Belief Propagation

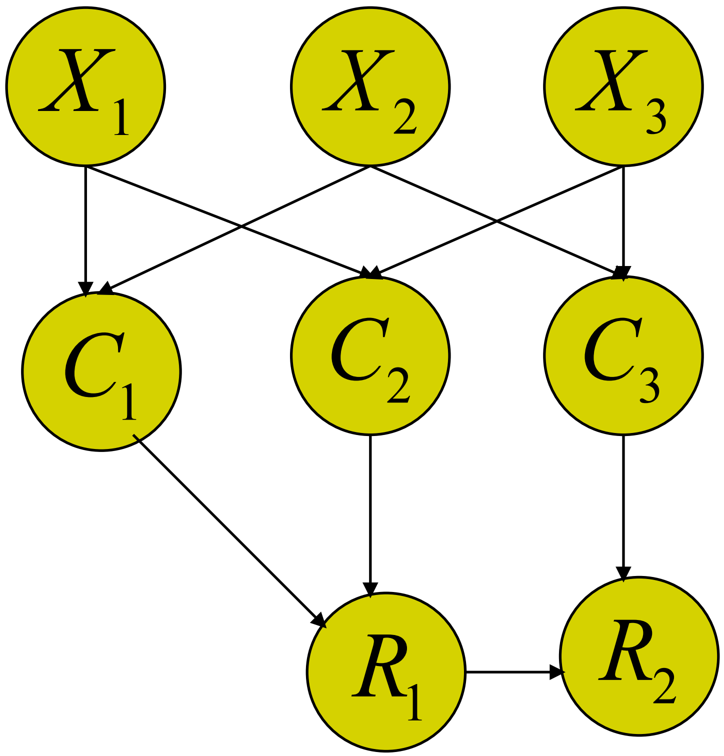
What is Belief Propagation (BP)?

BP is a specific instance of a general class of methods that exist for **approximate inference** in Bayes Nets (**variational methods**).

Simplified Bayes Net is the key idea of BP.

Simplification yields **faster/tractable inference at the cost of accuracy**.

An Example & Motivation

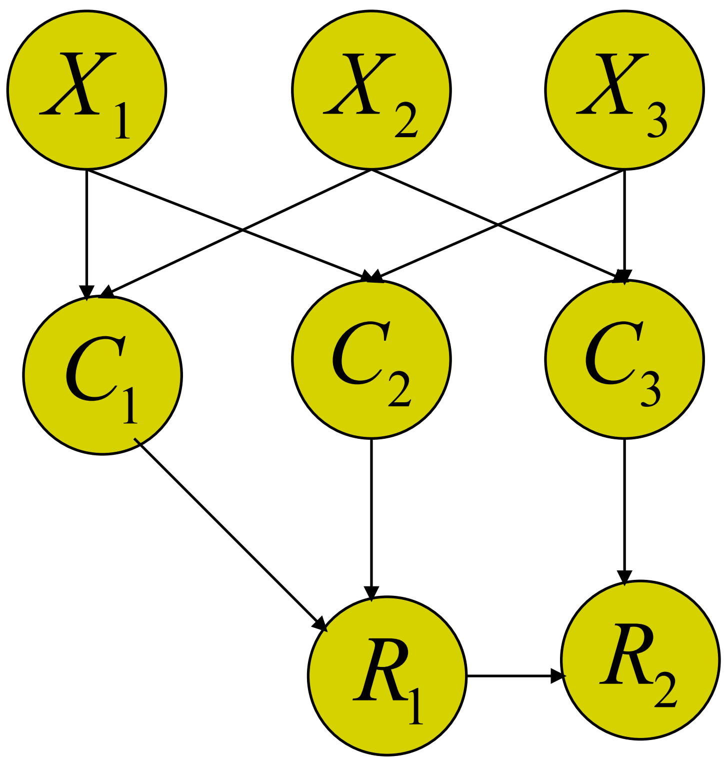


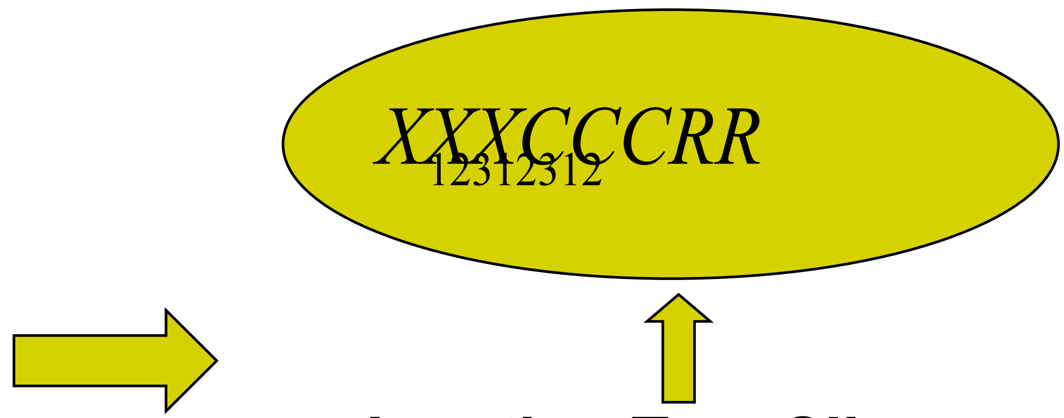
2-SAT problem as a Bayes Net.

Try applying Junction Tree Algorithm and …

An Example & Motivation (contd.)

...and Junction Tree Algorithm yields :





Junction Tree Clique

We get one huge clique.

Same as having a full joint table.

Defeats purpose of Bayes Net and so …

Accuracy Sacrifice = Possible Solution (Belief Propagation)

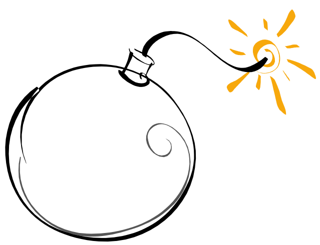
… Belief Propagation (BP) to the rescue

Two main steps :

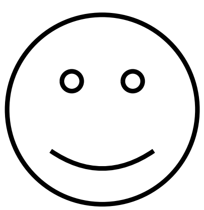
(1) Simplified Graph Construction

(2) Message Passing until convergence

Simplification? So what?

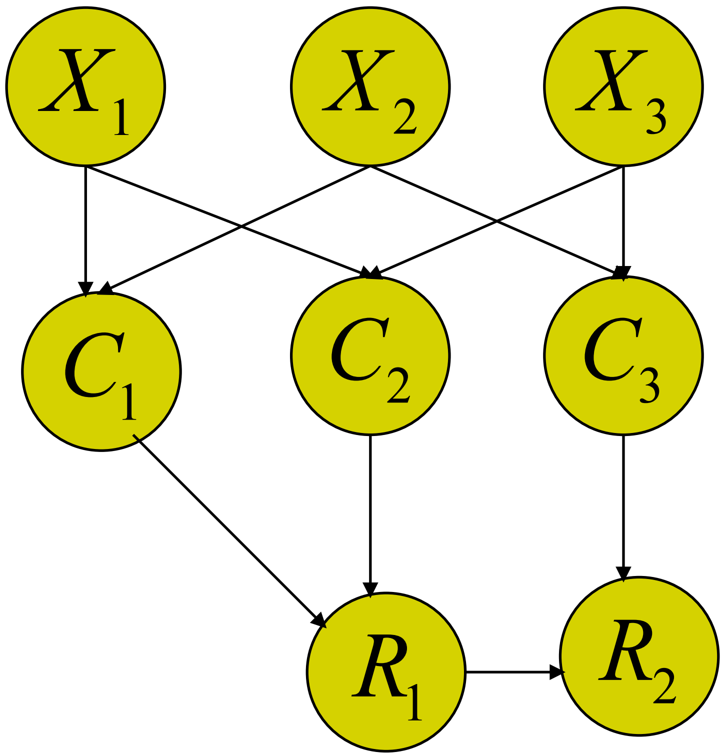


Caveat : BP may not converge.



Good News : Seems to work well in practice.

Simplified Graph Construction



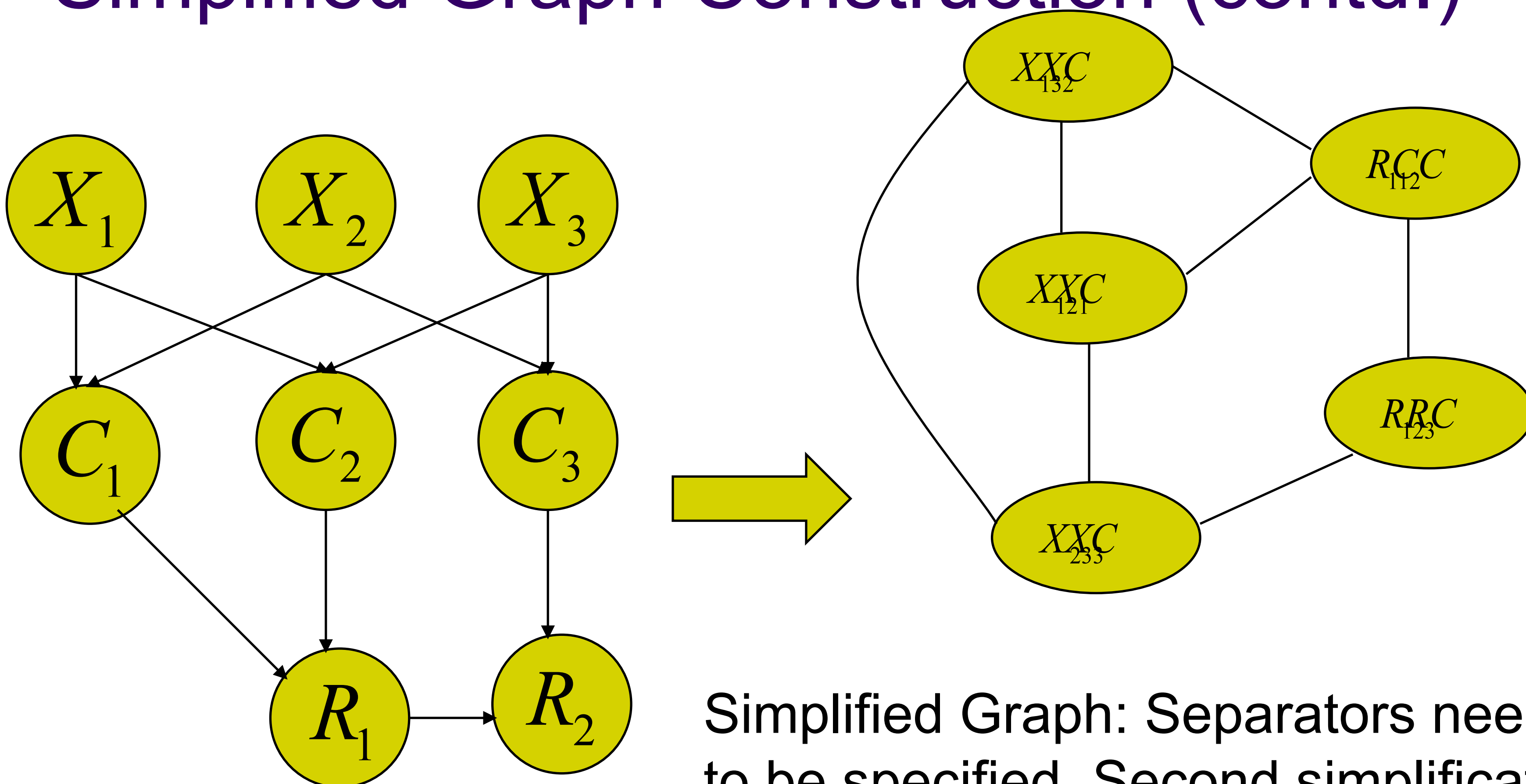
We will build a "clique" graph similar to Junction Tree Algorithm, but …

… without triangulation, and …

… need to have a "home" for all CPTs

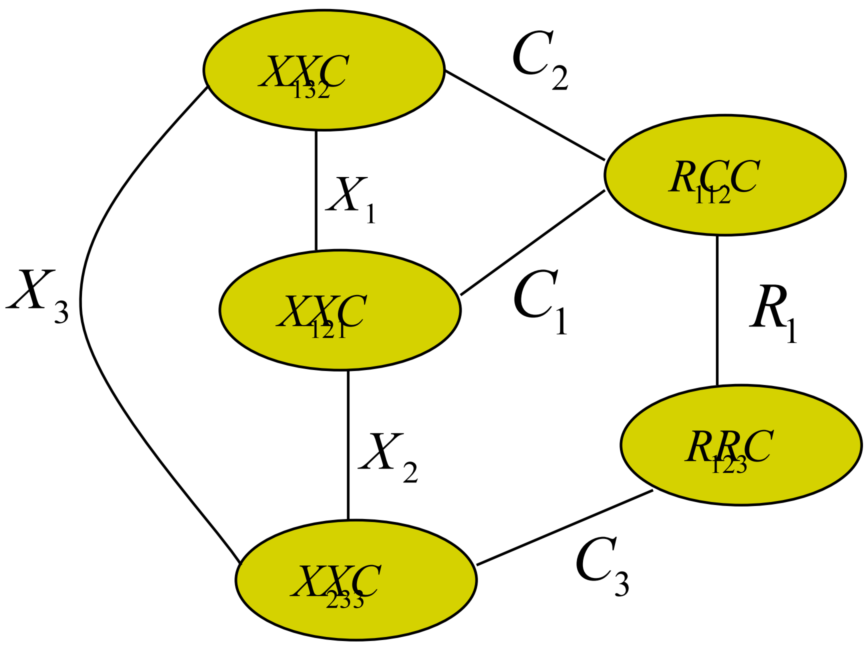
The simplified graph is …

Simplified Graph Construction (contd.)



Simplified Graph: Separators need to be specified. Second simplification is that the connecting arc need not have all the separator variables. By doing this we get …

Simplified Graph Construction (contd.)

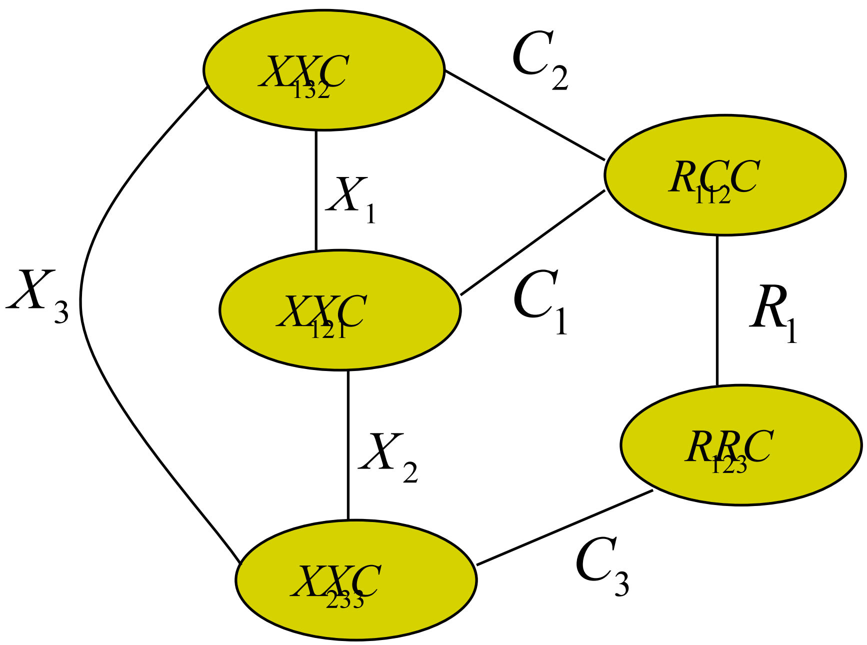


Here all separator variables are specified. This is a specific flavor of BP called **Loopy Belief Propagation** (LBP).

Loops are allowed in LBP.

Now we need to do …

Message Passing



Pass messages, just as in Junction Tree Algorithm.

Messages are nothing but CPTs marginalized down to the separator variable(s).

Graphical Models

- Diagrams
  - Nodes: random variables
  - Edges: statistical dependencies among random variables

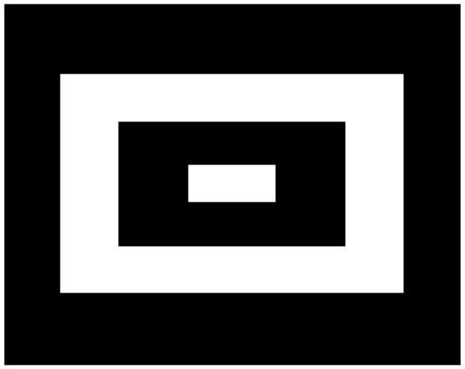
- Advantages:
  1. Better visualization
     - conditional independence properties
     - new models design
  2. Factorization

Graphical Models types

- Directed
  - causal relationships
  - e.g. Bayesian networks

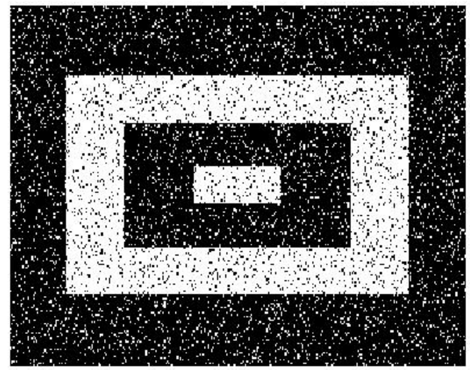
- Undirected
  - no constraints imposed on causality of events ("weak dependencies")
  - Markov Random Fields (MRFs)

Example MRF Application: Image Denoising



Original image

(Binary)



Noisy image

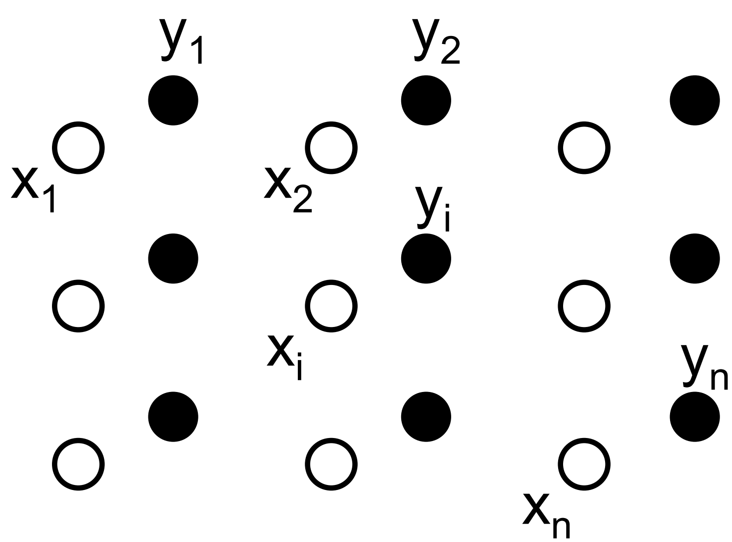
e.g. 10% of noise

- Question: How can we retrieve the original image given the noisy one?

MRF formulation

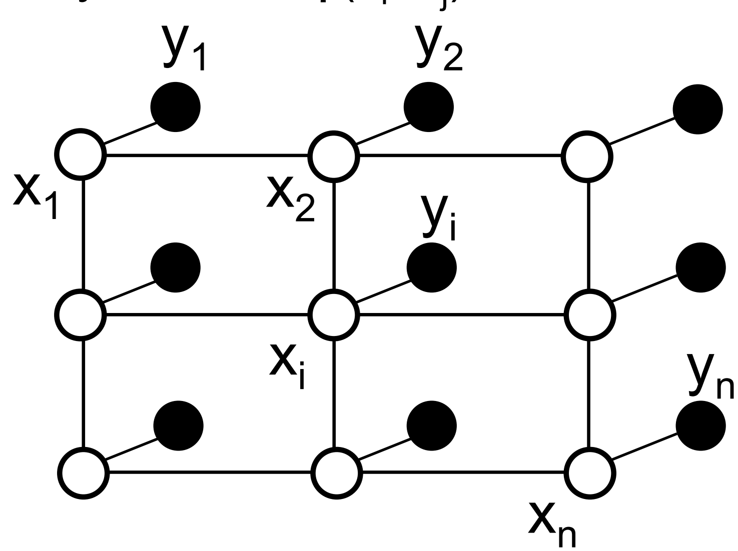
- <u>Nodes</u>
  - For each pixel i,
    - x_i : latent variable (value in original image)
    - y_i : observed variable (value in noisy image)

    → $x_i, y_i \in \{0,1\}$

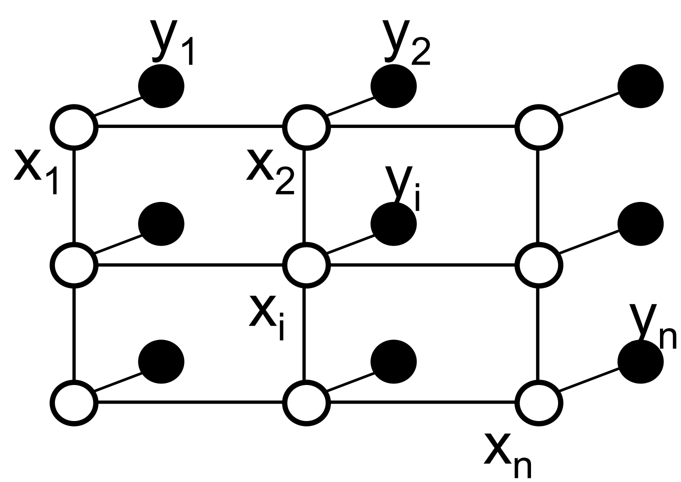


MRF formulation

- <u>Edges</u>
  - x_i, y_i of each pixel i correlated
    - local evidence function $\phi(x_i, y_i)$
    - E.g. $\phi(x_i, y_i) = 0.9$ (if $x_i = y_i$) and $\phi(x_i, y_i) = 0.1$ otherwise (10% noise)
  - Neighboring pixels, similar value
    - compatibility function $\psi(x_i, x_j)$



MRF formulation



$$P(x_1, x_2, \ldots, x_n) = (1/Z) \prod_{(ij)} \psi(x_i, x_j) \prod_i \phi(x_i, y_i)$$

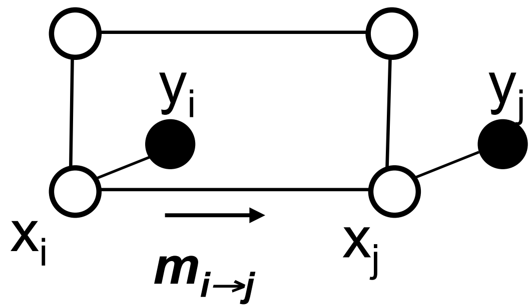
- **Question**: What are the marginal distributions for x_i, i = 1, …,n?

Belief Propagation

- Goal: compute marginals of the latent nodes of underlying graphical model
- Attributes:
  - iterative algorithm
  - message passing between neighboring latent variables nodes
- Question: Can it also be applied to directed graphs?
- Answer: Yes, but here we will apply it to MRFs

Belief Propagation Algorithm

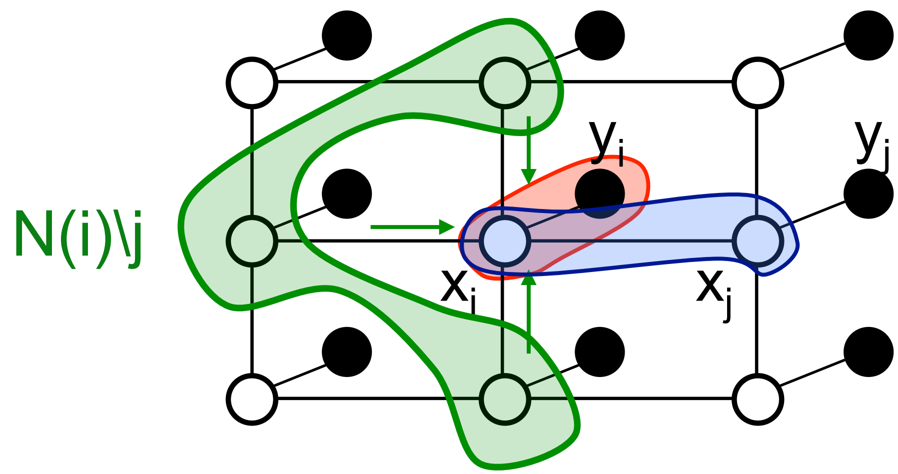
1) Select random neighboring latent nodes x_i, x_j
2) Send message $m_{i \to j}$ from x_i to x_j



3) Update belief about marginal distribution at node x_j
4) Go to step 1, until convergence
    - How is convergence defined?

Step 2: Message Passing

- **Message $m_{i \to j}$ from x_i to x_j** : what node x_i thinks about the marginal distribution of x_j

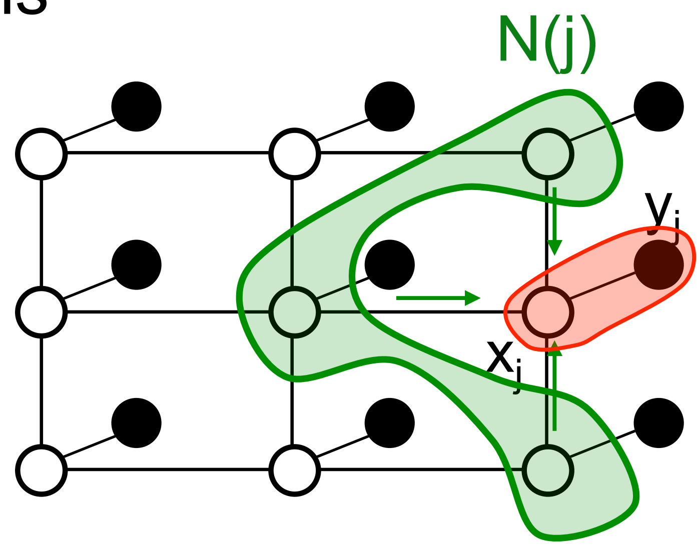


$$m_{i \to j}(x_j) = \sum_{(x_i)} \phi(x_i, y_i)\, \psi(x_i, x_j) \prod_{k \in N(i) \setminus j} m_{k \to i}(x_i)$$

- Messages initially uniformly distributed

Step 3: Belief Update

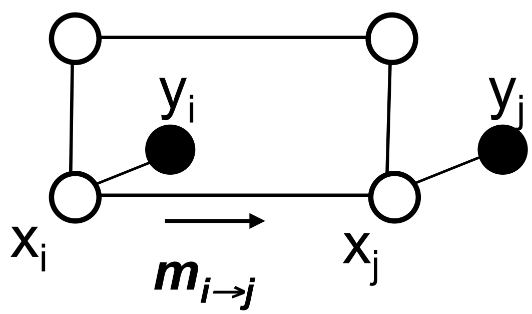
- **Belief $b(x_j)$**: what node x_j thinks its marginal distribution is



$$b(x_j) = k\, \phi(x_j, y_j) \prod_{q \in N(j)} m_{q \to j}(x_j)$$

Belief Propagation Algorithm

1) Select random neighboring latent nodes x_i, x_j
2) Send message $m_{i \to j}$ from x_i to x_j



3) Update belief about marginal distribution at node x_j
4) Go to step 1, until convergence

Example

- Compute belief at node 1.

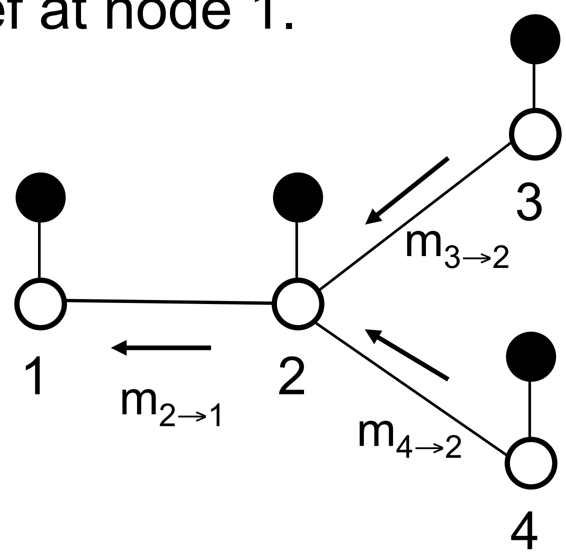


Fig. 12 (Yedidia et al.)

$$b_1(x_1) = k\phi_1(x_1)m_{21}(x_1)$$

$$b_1(x_1) = k\phi_1(x_1)\sum \psi_{12}(x_1, x_2)\phi_2(x_2)m_{32}(x_2)m_{42}(x_2)$$

$$b_1(x_1) = k\phi_1(x_1)\sum_{x_2} \psi_{12}(x_1, x_2)\phi_2(x_2)\sum_{x_3} \phi_3(x_3)\psi_{23}(x_2, x_3)\sum_{x_4} \phi_4(x_4)\psi_{24}(x_2, x_4)$$