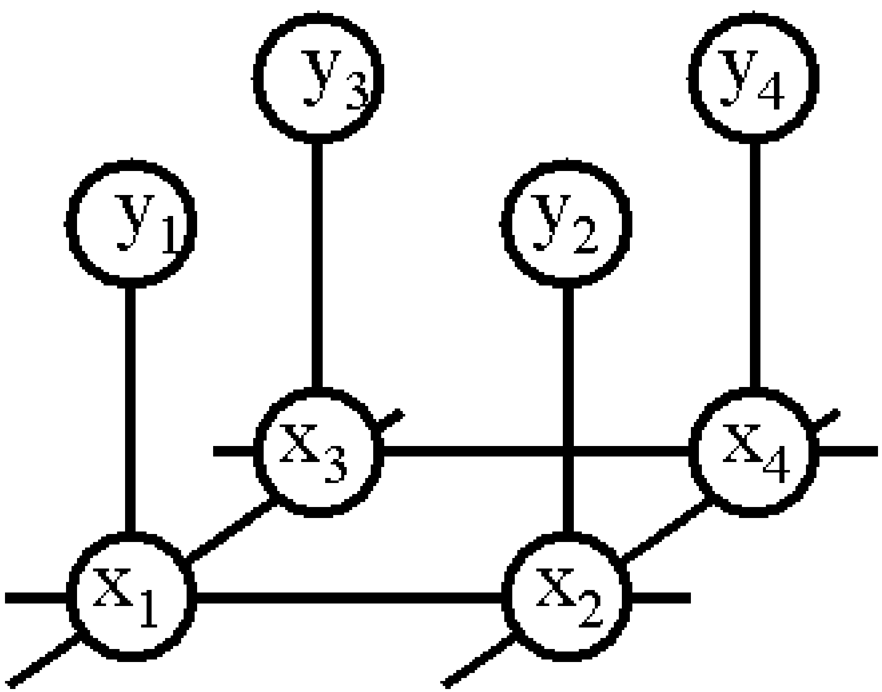
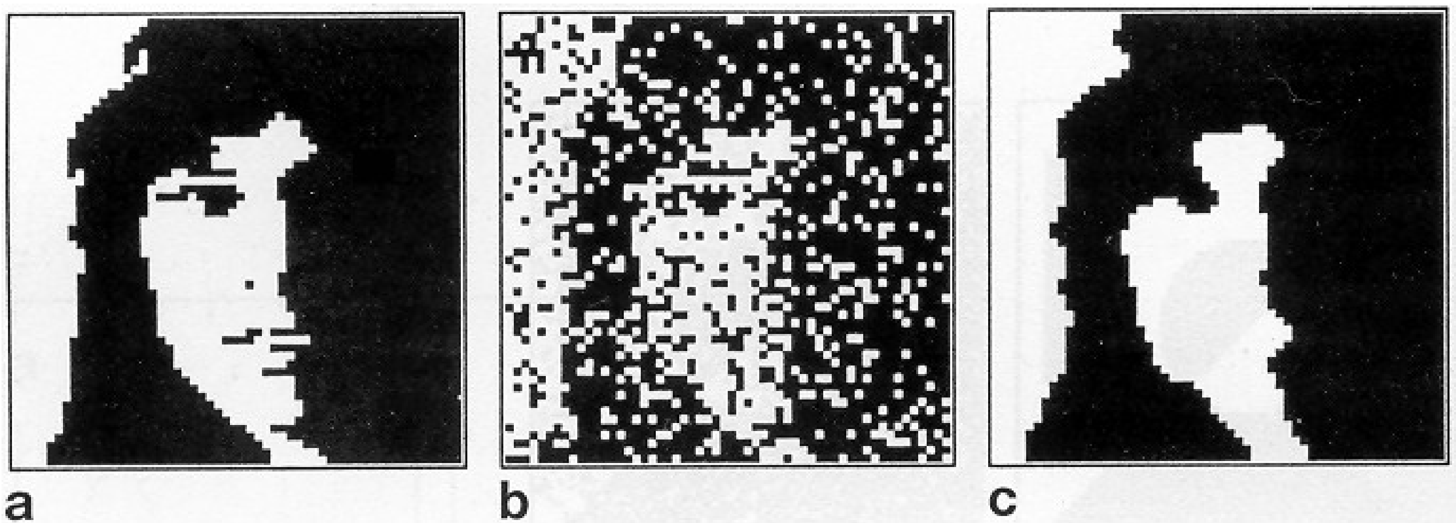
Markov Random Field

Markov Random Fields

- Allows rich probabilistic models for images.
- But built in a local, modular way. Learn local relationships, get global effects out.



MRF nodes as pixels



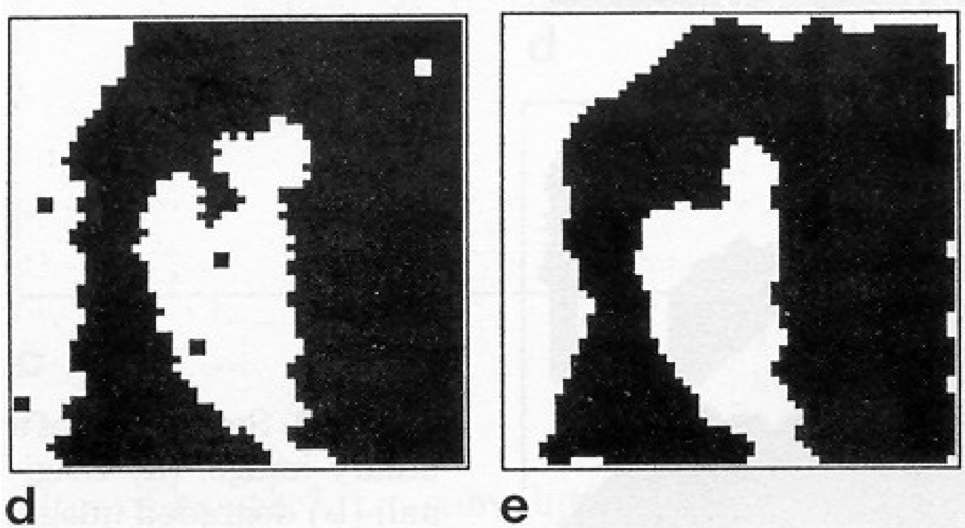
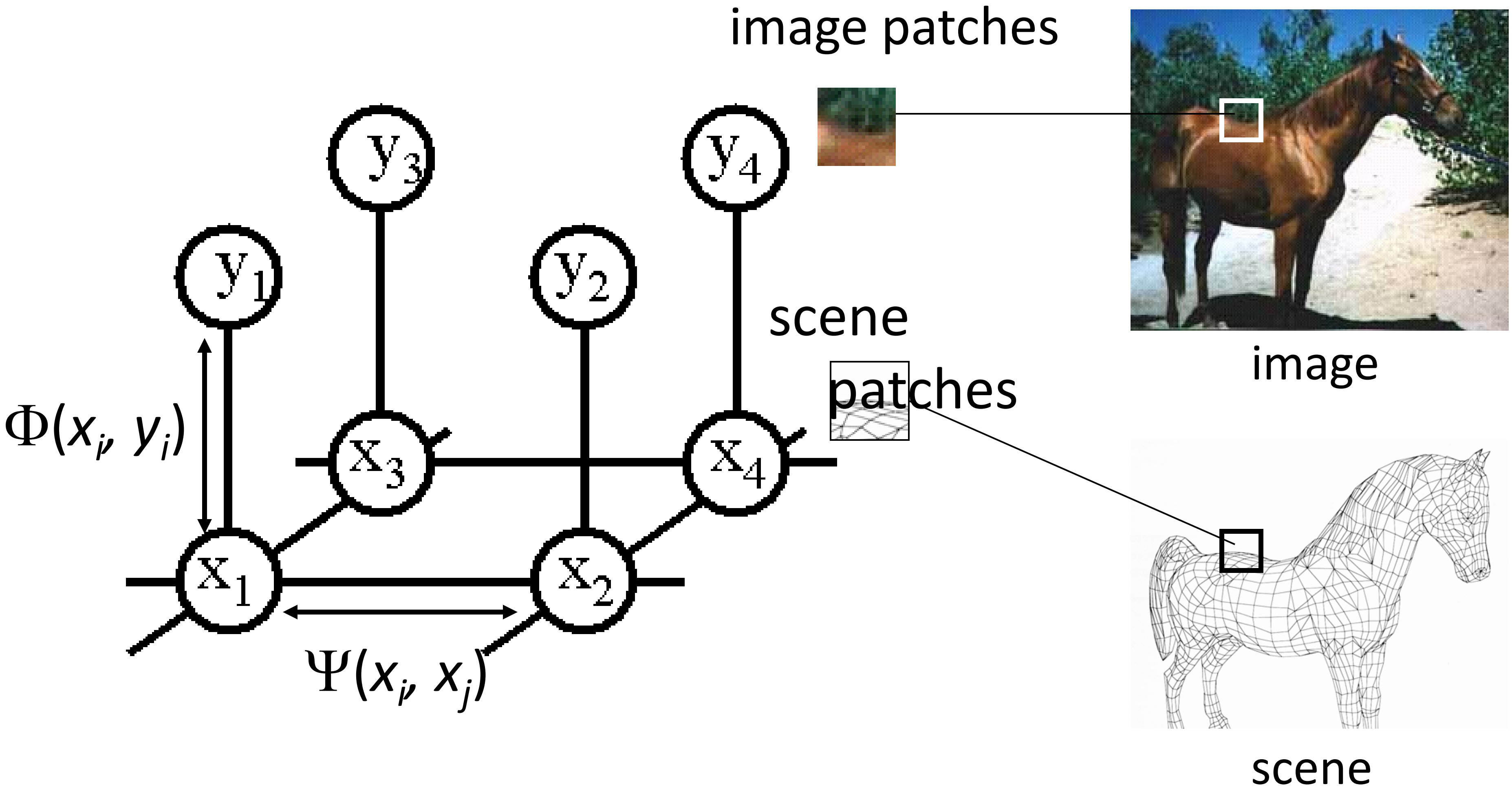


Fig. 2.3. Smoothing with the wrong prior. (**a**) Original, (**b**) degraded image, (**c**) MAP estimate $\beta = 1$, (**d**) MAP estimate $\beta = 0.3$, (**e**) median filter

MRF nodes as patches



MRF overview

- A statistical theory for analyzing spatial & contextual dependencies of physical phenomena.
- A Bayesian labeling problem
- A method to establish the probabilistic distributions of interacting labels
- Widely used in image processing and computer vision

Properties of MRF

- Not *ad hoc*, can be solved based on sound mathematical principles (maximum a posterior probability, MAP)
- Incorporating prior contextual information
- Using local properties, which can be implemented in parallel

An example: image restoration

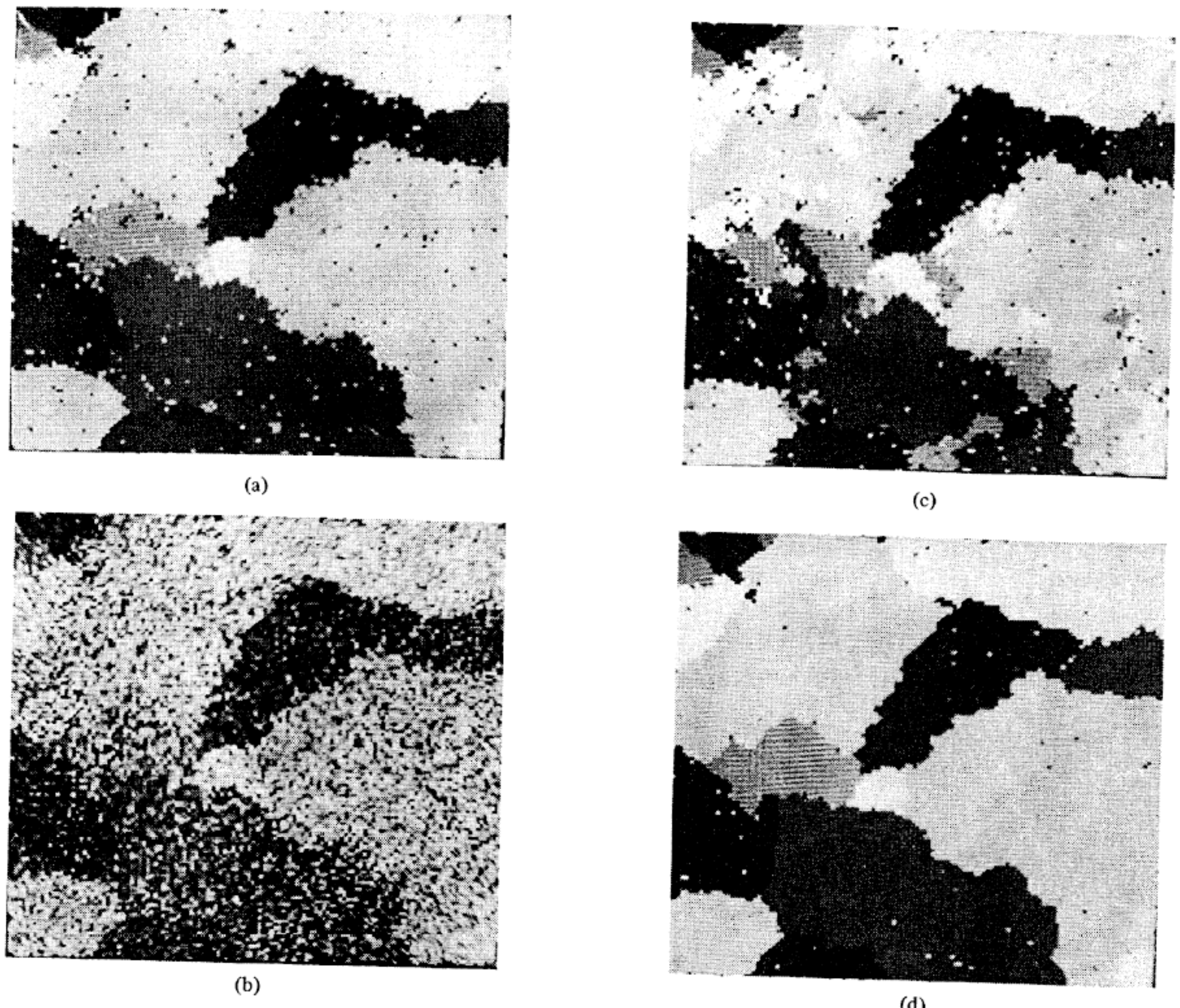


Fig. 2. (a) Original image: Sample from MRF. (b) Degraded image: Additive noise. (c) Restoration: 25 iterations. (d) Restoration: 300 iterations.

Applications

- Restore degraded and noisy images
- Infer the true pixels from noisy ones

Image restoration process

- Build the neighborhood systems and cliques
- Define the clique potentials for prior probability
- Derive the likelihood energy
- Compute the posterior energy
- Solve the MAP

Definition for symbols

S *= set of sites or nodes*

N *= neighbors*

(S, N) *= a nondirected graph*

f *= hidden “true “ pixel*

r *= observed “noisy “ pixel*

Neighborhood Systems

A neighborhood system for S is defined as

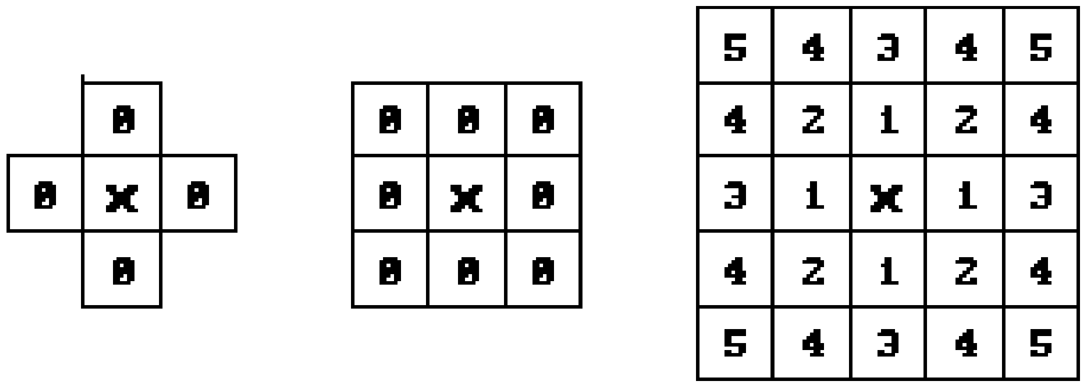
$$N = \{N_i \mid \forall_i \in S\}$$

$$N_i = \{i' \in S \mid dist(pixel_{i'}, pixel_i) \leq d, i' \neq i\}$$

where N_i is the set of sites neighboring i .
The neighboring relationship has the following properties:

(1) a site is not neighboring to itself
(2) the neighboring relationship is mutual

Neighborhood Systems



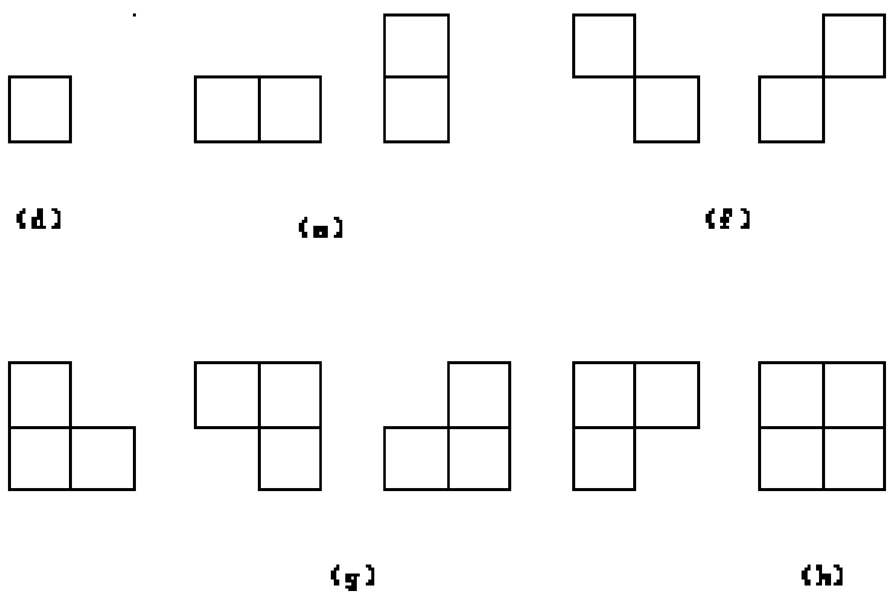
(a) (b) (c)

4 neighbors 8 neighbors 24 neighbors

Cliques

A clique is defined as a subset of sites in S , where every pair of sites are neighbors of each other. The collections of single-site, double-site, and triple-site cliques are denoted by C_1 ,C_2 , and C_3 ,...

A collection of cliques is $C = C_1 \cup C_2 \cup C_3 \ldots$



Markov random fields

- Positivity:

$$P(f) > 0, \forall f \in F$$

- Markovianity:

$$P(f_i \mid f_{S-\{i\}}) = P(f_i \mid f_{N_i})$$

- Homogeneity: probability independent of positions of sites

- Isotropy: probability independent of orientations of sites

Network joint probability

$$P(x, y) = \frac{1}{Z} \prod_{i,j} \Psi(\underbrace{x_i, x_j}_{\text{neighboring scene nodes}}) \prod_i \Phi(x_i, y_i)$$

scene

image

Scene-scene compatibility function

Image-scene compatibility function

local observations

Outline of MRF section

- Inference in MRF’s.
  - Gibbs sampling, simulated annealing
  - Iterated conditional modes (ICM)
  - Variation methods
  - Belief propagation

Gibbs Sampling and Simulated Annealing

- Gibbs sampling:
  - A way to generate random samples from a (potentially very complicated) probability distribution.

$$P(f) = Z^{-1} \times e^{-\frac{1}{T}U(f)}, Z = \sum_{f \in F} e^{-\frac{1}{T}U(f)}, U(f) = \sum_{c \in C} V_c(f)$$

$$U(f) = \sum_{c \in C} V_c(f) = \sum_{\{i\} \in C_1} V_1(f_i) + \sum_{\{i,j\} \in C_2} V_2(f_i, f_j) + \ldots$$

- Simulated annealing:
  - A schedule for modifying the probability distribution so that, at "zero temperature", you draw samples only from the MAP solution.

$$P(x) = \frac{1}{Z} \exp(-E(x)/kT)$$

Sampling from a 1-d function

1. Discretize the density function

$f(x)$

$f(k)$

$f(k)$

$F(k)$

2. Compute distribution function from density function

3. Sampling

draw α ~ U(0,1);

for k = 1 to n

if $F(k) \geq \alpha$

break;

$x = x_0 + k\mu,$

μ : random variable

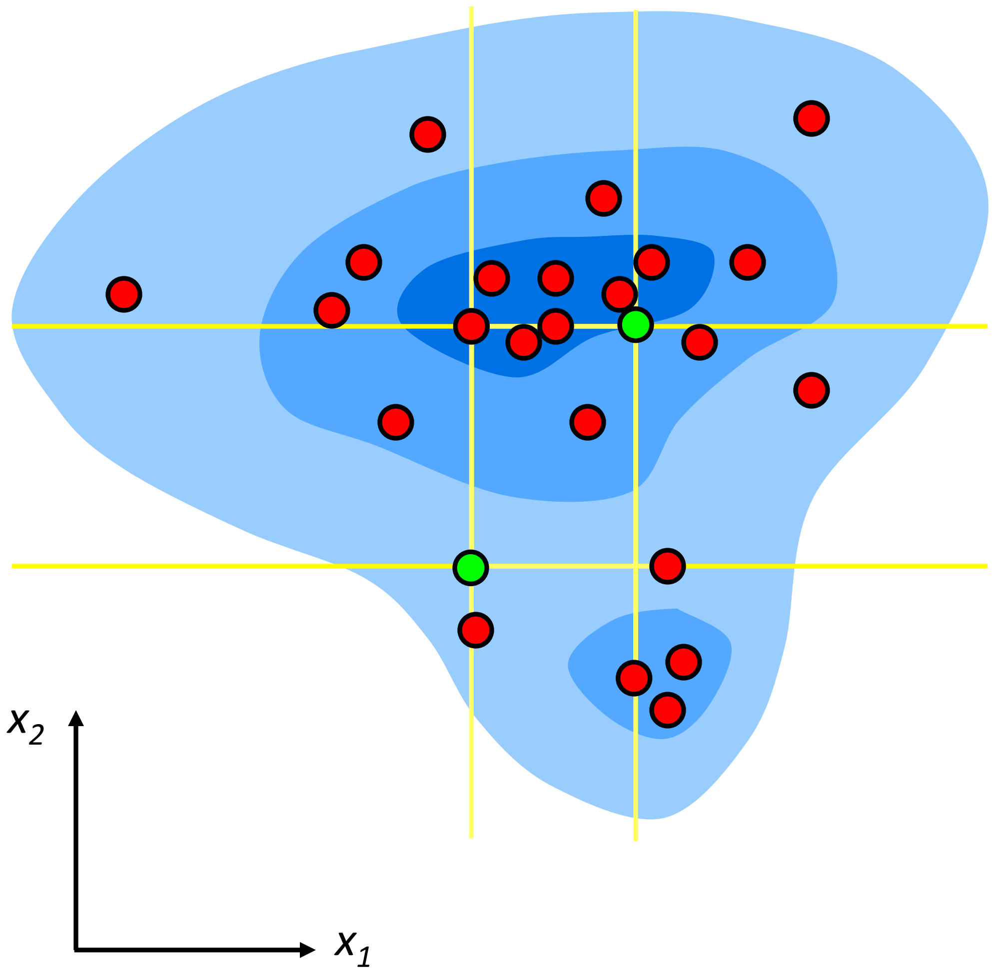
Gibbs Sampling

$$x_1^{(t+1)} \sim \pi(x_1 \mid x_2^{(t)}, x_3^{(t)}, \cdots, x_K^{(t)})$$

$$x_2^{(t+1)} \sim \pi(x_2 \mid x_1^{(t+1)}, x_3^{(t)}, \cdots, x_K^{(t)})$$

$$\vdots$$

$$x_K^{(t+1)} \sim \pi(x_K \mid x_1^{(t+1)}, \cdots, x_{K-1}^{(t+1)})$$



Gibbs sampling as simulated annealing

Simulated annealing as you gradually lower the "temperature" of the probability distribution ultimately giving zero probability to all but the MAP estimate.

What's good about it: finds global MAP solution.

What's bad about it: takes forever. Gibbs sampling is in the inner loop.

Iterated conditional modes

- For each node:
  - Condition on all the neighbors
  - Find the mode
  - Repeat.

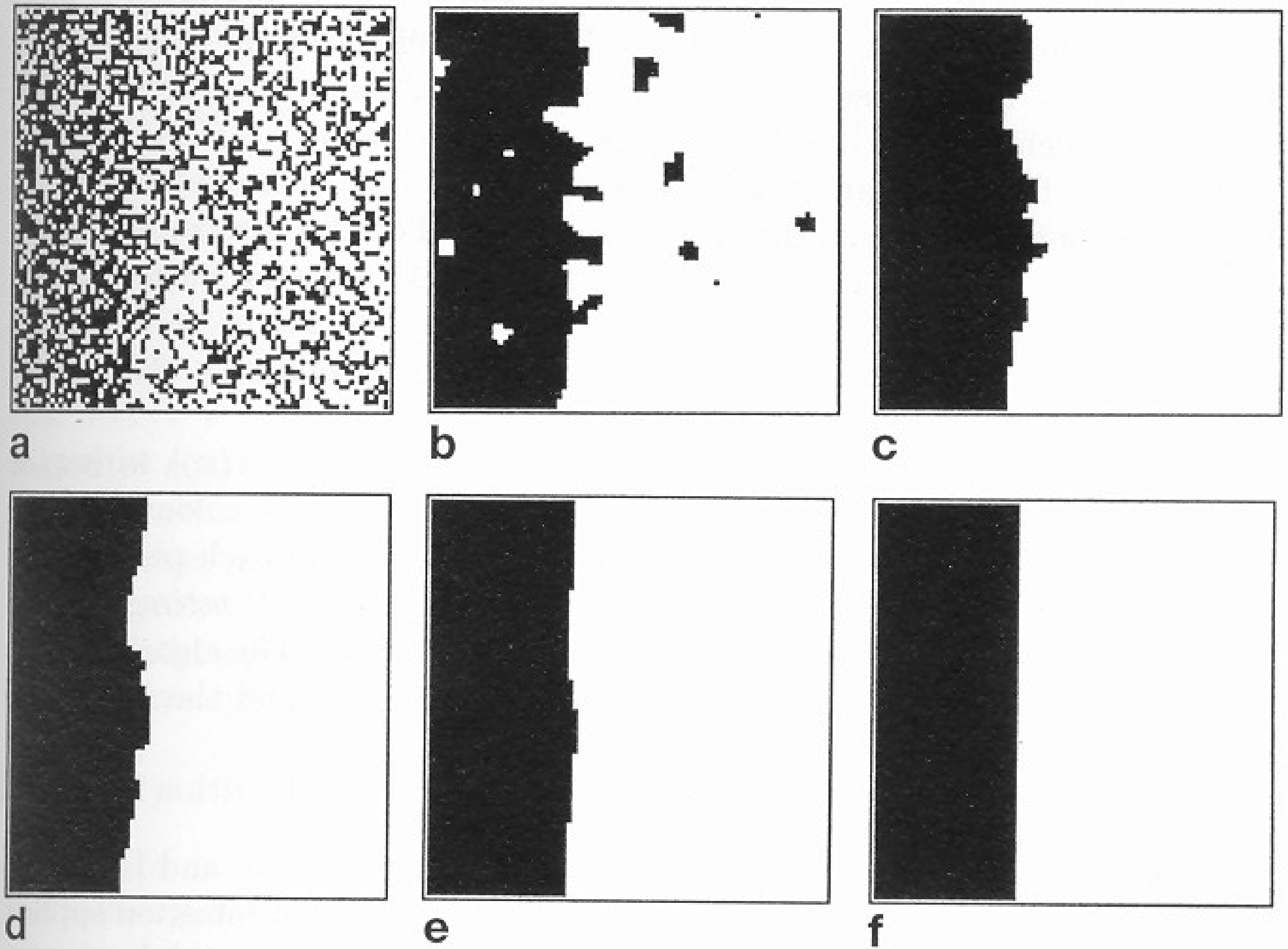
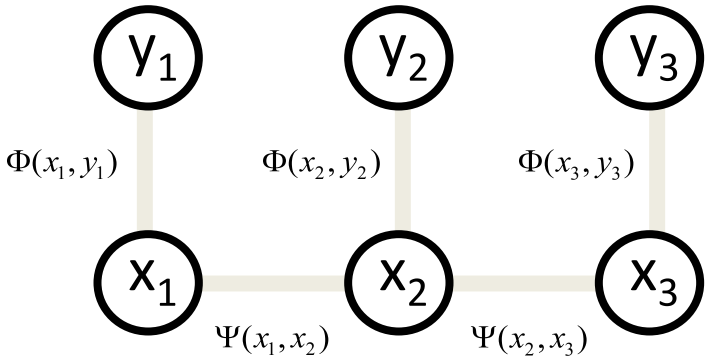


Fig. 6.2. Various steps of ICM

Variational methods

- Reference: Tommi Jaakkola's tutorial on variational methods,
  http://www.ai.mit.edu/people/tommi/
- Example: mean field
  - For each node
    - Calculate the expected value of the node, conditioned on the mean values of the neighbors.

Derivation of belief propagation



$$x_{1MMSE} = \underset{x_1}{\text{mean}}\ \underset{x_2}{\text{sum}}\ \underset{x_3}{\text{sum}}\ P(x_1, x_2, x_3, y_1, y_2, y_3)$$

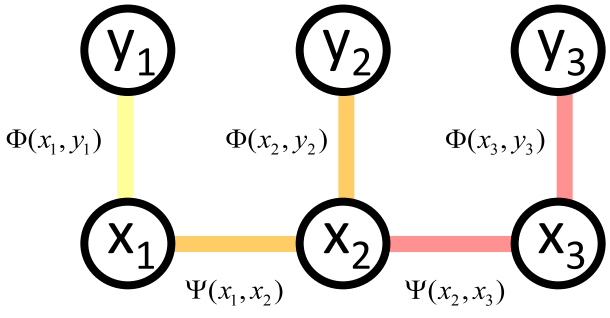
The posterior factorizes

$$x_{1MMSE} = \underset{x_1}{\text{mean}}\ \underset{x_2}{\text{sum}}\ \underset{x_3}{\text{sum}}\ P(x_1, x_2, x_3, y_1, y_2, y_3)$$

$$= \underset{x_1}{\text{mean}}\ \underset{x_2}{\text{sum}}\ \underset{x_3}{\text{sum}}\ \Phi(x_1, y_1)$$

$$\Phi(x_2, y_2)\ \Psi(x_1, x_2)$$

$$\Phi(x_3, y_3)\ \Psi(x_2, x_3)$$

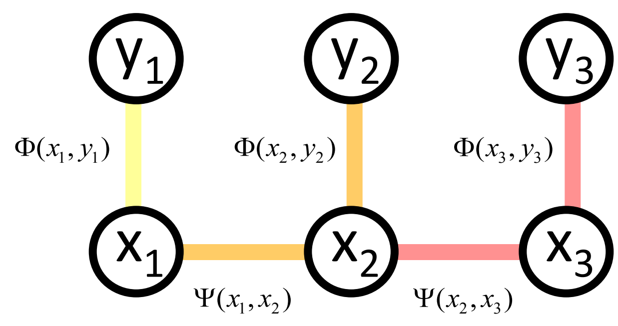


Propagation rules

$$x_{1MMSE} = \underset{x_1}{\text{mean}} \ \underset{x_2}{\text{sum}} \ \underset{x_3}{\text{sum}} \ P(x_1, x_2, x_3, y_1, y_2, y_3)$$

$$x_{1MMSE} = \underset{x_1}{\text{mean}} \ \underset{x_2}{\text{sum}} \ \underset{x_3}{\text{sum}} \ \Phi(x_1, y_1) \ \Phi(x_2, y_2) \ \Psi(x_1, x_2) \ \Phi(x_3, y_3) \ \Psi(x_2, x_3)$$

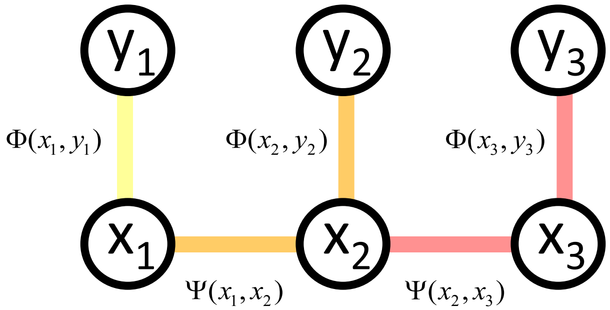
$$x_{1MMSE} = \underset{x_1}{\text{mean}} \ \Phi(x_1, y_1) \ \underset{x_2}{\text{sum}} \ \Phi(x_2, y_2) \ \Psi(x_1, x_2) \ \underset{x_3}{\text{sum}} \ \Phi(x_3, y_3) \ \Psi(x_2, x_3)$$



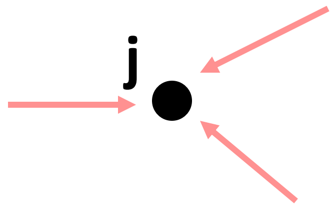
Propagation rules

$$x_{1MMSE} = \operatorname*{mean}_{x_1} \Phi(x_1, y_1) \operatorname*{sum}_{x_2} \Phi(x_2, y_2)\, \Psi(x_1, x_2) \operatorname*{sum}_{x_3} \Phi(x_3, y_3)\, \Psi(x_2, x_3)$$

$$M_1^2(x_1) = \operatorname*{sum}_{x_2} \Psi(x_1, x_2)\ \Phi(x_2, y_2)\ M_2^3(x_2)$$

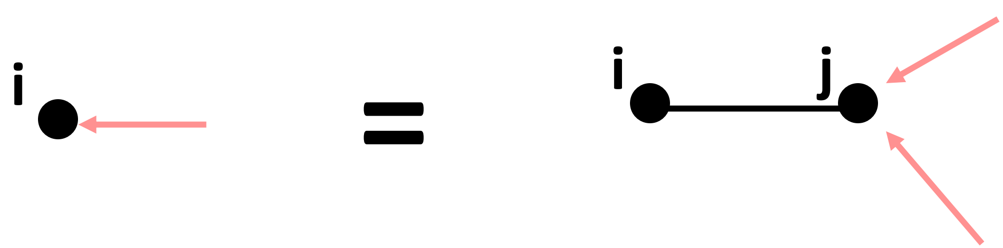


Belief, and message updates



$$b_j(x_j) = \prod_{k \in N(j)} M_j^k(x_j)$$

$$M_i^j(x_i) = \sum_{x_j} \psi_{ij}(x_i, x_j) \prod_{k \in N(j) \setminus i} M_j^k(x_j)$$

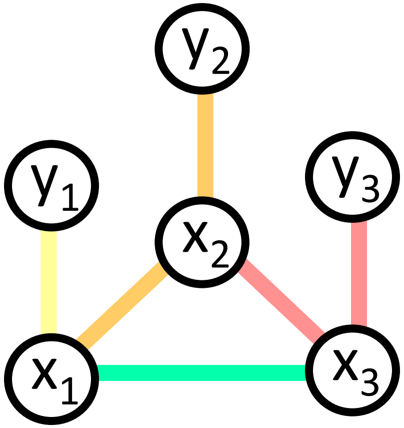


Optimal solution in a chain or tree: Belief Propagation

- “Do the right thing” Bayesian algorithm.
- For Gaussian random variables over time: Kalman filter.
- For hidden Markov models: forward/backward algorithm (and MAP variant is Viterbi).

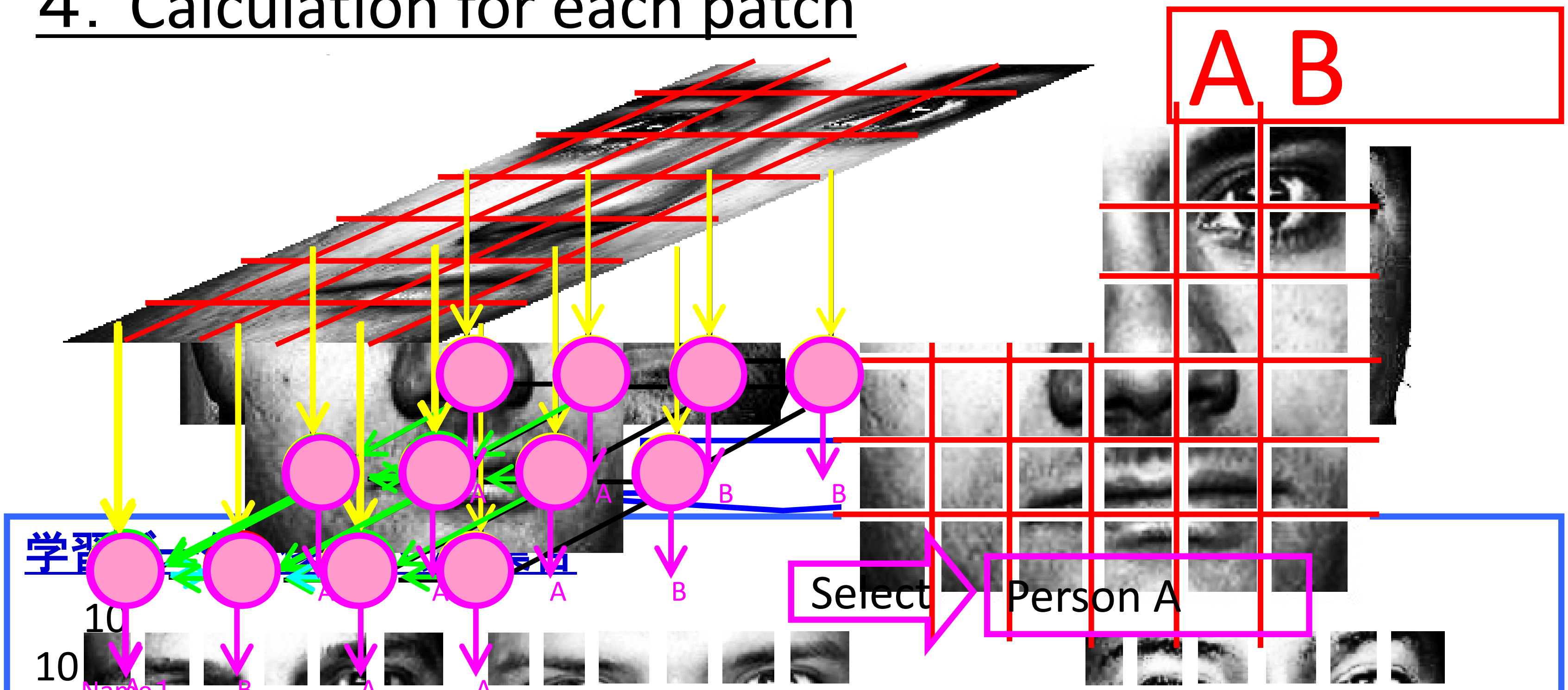
No factorization with loops!

$$x_{1MMSE} = \underset{x_1}{\text{mean}}\, \Phi(x_1, y_1)\, \underset{x_2}{\text{sum}}\, \Phi(x_2, y_2)\, \Psi(x_1, x_2)\, \underset{x_3}{\text{sum}}\, \Phi(x_3, y_3)\, \Psi(x_2, x_3)\ \Psi(x_1, x_3)$$



Example: Belief Propagation Calculation

4. Calculation for each patch



$$\phi(x_i, y_i) = P(y_i \mid x_i) = \frac{1}{\sqrt{2\pi\sigma_{xi}^2}} \exp\left(- \frac{\min_{y \in D, id(y)=x} d(y, y_i)}{2\sigma_{xi}^2} \right)$$

$$\psi(x_i, x_j) = P(x_i, x_j) = \frac{1}{Z} \exp\left(- \frac{\delta(x_i - x_j)}{\sigma^2} \right)$$

…

$$m_{ij}(x_j) = \sum_{xi} \psi(x_i, x_j) \phi(x_i, y_i) \prod_{k \in N(i) \backslash j} m_{ki}(x_i)$$

$$x_{iMAP} = \arg\max_{x_i} \phi(x_i, y_i) \prod_{j \in N(i)} m_{ji}(x_i)$$

Statistical mechanics interpretation

U - TS = Free energy

U = avg. energy =

$$\sum_{states} p(x_1, x_2, \ldots) E(x_1, x_2, \ldots)$$

T = temperature

S = entropy =

$$-\sum_{states} p(x_1, x_2, \ldots) \ln p(x_1, x_2, \ldots)$$

Free energy formulation

Defining

$$\Psi_{ij}(x_i, x_j) = e^{-E(x_i, x_j)/T} \qquad \Phi_i(x_i) = e^{-E(x_i)/T}$$

then the probability distribution $P(x_1, x_2, \ldots)$

that minimizes the F.E. is precisely

the true probability of the Markov network,

$$P(x_1, x_2, \ldots) = \prod_{ij} \Psi_{ij}(x_i, x_j) \prod_i \Phi_i(x_i)$$

Approximating the Free Energy

Exact:

$$F[p(x_1, x_2, \ldots, x_N)]$$

Mean Field Theory:

$$F[b_i(x_i)]$$

Bethe Approximation :

$$F[b_i(x_i), b_{ij}(x_i, x_j)]$$

Kikuchi Approximations:

$$F[b_i(x_i), b_{ij}(x_i, x_j), b_{ijk}(x_i, x_j, x_k), \ldots.]$$

Bethe Approximation

On tree-like lattices, exact formula:

$$p(x_1, x_2, \ldots, x_N) = \prod_{(ij)} p_{ij}(x_i, x_j) \prod_i [p_i(x_i)]^{1-q_i}$$

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

Gibbs Free Energy

$$F_{Bethe}(b_i, b_{ij}) + \sum_{(ij)} \gamma_{ij} \{ \sum_{x_i, x_j} b_{ij}(x_i, x_j) - 1 \}$$

$$+ \sum_{x_j} \sum_{(ij)} \lambda_{ij}(x_j) \{ \sum_{x_i} b_{ij}(x_i, x_j) - b_j(x_j) \}$$

Gibbs Free Energy

$$F_{Bethe}(b_i, b_{ij}) + \sum_{(ij)} \gamma_{ij}\{\sum_{x_i, x_j} b_{ij}(x_i, x_j) - 1\}$$

$$+ \sum_{x_j} \sum_{(ij)} \lambda_{ij}(x_j)\{\sum_{x_i} b_{ij}(x_i, x_j) - b_j(x_j)\}$$

Set derivative of Gibbs Free Energy w.r.t. b_{ij}, b_i terms to zero:

$$b_{ij}(x_i, x_j) = k\,\Psi_{ij}(x_i, x_j)\exp\left(\frac{-\lambda_{ij}(x_i)}{T}\right)$$

$$b_i(x_i) \qquad = k\,\Phi(x_i)\exp\left(\frac{\sum_{j \in N(i)} \lambda_{ij}(x_i)}{T}\right)$$

Belief Propagation = Bethe

Lagrange multipliers $\lambda_{ij}(x_j)$

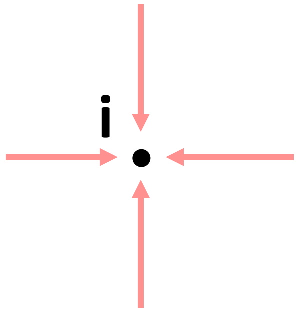
enforce the constraints $b_j(x_j) = \sum_{x_i} b_{ij}(x_i, x_j)$

Bethe stationary conditions = message update rules

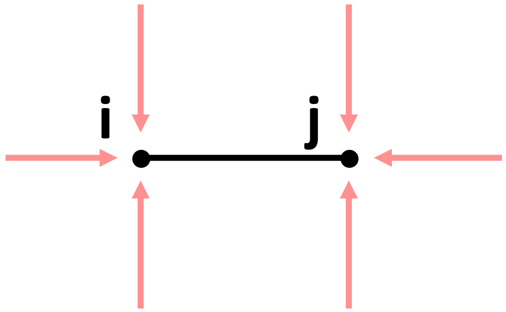
with $$\lambda_{ij}(x_j) = T \ln \prod_{k \in N(j) \backslash i} M_j^k(x_j)$$

Region marginal probabilities

$$b_i(x_i) = k\,\Phi(x_i)\prod_{k\in N(i)} M_i^k(x_i)$$

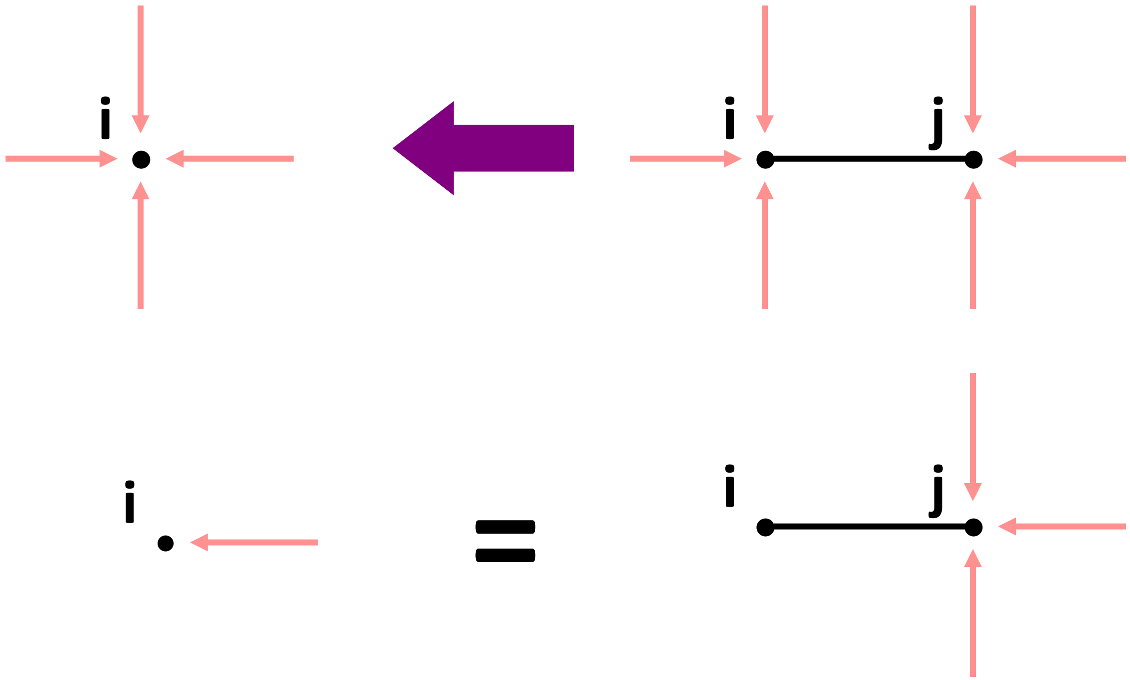


$$b_{ij}(x_i, x_j) = k\,\Psi(x_i, x_j)\prod_{k\in N(i)\backslash j} M_i^k(x_i)\prod_{k\in N(j)\backslash i} M_j^k(x_j)$$



Belief propagation equations

Belief propagation equations come from the marginalization constraints.



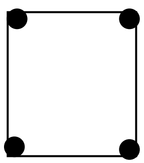
$$M_i^j(x_i) = \sum_{x_j} \psi_{\mathrm{ij}}(x_i, x_j) \prod_{k \in N(j) \setminus i} M_j^k(x_j)$$

Results from Bethe free energy analysis

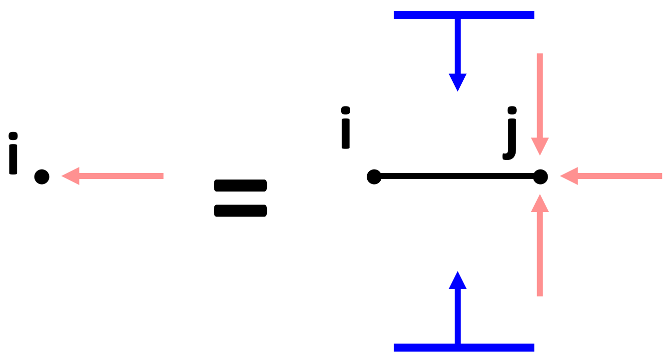
- Fixed point of belief propagation equations iff. Bethe approximation stationary point.
- Belief propagation always has a fixed point.
- Connection with variational methods for inference: both minimize approximations to Free Energy,
  - **variational**: usually use primal variables.
  - **belief propagation**: fixed pt. equs. for dual variables.
- Kikuchi approximations lead to more accurate belief propagation algorithms.
- Other Bethe free energy minimization algorithms—Yuille, Welling, etc.

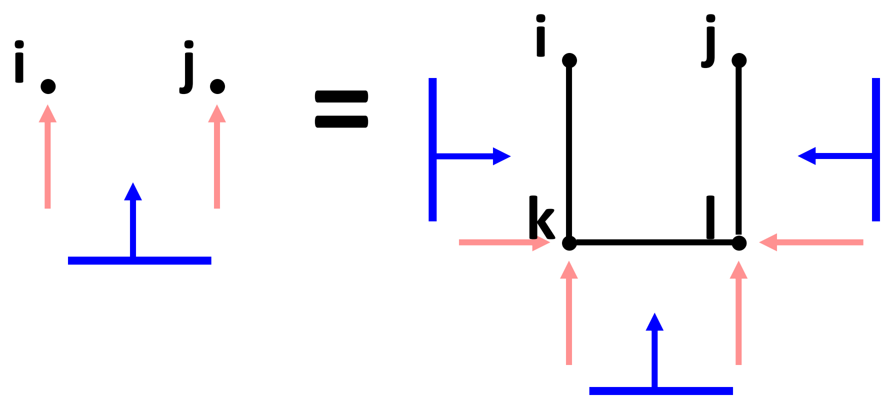
Kikuchi message-update rules

Groups of nodes send messages to other groups of nodes.

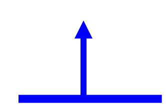


Typical choice for Kikuchi cluster.





Update for messages 

Update for messages 

Generalized belief propagation