}_1^n$ and randomly chosen graph G

Proof sketch: Main ideas for necessary conditions

- based on assessing difficulty of graph selection over various sub-ensembles $\mathcal{G} \subseteq \mathcal{G}_{p,d}$
- choose $G \in \mathcal{G}$ u.a.r., and consider multi-way hypothesis testing problem based on the data $\mathbf{X}_1^n = \{X_1, \ldots, X_n\}$
- for any graph estimator $\psi : \mathcal{X}^n \to \mathcal{G}$, Fano's inequality implies that

$$\mathbb{Q}[\psi(\mathbf{X}_1^n) \neq G] \geq 1 - \frac{I(\mathbf{X}_1^n; G) + \log 2}{\log |\mathcal{G}|}$$

  where $I(\mathbf{X}_1^n; G)$ is mutual information between observations $\mathbf{X}_1^n$ and randomly chosen graph G

- remaining steps:

  1. Construct "difficult" sub-ensembles $\mathcal{G} \subseteq \mathcal{G}_{p,d}$
  2. Compute or lower bound the log cardinality $\log |\mathcal{G}|$.
  3. Upper bound the mutual information $I(\mathbf{X}_1^n; G)$.

Summary

- simple ℓ_1-regularized neighborhood selection:
  - polynomial-time method for learning neighborhood structure
  - natural extensions (using block regularization) to higher order models

- information-theoretic limits of graph learning

Some papers:

- Ravikumar, W. & Lafferty (2010). High-dimensional Ising model selection using ℓ_1-regularized logistic regression. *Annals of Statistics*.
- Santhanam & W (2012). Information-theoretic limits of selecting binary graphical models in high dimensions, *IEEE Transactions on Information Theory*.

Two straightforward ensembles

Two straightforward ensembles

1. Naive bulk ensemble: All graphs on p vertices with max. degree d (i.e., $\mathcal{G} = \mathcal{G}_{p,d}$)

Two straightforward ensembles

1. Naive bulk ensemble: All graphs on p vertices with max. degree d (i.e., $\mathcal{G} = \mathcal{G}_{p,d}$)
   - simple counting argument: $\log |\mathcal{G}_{p,d}| = \Theta\big(pd \log(p/d)\big)$
   - trivial upper bound: $I(\mathbf{X}_1^n; G) \leq H(\mathbf{X}_1^n) \leq np$.
   - substituting into Fano yields necessary condition $n = \Omega(d \log(p/d))$
   - this bound independently derived by different approach by Bresler et al. (2008)

Two straightforward ensembles

1. Naive bulk ensemble: All graphs on p vertices with max. degree d (i.e., $\mathcal{G} = \mathcal{G}_{p,d}$)
   - simple counting argument: $\log|\mathcal{G}_{p,d}| = \Theta\big(pd\log(p/d)\big)$
   - trivial upper bound: $I(\mathbf{X}_1^n; G) \leq H(\mathbf{X}_1^n) \leq np$.
   - substituting into Fano yields necessary condition $n = \Omega(d\log(p/d))$
   - this bound independently derived by different approach by Bresler et al. (2008)

2. Small weight effect: Ensemble $\mathcal{G}$ consisting of graphs with a single edge with weight $\theta = \theta_{\min}$

Two straightforward ensembles

1. Naive bulk ensemble: All graphs on p vertices with max. degree d (i.e., $\mathcal{G} = \mathcal{G}_{p,d}$)
   - simple counting argument: $\log|\mathcal{G}_{p,d}| = \Theta\big(pd\log(p/d)\big)$
   - trivial upper bound: $I(\mathbf{X}_1^n; G) \le H(\mathbf{X}_1^n) \le np$.
   - substituting into Fano yields necessary condition $n = \Omega(d\log(p/d))$
   - this bound independently derived by different approach by Bresler et al. (2008)

2. Small weight effect: Ensemble $\mathcal{G}$ consisting of graphs with a single edge with weight $\theta = \theta_{\min}$
   - simple counting: $\log|\mathcal{G}| = \log\binom{p}{2}$
   - upper bound on mutual information:

$$I(\mathbf{X}_1^n; G) \le \frac{1}{\binom{p}{2}} \sum_{(i,j),(k,\ell)\in E} D\big(\theta(G^{ij})\|\theta(G^{k\ell})\big).$$

   - upper bound on symmetrized Kullback-Leibler divergences:

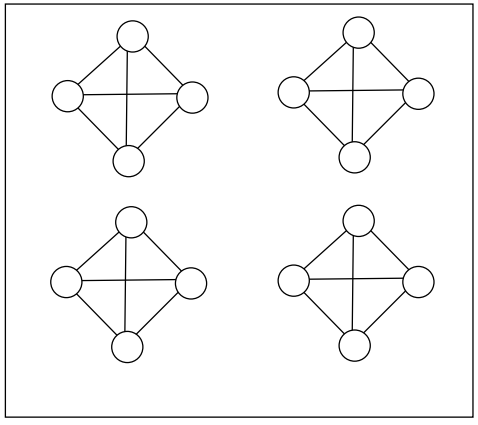
$$D\big(\theta(G^{ij})\|\theta(G^{k\ell})\big) + D\big(\theta(G^{k\ell})\|\theta(G^{ij})\big) \le 2\theta_{\min}\tanh(\theta_{\min}/2)$$

   - substituting into Fano yields necessary condition $n = \Omega\big(\frac{\log p}{\theta_{\min}\tanh(\theta_{\min}/2)}\big)$

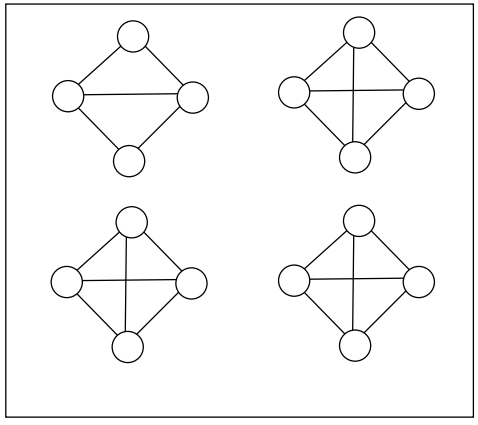
A harder d-clique ensemble

Constructive procedure:

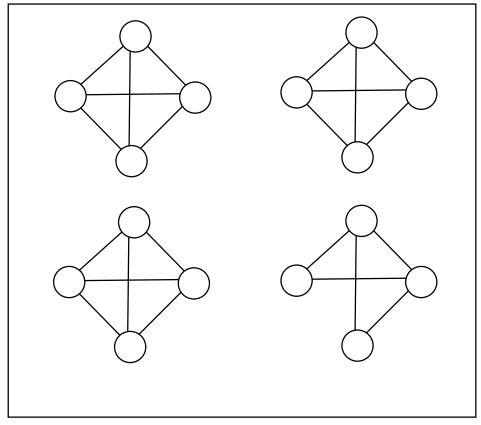
1. Divide the vertex set V into $\lfloor \frac{p}{d+1} \rfloor$ groups of size $d+1$.
2. Form the base graph $\overline{G}$ by making a $(d+1)$-clique within each group.
3. Form graph G^{uv} by deleting edge (u, v) from $\overline{G}$.
4. Form Markov random field $\mathbb{Q}_{\theta(G^{uv})}$ by setting $\theta_{st} = \theta_{\min}$ for all edges.



(a) Base graph $\overline{G}$



(b) Graph G^{uv}



(c) Graph G^{st}

- For $d \leq p/4$, we can form

$$|\mathcal{G}| \geq \lfloor \frac{p}{d+1} \rfloor \binom{d+1}{2} = \Omega(dp)$$

such graphs.