

UNDERSTANDING BELIEF PROPOGATION AND ITS GENERALIZATIONS

JONATHAN YEDIDIA, WILLIAM FREEMAN, YAIR WEISS

2001– MERL TECH REPORT

Kristin Branson and Ian Fasel

June 11, 2003

1. Inference

- Inference problems in **networks** include:
 - **Conditional Probability Query**: What is $p(X_i | X_E = x_E^*)$?
 - **Most Probable Explanation**: What is $\operatorname{argmax}_{x_U} p(X_U = x_U | X_E = x_E^*)$?
- Inference problems also arise in **statistical physics**.
- In general, these problems are intractable.

1.1. Inference Approximations

Pearl's Belief Propagation (BP) algorithm solves inference

- **Exactly** for tree networks and
- **Approximately** for other networks.
- BP is not well-understood for loopy networks.

BP is closely connected to the Bethe approximation of statistical physics.

1.2. Purpose

Our goal in this paper is to

- Gain understanding of the BP approximation and
- How to make it more exact

by exploring equivalent approximations in statistical physics.

This has led to an improved inference algorithm, **Generalized BP**.

Outline

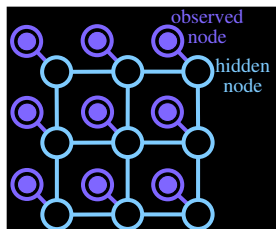
- The Settings.
 - Pairwise Markov Random Fields.
 - The Potts and Ising Models.
- The Approximations.
 - Belief Propagation.
 - The Mean-Field Approximation.
 - The Bethe Approximation.
 - The connection between the Bethe and BP approximations.
 - The Kikuchi Approximation.
- Generalized Belief Propagation.

2.

The Settings

2.1. Pairwise Markov Random Fields

A pairwise MRF is an undirected network with cliques of size two, e.g.:



With hidden variables x and observed variables y ,

$$p(x|y) = \frac{1}{Z} \prod_{(i,j)} \psi_{ij}(x_i, x_j) \prod_i \phi_i(x_i, y_i).$$

Any MRF can be converted into this form (Weiss, 2001).
We want to calculate $p(x_i|y)$ or $\operatorname{argmax}_x p(x|y)$.

2.2. The Potts Model

The pairwise MRF can be brought into a form recognizable to physicists as the **Potts model**:

$$p(x|y) = \frac{1}{Z} \exp \left[\left(\sum_{(i,j)} J_{ij}(x_i, x_j) + \sum_i h_i(x_i) \right) / T \right]$$

by setting

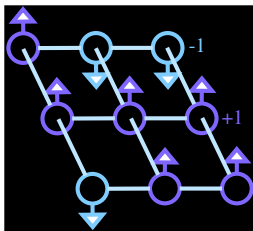
- The **interaction** $J_{ij}(x_i, x_j) = \lg \psi_{ij}(x_i, x_j)$,
- The **field** $h_i(x_i) = \lg \phi_i(x_i, y_i)$, and
- the temperature $T = 1$.

2.3. The Ising Model

The Ising model is a special case of the Potts model in which

- Each variable is **binary**,
- Interactions J_{ij} are **symmetric**,
- The distribution can be expressed as:

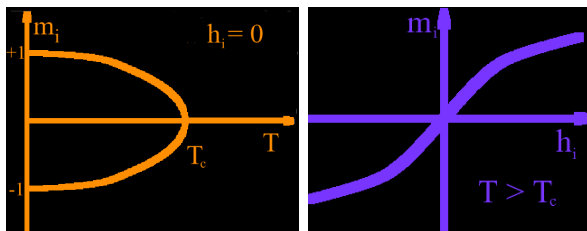
$$p(x) = \frac{1}{Z} \exp \left[\left(\sum_{(i,j)} J_{ij} x_i x_j + \sum_i h_i x_i \right) / T \right]$$



2.3.1. Use of the Ising Model

The Ising model was invented to describe **phase transitions in magnetics**.

- J encourages neighboring particle to have equal values.
- h represents an external magnetic field.
- The magnetization of a particle is its expected value, $m_i = p(x_i = +1) - p(x_i = -1)$.

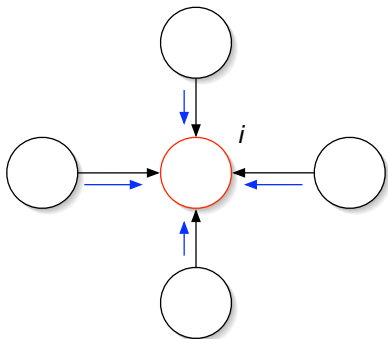


Phase transitions (**Veytsman and Kotelyanskii, 1997**).

3.

The Approximations

3.1. Standard Belief Propagation



Beliefs: $b_i(x_i)$

Messages: $m_{ji}(x_i)$

$$b_i(x_i) = k\phi_i(x_i) \prod_{j \in N(i)} m_{ji}(x_i)$$

The “belief” is the BP approximation of the marginal probability.

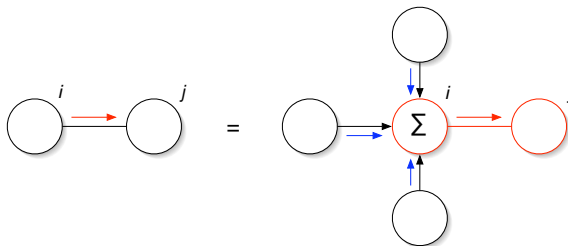
3.1.1. BP Message-update rules

To get marginal beliefs $b_i(x_i)$, sum over other variables:

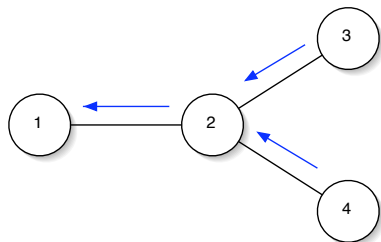
$$b_i(x_i) = \sum_{X_a \setminus x_i} b_a(X_a)$$

So we write the messages as:

$$m_{ij}(x_j) \leftarrow \sum_{x_i} \phi_i(x_i) \psi_{ij}(x_i, x_j) \prod_{k \in N(i) \setminus j} m_{ki}(x_i)$$



3.1.2. BP is Exact for Trees



$$\begin{aligned} b_i(x_i) &\propto m_{21}(x_1) \\ &\propto \sum_{x_2} \psi_{12}(x_1, x_2) m_{23}(x_3) m_{24}(x_4) \\ &\propto \sum_{x_2, x_3, x_4} \psi_{12}(x_1, x_2) \psi_{23}(x_2, x_3) \psi_{24}(x_2, x_4) \end{aligned}$$

3.2. Mean-Field Approximations

The true distribution $p(x|y)$ is approximated by the “closest” distribution $b(x)$ in the family \mathcal{Q} .

What is the definition of “closest”?

- The **Maximum-Likelihood** definition is

$$b^*(\cdot) = \operatorname{argmin}_{b(\cdot)} KL(p(\cdot|y) \parallel b(\cdot))$$

For reasonable \mathcal{Q} , computing $b^*(\cdot)$ is intractable.

- The **Mean-Field** definition is

$$b(\cdot) = \operatorname{argmin}_{b(\cdot)} KL(b(\cdot) \parallel p(\cdot|y)).$$

3.3. Why This Measure?

- For some reasonable \mathcal{Q} , we can compute $b(\cdot)$.
- If $p \in \mathcal{Q}$, we will exactly compute $b(x) = p(x)$.
- Minimizing $KL(b(\cdot) \parallel p(\cdot|y))$ yields the best lower bound on the log-likelihood $l(p; y)$:

$$\begin{aligned}l(p; y) &= \sum_x b(x) \lg p(y) \\&= \sum_x b(x) \lg \frac{p(x,y)b(x)}{b(x)p(x|y)} \\&= \sum_x b(x) \lg \frac{p(x,y)}{b(x)} + \sum_x b(x) \lg \frac{b(x)}{p(x|y)} \\&= \left\langle \lg \frac{p(x,y)}{b(x)} \right\rangle_b + KL(b(\cdot) \parallel p(\cdot|y)) \\&\geq \left\langle \lg \frac{p(x,y)}{b(x)} \right\rangle_b\end{aligned}$$

(Tanaka, 2001; Jaakkola, 2001)

3.4. The Gibbs Free Energy

In physics, this distance is the **Gibbs Free Energy**:

$$\begin{aligned} G(b(\cdot), y) &= KL(b(\cdot) \parallel p(\cdot|y)) \\ &= \sum_x b(x) E(x, y) + \sum_x b(x) \lg b(x) + \lg Z \end{aligned}$$

where the energy

$$E(x, y) = - \sum_{(i,j)} \lg \psi_{ij}(x_i, x_j) - \sum_i \lg \phi_i(x_i, y_i)$$

(thus $p(x|y) = \frac{1}{Z} \exp[-E(x, y)]$).

3.4.1. The Helmholtz Free Energy

Minimizing the Gibbs Free Energy is equivalent to minimizing the variational free energy

$$F(b(\cdot), y) = \sum_x b(x)E(x, y) + \sum_x b(x) \lg b(x).$$

If $b(\cdot) = p(\cdot|y)$, then the variational free energy achieves the **Helmholtz Free Energy**

$$F(y) = -\lg Z = \sum_x b(x)E(x, y) + \sum_x b(x) \lg b(x).$$

Otherwise, $F(b(\cdot), y)$ is an upperbound on $F(y)$.

Minimizing $G(b(\cdot), y)$ thus minimizes this upperbound (**Yedidia, 2001**).

3.5. The Naive MF Approximation

The naive MF approximation restricts \mathcal{Q} to be the set of distributions factorizable as

$$b(x) = \prod_i b_i(x_i).$$

The Gibbs free energy is then

$$\begin{aligned} G(b_i, y) &= - \sum_{(i,j)} \sum_{x_i, x_j} b_i(x_i) b_j(x_j) \lg \psi_{ij}(x_i, x_j) \\ &\quad - \sum_i \sum_{x_i} b_i(x_i) \lg \phi_i(x_i, y_i) \\ &\quad + \sum_i \sum_{x_i} b_i(x_i) \lg b_i(x_i) + \lg Z \end{aligned}$$

3.6. The Naive MF Algorithm

G_{MF} is minimized (subj to $\sum_{x_i} b_i(x_i) = 0$) by setting the derivative wrt $b_i(x_i)$ to 0, yielding

$$b_i(x_i) = \alpha \phi_i(x_i, y_i) \exp \left[\sum_{j \in N_i} \sum_{x_j} b_j(x_j) \lg \psi_{ij}(x_i, x_j) \right],$$

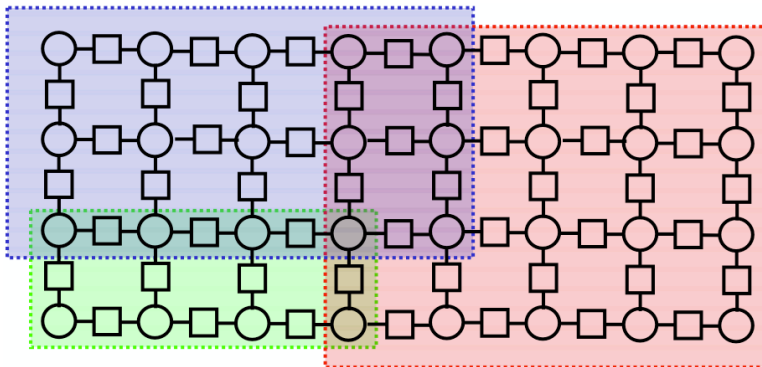
where α is a normalization constant.

Each estimate $b_i(x_i)$ is iteratively updated until convergence. $p(x_i|y)$ is approximated by the steady-state $b_i(x_i)$ (Weiss, 2001).

4. Region Based Inference Approximations

Naive MF restricts Q to distributions expressible in terms of **single-node beliefs**.

A better approximation is to restrict Q to distributions expressible in terms of **node-cluster beliefs**.



4.1. The Bethe Approximation

The Bethe approximation restricts Q to $b(x)$ that can be expressed in terms of the one- and two-node beliefs:

$$b(x) = \prod_{(i,j)} b_{ij}(x_i, x_j) \prod_i b_i(x_i)^{1-q_i}$$

The Gibbs free energy is then:

$$G_{Bethe} = \sum_{x_i, x_j} \sum_{x_i, x_j} b_{ij}(x_i, x_j) (E_{ij}(x_i, x_j) + \ln b_{ij}(x_i, x_j)) \\ + \sum_i (1 - q_i) \sum_{x_i} b_i(x_i) \ln b_i(x_i)$$

4.1.1. Minimizing the Bethe Free Energy

Minimize by constrained optimization using Lagrange multipliers.

$$L = G_{Bethe} + \sum_i \lambda_i \left\{ \sum_{x_i} b_i(x_i) = 1 \right\} \\ + \sum_a \sum_{i \in N(a)} \sum_{x_i} \lambda_{ai}(x_i) \left\{ \sum_{X_a \setminus x_i} b_a(X_a) = b_i(x_i) \right\}$$

Results in belief equations:

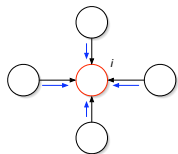
$$\frac{\partial L}{\partial b_i(x_i)} = 0 \Rightarrow b_i(x_i) \propto \exp \left(\frac{1}{d_i - 1} \sum_{a \in N(i)} \lambda_{ai}(x_i) \right) \\ \frac{\partial L}{\partial b_{ai}(x_a)} = 0 \Rightarrow b_a(X_a) \propto \exp \left(-E_a(X_a) + \sum_{a \in N(i)} \lambda_{ai}(x_i) \right)$$

4.1.2. Equivalence of BP to the Bethe Free Energy

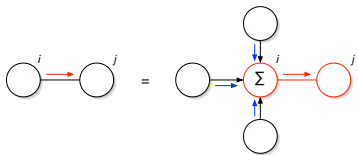
Identify:

$$\lambda_{i,j}(x_j) = \ln \prod_{k \in N(j) \setminus i} m_{kj}(x_j)$$

To obtain BP equations:



$$b_i(X_i) \propto \prod_{a \in N(i)} m_{a \rightarrow i}(x_i)$$



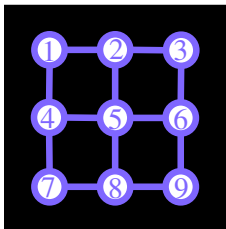
$$m_{ij}(x_j) \leftarrow \sum_{x_i} \phi_i(x_i) \psi_{ij}(x_i, x_j) \prod_{k \in N(i) \setminus j} m_{ki}(x_i)$$

4.2. Generalizing the Bethe Approximation

The Bethe approximation restricts Q to $b(x)$ that can be expressed in terms of the one- and two-node beliefs:

$$b(x) = \prod_{(i,j)} b_{ij}(x_i, x_j) \prod_i b_i(x_i)^{1-q_i}.$$

Another way to write the Bethe Approximation:
Let $\mathcal{C} = \{\{i, j\}\}$.

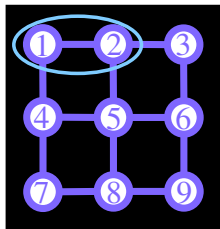


4.2. Generalizing the Bethe Approximation

The Bethe approximation restricts Q to $b(x)$ that can be expressed in terms of the one- and two-node beliefs:

$$b(x) = \prod_{(i,j)} b_{ij}(x_i, x_j) \prod_i b_i(x_i)^{1-q_i}.$$

Another way to write the Bethe Approximation:
Let $\mathcal{C} = \{\{i, j\}\}$.

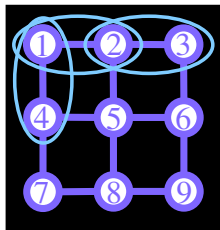


4.2. Generalizing the Bethe Approximation

The Bethe approximation restricts Q to $b(x)$ that can be expressed in terms of the one- and two-node beliefs:

$$b(x) = \prod_{(i,j)} b_{ij}(x_i, x_j) \prod_i b_i(x_i)^{1-q_i}.$$

Another way to write the Bethe Approximation:
Let $\mathcal{C} = \{\{i, j\}\}$.

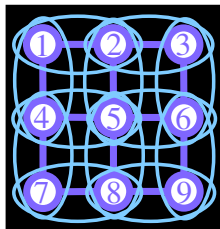


4.2. Generalizing the Bethe Approximation

The Bethe approximation restricts Q to $b(x)$ that can be expressed in terms of the one- and two-node beliefs:

$$b(x) = \prod_{(i,j)} b_{ij}(x_i, x_j) \prod_i b_i(x_i)^{1-q_i}.$$

Another way to write the Bethe Approximation:
Let $\mathcal{C} = \{\{i, j\}\}$.



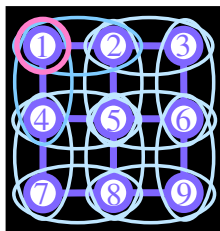
4.2. Generalizing the Bethe Approximation

The Bethe approximation restricts Q to $b(x)$ that can be expressed in terms of the one- and two-node beliefs:

$$b(x) = \prod_{(i,j)} b_{ij}(x_i, x_j) \prod_i b_i(x_i)^{1-q_i}.$$

Another way to write the Bethe Approximation:

Let $\mathcal{C} = \{\{i, j\}\}$ and $R = \{C_i\} \cup \{C_i \cap C_j\}$.



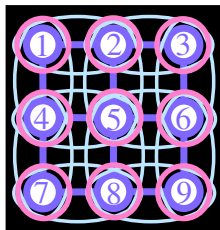
4.2. Generalizing the Bethe Approximation

The Bethe approximation restricts Q to $b(x)$ that can be expressed in terms of the one- and two-node beliefs:

$$b(x) = \prod_{(i,j)} b_{ij}(x_i, x_j) \prod_i b_i(x_i)^{1-q_i}.$$

Another way to write the Bethe Approximation:

Let $\mathcal{C} = \{\{i, j\}\}$ and $R = \{C_i\} \cup \{C_i \cap C_j\}$.



4.2. Generalizing the Bethe Approximation

The Bethe approximation restricts \mathcal{Q} to $b(x)$ that can be expressed in terms of the one- and two-node beliefs:

$$b(x) = \prod_{(i,j)} b_{ij}(x_i, x_j) \prod_i b_i(x_i)^{1-q_i}.$$

Another way to write the Bethe Approximation:

Let $\mathcal{C} = \{\{i, j\}\}$ and $R = \{C_i\} \cup \{C_i \cap C_j\}$.

Define $c_r = 1 - \sum_{s \in \text{super}(r)} c_s$ and $\text{super}(r) = \{s \mid r \subset s\}$.

- If $r = \{i, j\}$, $\text{super}(r) = \{\}$, so $c_r = 1$.
- If $r = \{i\}$, $\text{super}(r) = \{\{i, \cdot\}\}$, so $c_r = 1 - q_i$.

Thus,

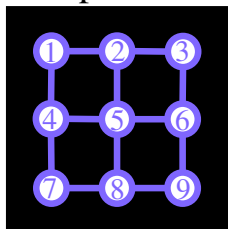
$$b(x) = \prod_{r \in R} b_r(x_r)^{c_r},$$

4.3. Kikuchi Approximations

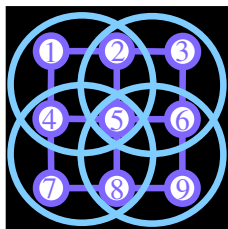
The Kikuchi approximation allows \mathcal{C} to be *any* set of clusters of nodes s.t. every edge and node of the MRF is in some cluster, and sets

$$R = \mathcal{C} \cup \{C_i \cap C_j\} \cup \{(C_i \cap C_j) \cap (C_k \cap C_l)\} \cup \dots$$

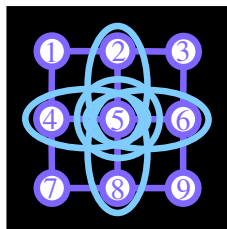
Example:



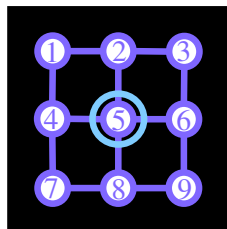
Original MRF



\mathcal{C}



$\{C_i \cap C_j\}$



$\{(C_i \cap C_j) \cap (C_k \cap C_l)\}$

4.3. Kikuchi Approximations

The Kikuchi approximation restricts \mathcal{Q} to $b(x)$ that can be expressed in terms of cluster beliefs:

$$b(x) = \prod_{r \in R} b_r(x_r)^{1-c_r}.$$

$p(\cdot|y)$ is approximated by

$$b(\cdot) = \operatorname{argmin}_{b(\cdot) \in \mathcal{Q}} KL(b(\cdot) \parallel p(\cdot|y)).$$

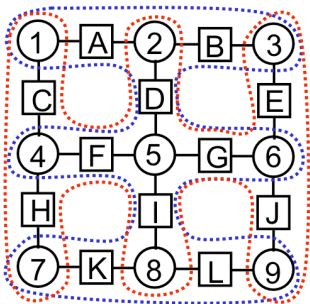
The Kikuchi approximation can be made exact, in which case it is equivalent to the junction tree algorithm.

4.4. Other Approximations

In general, it is possible to define a set of rules for constructing “valid” region approximations.

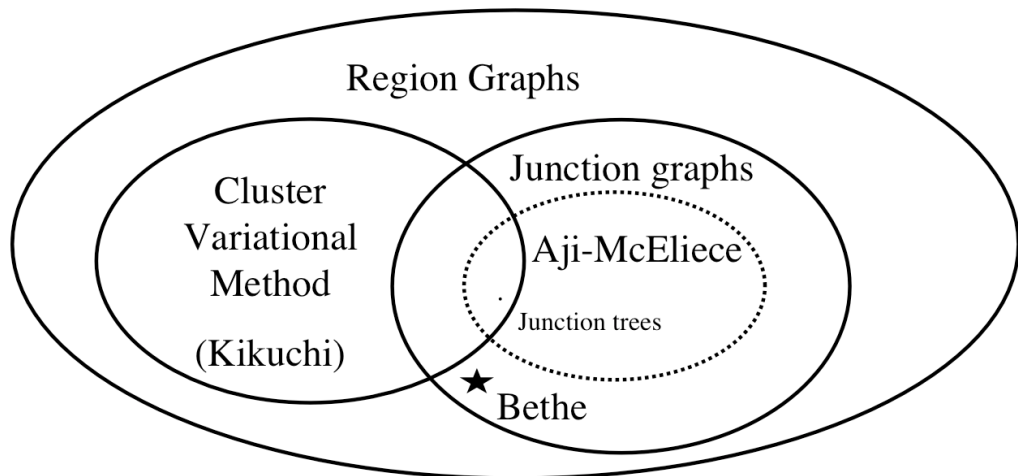
- Bethe restricts regions to neighboring pairs
- Kikuchi restricts regions to be connected clusters of nodes

Example of a region that cannot be created using other methods:



Many others...!

4.4. Other Approximations



Other region based approximations exist, given a few rules for constructing “valid” approximations (involving counting numbers, etc.)

5. Generalizing BP

Given a valid region-based approximation, we can construct a generalized belief propagation (GBP) algorithm.

Belief in a region is the product of:

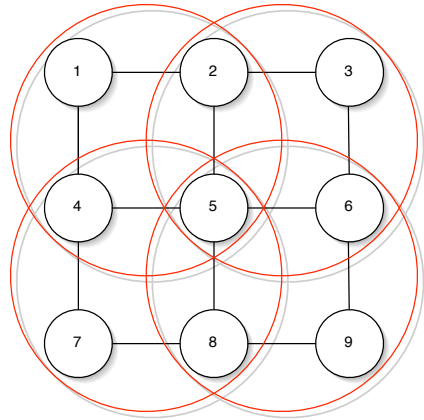
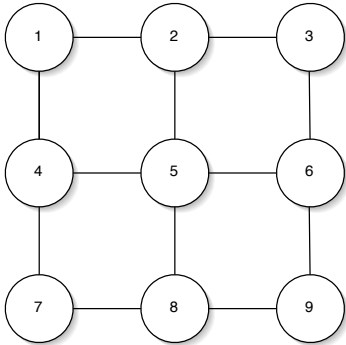
- Local information (factors in a region)
- Messages from parent regions
- Messages into descendant regions from parents who are not descendants

Message update rule obtained by enforcing marginalization constraints.

We do this for Cluster Variational Method (Kikuchi) as an example...

5.1. Constructing clusters

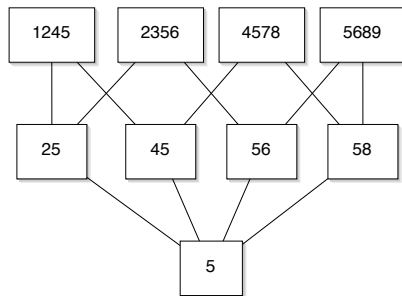
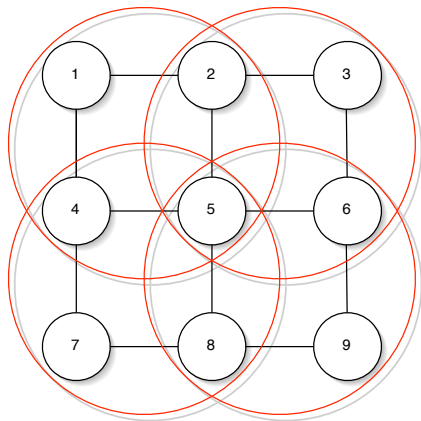
First, construct clusters and find intersection regions.



(assign “counting numbers” at this point)

5.2. Construct *region graph*

A hierarchy of regions and their “direct” sub-regions



(This example is the region graph using Kikuchi method.)

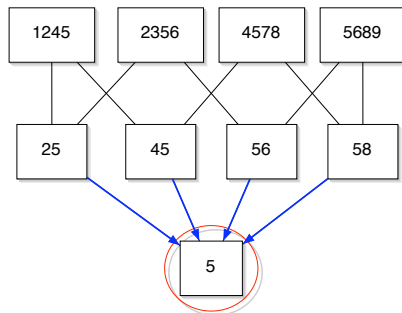
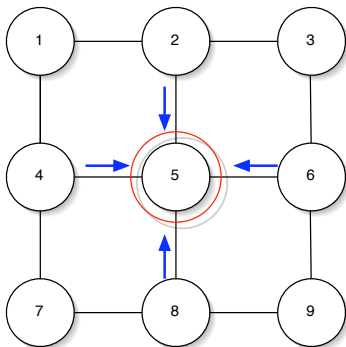
5.3. Construct Belief Equations

Belief equations for every region r are proportional to the product of

- local information (factors in a region)
- messages from parent regions
- messages into descendant regions from parents who are not descendants

5.3.1. Constructing Belief Equations

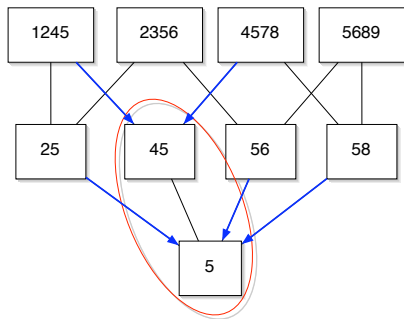
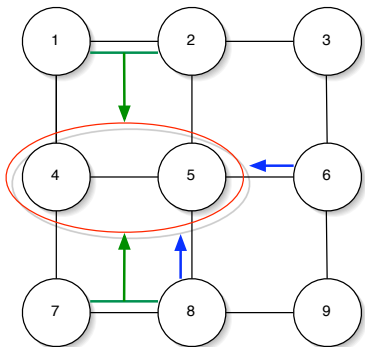
Step 3: Construct belief equations for each region



$$b_5(x_5) = k[\phi_5][m_{2 \rightarrow 5} m_{4 \rightarrow 5} m_{6 \rightarrow 5} m_{8 \rightarrow 5}]$$

5.3.2. Constructing Belief Equations

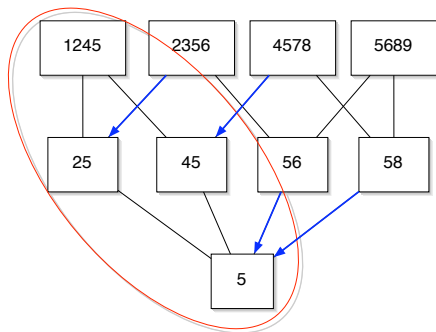
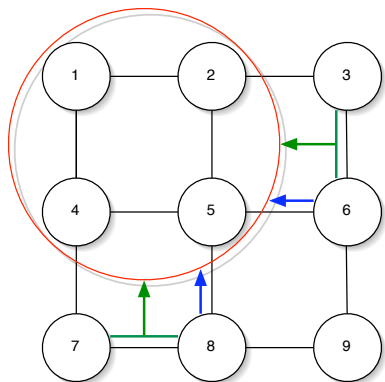
Step 3: Construct belief equations for each region



$$b_{45} = k[\phi_4\phi_5\psi_{45}][m_{12\rightarrow45}m_{78\rightarrow45}m_{2\rightarrow5}m_{6\rightarrow5}m_{8\rightarrow5}]$$

5.3.3. Constructing Belief Equations

Step 3: Construct belief equations for each region



$$b_{1245} = k[\phi_1\phi_2\phi_4\phi_5\psi_{12}\psi_{14}\psi_{25}\psi_{45}][m_{36\rightarrow 25}m_{78\rightarrow 45}m_{6\rightarrow 5}m_{8\rightarrow 5}]$$

5.4. Message Update Rule

Use marginalization constraints to define message update rules.

$$b_5(x_5) = \sum_{x_4} b_{45}(x_4, x_5)$$

Combining previous belief equations:

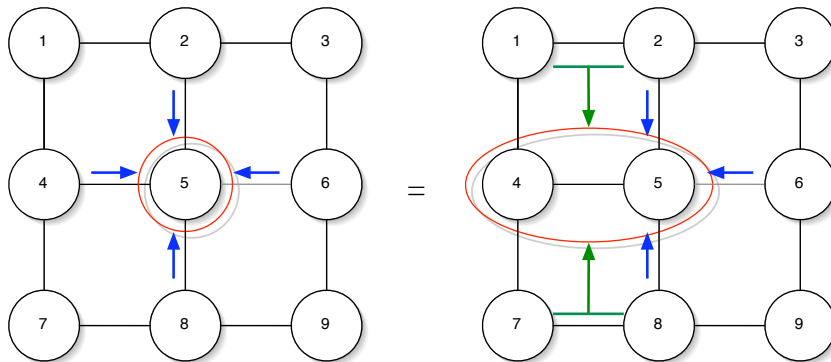
$$\begin{aligned} & k_1[\phi_5][m_{2 \rightarrow 5} m_{4 \rightarrow 5} m_{6 \rightarrow 5} m_{8 \rightarrow 5}] \\ = & \sum_{x_4} k_2[\phi_4 \phi_5 \psi_{45}][m_{12 \rightarrow 45} m_{78 \rightarrow 45} m_{2 \rightarrow 5} m_{6 \rightarrow 5} m_{8 \rightarrow 5}] \end{aligned}$$

Gives us:

$$m_{4 \rightarrow 5}(x_5) = k \sum_{x_4} \phi_4(x_4) \psi_{45}(x_4, x_5) m_{12 \rightarrow 45}(x_4, x_5) m_{78 \rightarrow 25}(x_2, x_5)$$

5.4.1. Message Update Rule

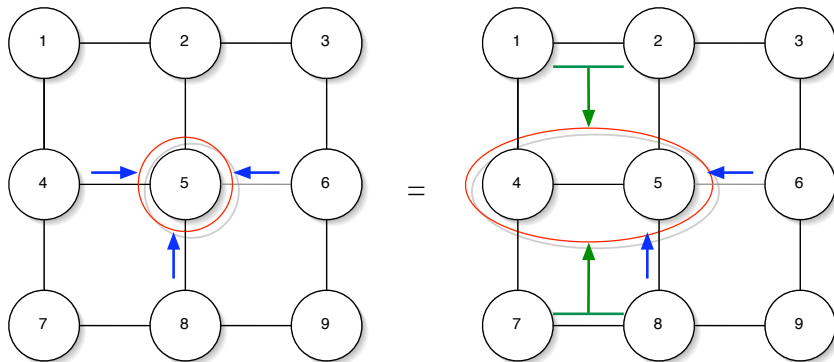
Use marginalization constraints to define message update rules:



$$b_5 = \sum_{x_4} b_{45}(x_4, x_5)$$

5.4.2. Message Update Rule

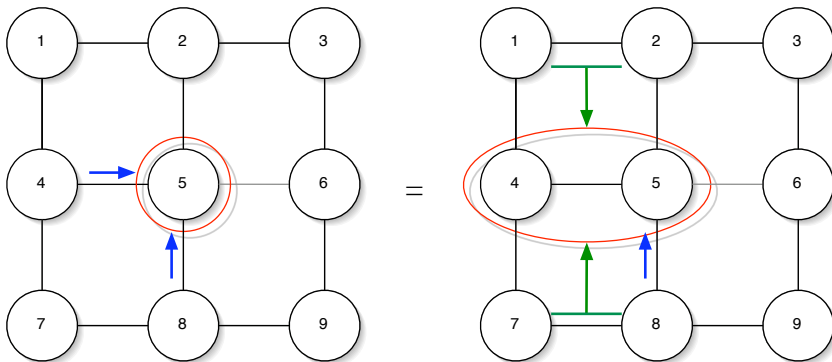
Use marginalization constraints to define message update rules:



$$b_5 = \sum_{x_4} b_{45}(x_4, x_5)$$

5.4.3. Message Update Rule

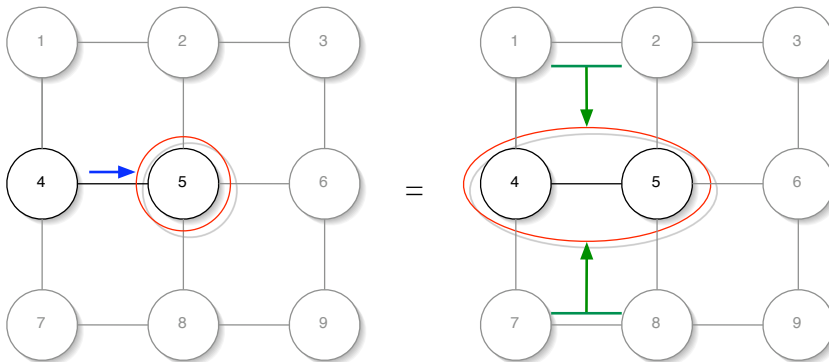
Use marginalization constraints to define message update rules:



$$b_5 = \sum_{x_4} b_{45}(x_4, x_5)$$

5.4.4. Message Update Rule

Use marginalization constraints to define message update rules:



$$m_{4 \rightarrow 5}(x_5) = k \sum_{x_4} \phi_4(x_4) \psi_{45}(x_4, x_5) m_{12 \rightarrow 45}(x_4, x_5) m_{78 \rightarrow 25}(x_2, x_5)$$

5.5. Run GBP algorithm

The GBP algorithm runs in the same way as BP:

1. Initialize messages to unbiased states
2. Iterate through message update rules until (hopefully) convergence

Occasionally, helpful to move only part-way to new values of messages at each iteration.

5.6. Generalized Belief Propagation

- Theorems:
 - Stationary points of Bethe approximation are fixed points of BP
 - Stationary points of Kikuchi approximation are fixed points of GBP.

 - When region graph is a tree, message-passing algorithm is exact
- Empirically, more likely to converge than BP
- Can be nearly as fast, but much more accurate (depending on details of region graph)

6. Conclusions

- Standard BP is equivalent to minimizing Bethe Free Energy.
- The Bethe and Junction Tree approximations are both sub-classes of the Kikuchi approximation.
- GBP is equivalent to minimizing the Kikuchi Free Energy.
- GBP is exact when region graph is a tree.

References

Jaakkola, T. (2001). Tutorial on variational approximation methods. In Opper, M. and Saad, D., editors, *Advanced Mean Field Methods: Theory and Practice*, chapter 10, pages 129–160. MIT Press.

Kappen, H. and Wiegerninck, W. (2001). Mean field theory for graphical models. In Opper, M. and Saad, D., editors, *Advanced Mean Field Methods: Theory and Practice*, chapter 4, pages 37–50. MIT Press.

Opper, M. and Winther, O. (2001). From naive mean field theory to the tap equations. In Opper, M. and Saad, D., editors, *Advanced Mean Field Methods: Theory and Practice*, chapter 2, pages 7–20. MIT Press.

Tanaka, T. (2001). Information geometry of mean-field approximation. In Opper, M. and Saad, D., editors, *Advanced Mean Field Methods: Theory and Practice*, chapter 17, pages 259–273. MIT Press.

Veytsman, B. and Kotelyanskii, M. (1997). Website: Ising model and its applications at <http://www.plmsc.psu.edu/www/matsc597c-1997/phases/lecture3/>.

- Weiss, Y. (2001). Comparing the mean field method and belief propagation for approximate inference in mrfs. In Opper, M. and Saad, D., editors, *Advanced Mean Field Methods: Theory and Practice*, chapter 15, pages 229–240. MIT Press.
- Yedidia, J. (2001). An idiosyncratic journey beyond mean field theory. In Opper, M. and Saad, D., editors, *Advanced Mean Field Methods: Theory and Practice*, chapter 3, pages 21–36. MIT Press.