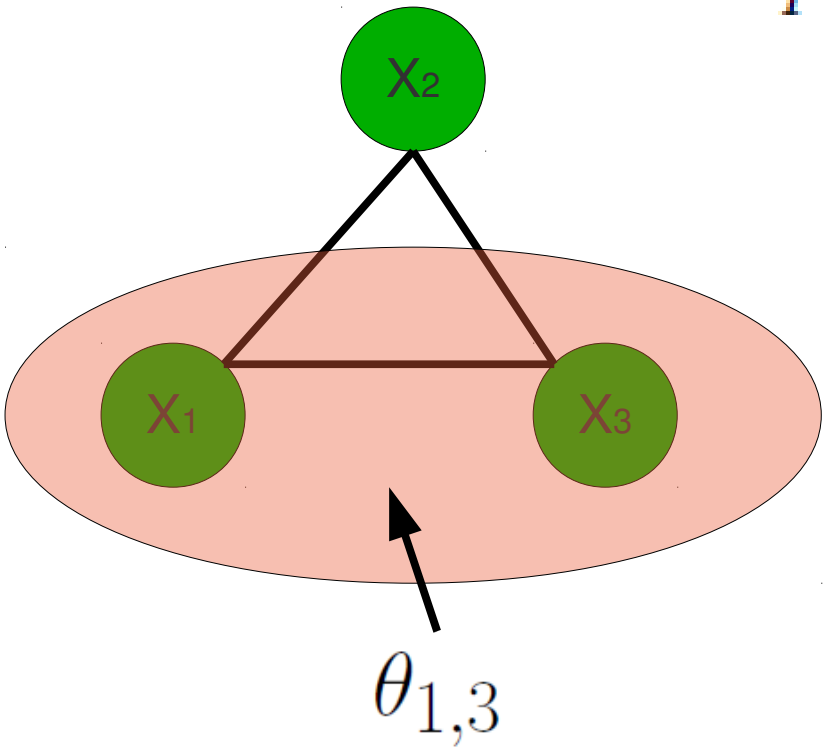
Akshat Kumar & Shlomo Zilberstein.

MAP Estimation for graphical Models by Likelihood Maximization.

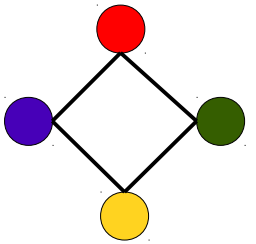
MAP problem for MRF



$$p(\boldsymbol{x};\boldsymbol{\theta}) = \frac{1}{Z} e^{\sum_{ij \in E} \theta_{ij}(x_i, x_j)}$$

Task: find most probable assignments to all variables = MPE

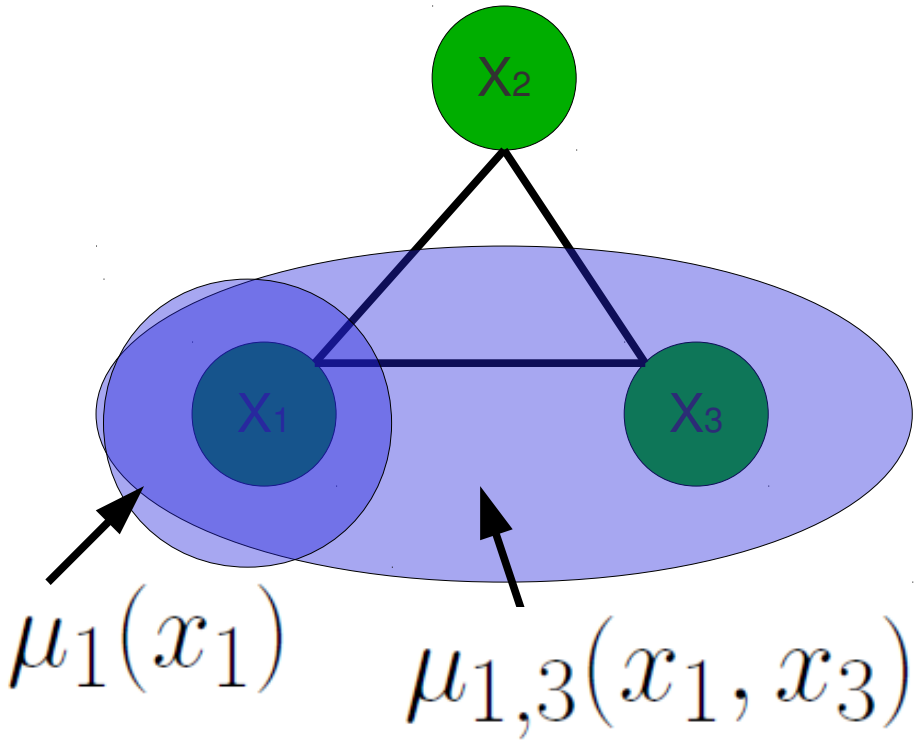
Paper in a nutshell



- We want to solve MAP problem for Markov Random Fields
- Can formulate as Linear Programming problem
  - Associate a marginal polytop with the MAP problem
  - Exponentially large constraints – difficult to solve
- Well-known LP relaxations to find the upper bound exist
  - Outer bound on the marginal polytop
- Authors propose relaxation to get a **lower-bound**
  - Inner bound on the marginal polytop
  - Not convex – formulate as a **mixture of BN**, solve with **EM**
- → Get fast and pretty accurate approximation method

MAP problem for MRF as LP

$$p(\boldsymbol{x};\boldsymbol{\theta}) = \frac{1}{Z} e^{\sum_{ij\in E} \theta_{ij}(x_i,x_j)}$$

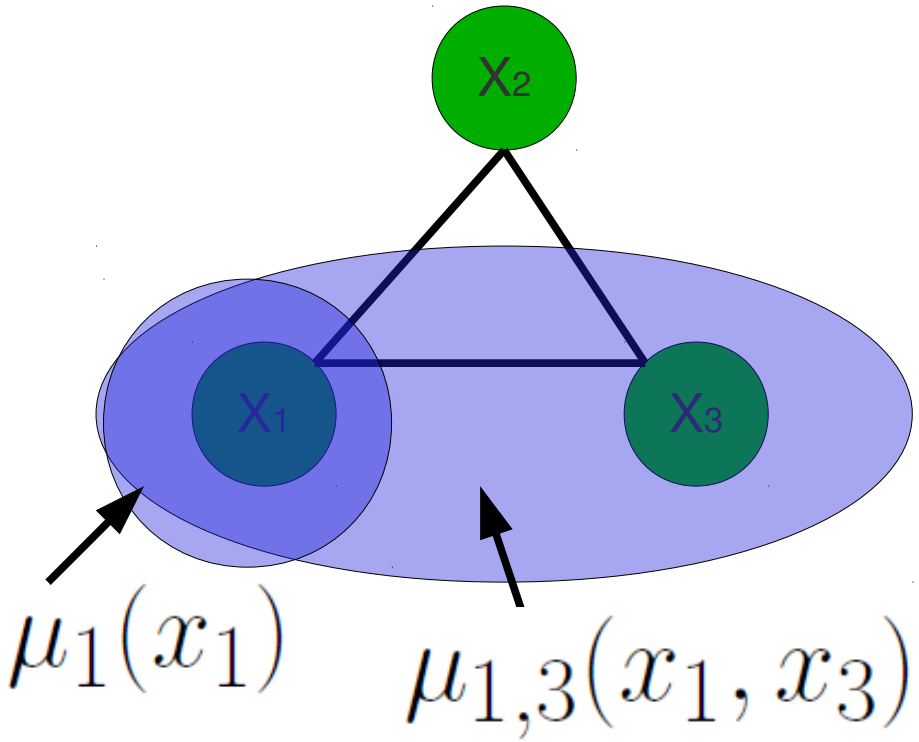


Introduce $\boldsymbol{\mu}$ - a vector of marginal probabilities for each node and edge of MRF **coming from some joint probability p**

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

$$\mathcal{M}(G) = \{\boldsymbol{\mu} \mid \exists p(\boldsymbol{x}) \; s.t. \;\; p(x_i, x_j) = \mu_{ij}(x_i, x_j), \;\; p(x_i) = \mu_i(x_i)\}$$

MAP problem for MRF as LP



$$p(\boldsymbol{x};\boldsymbol{\theta}) = \frac{1}{Z} e^{\sum_{ij\in E} \theta_{ij}(x_i,x_j)}$$

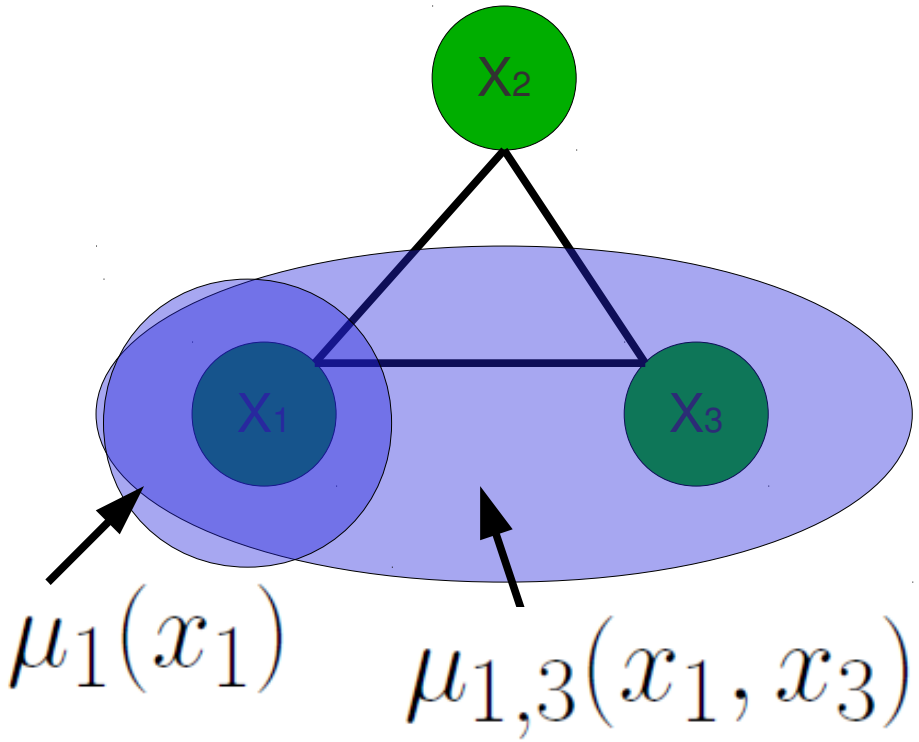
$$\mathcal{M}(G) = \{\boldsymbol{\mu} \mid \exists p(\boldsymbol{x}) \; s.t. \; p(x_i, x_j) = \mu_{ij}(x_i, x_j), \; p(x_i) = \mu_i(x_i)\}$$

MAP problem is equivalent to solving:

$$\max_{\boldsymbol{x}} f(\boldsymbol{x};\boldsymbol{\theta}) = \max_{\boldsymbol{\mu}\in\mathcal{M}(G)} \boldsymbol{\mu}\cdot\boldsymbol{\theta} = \max_{\boldsymbol{\mu}\in\mathcal{M}(G)} \sum_{ij\in E} \sum_{x_i x_j} \theta_{ij}(x_i, x_j)\mu_{ij}(x_i, x_j)$$

MAP problem for MRF as LP

$$p(\boldsymbol{x};\boldsymbol{\theta}) = \frac{1}{Z} e^{\sum_{ij \in E} \theta_{ij}(x_i, x_j)}$$



$\boldsymbol{\mu}$ should satisfy constraints

$$\mathcal{M}(G) = \{\boldsymbol{\mu} \mid \exists p(\boldsymbol{x}) \; s.t. \; p(x_i, x_j) = \mu_{ij}(x_i, x_j), \; p(x_i) = \mu_i(x_i)\}$$

MAP problem is equivalent to solving:

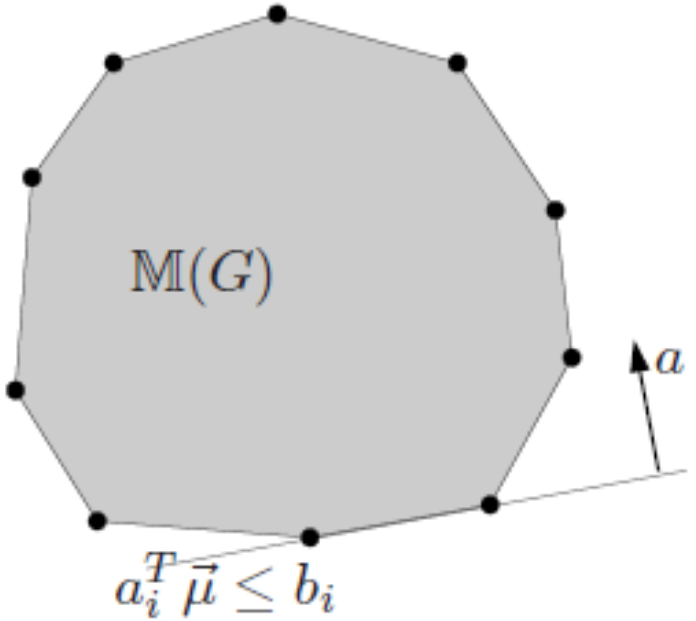
$$\max_{\boldsymbol{x}} f(\boldsymbol{x};\boldsymbol{\theta}) = \max_{\boldsymbol{\mu} \in \mathcal{M}(G)} \boldsymbol{\mu} \cdot \boldsymbol{\theta} = \max_{\boldsymbol{\mu} \in \mathcal{M}(G)} \sum_{ij \in E} \sum_{x_i x_j} \theta_{ij}(x_i, x_j) \mu_{ij}(x_i, x_j)$$

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 $(X_1, \ldots, X_N) \in \{0, 1, \ldots, m-1\}^N$.



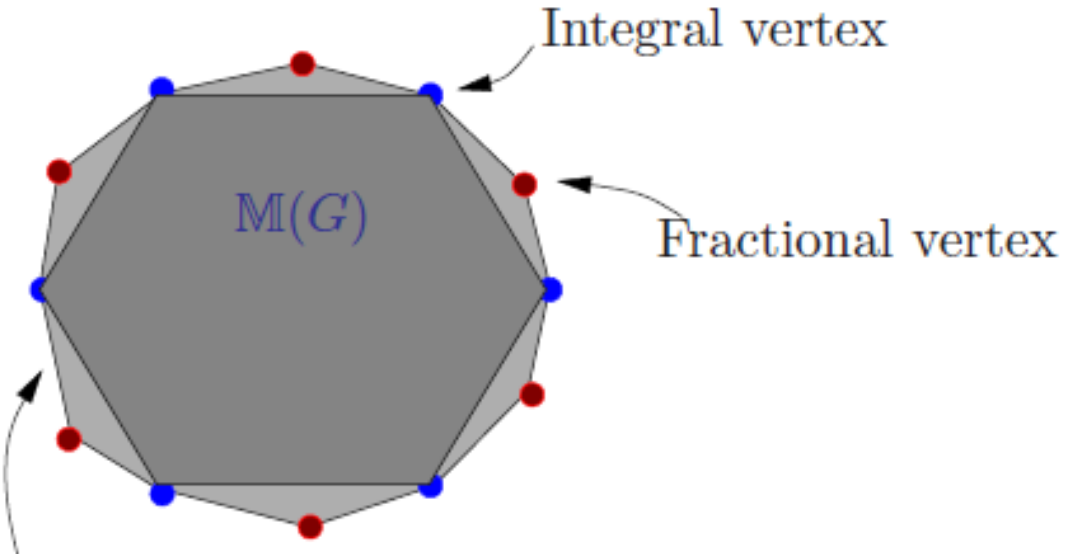
- polytope in $d = m|V| + m^2|E|$ dimensions (m per vertex, m^2 per edge)
- with m^N vertices

MAP problem for MRF as LP relaxation (upper bound)

- We don't assume anymore that **μ** comes from some distribution ***p***
- Only pairwise and singleton consistently is required

$$\sum_{x_i} \mu_i(x_i) = 1 \; \forall i \in V \; , \sum_{\hat{x}_i} \mu_{ij}(\hat{x}_i, x_j) = \mu_j(x_j)$$

$$\sum_{\hat{x}_j} \mu_{ij}(x_i, \hat{x}_j) = \mu_i(x_i) \; \forall (i,j) \in E \qquad (3)$$



Marginal polytope is expressed by:

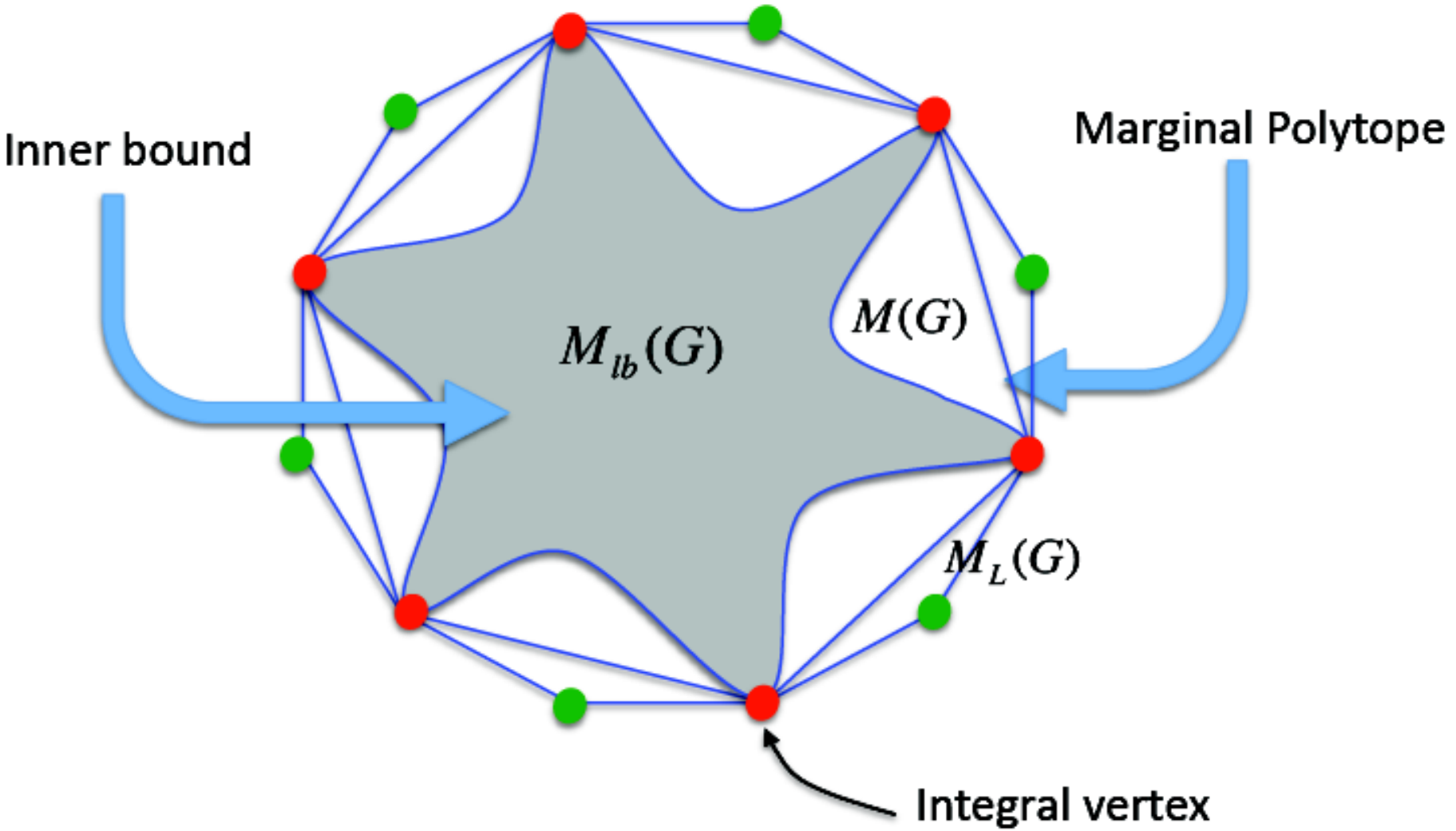
$$\mathcal{M}_L(G) = \{\mu \geq 0 \mid \text{The conditions of Eq. 3 hold}\}$$

Inner bound on Marginal Polytope [Ravikumar and Lafferty, 2006]

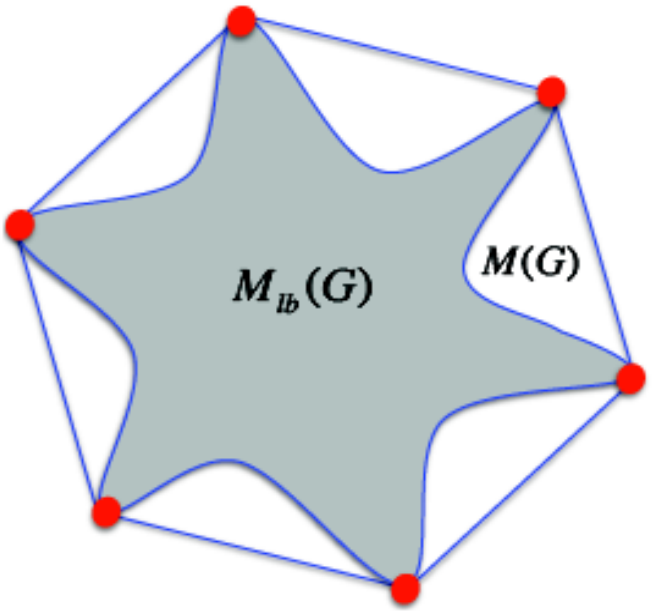
- No requirement on the structure of global distribution p
- Mean-field structure: $p(\mathbf{x}) = \prod_{i=1}^{n} p_i(x_i)$
- Inner bound $\mathcal{M}_{lb}(G) = \{\mu\}$ s.t.
  $\mu_i(x_i) = p_i(x_i)$
  $\mu_{ij}(x_i, x_j) = p_i(x_i)p_j(x_j)$

Objective: $\max_{\mu \in \mathcal{M}_{lb}(G)} \sum_{ij \in E} \sum_{x_i x_j} \theta_{ij}(x_i, x_j)\mu_i(x_i)\mu_j(x_j)$

Relationship among formulations



Properties of Inner Bound $\mathcal{M}_{lb}(G)$



- Mean field structure: exact or approximate?
- Fact1: $\mathcal{M}_{lb}(G)$ touches $\mathcal{M}(G)$ at integral solution
- Fact2: Optimal solution of $\mathcal{M}(G)$ is also integral
- Optimization over inner bound is exact!

Properties of Inner Bound $\mathcal{M}_{lb}(G)$

Some advantages of the inner bound approach

- Example problem: 170 variables, 150 domain size, 3167 edges
- Constraints in $\mathcal{M}(G)$ exponential
- In outer bound: ≈ 71 million variables, ≈ 1 million constraints
- In inner bound: 25500 variables, 170 constraints
- Only normalization constraints, linear in the number of nodes

No free lunch – convexity is lost

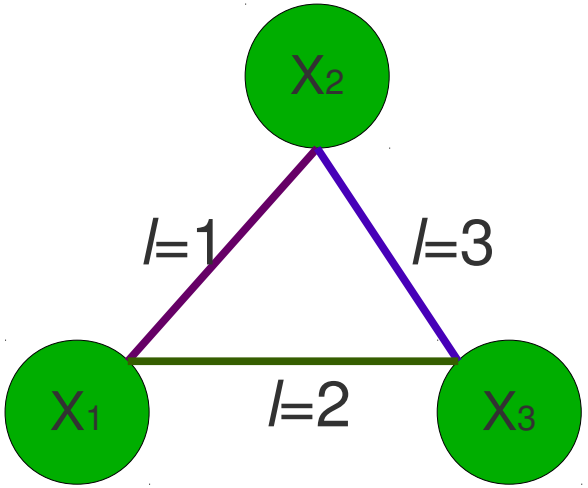
Optimization over the inner bound

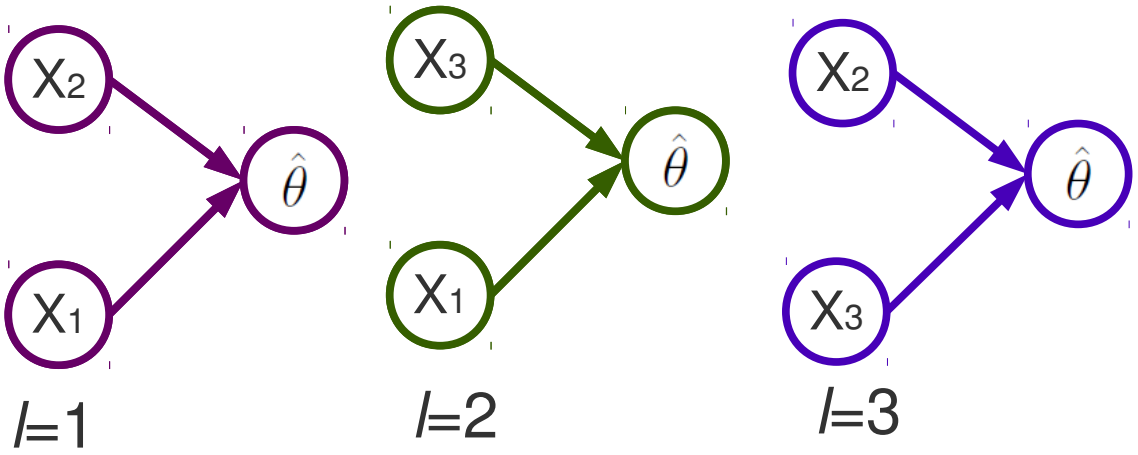
Objective function to optimize:

$\max \sum_{ij \in E} \sum_{x_i x_j} \theta_{ij}(x_i, x_j) p_i(x_i) p_j(x_j)$
subject to normalization constraint for each variable

- A general nonlinear optimization problem
- Can solve using generic NLP solvers, may not be efficient
- We present an alternate characterization of MAP using Bayes net mixture
- MAP $\equiv$ probabilistic inference over this mixture

MAP as mixture of Bayes Nets



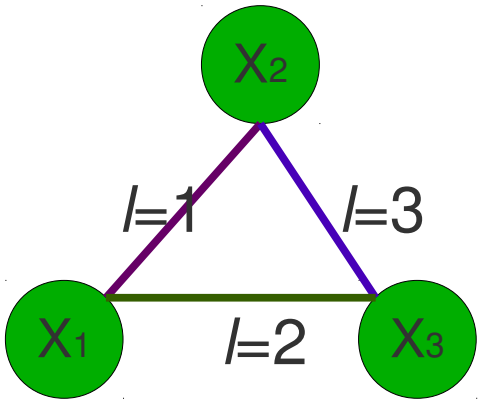


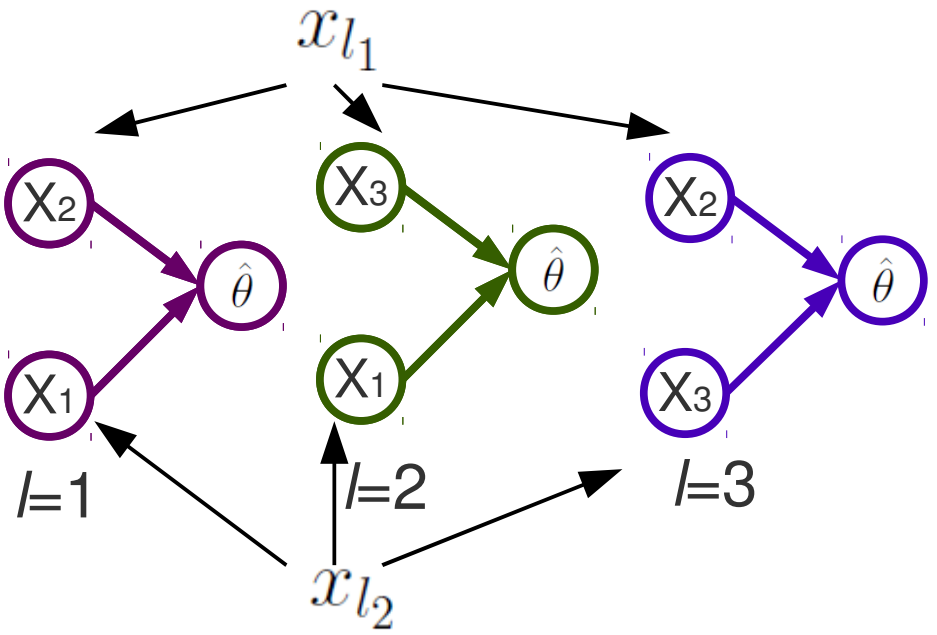
$$p(l = i) = \frac{1}{k}, i \in [1; k]$$

We want to estimate marginal probabilities:

$$\mathbf{p} = \langle p_1, \ldots, p_n \rangle, p_i = p(x_i)$$

MAP as mixture of Bayes Nets





$$P(\hat{\theta} = 1 | x_{l_1}, x_{l_2}, l) = \frac{\theta_l(x_{l_1}, x_{l_2}) - \theta_{min}}{\theta_{max} - \theta_{min}}$$

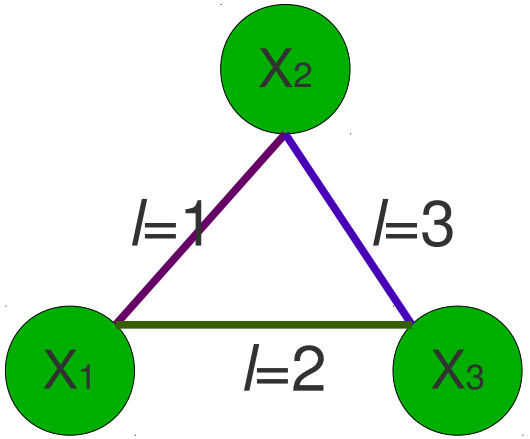
Example: for *l*=1

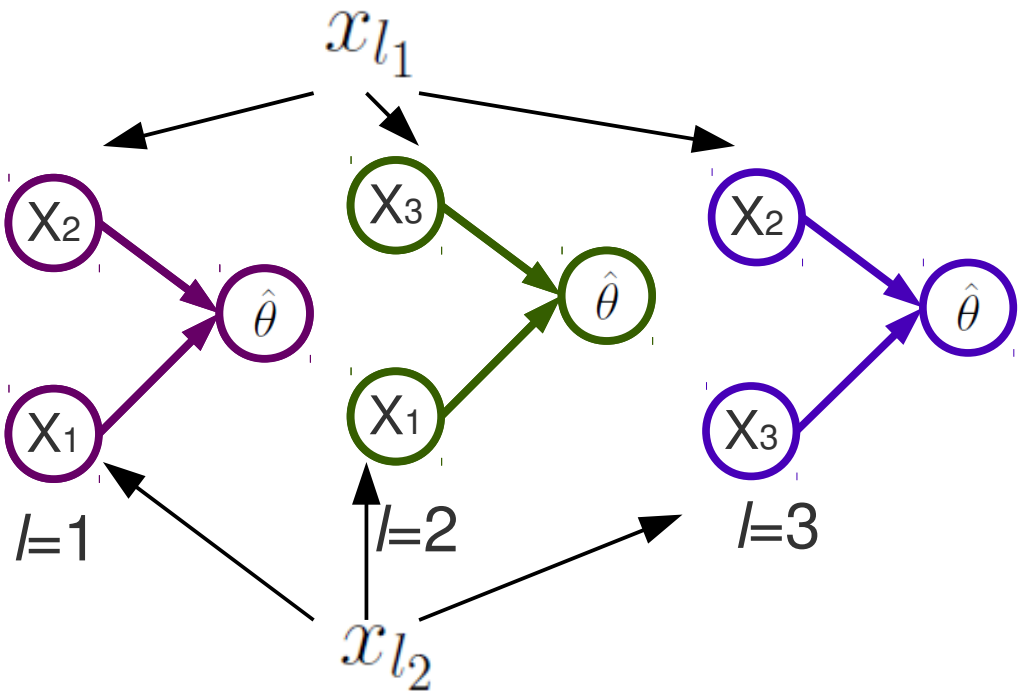
$$P(\hat{\theta} = 1 | x_1, x_2, l = 1) = (\theta_{12}(x_1, x_2) - \theta_{min}) / (\theta_{max} - \theta_{min})$$

Full joint for a particular Bayes Net indicated by *l*:

$$P(\hat{\theta}, x_{l_1}, x_{l_2} | l; \boldsymbol{p}) = P(\hat{\theta} | x_{l_1}, x_{l_2}, l) p_{l_1}(x_{l_1}; \boldsymbol{p}) p_{l_2}(x_{l_2}; \boldsymbol{p})$$

Some notation:



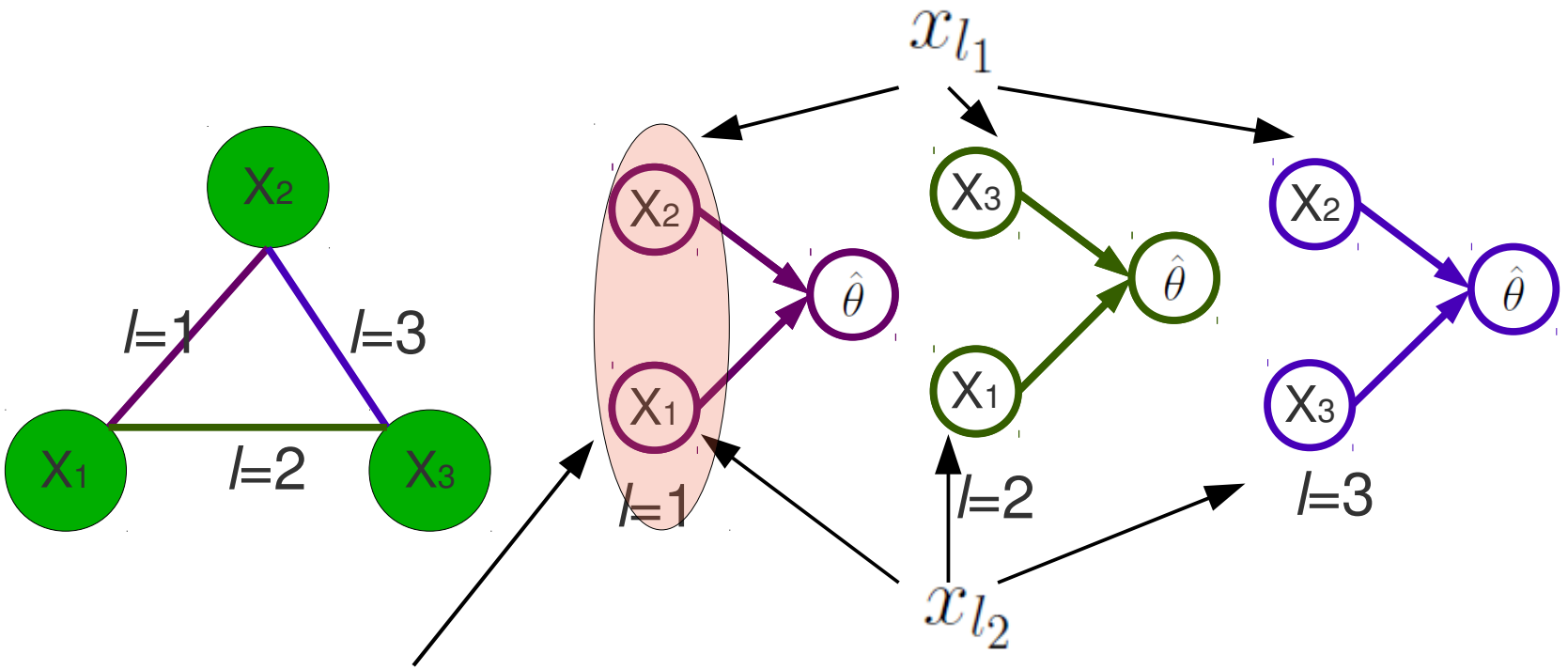


$$x_l = (x_{l_1}, x_{l_2})$$

$$\hat{\theta}_{x_l} = P(\hat{\theta} = 1 | x_{l_1}, x_{l_2}, l)$$

$$\theta_l :$$

Some notation:

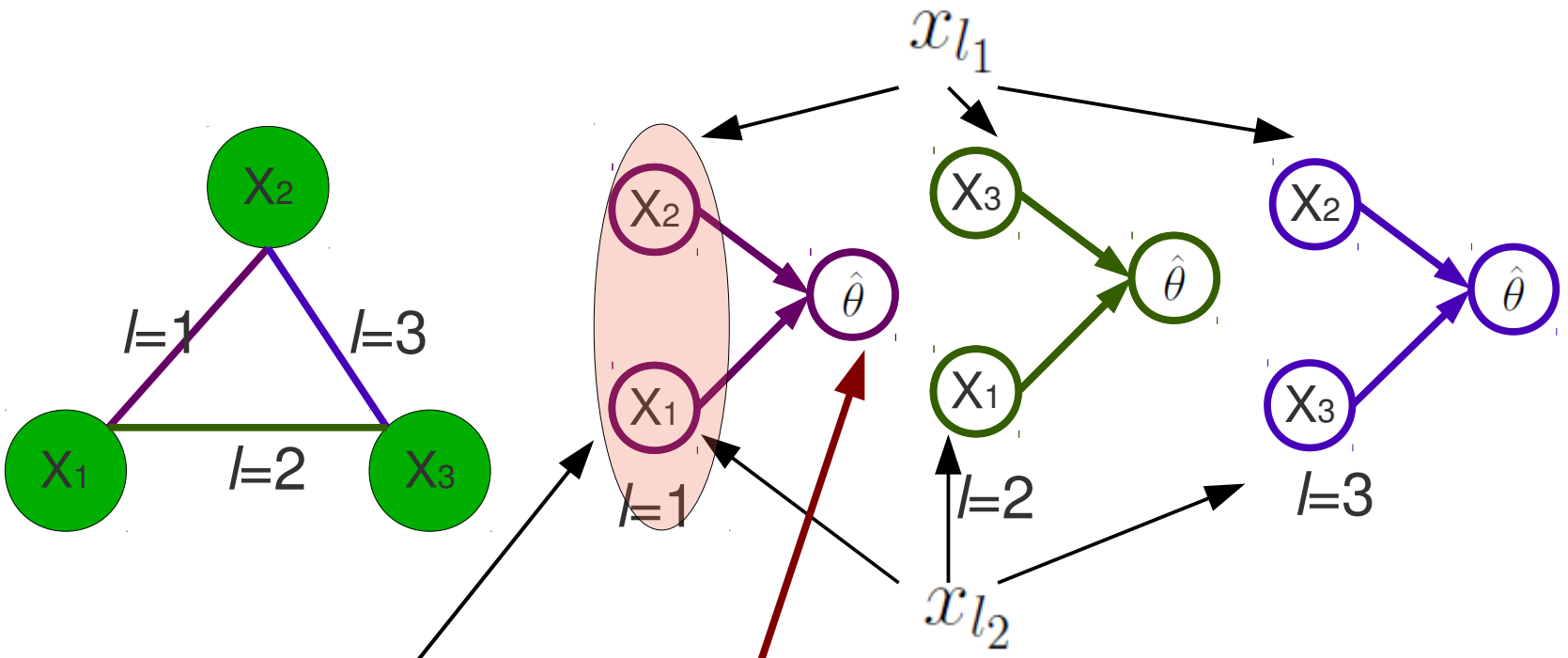


$$x_l = (x_{l_1}, x_{l_2})$$

$$\hat{\theta}_{x_l} = P(\hat{\theta} = 1 | x_{l_1}, x_{l_2}, l)$$

$$\theta_l :$$

Some notation:

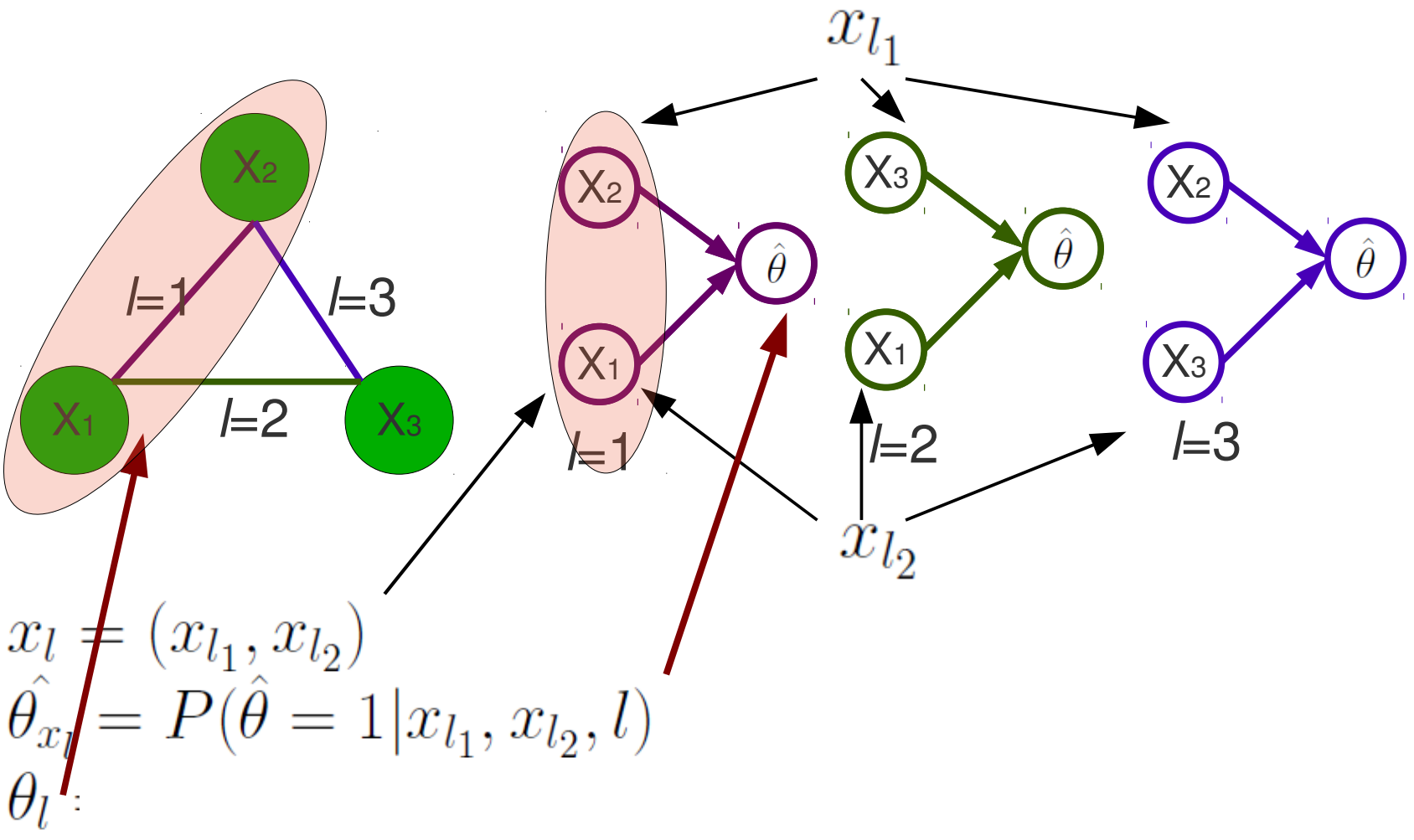


$$x_l = (x_{l_1}, x_{l_2})$$

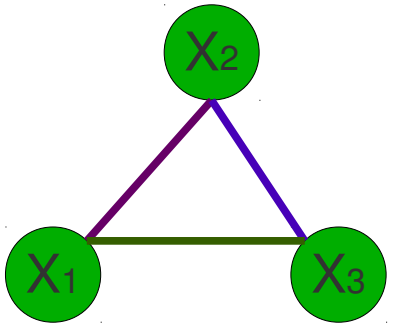
$$\hat{\theta}_{x_l} = P(\hat{\theta} = 1 | x_{l_1}, x_{l_2}, l)$$

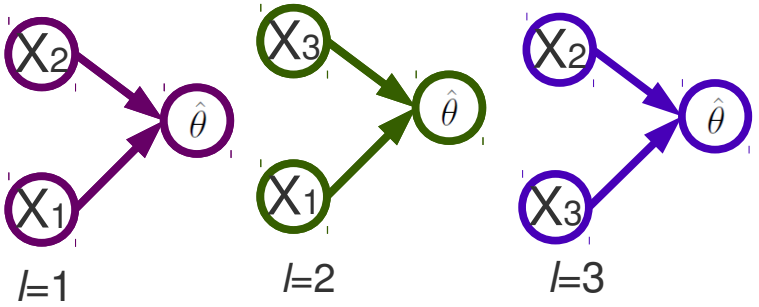
$$\theta_l :$$

Some notation:



$$x_l = (x_{l_1}, x_{l_2})$$
$$\hat{\theta}_{x_l} = P(\hat{\theta} = 1 | x_{l_1}, x_{l_2}, l)$$





Theorem

Maximizing the likelihood of observing $\hat{\theta} = 1$ in the mixture optimizes the original objective function.

Objective: $\max \sum_{ij \in E} \sum_{x_i x_j} \theta_{ij}(x_i, x_j) p_i(x_i) p_j(x_j)$

$$\max_{\boldsymbol{x}} f_{lb}(\boldsymbol{x}; \boldsymbol{\theta}) = \max_{\boldsymbol{\mu} \in \mathcal{M}_{lb}(G)} \boldsymbol{\mu} \cdot \boldsymbol{\theta} = \max_{\boldsymbol{\mu} \in \mathcal{M}_{lb}(G)} \sum_{ij \in E} \sum_{x_i x_j} \theta_{ij}(x_i, x_j) \mu_i(x_i) \mu_j(x_j)$$

Theorem

Maximizing the likelihood of observing $\hat{\theta} = 1$ in the mixture optimizes the original objective function.

Objective: $\max \sum_{ij \in E} \sum_{x_i x_j} \theta_{ij}(x_i, x_j) p_i(x_i) p_j(x_j)$

Proof.

Likelihood for a single Bayes net

$$L_{ij} = P(\hat{\theta} = 1 | (i,j)) = \sum_{x_i x_j} \hat{\theta}_{ij}(x_i, x_j) p_i(x_i) p_j(x_j) = \sum_{x_l} \hat{\theta}_{x_l} p_{l_1}(x_{l_1}; p) p_{l_2}(x_{l_2}; p)$$

For the complete mixture

$$L = \sum_{(i,j)} P(l) L_{ij} = \frac{1}{|E|} \sum_{(i,j)} \sum_{x_i x_j} \hat{\theta}_{ij}(x_i, x_j) p_i(x_i) p_j(x_j)$$

Objective: $\sum_{ij \in E} \sum_{x_i x_j} \theta_{ij}(x_i, x_j) p_i(x_i) p_j(x_j) = \frac{1}{k} \sum_{l} \sum_{x_l} \hat{\theta}_{x_l} p_{l_1}(x_{l_1}; p) p_{l_2}(x_{l_2}; p)$

$L \propto$ *Objective*

□

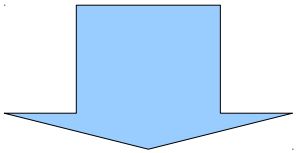
That's the likelihood of BN mixture from previous slide:

$$\sum_{l} \sum_{x_l} \theta_l(x_l) p_{l_1}(x_{l_1}; p) p_{l_2}(x_{l_2}; p) = k(\theta_{min} + (\theta_{max} - \theta_{min}) L^p).$$

And that's our optimization criterion for lower bound:

$$\max_{\mu \in \mathcal{M}_{lb}(G)} \sum_{ij \in E} \sum_{x_i x_j} \theta_{ij}(x_i, x_j) \mu_i(x_i) \mu_j(x_j)$$

Recalling that $\theta_l(x_l) = \theta_{i,j}(x_i, x_j)$ and $p_{l_1}(x_{l_1}; \mathbf{p}) = \mu_i(x_i)$



These are the same things

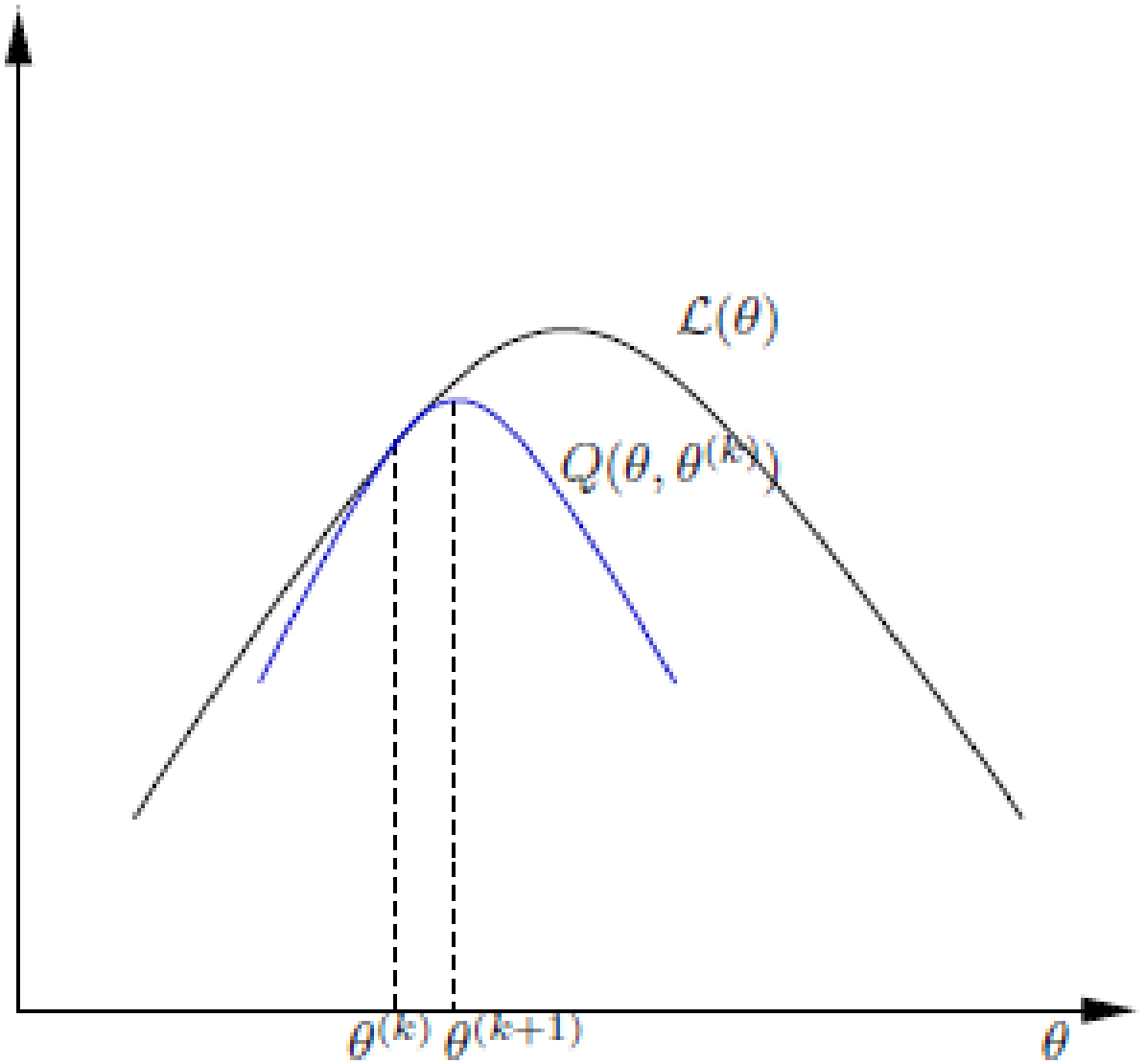
So, can maximize L^p and get MAP lower bound!

Let's use EM for that

The EM Algorithm

- Maximum likelihood parameter estimates
  - One definition of the "best" knob settings.
  - Often impossible to find directly.
- The EM algorithm:
  - Finds ML parameters when the original (hard) problem can be broken up into two (easy) pieces, i.e., finds the ML estimates of parameters for data in which some variables are unobserved.
  - A iterative method which consists of "Expectation step" (E-step) and "Maximization step" (M-step).
    1. Estimate some "missing" or "unobserved" data from observed data and current parameters (E-step)
    2. Using this "complete" data, find the maximum likelihood parameter estimates (M-step).

Graphical illustration for auxiliary function



EM for MAP Estimation

Basic idea

- Treat all variables x_i as hidden
- Observed data $\hat{\theta} = 1$
- Estimate μ_i for each variable i
- Update equations easily implemented using graph-based message passing

Update rules

p – old parameters
p* - new parameters

Expected complete log-likelihood:

$$Q(p, p^\star) = \sum_l \sum_{x_l} P(\hat{\theta} = 1, x_l, l; p) \log P(\hat{\theta} = 1, x_l, l; p^\star)$$

The full joint is given by:

$$P(\hat{\theta} = 1, x_l, l; p) \;=\; P(\hat{\theta} = 1 | x_l, l) P(x_l | l; p) P(l) \;=\; \frac{1}{k} \hat{\theta}_{x_l} p_{l_1}(x_{l_1}; p) p_{l_2}(x_{l_2}; p)$$

We will omit the parameter p whenever the expression is unambiguous. Taking the log, we get:

$$\log P(\hat{\theta} = 1, x_l, l; p) = \langle \text{Ind. terms of } p \rangle + \log p_{l_1}(x_{l_1}) + \log p_{l_2}(x_{l_2}) \tag{12}$$

Substituting the above equation into the definition of $Q(p, p^\star)$ (Eq. 11) and discarding the terms which are independent of $p^\star$, we get:

$$Q(p, p^\star) = \frac{1}{k} \sum_l \sum_{x_l} \hat{\theta}_{x_l} p_{l_1}(x_{l_1}) p_{l_2}(x_{l_2}) \{ \log p^\star_{l_1}(x_{l_1}) + \log p^\star_{l_2}(x_{l_2}) \} \tag{13}$$

Upon simplifying the above equation by grouping together the terms associated with the variables x_i of the MRF, we get:

$$Q(p, p^\star) = \frac{1}{k} \sum_{i=1}^{n} \sum_{x_i} p_i(x_i) \log p^\star_i(x_i) \sum_{j \in Ne(i)} \sum_{x_j} \hat{\theta}_{x_i x_j} p_j(x_j) \tag{14}$$

$$p^\star_i(x_i) = \frac{p_i(x_i) \sum_{j \in Ne(i)} \sum_{x_j} \hat{\theta}_{x_i x_j} p_j(x_j)}{C_i}$$

Update rules

p – old parameters
p* - new parameters

Expected complete log-likelihood:

$$Q(p, p^\star) = \sum_l \sum_{x_l} P(\hat{\theta} = 1, x_l, l; p) \log P(\hat{\theta} = 1, x_l, l; p^\star)$$

The full joint is given by:

$$P(\hat{\theta} = 1, x_l, l; p) \;=\; P(\hat{\theta} = 1 | x_l, l) P(x_l | l; p) P(l) \;=\; \frac{1}{k} \hat{\theta}_{x_l} p_{l_1}(x_{l_1}; p) p_{l_2}(x_{l_2}; p)$$

We will omit the parameter p whenever the expression is unambiguous. Taking the log, we get:

$$\log P(\hat{\theta} = 1, x_l, l; p) = \langle \text{Ind. terms of } p \rangle + \log p_{l_1}(x_{l_1}) + \log p_{l_2}(x_{l_2}) \tag{12}$$

Substituting the above equation into the definition of $Q(p, p^\star)$ (Eq. 11) and discarding the terms which are independent of $p^\star$, we get:

$$Q(p, p^\star) = \frac{1}{k} \sum_l \sum_{x_l} \hat{\theta}_{x_l} p_{l_1}(x_{l_1}) p_{l_2}(x_{l_2}) \{ \log p^\star_{l_1}(x_{l_1}) + \log p^\star_{l_2}(x_{l_2}) \} \tag{13}$$

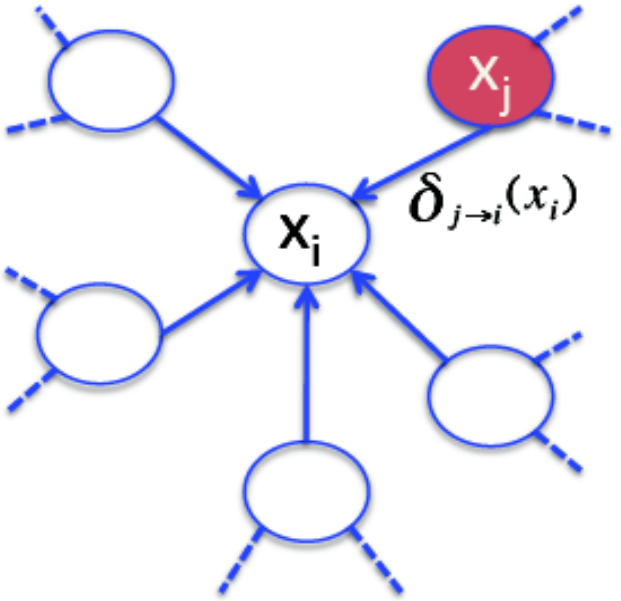
Upon simplifying the above equation by grouping together the terms associated with the variables x_i of the MRF, we get:

$$Q(p, p^\star) = \frac{1}{k} \sum_{i=1}^{n} \sum_{x_i} p_i(x_i) \log p^\star_i(x_i) \sum_{j \in Ne(i)} \sum_{x_j} \hat{\theta}_{x_i x_j} p_j(x_j) \tag{14}$$

M-step

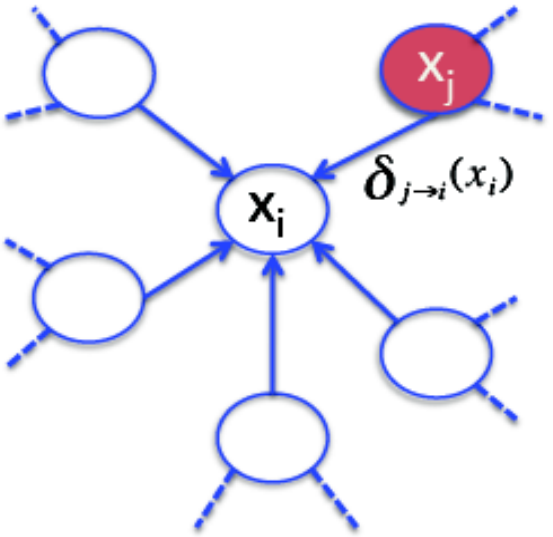
$$p^\star_i(x_i) = \frac{p_i(x_i) \sum_{j \in Ne(i)} \sum_{x_j} \hat{\theta}_{x_i x_j} p_j(x_j)}{C_i}$$

Message Passing in EM



- E-step of EM
- Node j sends message $\delta_{j\to i}(x_i)$ to its neighbor i
- $\delta_{j\to i}(x_i) = \sum_{x_j} \mu_j(x_j)\hat{\theta}_{ij}(x_i, x_j)$
- Weighted normalized reward
- Local message passing, scalable, efficient

M-Step in EM



- Each variable collects all incoming messages
- Update: $\mu^{\star}(x_i) = \mu_i(x_i)\frac{\sum_{k\in\ Ne(i)}\delta_{k\to i}(x_i)}{C_i}$
- Computational complexity $O(nd^2)$

Algorithm

Algorithm 1: Graph-based message passing for MAP estimation

input : Graph $G = (V, E)$ for the MRF and potentials θ for each edge
repeat
  foreach *node* $i \in V$ **do**

    EM: Send message $\delta_{i \to j}$ to each neighbor $j \in Ne(i)$

    $\delta_{i \to j}(x_j) \leftarrow \sum_{x_i} p_i(x_i) \hat{\theta}_{x_i x_j}$

    Set marginal probability to sum of incoming messages: $p_i^\star(x_i) = p_i(x_i) \frac{\sum_{k \in Ne(i)} \delta_{k \to i}(x_i)}{C_i}$

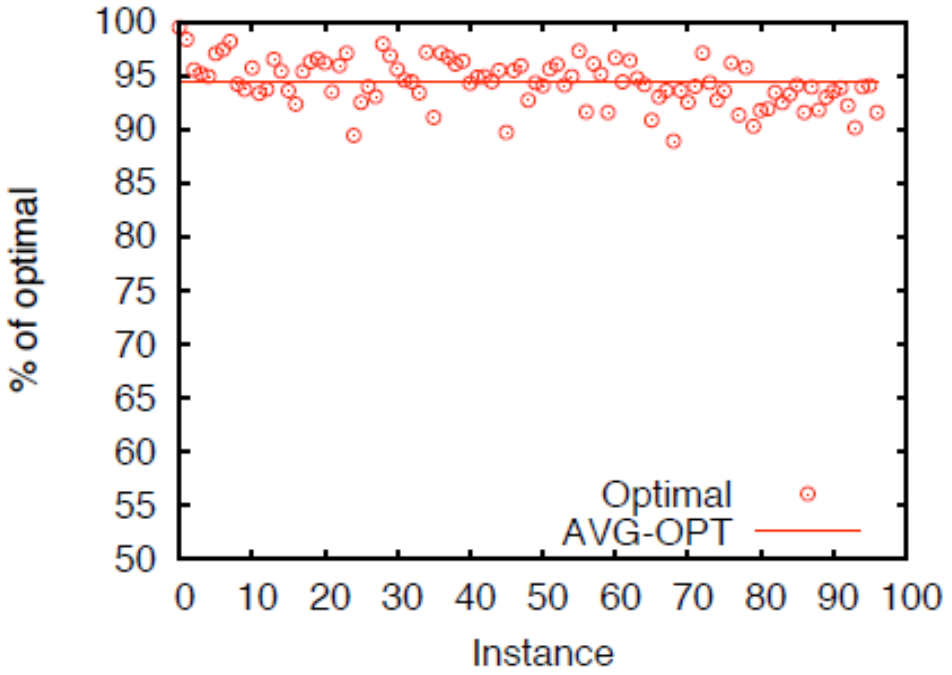
  until *stopping criterion is satisfied*
EM : Return complete assignment x s.t. $x_i = \text{argmax}_{\hat{x}_i} p_i(\hat{x}_i)$

Complexity of computing a single message is $O(d^2)$, were d - domain size

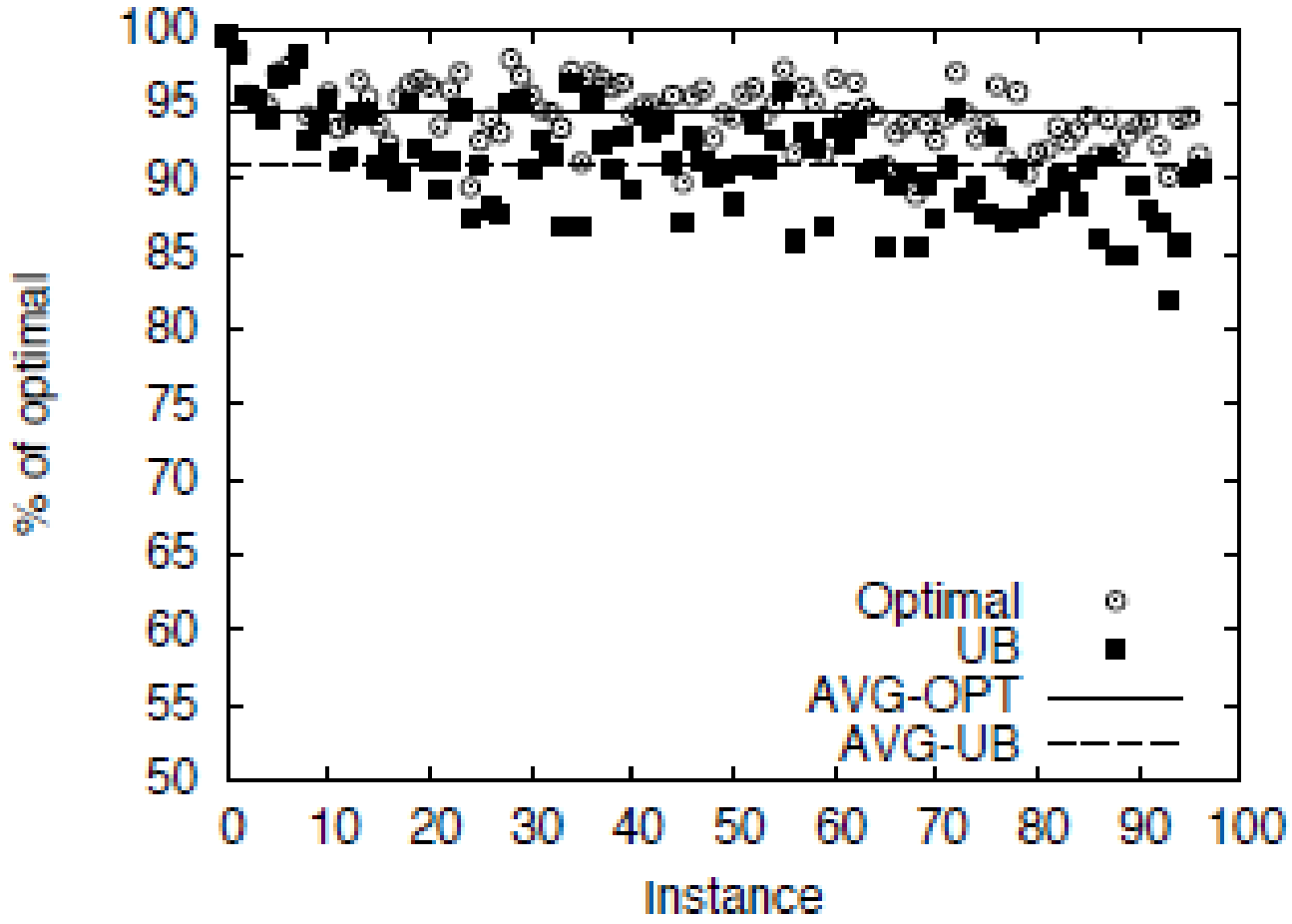
Independent of degree ???

Experiments

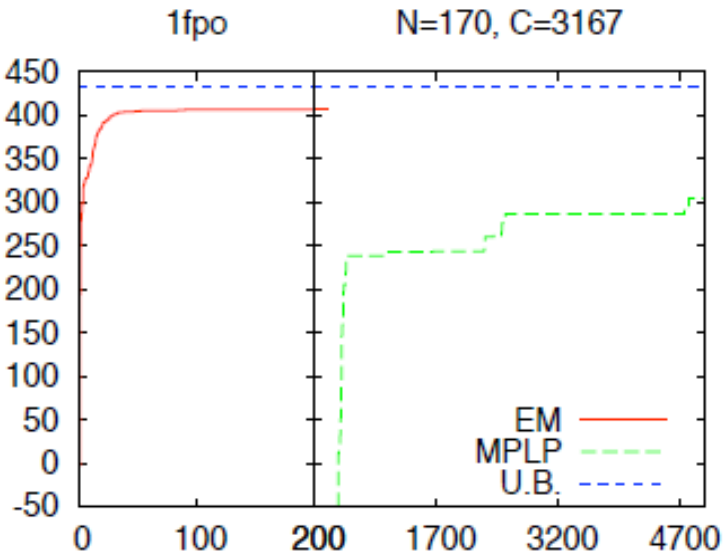
Protein design dataset – 97 instances
Comparisons with Max-Product LP [Sontag et al., 2008],
Max-Product [Pearl, 1988]
Mean running time MPLP– 9.7 hours, EM only 300 seconds.
EM produces better solution quality much faster than MPLP

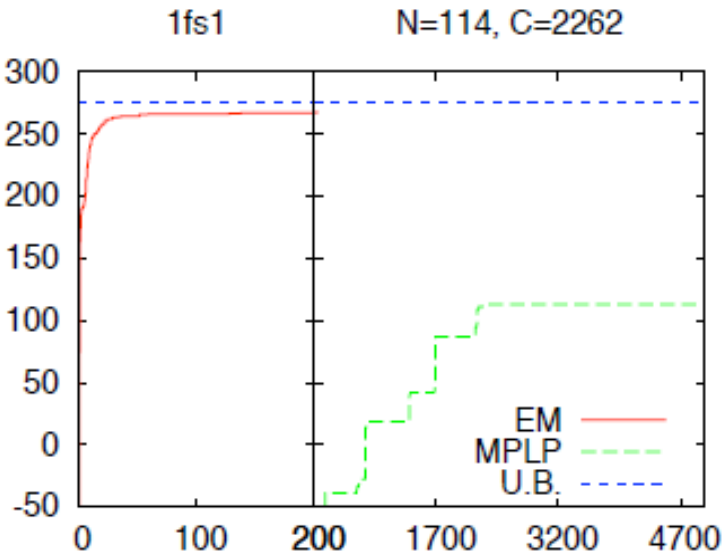


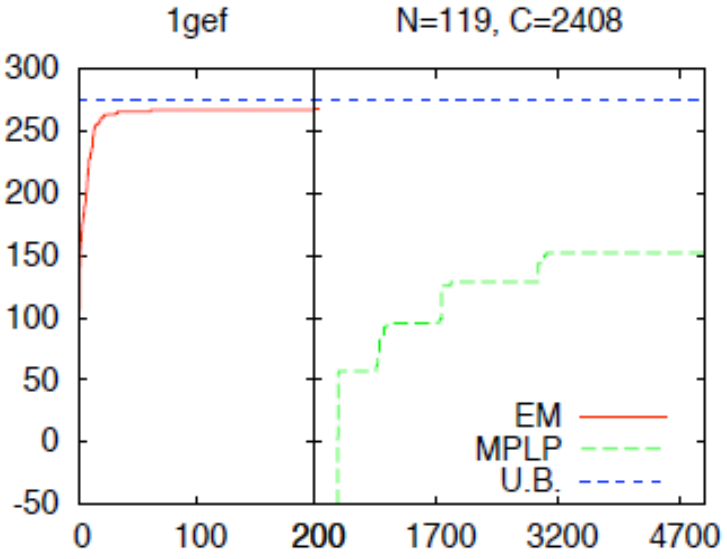
Experiment: solution quality

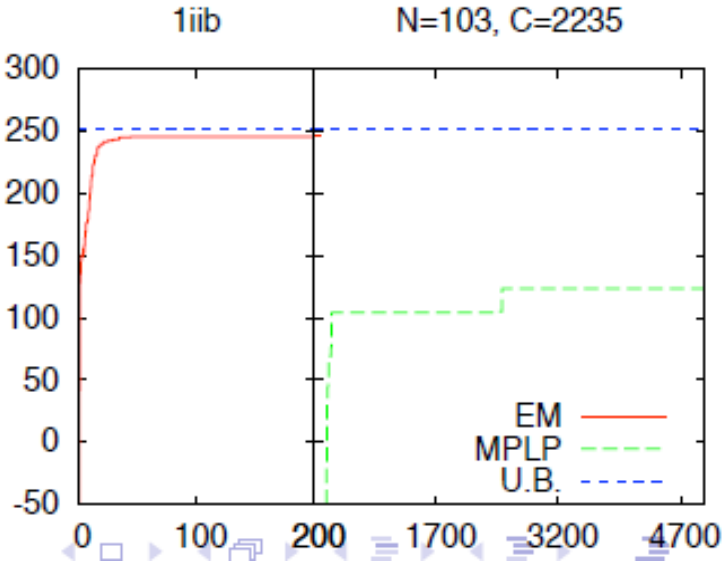


Experiments

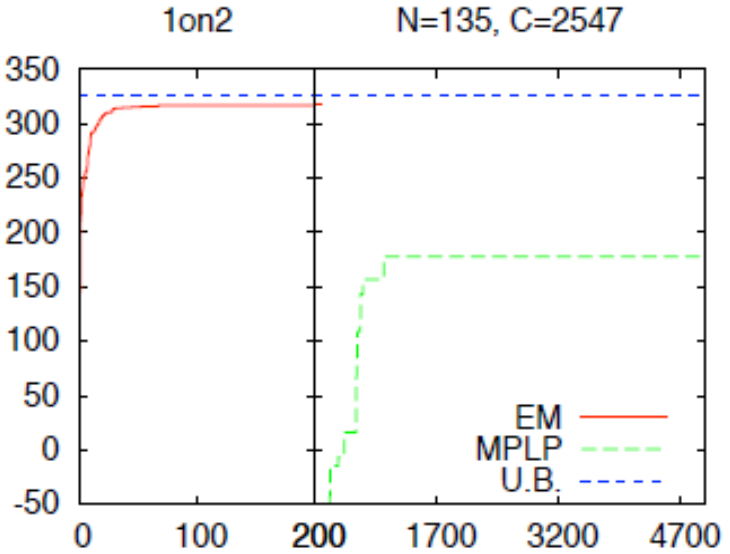


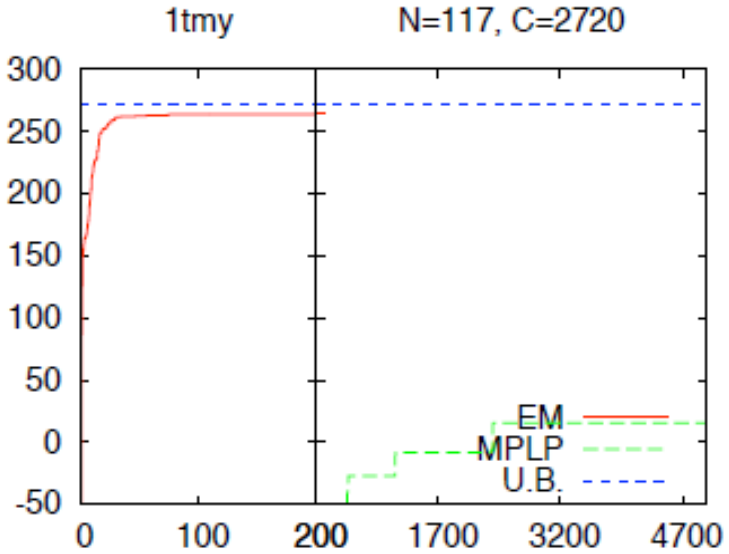


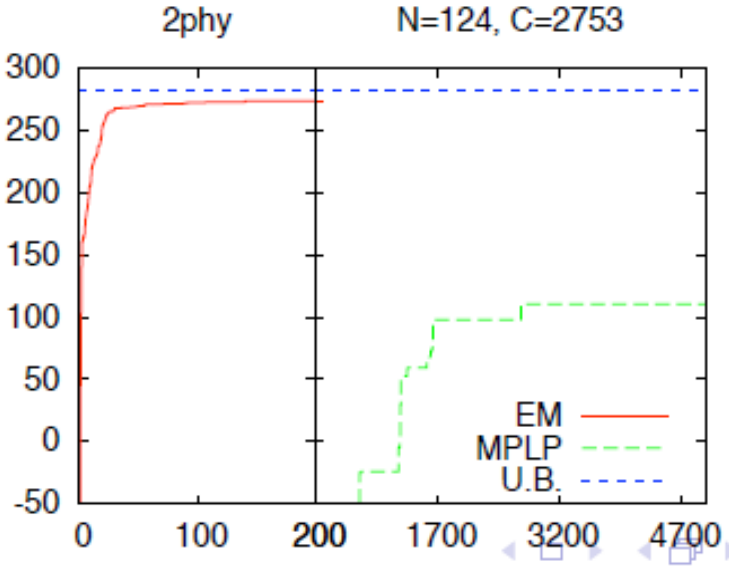




Experiments







Experiments: compare with max-product

- Comparisons with Max-Product
- 1500 iterations
- Gives decent quality, but takes much longer than EM (due to high degree of graph)

Instance	MP Quality	Time/Iteration	EM Quality	Time/Iteration	U.B.
1fs1	268.3	3628.5/344	267.6	202.3/1220	276.4
1gef	239.8	13938.9/1331	267.3	71.6/428	276.1
1bkb	272.4	10928.9/965	288.2	250.1/1462	292.8
1iib	236.2	11493.1/1099	245.7	78.1/442	251.1
1on2	314.7	11628.34/1226	317.1	146.6/807	327.23
1tmy	202.9	99.5/8	264.8	222.4/1067	272.1
1or7	368.1	234.9/22	410.2	240.8/1087	419.3
1fpo	406.2	9072.7/791	407.1	263.6/1125	434

Why is this method good?

- Average solution quality within **95% of optimal** (for protein design)
- Increases lower bound on MAP **rapidly**
- Embarassingly **parallel**
  - (assuming shared memory? o.w. too much communication)
- Doesn't work:
  - $\Theta_{max} - \Theta_{min}$ is large