Graphical Models

Loopy BP and Bethe Free Energy

Siamak Ravanbakhsh

Winter 2018

Learning objective

- loopy belief propagation
- its variational derivation: Bethe approximation

So far...

- exact inference:
  - variable elimination
  - equivalent to belief propagation (BP) in a clique tree

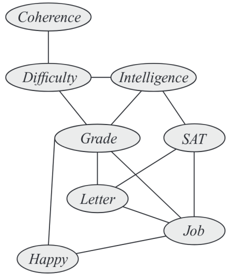
So far...

- exact inference:
  - variable elimination
  - equivalent to belief propagation (BP) in a clique tree

This class...

- what if the exact inference is too expensive? (i.e., the tree-width is large)
  - continue to use BP: **loopy BP**
  - why is this a good idea?
    - answer using variational interpretation

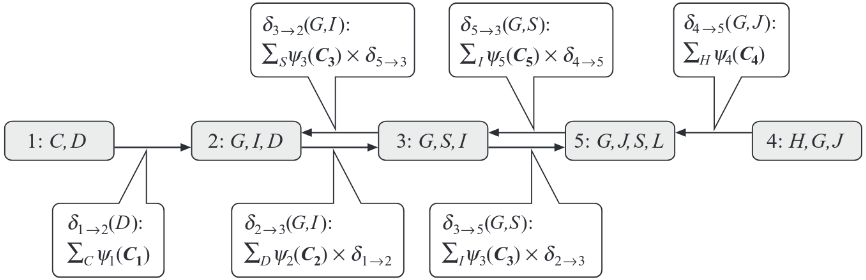
Recap: BP in clique trees



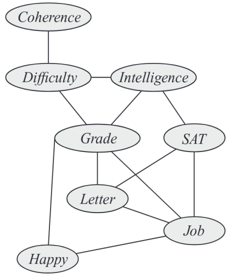
sum-product **BP** message update:

$$\delta_{i\to j}(\underbrace{S_{i,j}}_{\text{sepset}}) = \sum_{C_i - S_{i,j}} \psi_i(\underbrace{C_i}_{\text{cluster/clique}}) \prod_{k \in Nb_i - j} \delta_{k\to i}(S_{i,k})$$

- from leaves towards the root
- back to leaves



Recap: BP in clique trees

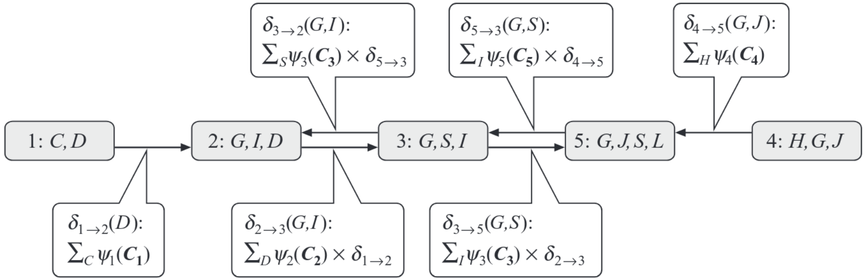


sum-product **BP** message update:

$$\delta_{i\to j}(\boxed{S_{i,j}}) = \sum_{C_i - S_{i,j}} \psi_i(\boxed{C_i}) \prod_{k\in Nb_i - j} \delta_{k\to i}(S_{i,k})$$

sepset cluster/clique

- from leaves towards the root
- back to leaves

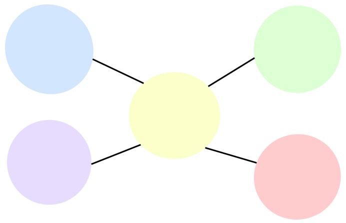


marginal (belief) for each cluster:

$$p_i(C_i) \propto \beta_i(C_i) = \psi_i(C_i) \prod_{k\in Nb_i} \delta_{k\to i}(S_{i,k})$$

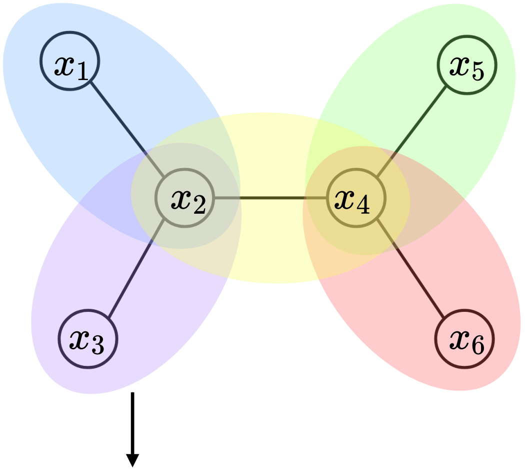
Clique-tree for tree structures

- pairwise potentials $\phi_{i,j}(x_i, x_j)$
- tree width = 1



one possible clique-tree

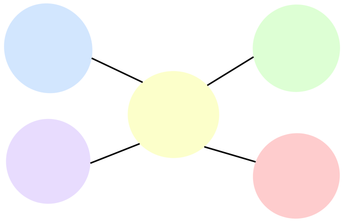
what are the sepsets?



one cluster per factor

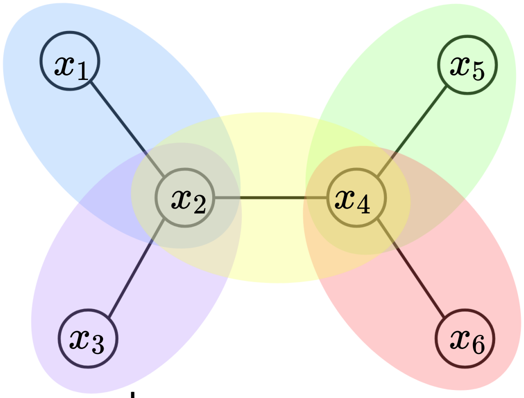
Clique-tree for tree structures

- pairwise potentials $\phi_{i,j}(x_i, x_j)$
- tree width = 1

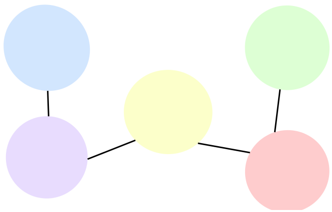


one possible clique-tree

what are the sepsets?



one cluster per factor



a different valid clique-tree

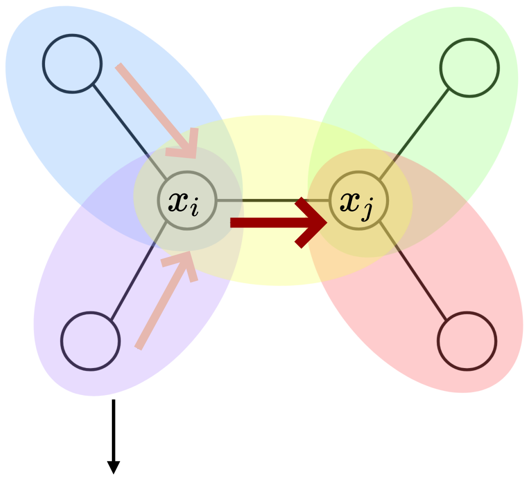
check for *running intersection property*

BP for tree structures

- pairwise potentials $\phi_{i,j}(x_i, x_j)$
- message update

$$\delta_{i\to j}(x_j) = \sum_{x_i} \phi_{i,j}(x_i, x_j) \prod_{k\in Nb_i - j} \delta_{k\to i}(x_i)$$

  - *from leaves towards **a** root*
  - *back to leaves*



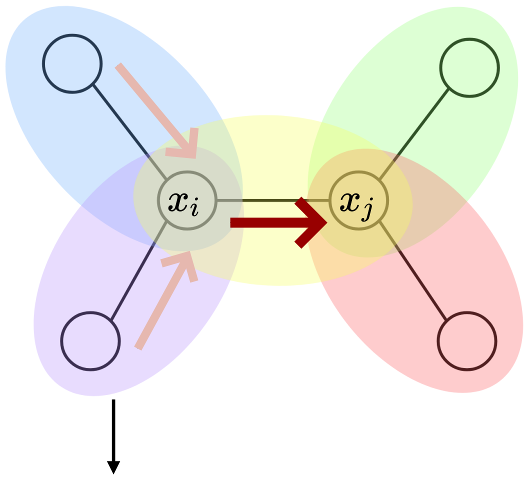
one cluster per factor

BP for **tree structures**

- pairwise potentials $\phi_{i,j}(x_i, x_j)$
- message update

$$\delta_{i\to j}(x_j) = \sum_{x_i} \phi_{i,j}(x_i, x_j) \prod_{k\in Nb_i - j} \delta_{k\to i}(x_i)$$

  - *from leaves towards **a** root*
  - *back to leaves*



one cluster per factor

- marginal (belief) for each cluster

$$p_i(x_i) \propto \prod_{k\in Nb_i} \delta_{k\to i}(x_i)$$

$$p_{i,j}(x_i, x_j) \propto \phi_{i,j}(x_i, x_j) \prod_{k\in Nb_i - j} \delta_{k\to i}(x_i) \prod_{k\in Nb_j - i} \delta_{k\to j}(x_j)$$

BP for tree structures: reparametrization

graphical model represents

$$* \quad p(\mathbf{x}) = \frac{1}{z} \prod_{i,j \in \mathcal{E}} \phi_{i,j}(x_i, x_j)$$

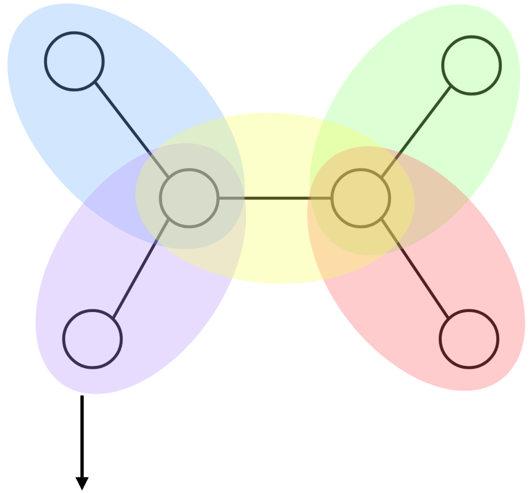
write it in terms of marginals

$$p(\mathbf{x}) = \frac{\prod_{i,j \in \mathcal{E}} p_{i,j}(x_i, x_j)}{\prod_i p_i^{|Nb_i|-1}}$$

why is this correct?

the denominator is adjusting for double-counts

substitute the marginals using BP messages to get (*)



one cluster per factor

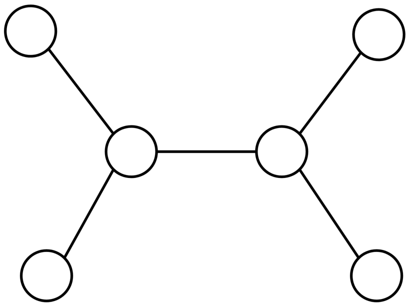
Variational interpretation

BP as I-projection

$$\arg\min_q D(q||p)$$

$$p(x) = \frac{1}{Z}\prod_k \phi_{i,j}(x_i, x_j)$$

$$q(x) = \frac{\prod_{i,j\in\mathcal{E}} q_{i,j}(x_i,x_j)}{\prod_i q_i(x_i)^{|Nb_i|-1}}$$

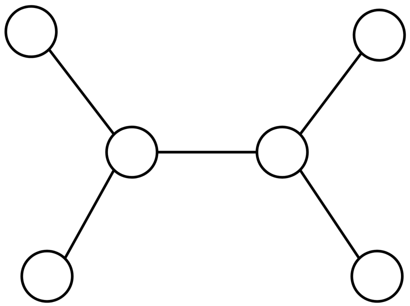


write q in terms of marginals of interest

minimization gives us the marginals $q_{i,j}, q_i$

Variational **free energy**

$$D(q\|p) = \sum_{\mathbf{x}} \underbrace{q(\mathbf{x})(\ln q(x)}_{-H(q)} - \underbrace{\ln p(x)}_{\mathbb{E}_q[\sum_{i,j} \ln \phi_{i,j}(x_i, x_j)] - \ln(Z)})$$



$$= -H(q) - \mathbb{E}_q[\sum_{i,j} \ln \phi_{i,j}(x_i, x_j)] + \ln Z$$ ignore: does not depend on q

I-projection is equivalent to $\arg\max_q \underbrace{H(q) + \mathbb{E}_q[\sum_{i,j} \ln \phi_{i,j}(x_i, x_j)]}_{\text{variational free energy}}$

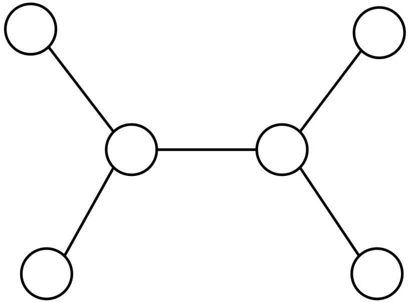
free energy is a lower-bound on $\ln Z$

Simplifying the free energy

$\arg\min_q D(q||p)$

$p(x) = \frac{1}{Z}\prod_k \phi_{i,j}(x_i, x_j)$

$q(x) = \frac{\prod_{i,j\in\mathcal{E}} q_{i,j}(x_i,x_j)}{\prod_i q_i(x_i)^{|Nb_i|-1}}$



$\equiv \arg\max_q H(q) + \mathbb{E}_q[\sum_{i,j} \ln \phi_{i,j}(x_i, x_j)]$

so far did not use the **decomposed form of q**

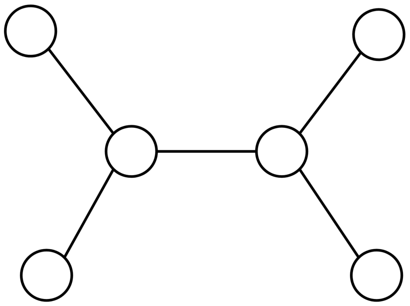
both entropy and energy involve summation over exponentially many terms

Simplifying the free energy

$$\arg\min_q D(q||p)$$

$$p(x) = \frac{1}{Z}\prod_k \phi_{i,j}(x_i, x_j)$$

$$q(x) = \frac{\prod_{i,j\in\mathcal{E}} q_{i,j}(x_i,x_j)}{\prod_i q_i(x_i)^{|Nb_i|-1}}$$



$$\equiv \arg\max_q H(q) + \mathbb{E}_q[\sum_{i,j} \ln\phi_{i,j}(x_i, x_j)]$$

$$\sum_{i,j\in\mathcal{E}}\sum_{x_{i,j}} q_{i,j}(x_i, x_j)\ln\phi_{i,j}(x_i, x_j)$$

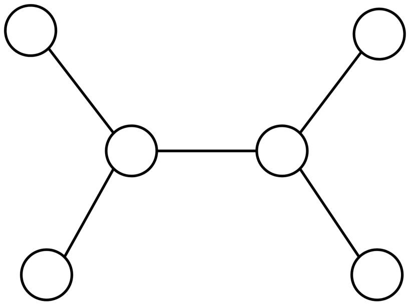
$$\sum_{i,j\in\mathcal{E}} H(q_{i,j}) - \sum_i(|Nb_i| - 1)H(q_i)$$ follows from the decomposition of q

Variational interpretation: marginal constraints

$$\arg\max_q H(q) + \mathbb{E}_q[\sum_{i,j} \ln \phi_{i,j}(x_i, x_j)]$$

$$\sum_{i,j \in \mathcal{E}} \sum_{x_{i,j}} q_{i,j}(x_i, x_j) \ln \phi_{i,j}(x_i, x_j)$$

$$\sum_{i,j \in \mathcal{E}} H(q_{i,j}) - \sum_i (|Nb_i| - 1) H(q_i)$$



marginals $q_{i,j}, q_i$ should be "valid" a real distribution with these marginals should exist

marginal polytope

$$\sum_{x_i} q_{i,j}(x_i, x_j) = q_j(x_j) \quad \forall i, j \in \mathcal{E}, x_j$$

for tree graphical models this **local** consistency is enough

Variational derivation of BP

$$\arg\max_{\{q\}} \sum_{i,j\in\mathcal{E}} H(q_{i,j}) - \sum_i (|Nb_i| - 1) H(q_i) + \sum_{i,j\in\mathcal{E}} \sum_{x_{i,j}} q_{i,j}(x_i, x_j) \ln \phi_{i,j}(x_i, x_j)$$

Variational derivation of BP

$$\arg\max_{\{q\}} \sum_{i,j\in\mathcal{E}} H(q_{i,j}) - \sum_i (|Nb_i| - 1) H(q_i) + \sum_{i,j\in\mathcal{E}} \sum_{x_{i,j}} q_{i,j}(x_i, x_j) \ln \phi_{i,j}(x_i, x_j)$$

$$\sum_{x_i} q_{i,j}(x_i, x_j) = q_j(x_j) \quad \forall i, j \in \mathcal{E}, x_j$$

$$q_{i,j}(x_i, x_j) \geq 0 \quad \forall i, j \in \mathcal{E}, x_i, x_j$$

$$\sum_{x_i} q_i(x_i) = 1 \quad \forall i$$

locally consistent
marginal distributions

Variational derivation of BP

$$\arg\max_{\{q\}} \sum_{i,j\in\mathcal{E}} H(q_{i,j}) - \sum_i (|Nb_i| - 1)H(q_i) + \sum_{i,j\in\mathcal{E}} \sum_{x_{i,j}} q_{i,j}(x_i, x_j) \ln \phi_{i,j}(x_i, x_j)$$

$$\sum_{x_i} q_{i,j}(x_i, x_j) = q_j(x_j) \quad \forall i, j \in \mathcal{E}, x_j$$

$$q_{i,j}(x_i, x_j) \geq 0 \quad \forall i, j \in \mathcal{E}, x_i, x_j$$

$$\sum_{x_i} q_i(x_i) = 1 \quad \forall i$$

locally consistent
marginal distributions

BP update is derived as "fixed-points" of the Lagrangian

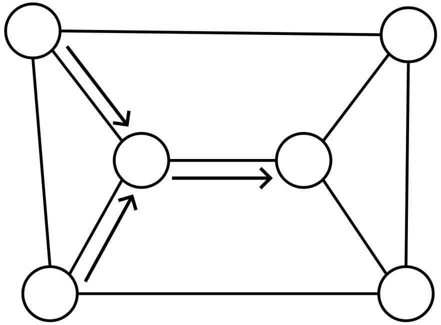
- BP **messages** are the (exponential form of the) **Lagrange multipliers**

What happens if there are loops?

We can still apply BP update:

$$\delta_{i\to j}(x_j) \propto \sum_{x_i} \psi_{i,j}(x_i, x_j) \prod_{k \in Nb_i - j} \delta_{k\to i}(x_k)$$

proportional to
normalize the message for numerical stability

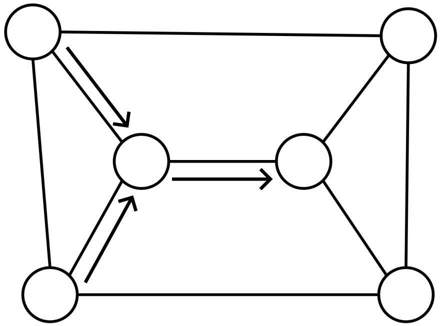


What happens if there are loops?

We can still apply BP update:

$$\delta_{i\to j}(x_j) \propto \sum_{x_i} \psi_{i,j}(x_i, x_j) \prod_{k\in Nb_i - j} \delta_{k\to i}(x_k)$$

proportional to
normalize the message for numerical stability



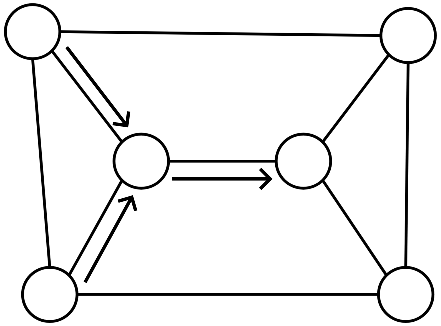
- update the messages **synchronously** or **sequentially**

What happens if there are loops?

We can still apply BP update:

$$\delta_{i\to j}(x_j) \propto \sum_{x_i} \psi_{i,j}(x_i, x_j) \prod_{k\in Nb_i - j} \delta_{k\to i}(x_k)$$

proportional to
normalize the message for numerical stability



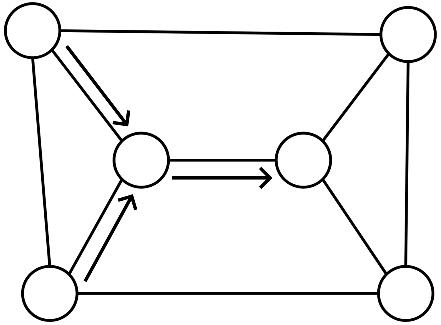
- update the messages **synchronously** or **sequentially**
- may not **converge** *(oscillating behavior)*

What happens if there are loops?

We can still apply BP update:

$$\delta_{i\to j}(x_j) \propto \sum_{x_i} \psi_{i,j}(x_i, x_j) \prod_{k\in Nb_i - j} \delta_{k\to i}(x_k)$$

proportional to
normalize the message for numerical stability



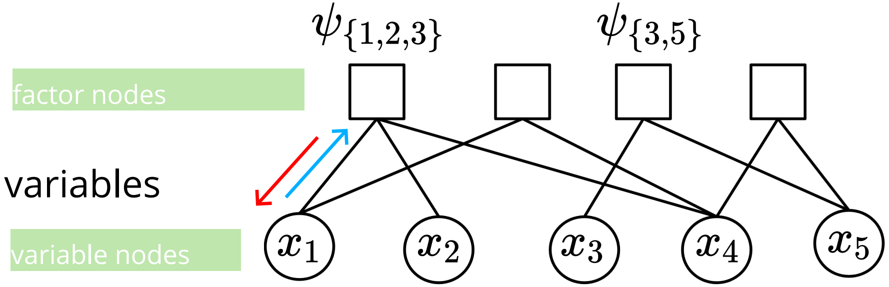
- update the messages **synchronously** or **sequentially**
- may not **converge** *(oscillating behavior)*
- even when convergent only gives an **approximation:**

$\hat{p}(x_i) \propto \prod_{k\in Nb_i} \delta_{k\to i}(x_i)$ is not (proportional to) the exact marginal $p(x_i)$

Loopy BP on factor graphs

$$p(\mathbf{x}) = \frac{1}{Z} \prod_I \psi_I(x_I)$$

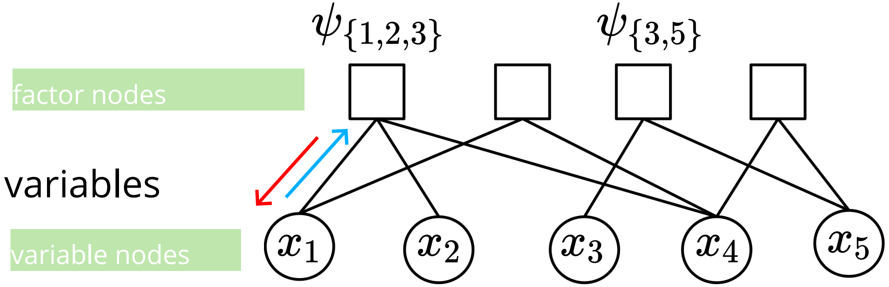
$I \subseteq \{1, \ldots, N\}$ is a subset of variables



Loopy BP on factor graphs

$p(\mathbf{x}) = \frac{1}{Z} \prod_I \psi_I(x_I)$

$I \subseteq \{1, \ldots, N\}$ is a subset of variables

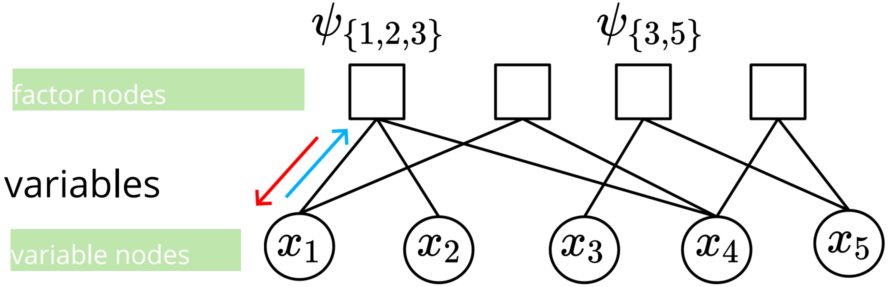


variable-to-factor message: $\delta_{i \to I}(x_i) \propto \prod_{J | i \in J, J \neq I} \delta_{J \to i}(x_i)$

Loopy BP on factor graphs

$p(\mathbf{x}) = \frac{1}{Z} \prod_I \psi_I(x_I)$

$I \subseteq \{1, \ldots, N\}$ is a subset of variables



variable-to-factor message: $\delta_{i \to I}(x_i) \propto \prod_{J | i \in J, J \neq I} \delta_{J \to i}(x_i)$

factor-to-variable message: $\delta_{I \to i}(x_i) \propto \sum_{x_{I-i}} \psi_I(x_I) \prod_{j \in I - i} \delta_{j \to I}(x_i)$

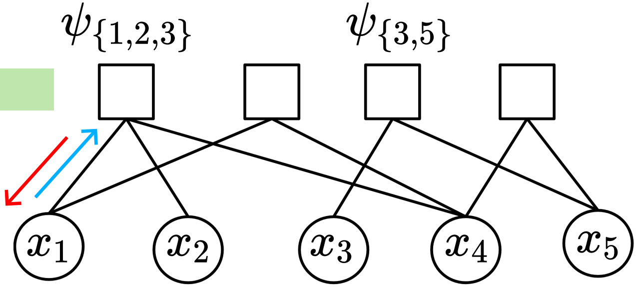
Loopy BP on factor graphs

$p(\mathbf{x}) = \frac{1}{Z} \prod_I \psi_I(x_I)$

$I \subseteq \{1, \ldots, N\}$ is a subset of variables

factor nodes

variable nodes



variable-to-factor message: $\delta_{i \to I}(x_i) \propto \prod_{J|i \in J, J \neq I} \delta_{J \to i}(x_i)$

factor-to-variable message: $\delta_{I \to i}(x_i) \propto \sum_{x_{I-i}} \psi_I(x_I) \prod_{j \in I-i} \delta_{j \to I}(x_i)$

after convergence: $\hat{p}(x_i) \propto \prod_{J|i \in J} \delta_{J \to i}(x_i)$

Loopy BP on factor graphs: complexity

variable-to-factor message:

- from each var to all neighbors

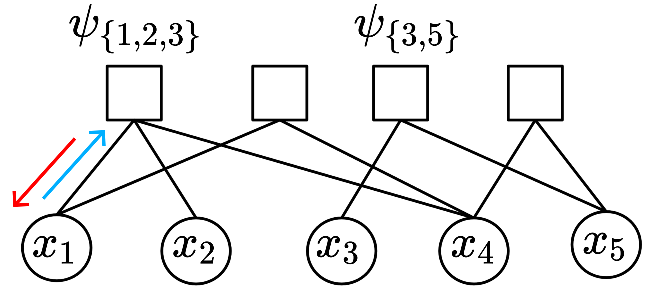
$$\delta_{i\to I}(x_i) \propto \prod_{J|i\in J, J\neq I} \delta_{J\to i}(x_i)$$

$$n d \Delta_{\max}{}^2$$

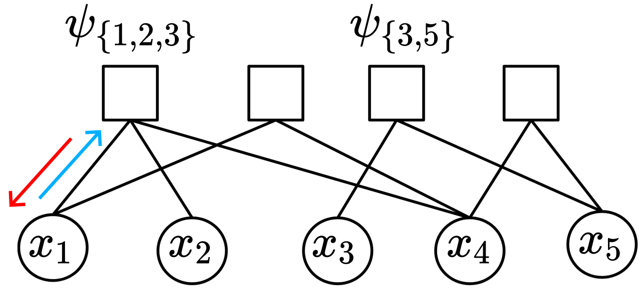
number of vars

domain size
(2 for binary)

max neighbours



Loopy BP on factor graphs: complexity



variable-to-factor message:

- from each var to all neighbors

$$\delta_{i\to I}(x_i) \propto \prod_{J|i\in J, J\neq I} \delta_{J\to i}(x_i)$$

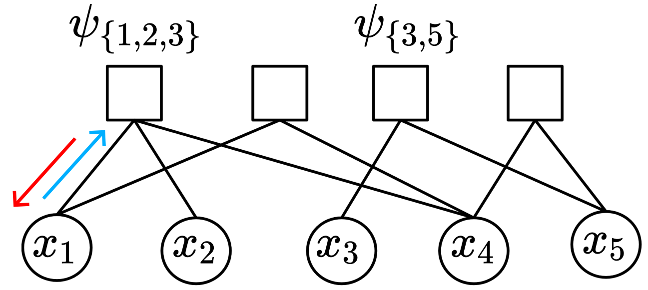
$$\underbrace{n}_{\text{number of vars}}\underbrace{d}_{\substack{\text{domain size}\\ \text{(2 for binary)}}}\underbrace{\Delta_{\max}}_{\text{max neighbours}}{}^{2}$$

factor-to-variable messages:

$$\underbrace{m}_{\text{number of factors}} d^{|\text{Scope}_{\max}|} \underbrace{|\text{Scope}_{\max}|}_{\text{vars in a factor}}$$

$$\delta_{I\to i}(x_i) \propto \sum_{x_{I-i}} \psi_I(x_I) \prod_{j\in I-i} \delta_{j\to I}(x_i)$$

Loopy BP on factor graphs: complexity



variable-to-factor message:

- from each var to all neighbors

$$\delta_{i\to I}(x_i) \propto \prod_{J|i\in J, J\neq I} \delta_{J\to i}(x_i)$$

$$nd\Delta_{\max}^{2}$$

number of vars

domain size
(2 for binary)

max neighbours

factor-to-variable messages:

$$md^{|\mathrm{Scope}_{\max}|}|\mathrm{Scope}_{\max}|$$

number of factors

vars in a factor

$$\delta_{I\to i}(x_i) \propto \sum_{x_{I-i}} \psi_I(x_I) \prod_{j\in I-i} \delta_{j\to I}(x_i)$$

(Loopy) BP has found many applications

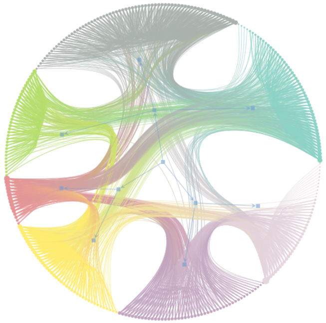
Machine Learning:

- clustering
- tensor factorization

www.jianxiongxiao.com

Vision:

- inpainting &denoising
- stereo matching



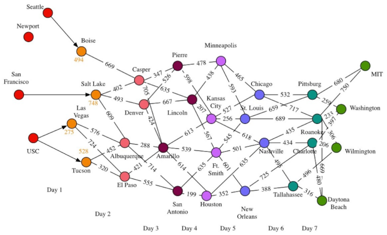
https://graph-tool.skewed.de

Social network analysis:

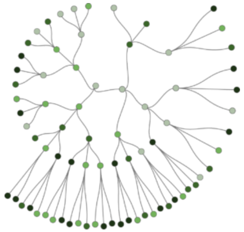
- stochastic block modelling

NLP and bioinformatics:

- Viterbi algorithm



Combinatorial optimization:



Application: LDPC coding using BP

low-density parity check

$x_1, \ldots, x_n$ are sent through a noisy channel

$y_1, \ldots, y_n$ are observerd

$$p(y_i = 1 \mid x_i = 1) = p(y_i = 0 \mid x_i = 0) = 1 - \epsilon$$

Application: LDPC coding using BP

low-density parity check

$x_1, \ldots, x_n$ are sent through a noisy channel

$y_1, \ldots, y_n$ are observerd

$p(y_i = 1 \mid x_i = 1) = p(y_i = 0 \mid x_i = 0) = 1 - \epsilon$

the message satisfies parity constraints:

$$\psi_{stu}(x_s, x_t, x_u) := \begin{cases} 1 & \text{if } x_s \oplus x_t \oplus x_u = 1 \\ 0 & \text{otherwise.} \end{cases}$$

Application: LDPC coding using BP

low-density parity check

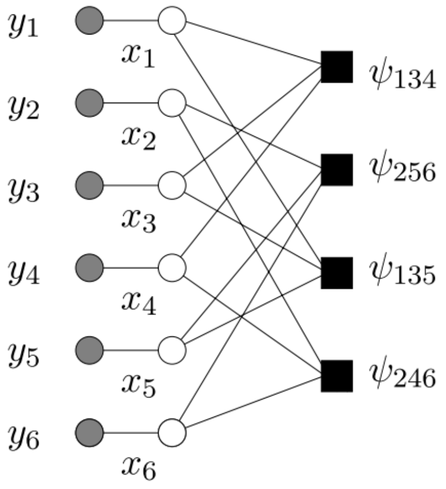
$x_1, \ldots, x_n$ are sent through a noisy channel

$y_1, \ldots, y_n$ are observerd

$p(y_i = 1 \mid x_i = 1) = p(y_i = 0 \mid x_i = 0) = 1 - \epsilon$

the message satisfies parity constraints:

$$\psi_{stu}(x_s, x_t, x_u) := \begin{cases} 1 & \text{if } x_s \oplus x_t \oplus x_u = 1 \\ 0 & \text{otherwise.} \end{cases}$$



joint dist. over unobserved message:

$$p(x \mid y) = \prod_{s,t,u} \psi(x_s, x_t, x_u) \prod_{i=1}^{n} (1 - \epsilon)\mathbb{I}(x_i = y_i) + \epsilon \mathbb{I}(x_i \neq y_i)$$

image: wainwright&jordan

Application: LDPC coding using BP

low-density parity check

joint dist. over unobserved message:

$$p(x \mid y) = \prod_{s,t,u} \psi(x_s, x_t, x_u) \prod_{i=1}^{n} (1-\epsilon)\mathbb{I}(x_i = y_i) + \epsilon\mathbb{I}(x_i \neq y_i)$$

inference problems

- most likely joint assignment

$$x^* = \arg\max_x p(x \mid y)$$

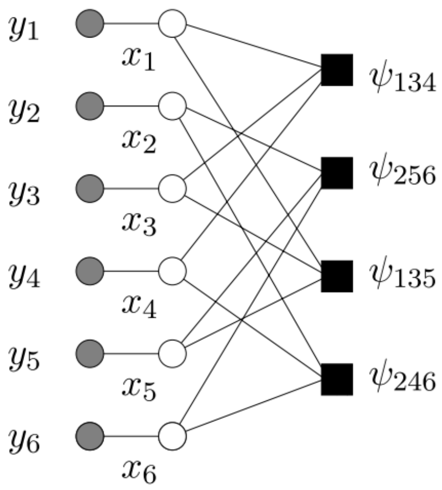


image: wainwright&jordan

Application: LDPC coding **using BP**

low-density parity check

joint dist. over unobserved message:

$$p(x \mid y) = \prod_{s,t,u} \psi(x_s, x_t, x_u) \prod_{i=1}^{n} (1-\epsilon)\mathbb{I}(x_i = y_i) + \epsilon \mathbb{I}(x_i \neq y_i)$$

inference problems

- most likely joint assignment

  $$x^* = \arg\max_x p(x \mid y)$$

- max-marginals $\quad x_i^* = \arg\max_{x_i} p(x_i \mid y)$
  - calculate the marginals $p(x_i \mid y) \forall i$
  - using **loopy BP**

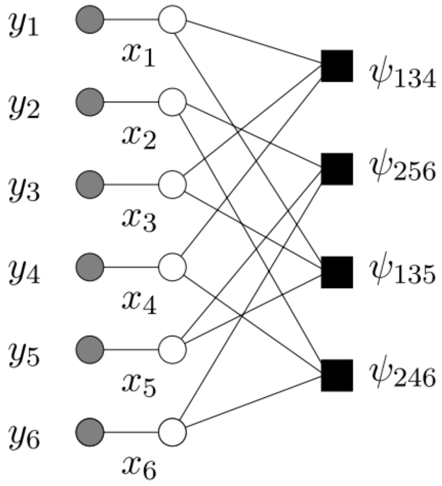


image: wainwright&jordan

Application: LDPC coding using BP

low-density parity check

joint dist. over unobserved message:

$$p(x \mid y) = \prod_{s,t,u} \psi(x_s, x_t, x_u) \prod_{i=1}^{n} (1-\epsilon)\mathbb{I}(x_i = y_i) + \epsilon\mathbb{I}(x_i \neq y_i)$$

inference problems

- Most likely **joint** assignment

$$x^* = \arg\max_x p(x \mid y)$$

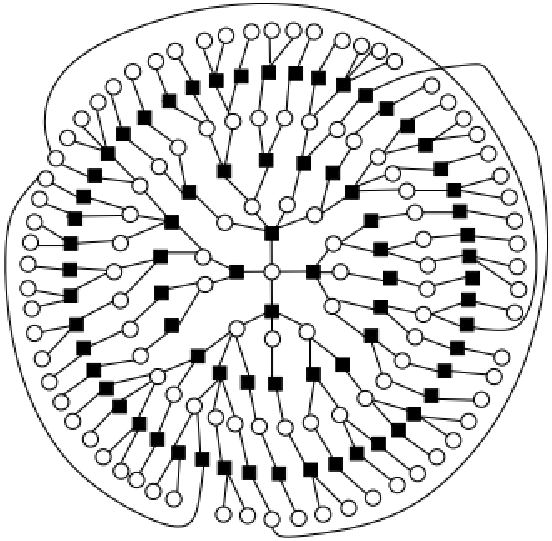


image: wainwright&jordan

Application: LDPC coding using BP

low-density parity check

joint dist. over unobserved message:

$$p(x \mid y) = \prod_{s,t,u} \psi(x_s, x_t, x_u) \prod_{i=1}^{n} (1-\epsilon)\mathbb{I}(x_i = y_i) + \epsilon\mathbb{I}(x_i \neq y_i)$$

inference problems

- Most likely **joint** assignment

$$x^* = \arg\max_x p(x \mid y)$$

- Max-marginals $x_i^* = \arg\max_{x_i} p(x_i \mid y)$
  - calculate the marginals $p(x_i \mid y) \forall i$
  - using **loopy BP**

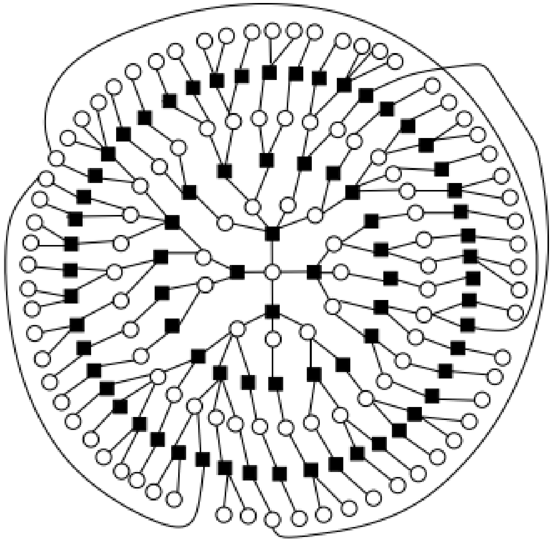


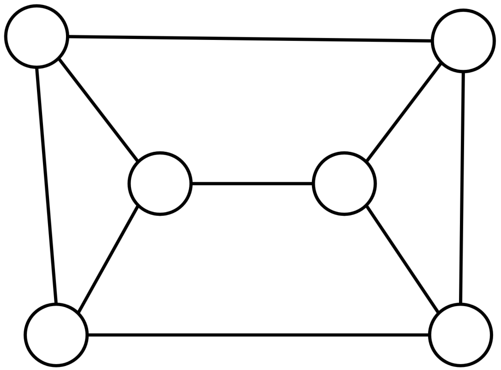
image: wainwright&jordan

Loops and variational interepretation

$$\arg\max_q H(q) + \mathbb{E}_q[\sum_{i,j} \ln \phi_{i,j}(x_i, x_j)]$$

$$\sum_{i,j\in\mathcal{E}} \sum_{x_{i,j}} q_{i,j}(x_i, x_j) \ln \phi_{i,j}(x_i, x_j)$$

$$\sum_{i,j\in\mathcal{E}} H(q_{i,j}) - \sum_i (|Nb_i| - 1) H(q_i)$$

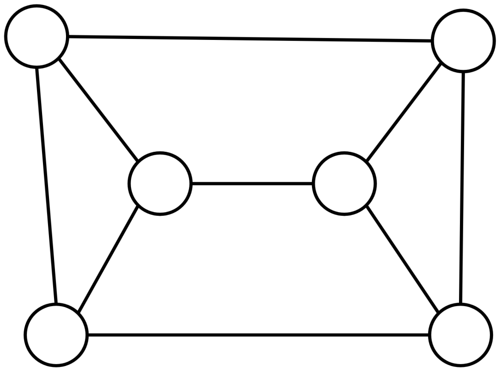


Loops and variational interepretation

$$\arg\max_q H(q) + \mathbb{E}_q[\textstyle\sum_{i,j} \ln \phi_{i,j}(x_i, x_j)]$$

$$\textstyle\sum_{i,j\in\mathcal{E}} \sum_{x_{i,j}} q_{i,j}(x_i, x_j) \ln \phi_{i,j}(x_i, x_j)$$

$$\textstyle\sum_{i,j\in\mathcal{E}} H(q_{i,j}) - \sum_i (|Nb_i| - 1) H(q_i)$$



the entropy term is not exact anymore

- called **Bethe approximation** to the entropy
- generally not convex anymore *(multiple fixed points)*

Loops and variational interepretation

$$\arg\max_q H(q) + \mathbb{E}_q[\textstyle\sum_{i,j} \ln \phi_{i,j}(x_i, x_j)]$$

$$\mathbb{L} : \boxed{\textstyle\sum_{x_i} q_{i,j}(x_i, x_j) = q_j(x_j) \quad \forall i, j \in \mathcal{E}, x_j}$$

Loops and variational interepretation

$$\arg\max_q H(q) + \mathbb{E}_q[\sum_{i,j} \ln \phi_{i,j}(x_i, x_j)]$$

$$\mathbb{L} : \boxed{\sum_{x_i} q_{i,j}(x_i, x_j) = q_j(x_j) \quad \forall i, j \in \mathcal{E}, x_j}$$

the entropy term is not exact anymore

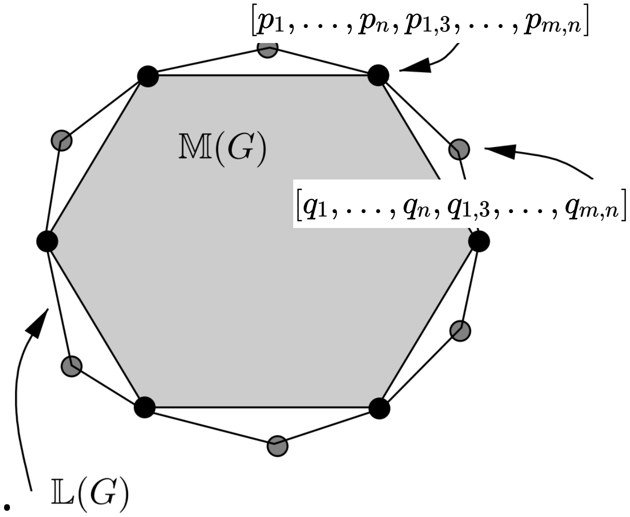
Local consistency constraints are inadequate:

- locally consistent $q_{i,j}, q_i$ may not be marginals for any joint dist.
  - i.e., local consistency polytope is an outer bound on the marginal polytope

Loops and variational interepretation

$$\arg\max_q H(q) + \mathbb{E}_q[\sum_{i,j} \ln \phi_{i,j}(x_i, x_j)]$$

$$\mathbb{L} : \sum_{x_i} q_{i,j}(x_i, x_j) = q_j(x_j) \quad \forall i, j \in \mathcal{E}, x_j$$



the entropy term is not exact anymore

Local consistency constraints are inadequate:

- locally consistent $q_{i,j}, q_i$ may not be marginals for any joint dist.
  - i.e., local consistency polytope is an outer bound on the marginal polytope

Variations on BP

$$\arg\max_q H(q) + \mathbb{E}_q[\sum_{i,j} \ln \phi_{i,j}(x_i, x_j)]$$

- the entropy term is not exact anymore:
  - improved entropy approximations (e.g., region-based, convex)
- local consistency constraints are inadequate
  - tighter constraints (e.g., marginal consistency of larger clusters)

Variations on BP: cluster-graph

cluster-graph generalizes clique-tree

- clusters are not necessarily max-cliques
- running intersection property
- family-preserving property
- $S_{i,j} \boxed{\subseteq} C_i \cap C_j$

instead of = in clique-tree

Variations on BP: cluster-graph

cluster-graph generalizes clique-tree

- clusters are not necessarily max-cliques
- running intersection property
- family-preserving property
- $S_{i,j} \boxed{\subseteq} C_i \cap C_j$

instead of = in clique-tree

similar reparametrization:

$$p(\mathbf{x}) \boxed{\propto} \frac{\prod_i \hat{p}(C_i)}{\prod_{i,j} \hat{p}(S_{i,j})}$$

instead of = in clique-tree

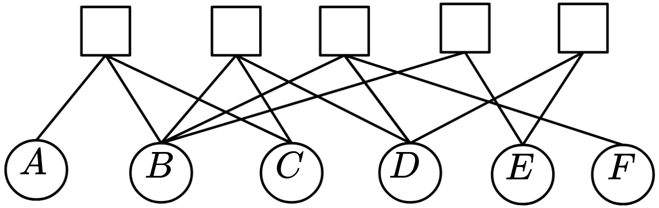
Variations on BP: cluster-graph

cluster-graph generalizes clique-tree

- clusters are not necessarily max-cliques
- running intersection property
- family-preserving property
- $\boxed{S_{i,j} \subseteq C_i \cap C_j}$

instead of = in clique-tree

a factor-graph



similar reparametrization:

$$p(\mathbf{x}) \propto \frac{\prod_i \hat{p}(C_i)}{\prod_{i,j} \hat{p}(S_{i,j})}$$

instead of = in clique-tree

Variations on BP: cluster-graph

cluster-graph generalizes clique-tree

- clusters are not necessarily max-cliques
- running intersection property
- family-preserving property
- $S_{i,j} \subseteq C_i \cap C_j$

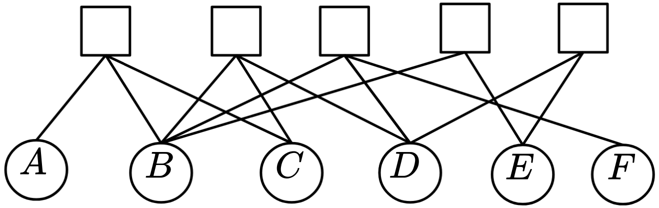
instead of = in clique-tree

similar reparametrization:

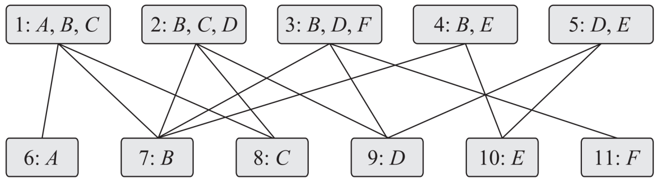
$$p(\mathbf{x}) \propto \frac{\prod_i \hat{p}(C_i)}{\prod_{i,j} \hat{p}(S_{i,j})}$$

instead of = in clique-tree

a factor-graph



corresponding cluster-graph (the same BP updates)



Variations on BP: cluster-graph

cluster-graph generalizes clique-tree

- clusters are not necessarily max-cliques
- running intersection property
- family-preserving property
- $S_{i,j} \subseteq C_i \cap C_j$

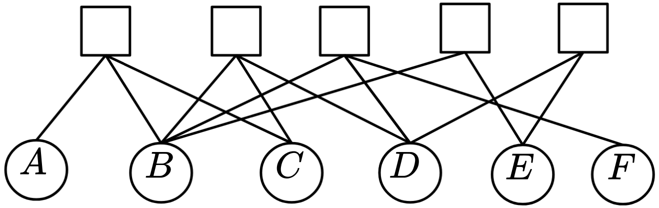
instead of = in clique-tree

similar reparametrization:

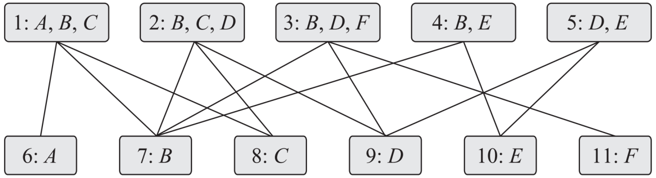
$$p(\mathbf{x}) \propto \frac{\prod_i \hat{p}(C_i)}{\prod_{i,j} \hat{p}(S_{i,j})}$$

instead of = in clique-tree

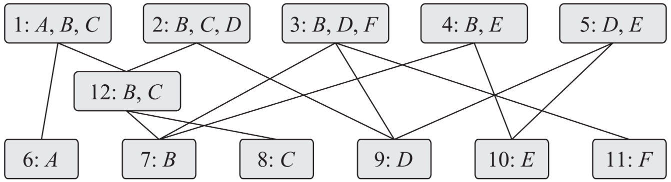
a factor-graph



corresponding cluster-graph (the same BP updates)

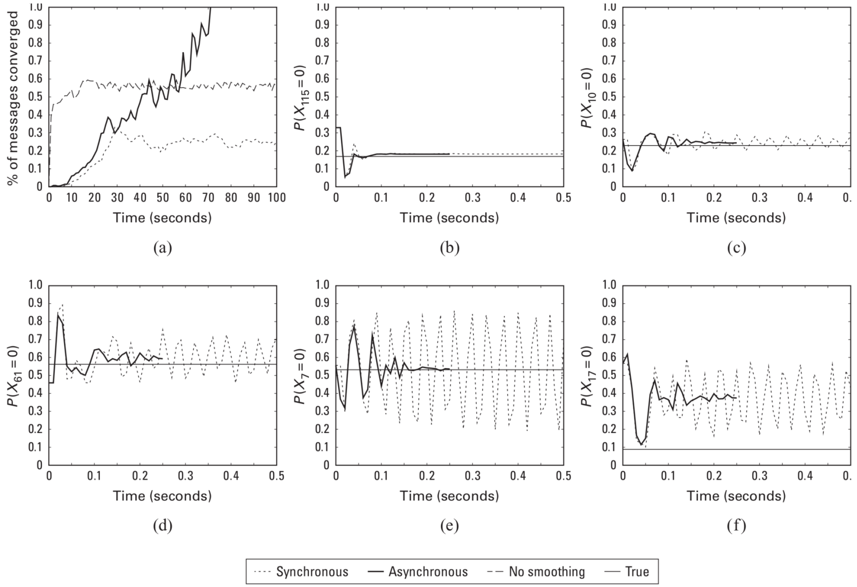


improved cluster-graph (better entropy approximation + marginal constraint)



BP in practice

11 x 11 Ising grid



- works well when:
  - locally tree-like graphs
  - dense graphs with weak interactions

- sequential update works better than parallel update

- improved convergence by damping (smoothing) the update

$$\delta_{i\to I}^{(t+1)}(x_i) \propto (1-\alpha)\delta_{i\to I}^{(t)}(x_i) + \alpha \prod_{J|i\in J, J\neq I} \delta_{J\to i}^{(t)}(x_i)$$

Summary

belief propagation: efficient deterministic inference

- exact in clique-tree = variable elimination
  - application of distributive law

Summary

belief propagation: efficient deterministic inference

- exact in clique-tree = variable elimination
  - application of distributive law
- optimization perspective:
  - KL-divergence minimization

Summary

belief propagation: efficient deterministic inference

- exact in clique-tree = variable elimination
  - application of distributive law
- optimization perspective:
  - KL-divergence minimization
- works well in (cluster) graphs with loops (large tree-width):
  - approximate objective (Bethe free energy) and constraints