

Approximate inference for Bayesian machine learning

Yuan (Alan) Qi

Department of CS and Statistics

Purdue University

with special thanks to Tom Minka

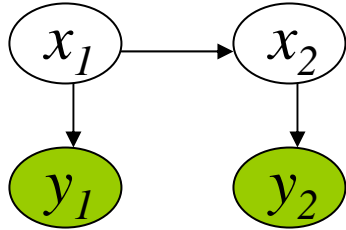
Outline: Section 1

- Graphical models
- Bayesian integration
- Message Passing
 - Example of message passing
 - Interpreting message passing
 - Divergence measures
 - Message passing from a divergence measure
 - Big picture

Outline: Section 1

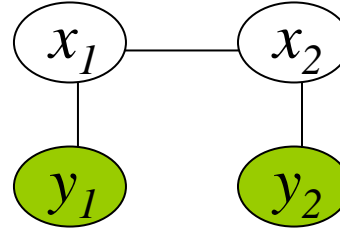
- Graphical models
- Bayesian integration
- Message Passing
 - Example of message passing
 - Interpreting message passing
 - Divergence measures
 - Message passing from a divergence measure
 - Big picture

Graphical Models



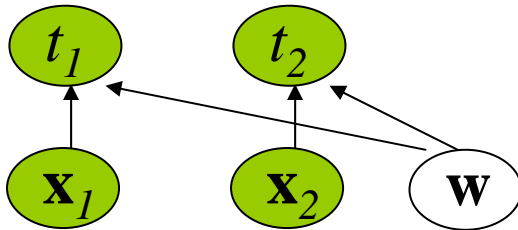
$$p(\mathbf{x}, \mathbf{y}) = \prod_i p(\mathbf{x}_i | \mathbf{x}_{pa(i)}) \prod_j p(\mathbf{y}_j | \mathbf{x}_{pa(j)})$$

Bayesian networks



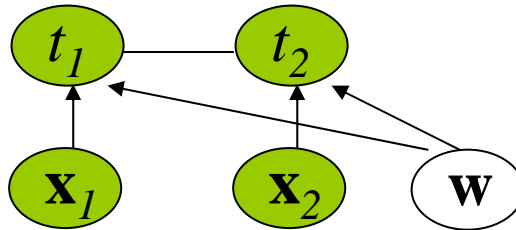
$$p(\mathbf{x}, \mathbf{y}) = \frac{1}{Z} \prod_a \phi_a(\mathbf{x}, \mathbf{y})$$

Markov networks



$$p(\mathbf{t} | \mathbf{x}, \mathbf{w}) = \prod_i p(t_i | \mathbf{x}_i, \mathbf{w})$$

conditional classification



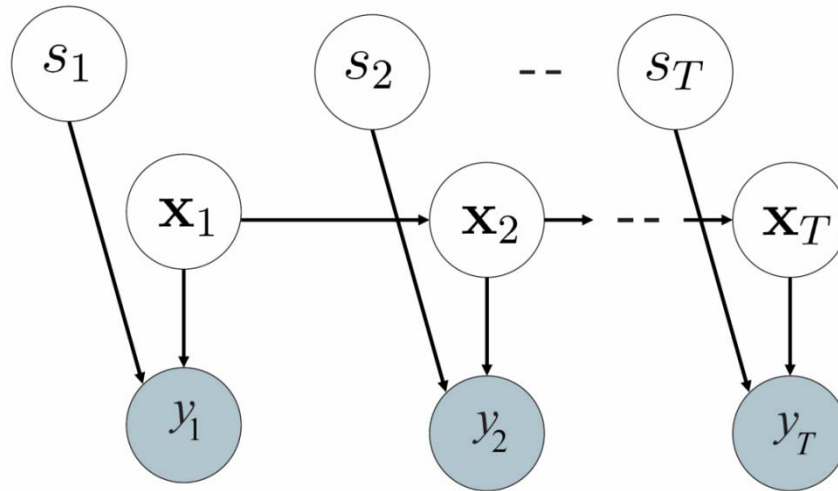
$$p(\mathbf{t} | \mathbf{w}, \mathbf{x}) = \frac{1}{Z(\mathbf{w})} \prod_a \phi_a(\mathbf{x}, \mathbf{y}, \mathbf{t})$$

conditional random fields

INFERENCE

LEARNING

Example: hybrid Bayesian network

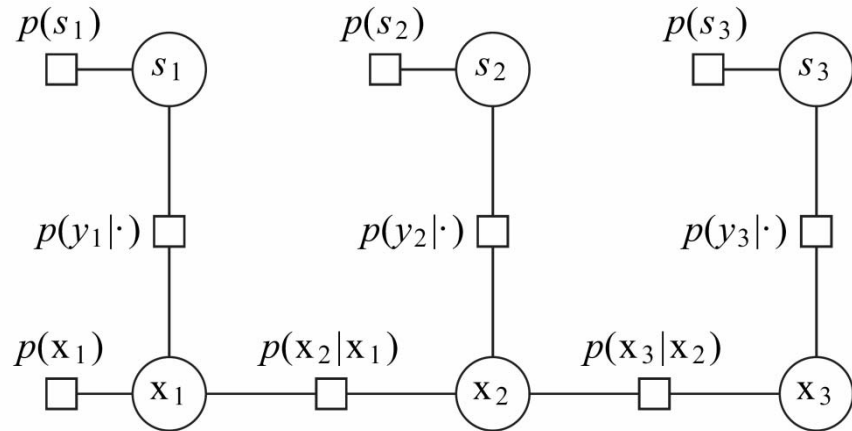


$$\mathbf{x}_t = \mathbf{F}\mathbf{x}_{t-1} + \mathbf{g}_t v_t$$

$$y_t = s_t \mathbf{h}^H \mathbf{x}_t + w_t$$

Factor graph representation

- The probabilistic distribution of the variables (circles) are proportional to the product of factors (rectangles).



- Unifying framework for directed and undirected graphical models

$$\begin{aligned} & p(s_{1:T}, \mathbf{x}_{1:T} | y_{1:T}) \\ & \propto p(s_1)p(\mathbf{x}_1)p(y_1 | s_1, \mathbf{x}_1) \cdot \\ & \cdot \prod_{t=2}^T p(\mathbf{x}_t | \mathbf{x}_{t-1})p(s_t)p(y_t | s_t, \mathbf{x}_t) \end{aligned}$$

Two key questions



- Choose/design Bayesian model
- Calculate posterior distribution for inference and learning

Outline: Section 1

- Graphical models
- **Bayesian integration**
- Message Passing
 - Example of message passing
 - Interpreting message passing
 - Divergence measures
 - Message passing from a divergence measure
 - Big picture

Bayesian Integration: Monte Carlo

- Markov Chain Monte Carlo (MCMC), e.g., M-H algorithm, Gibbs sampler
- (Adaptive) Importance sampling
- Sequential Importance sampling, e.g., particle filter and smoothers
- Recent development: Wang-Landau Algorithm, auxiliary variables (Swendsen and Wang) and data augmentation, EAV, parallel tempering, etc.

Monte Carlo Methods

- Obtain exact posterior distributions, given enough number of samples.
- Can be computationally expensive for large-scale, high-dimensional problems
- When converge?

Deterministic approximation

- Quadrature: similar to importance sampling. Deterministic choice of nodes (samples) and weight
- Laplace's method: fitting a Gaussian to the mode of the posterior

Message-passing algorithms

- Computationally efficient
- High accuracy for a variety of probabilistic models
- Tuning inference problem into a distributed optimization

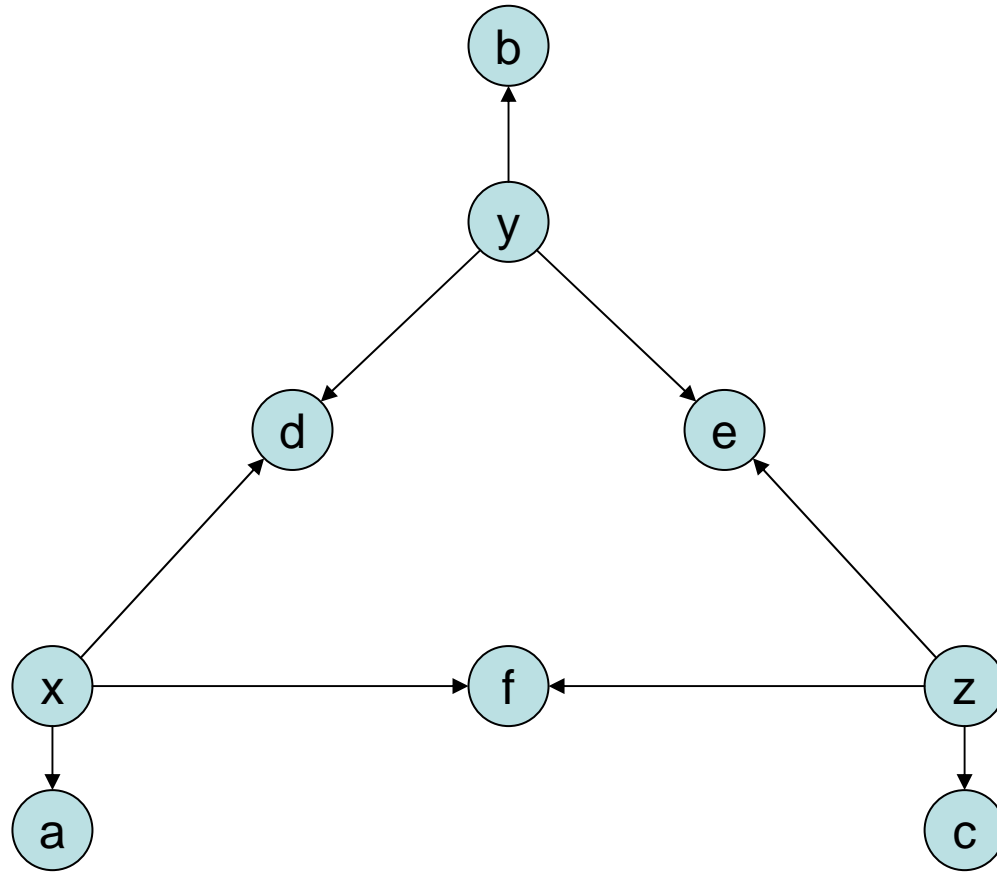
Message-Passing Algorithms

Mean-field	MF	[Peterson,Anderson 87]
Loopy belief propagation	BP	[Frey,MacKay 97]
Expectation propagation	EP	[Minka 01]
Tree-reweighted message passing	TRW	[Wainwright,Jaakkola,Willsky 03]
Fractional belief propagation	FBP	[Wiegerinck,Heskes 02]
Power EP	PEP	[Minka 04]

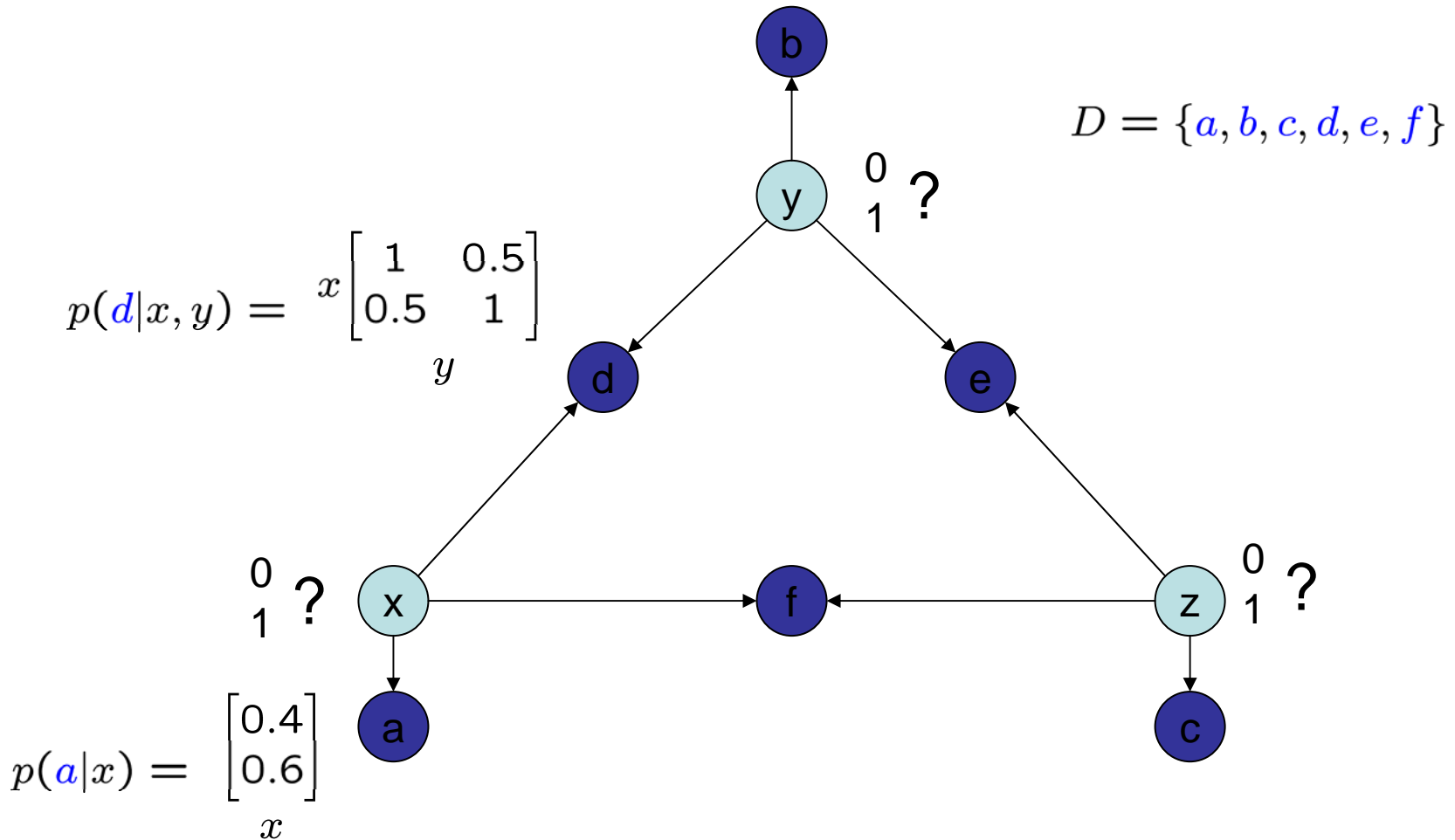
Outline: Section 1

- Graphical models
- Bayesian integration
- **Message Passing**
 - Example of message passing
 - Interpreting message passing
 - Divergence measures
 - Message passing from a divergence measure
 - Big picture

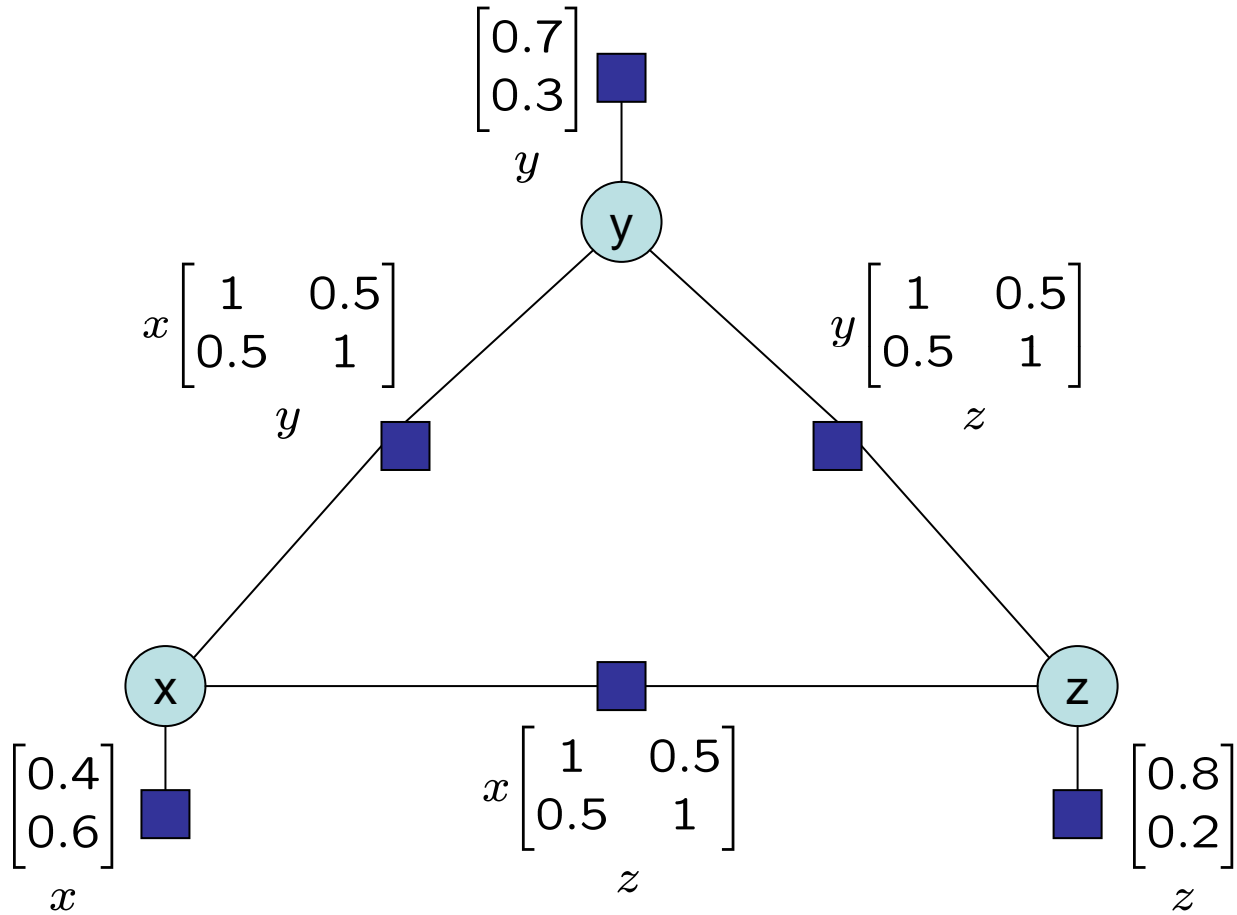
Estimation Problem



Estimation Problem



Estimation Problem



Estimation Problem

$$p(x, y, z, D) = x \begin{matrix} \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix} \\ y \end{matrix} y \begin{matrix} \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix} \\ z \end{matrix} x \begin{matrix} \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix} \\ z \end{matrix} \begin{bmatrix} 0.4 \\ 0.6 \end{bmatrix} \begin{matrix} \begin{bmatrix} 0.7 \\ 0.3 \end{bmatrix} \\ y \end{matrix} \begin{matrix} \begin{bmatrix} 0.8 & 8 \\ 0.2 & 2 \end{bmatrix} \\ z \end{matrix}$$

$$p(0, 0, 0, D) = 0.224$$

$$p(0, 0, 1, D) = 0.014$$

$$p(0, 1, 0, D) = 0.024$$

$$p(0, 1, 1, D) = 0.006$$

$$p(1, 0, 0, D) = 0.084$$

$$p(1, 0, 1, D) = 0.021$$

$$p(1, 1, 0, D) = 0.036$$

$$p(1, 1, 1, D) = 0.036$$

Queries:

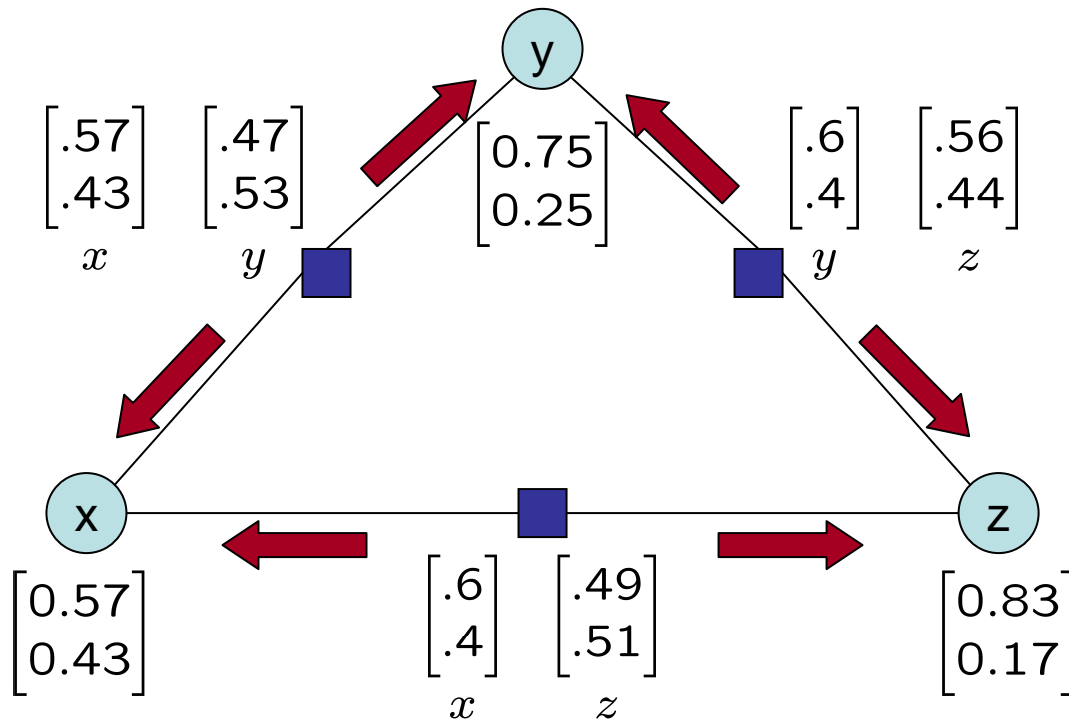
$$p(x, D) = \sum_{y,z} p(x, y, z, D)$$

$$p(D) = \sum_{x,y,z} p(x, y, z, D)$$

$$(x^*, y^*, z^*) = \operatorname{argmax} p(x, y, z, D)$$

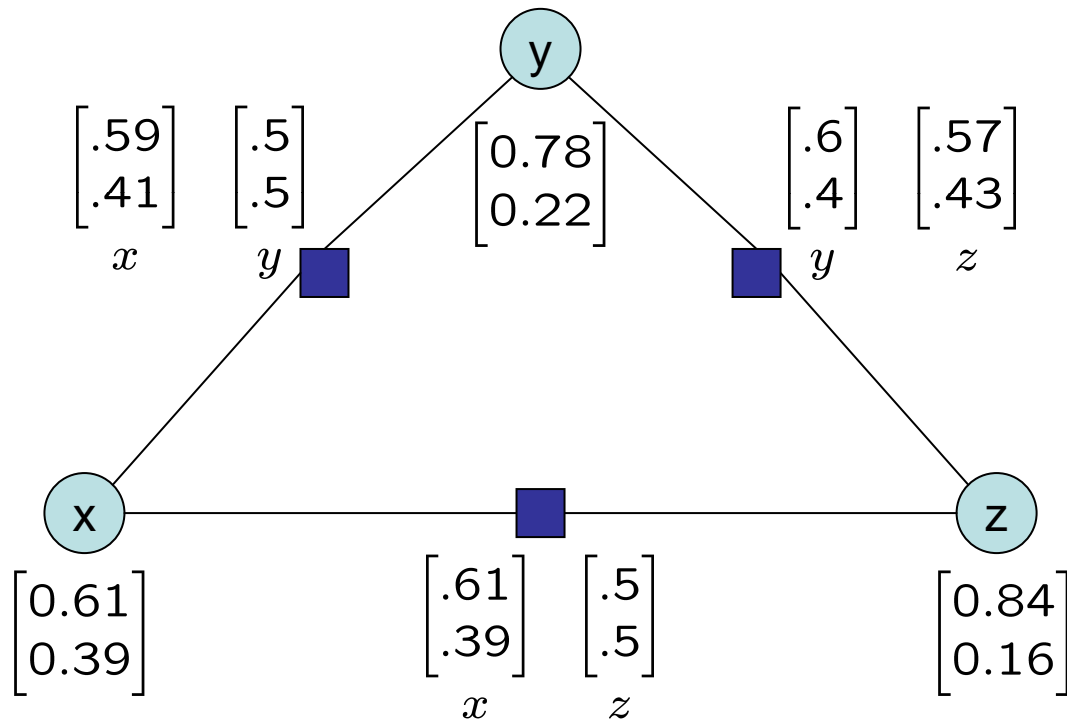
Want to do these *quickly*

Belief Propagation



Belief Propagation

Final



Belief Propagation

Marginals: $\begin{bmatrix} 0.60 \\ 0.40 \end{bmatrix}$ $\begin{bmatrix} 0.77 \\ 0.23 \end{bmatrix}$ $\begin{bmatrix} 0.83 \\ 0.17 \end{bmatrix}$ (Exact)
 x y z

$\begin{bmatrix} 0.61 \\ 0.39 \end{bmatrix}$ $\begin{bmatrix} 0.78 \\ 0.22 \end{bmatrix}$ $\begin{bmatrix} 0.84 \\ 0.16 \end{bmatrix}$ (BP)
 x y z

Normalizing constant: 0.45 (Exact)
0.44 (BP)

Argmax: (0,0,0) (Exact)
(0,0,0) (BP)

Outline: Section 1

- Graphical models
- Bayesian integration
- **Message Passing**
 - Example of message passing
 - **Interpreting message passing**
 - Divergence measures
 - Message passing from a divergence measure
 - Big picture

Message Passing = Distributed Optimization

- Messages represent a simpler distribution $q(x)$ that approximates $p(x)$
 - A *distributed* representation
- Message passing = optimizing q to fit p
 - q stands in for p when answering queries
- Parameters:
 - What type of distribution to construct (approximating family)
 - What cost to minimize (divergence measure)

How to make a message-passing algorithm

1. Pick an approximating family
 - fully-factorized, Gaussian, etc.
2. Pick a divergence measure
3. Construct an optimizer for that measure
 - usually fixed-point iteration
4. Distribute the optimization across factors

Outline: Section 1

- Graphical models
- Bayesian integration
- **Message Passing**
 - Example of message passing
 - Interpreting message passing
 - **Divergence measures**
 - Message passing from a divergence measure
 - Big picture

Let p, q be *unnormalized* distributions

Kullback-Leibler (KL) divergence

$$KL(p \parallel q) = \int p(x) \log \frac{p(x)}{q(x)} dx + \int (q(x) - p(x)) dx$$

Alpha-divergence (α is any real number)

$$D_\alpha(p \parallel q) = \frac{\int_x \alpha p(x) + (1 - \alpha) q(x) - p(x)^\alpha q(x)^{1-\alpha} dx}{\alpha(1 - \alpha)}$$

Asymmetric, convex	$D_\alpha(p \parallel q) = 0$	if $p = q$
	$D_\alpha(p \parallel q) > 0$	otherwise

Examples of alpha-divergence

$$D_{-1}(p \parallel q) = \frac{1}{2} \int_x \frac{(q(x) - p(x))^2}{p(x)} dx$$

$$D_0(p \parallel q) = KL(q \parallel p)$$

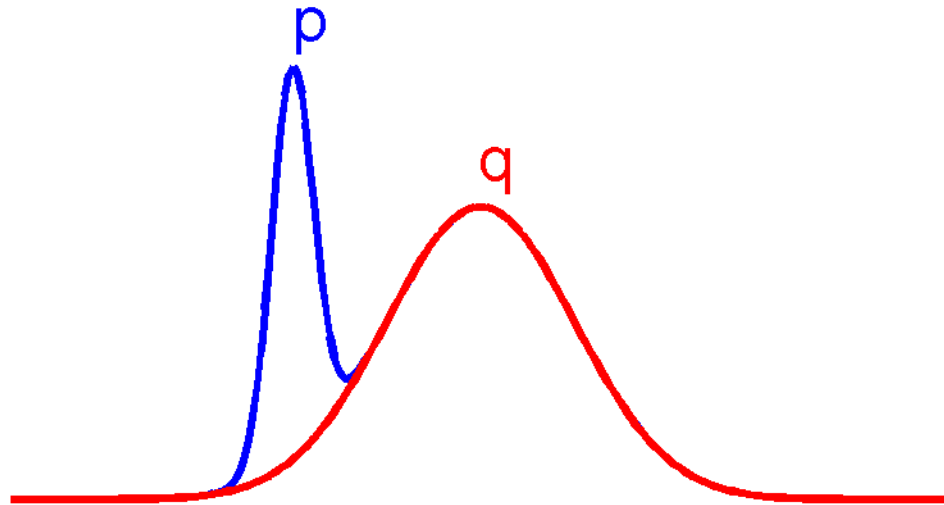
$$D_{\frac{1}{2}}(p \parallel q) = 2 \int_x \left(\sqrt{p(x)} - \sqrt{q(x)} \right)^2 dx$$

$$D_1(p \parallel q) = KL(p \parallel q)$$

$$D_2(p \parallel q) = \frac{1}{2} \int_x \frac{(p(x) - q(x))^2}{q(x)} dx$$

Minimum alpha-divergence

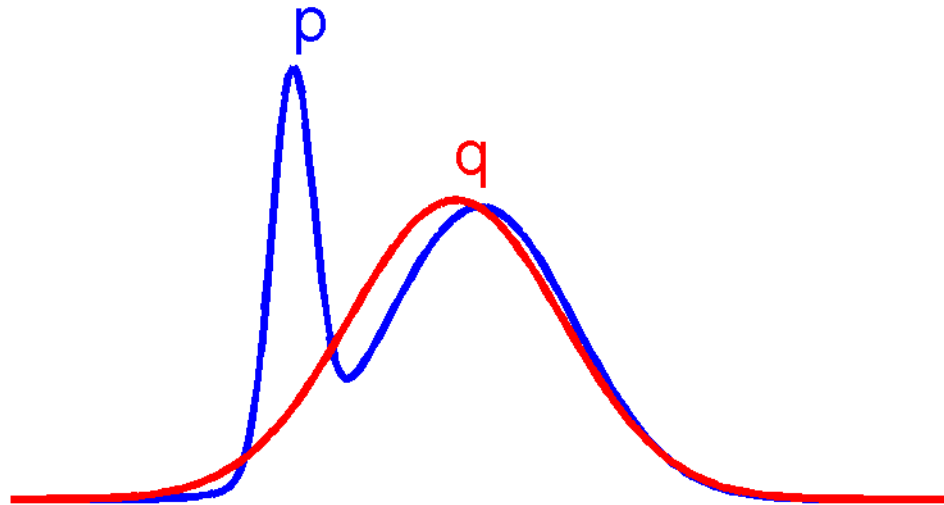
q is Gaussian, minimizes $D_\alpha(p||q)$



$$\alpha = -\infty$$

Minimum alpha-divergence

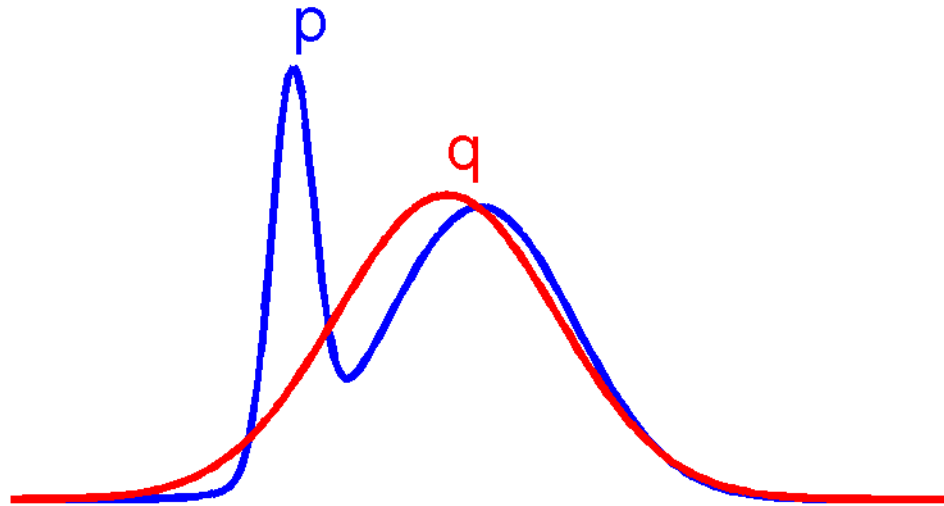
q is Gaussian, minimizes $D_\alpha(p||q)$



$$\alpha = 0$$

Minimum alpha-divergence

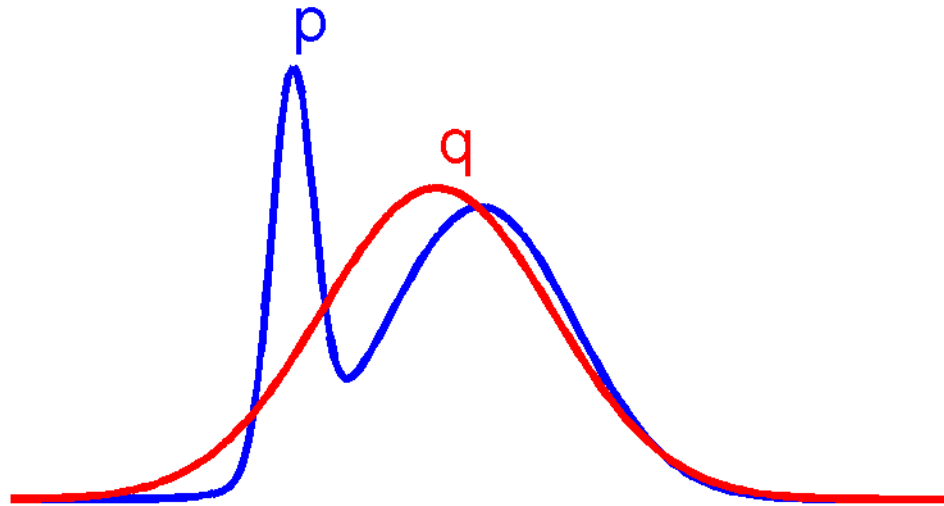
q is Gaussian, minimizes $D_\alpha(p||q)$



$\alpha = 0.5$

Minimum alpha-divergence

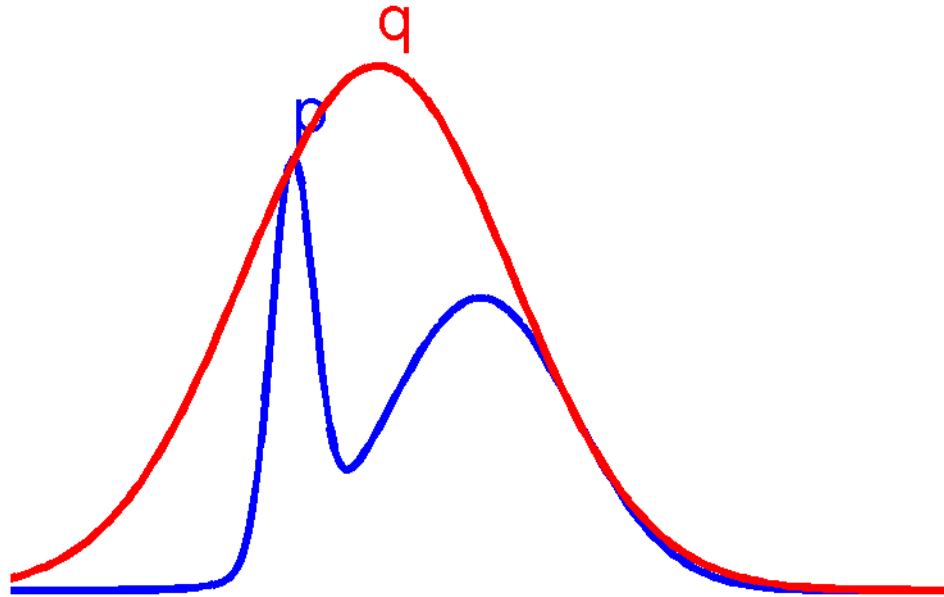
q is Gaussian, minimizes $D_\alpha(p||q)$



$$\alpha = 1$$

Minimum alpha-divergence

q is Gaussian, minimizes $D_\alpha(p||q)$



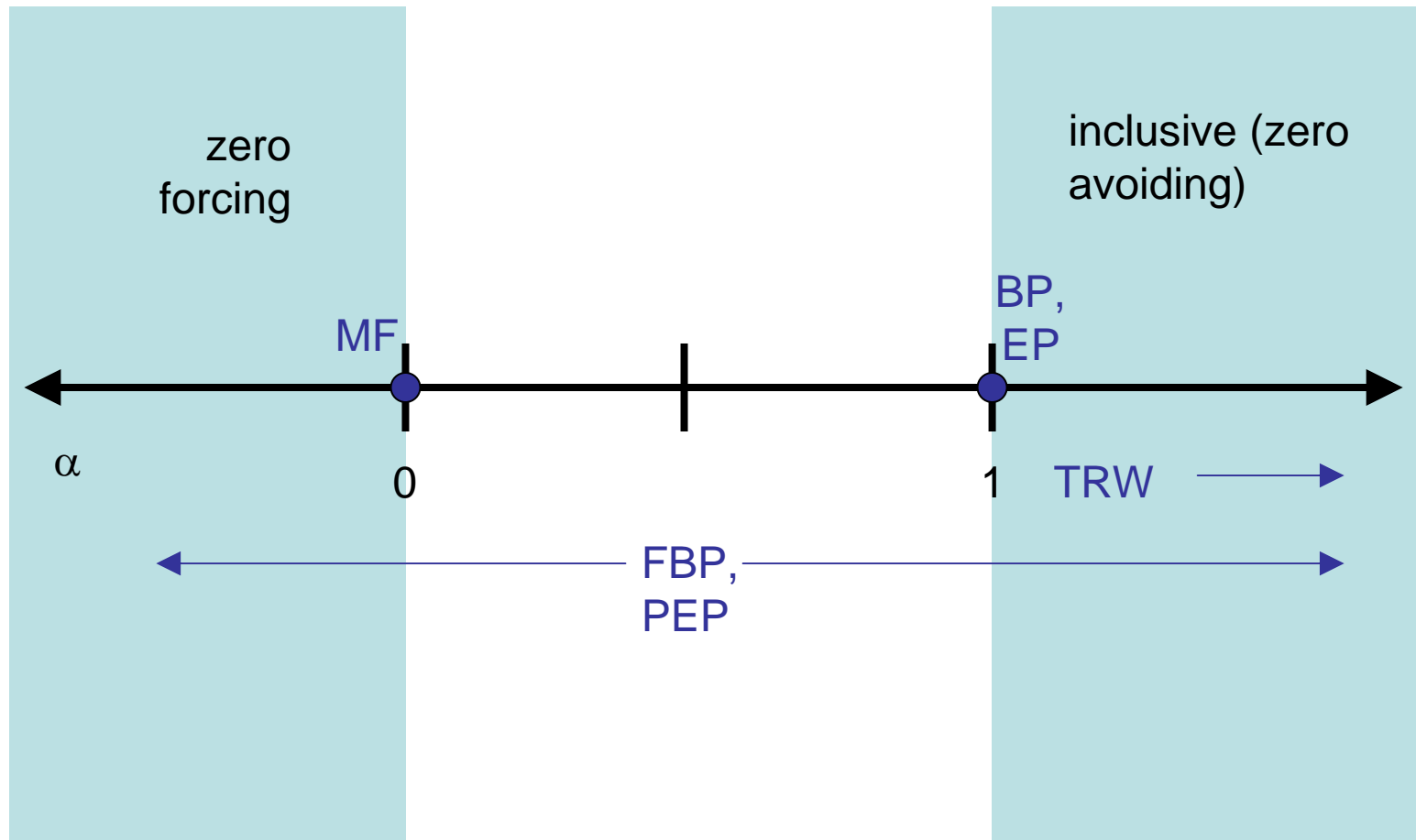
$$\alpha = \infty$$

Properties of alpha-divergence

- $\alpha \leq 0$ seeks the mode with largest mass (not tallest)
 - *zero-forcing*: $p(x)=0$ forces $q(x)=0$
 - underestimates the support of p
- $\alpha \geq 1$ stretches to cover everything
 - *inclusive*: $p(x)>0$ forces $q(x)>0$
 - overestimates the support of p

[Frey, Patrascu, Jaakkola, Moran 00]

Structure of alpha space



Other properties

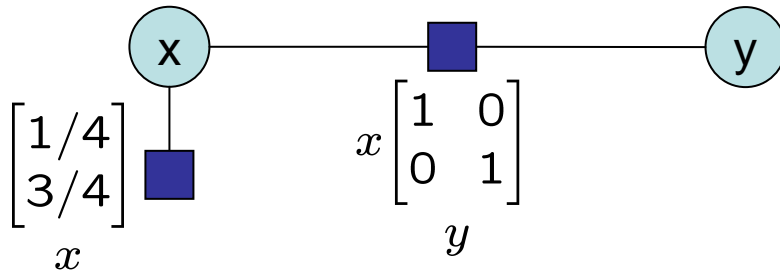
- If q is an exact minimum of alpha-divergence:
- Normalizing constant:

$$\int q(x) dx \leq \int p(x) dx \quad \text{if } \alpha < 1$$

$$\int q(x) dx = \int p(x) dx \quad \text{if } \alpha = 1$$

- If $\int q(x) dx \geq \int p(x) dx$ if $\alpha > 1$
 - Fully factorized q matches marginals of p

Two-node example



$$p(x, y) = \begin{bmatrix} 1/4 \\ 3/4 \end{bmatrix}_x \begin{matrix} x \\ y \end{matrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad p(y) = \begin{bmatrix} 1/4 \\ 3/4 \end{bmatrix}$$

$$q(x, y) = \begin{bmatrix} a \\ b \end{bmatrix}_x \begin{bmatrix} c \\ d \end{bmatrix}_y$$

- q is fully-factorized, minimizes α -divergence to p
- q has correct marginals only for $\alpha = 1$ (BP)

Two-node example

Bimodal
distribution

$$p(x, y) = \begin{matrix} x & \begin{bmatrix} 1/4 & 0 \\ 0 & 3/4 \end{bmatrix} \\ & y \end{matrix}$$

$\alpha = 1$ (BP)

$$q(x, y) = \begin{matrix} \begin{bmatrix} 1/4 \\ 3/4 \end{bmatrix} & \begin{bmatrix} 1/4 \\ 3/4 \end{bmatrix} \\ x & y \end{matrix} = \begin{matrix} x & \begin{bmatrix} 1/16 & 3/16 \\ 3/16 & 9/16 \end{bmatrix} \\ & y \end{matrix}$$

$\alpha = 0$ (MF)

$\alpha \leq 0.5$

$$q(x, y) = \begin{matrix} \begin{bmatrix} 0 \\ \sqrt{3}/2 \end{bmatrix} & \begin{bmatrix} 0 \\ \sqrt{3}/2 \end{bmatrix} \\ x & y \end{matrix} = \begin{matrix} x & \begin{bmatrix} 0 & 0 \\ 0 & 3/4 \end{bmatrix} \\ & y \end{matrix}$$

Good	Bad
<ul style="list-style-type: none"> •Marginals •Mass 	<ul style="list-style-type: none"> •Zeros •Peak heights
<ul style="list-style-type: none"> •Zeros •One peak 	<ul style="list-style-type: none"> •Marginals •Mass

Two-node example

Bimodal
distribution

$$p(x, y) = x \begin{bmatrix} 1/4 & 0 \\ 0 & 3/4 \end{bmatrix} y$$

Good	Bad
•Peak heights	•Zeros •Marginals

$$q(x, y) = \begin{matrix} \alpha = \infty \\ \begin{bmatrix} 1/2 \\ \sqrt{3}/2 \end{bmatrix} \\ x \end{matrix} \begin{matrix} \begin{bmatrix} 1/2 \\ \sqrt{3}/2 \end{bmatrix} \\ y \end{matrix} = x \begin{bmatrix} 1/4 & \sqrt{3}/4 \\ \sqrt{3}/4 & 3/4 \end{bmatrix} y$$

Lessons

- Neither method is inherently superior – depends on what you care about
- A factorized approx does not imply matching marginals (only for $\alpha=1$)
- Adding y to the problem can change the estimated marginal for x (though true marginal is unchanged)

Outline: Section 1

- Graphical models
- Bayesian integration
- **Message Passing**
 - Example of message passing
 - Interpreting message passing
 - Divergence measures
 - **Message passing from a divergence measure**
 - Big picture

Distributed divergence minimization

$$p(x, y, z) = x \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix} y \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix} z \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix} \begin{bmatrix} 0.4 \\ 0.6 \end{bmatrix} \begin{bmatrix} 0.7 \\ 0.3 \end{bmatrix} \begin{bmatrix} 0.8 \\ 0.2 \end{bmatrix}$$

$$qq(x, y, z) = \begin{bmatrix} .59 \\ .41 \end{bmatrix} \begin{bmatrix} .5 \\ .5 \end{bmatrix} \begin{bmatrix} .6 \\ .4 \end{bmatrix} \begin{bmatrix} .57 \\ .43 \end{bmatrix} \begin{bmatrix} .61 \\ .39 \end{bmatrix} \begin{bmatrix} .5 \\ .5 \end{bmatrix} \begin{bmatrix} 0.4 \\ 0.6 \end{bmatrix} \begin{bmatrix} 0.7 \\ 0.3 \end{bmatrix} \begin{bmatrix} 0.8 \\ 0.2 \end{bmatrix}$$

$$q(x, y, z) = \begin{bmatrix} 0.61 \\ 0.39 \end{bmatrix} \begin{bmatrix} 0.78 \\ 0.22 \end{bmatrix} \begin{bmatrix} 0.84 \\ 0.16 \end{bmatrix}$$

Distributed divergence minimization

- Write p as product of factors:

$$p(x) = \prod_a t_a(x)$$

- Approximate factors one by one:

$$t_a(x) \rightarrow \tilde{t}_a(x)$$

- Multiply to get the approximation:

$$q(x) = \prod_a \tilde{t}_a(x)$$

Global divergence to local divergence

- Global divergence:

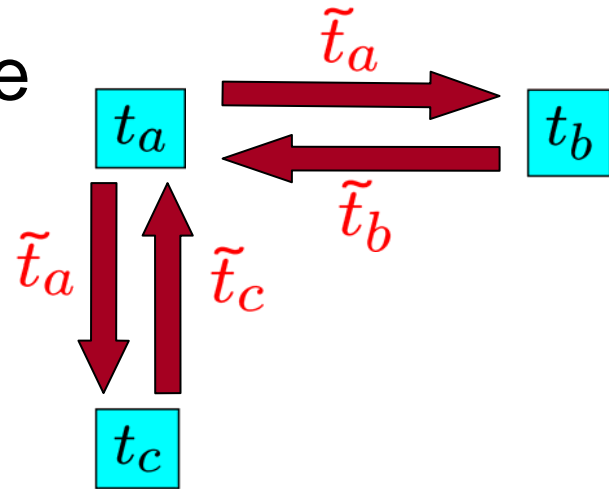
$$D(p(x) \parallel q(x)) = D(t_a(x) \prod_{b \neq a} t_b(x) \parallel \tilde{t}_a(x) \prod_{b \neq a} \tilde{t}_b(x))$$

- Local divergence:

$$D(t_a(x) \prod_{b \neq a} \tilde{t}_b(x) \parallel \tilde{t}_a(x) \prod_{b \neq a} \tilde{t}_b(x))$$

Message passing

- Messages are passed between *factors*
- Messages are factor approximations: $\tilde{t}_a(x)$
- Factor a receives $\tilde{t}_b(x), b \neq a$
 - Minimize local divergence to get $\tilde{t}_a(x)$
 - Send to other factors
 - Repeat until convergence
- Produces all 6 algs



Global divergence vs. local divergence



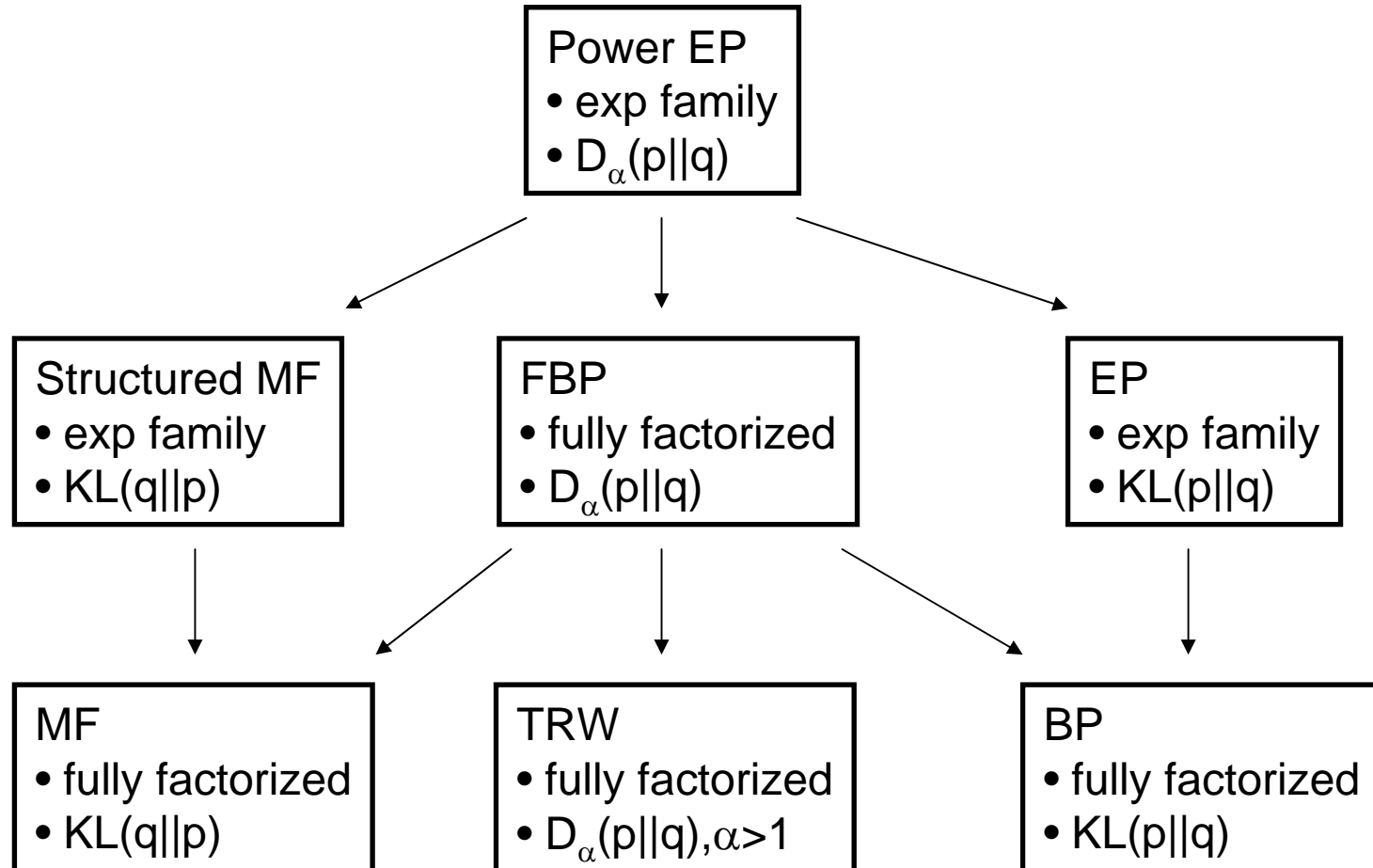
In general, local \neq global

- but results are similar
- BP doesn't minimize global KL, but comes close

Outline: Section 1

- Graphical models
- Bayesian integration
- **Message Passing**
 - Example of message passing
 - Interpreting message passing
 - Divergence measures
 - Message passing from a divergence measure
 - **Big picture**

Hierarchy of algorithms



Other Message Passing Algorithms

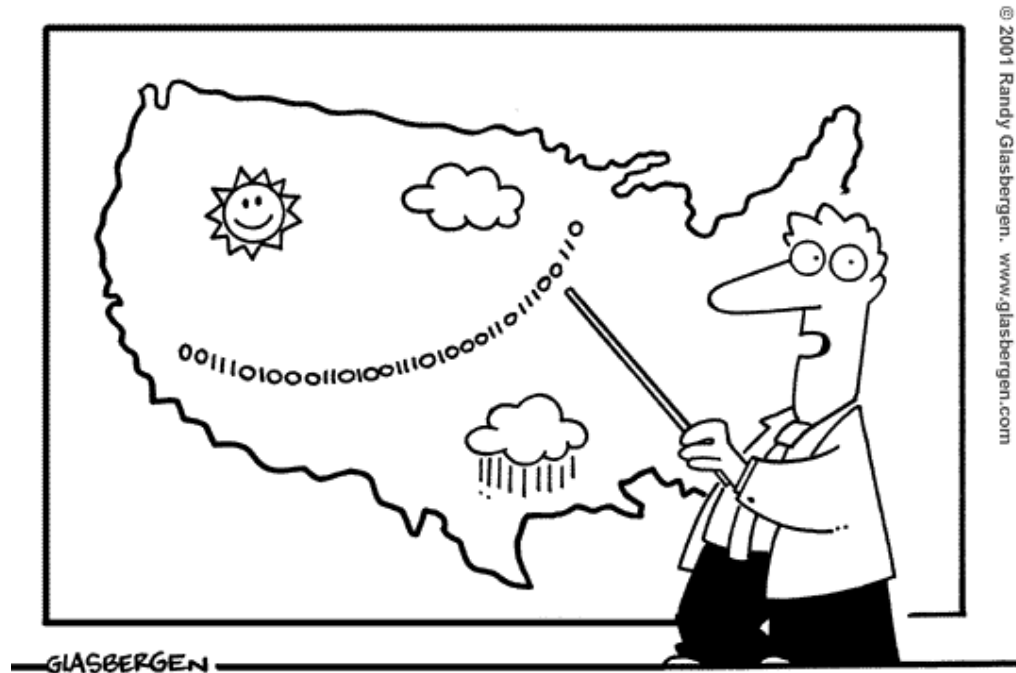
Do they correspond to divergence measures?

- Generalized belief propagation [Yedidia, Freeman, Weiss 00]
- Iterated conditional modes [Besag 86]
- Max-product belief revision
- TRW-max-product [Wainwright, Jaakkola, Willsky 02]
- Laplace propagation [Smola, Vishwanathan, Eskin 03]
- Penniless propagation [Cano, Moral, Salmerón 00]
- Bound propagation [Leisink, Kappen 03]

Outline: section 2

- Inference on graphical models
 - EP on hybrid dynamic networks for wireless signal detection
 - Tree-structured approximation for message passing on loopy graphs
 - Message approximation for detecting Protein-DNA binding sites
- Learning conditional graphical models
 - Bayesian conditional random fields for handwritten ink analysis and news group parsing

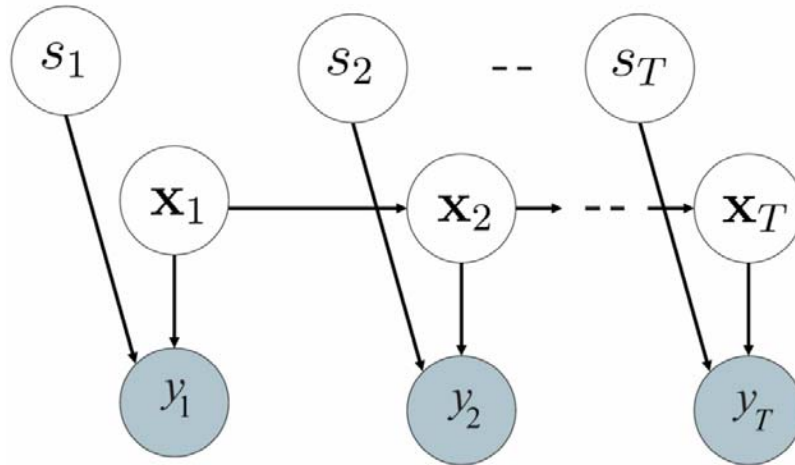
Wireless signal detection



"And on Friday, we'll have a large area of wireless data blowing in from the West..."

How to recover original information from received noisy wireless signal?

Hybrid Bayesian network modeling

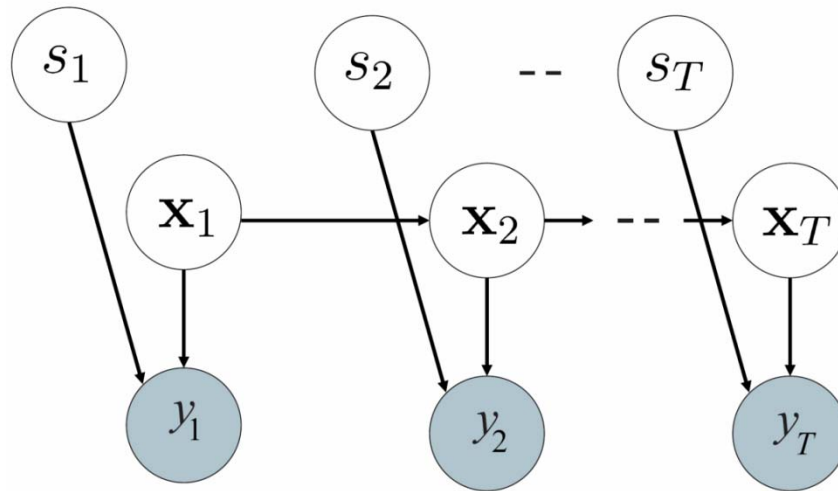


$$\mathbf{x}_t = \mathbf{F}\mathbf{x}_{t-1} + \mathbf{g}_t v_t \quad v_t \sim \mathcal{N}_c(0, 1)$$

$$y_t = s_t \mathbf{h}^H \mathbf{x}_t + w_t \quad w_t \sim \mathcal{N}_c(0, \sigma^2)$$

- y_t : Received noisy observation at time t
- s_t : Transmitted signal at time t
- \mathbf{x}_t : Channel coefficient for wireless communications

Signal detection: Inference on nonlinear dynamic systems



Bayesian Inference: compute posterior distribution

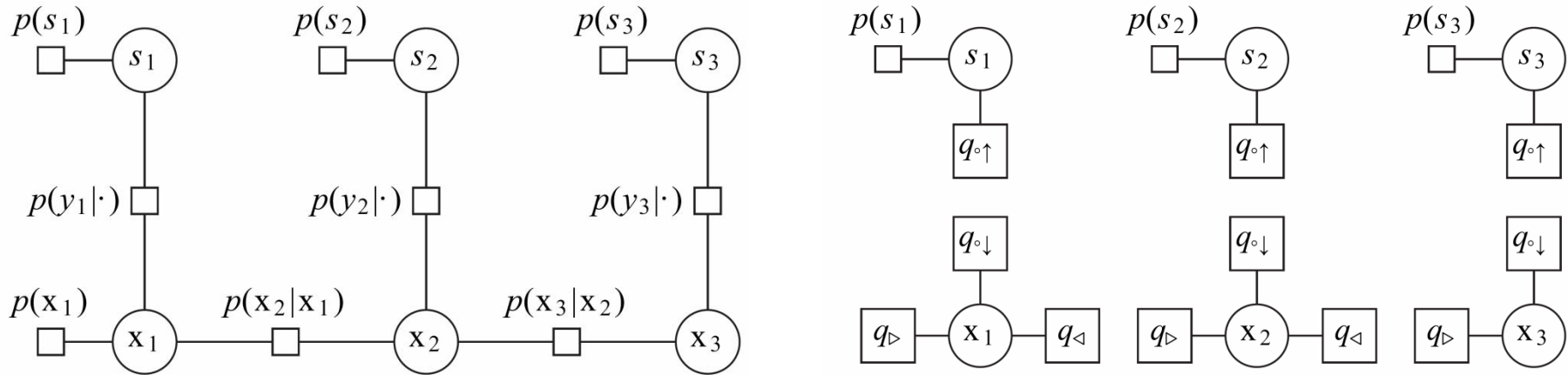
$$p(s_t, \mathbf{x}_t | y_{1:t+L-1})$$

based on received signal $y_{1:t+L-1} = [y_1, \dots, y_{t+L-1}]$

Previous Approaches

- Pilot-symbols aided approach (D'Andrea et al. 1995)
 - Using additional pilot symbols for estimating the channel coefficients
- Sequential Monte Carlo Approach: Rao-blackwellized particle filters and smoothers (Chen et al. 2000 and Wang et al. 2002)
 - Accurate estimation
 - Expensive computation (Infeasible in practice)

Factor graph & posterior distributions



The posteriors of the variables (circles) are proportional to the product of factors (rectangles).

Exact posterior distribution

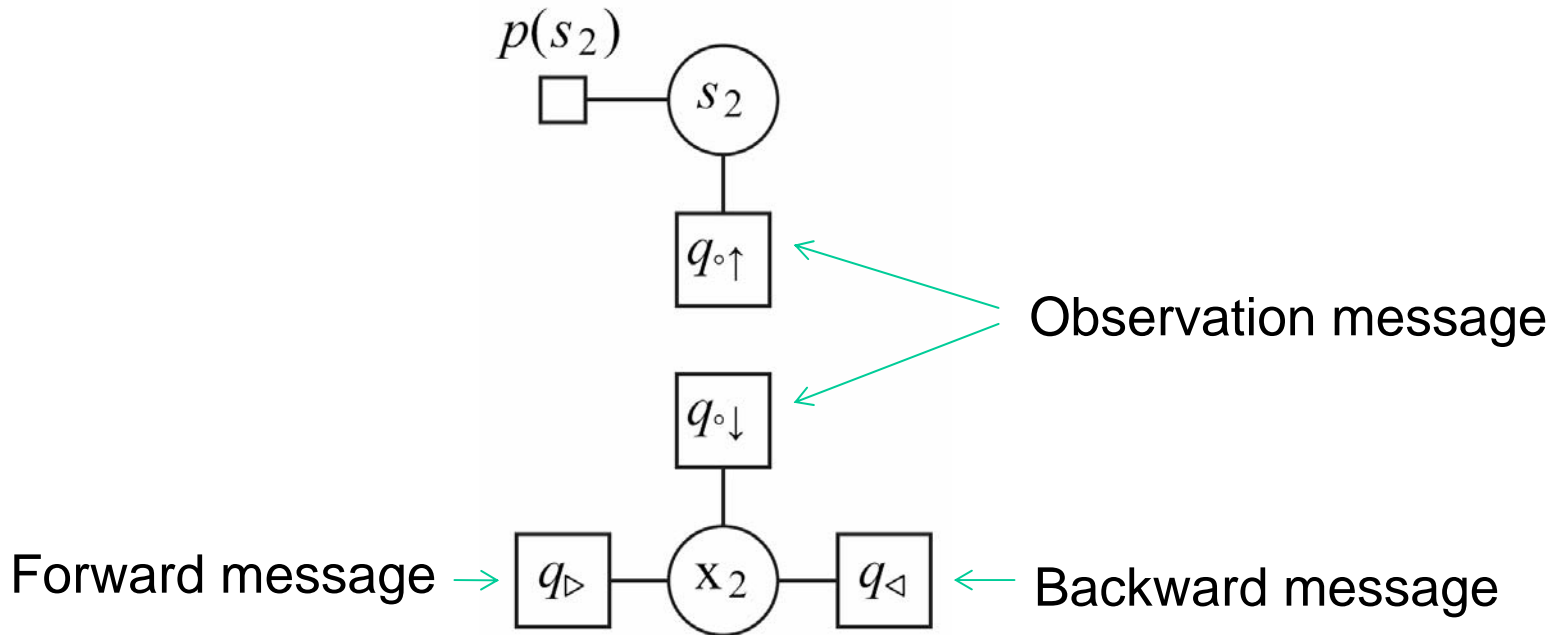
$$\begin{aligned}
 & p(s_{1:T}, \mathbf{x}_{1:T} | y_{1:T}) \\
 & \propto p(s_1) p(\mathbf{x}_1) p(y_1 | s_1, \mathbf{x}_1) \cdot \\
 & \quad \prod_{t=2}^T p(\mathbf{x}_t | \mathbf{x}_{t-1}) p(s_t) p(y_t | s_t, \mathbf{x}_t)
 \end{aligned}$$

Approximate posterior distribution

$$q(s_{1:T}, \mathbf{x}_{1:T}) \propto \prod_{t=1}^T q_{\triangleright}(\mathbf{x}_t) p(s_t) q_{o\uparrow}(s_t) q_{o\downarrow}(\mathbf{x}_t) q_{\triangleleft}(\mathbf{x}_t)$$

Here $q_{o\uparrow} \equiv q_o(s_t)$, and $q_{o\downarrow} \equiv q_o(\mathbf{x}_t)$.

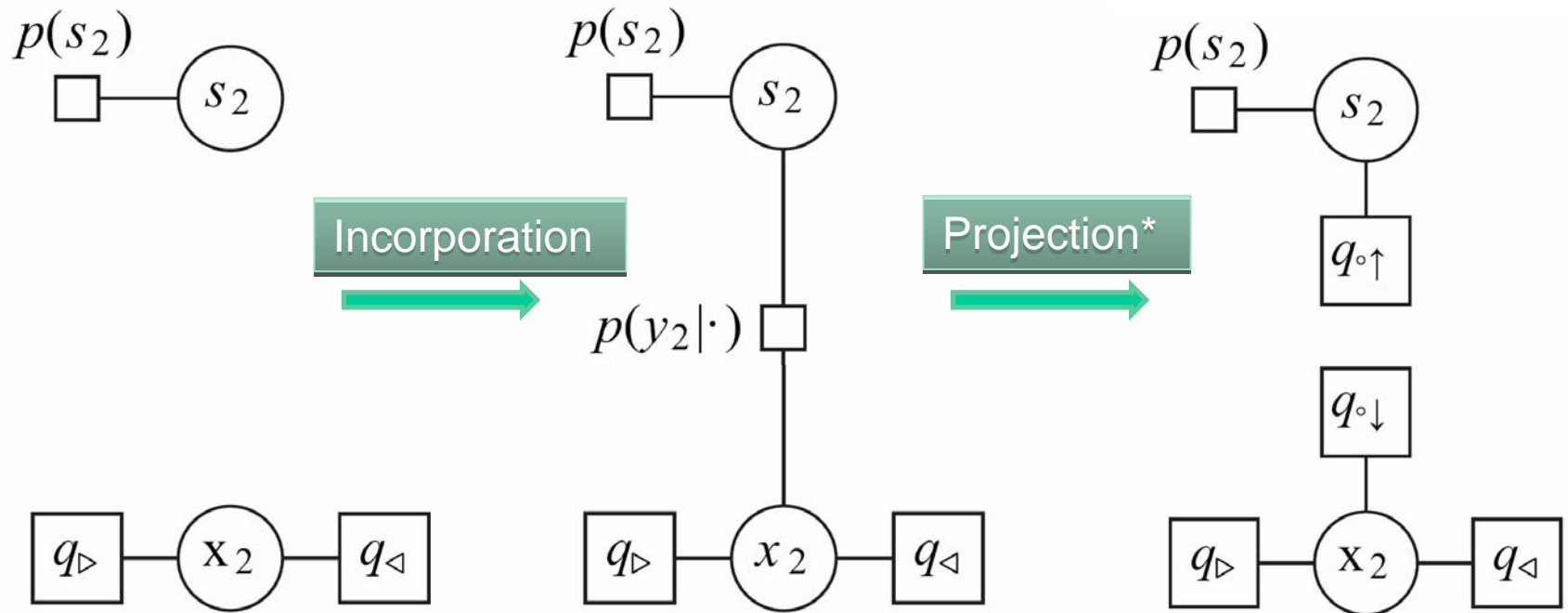
Message passing in factor graph



Approximate state belief of $x_2 =$ Forward message • Observation message
• backward message

How to compute messages to accurately approximate belief/posterior?

Expectation propagation (EP)

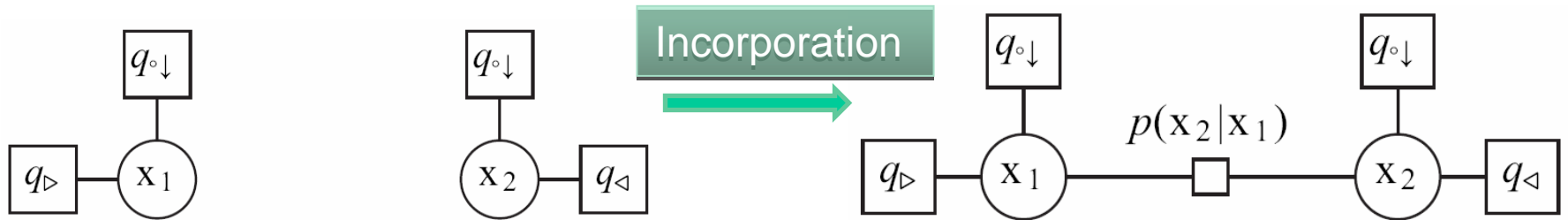


(a) $p(s_2)q_{\triangleright\triangleleft}(x_2)$ (b) $p(y_2|s_2, x_2)p(s_2)q_{\triangleright\triangleleft}(x_2)$ (c) $q(s_2)q(x_2)$

EP processes the observation distribution $p(y_2|s_2, x_2)$ and update messages.

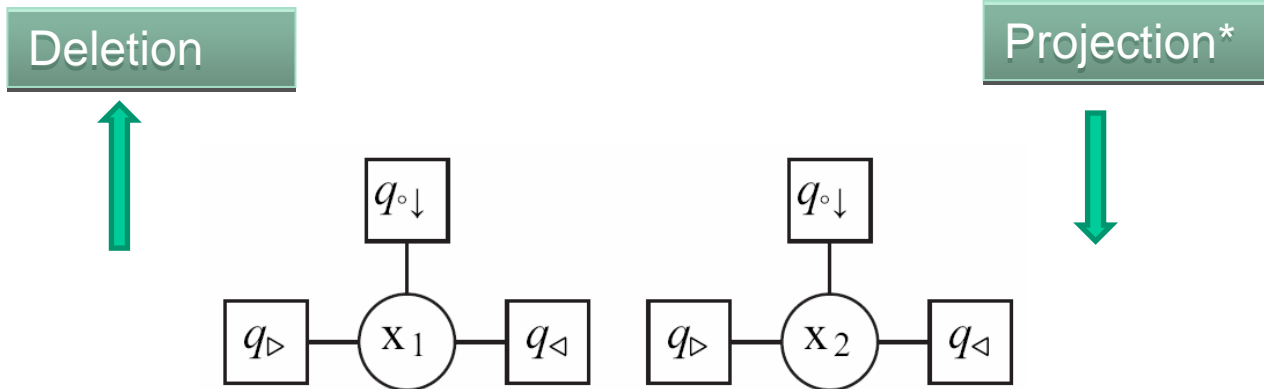
Projection: minimize $\zeta\mathbb{L}(p(y_t|s_t, \mathbf{x}_t)p(s_t)q_{\triangleright\triangleleft}(\mathbf{x}_t) \| q(s_t)q(\mathbf{x}_t))$

EP processes the transition distribution



(a) $q_{\triangleright\circ}(x_1)q_{\circ\triangleleft}(x_2)$

(b) $q_{\triangleright\circ}(x_1)p(x_2|x_1)q_{\circ\triangleleft}(x_2)$



(c) $q(x_1)q(x_2)$

Projection: $\min KL = 0 \Rightarrow$ No approximation

Computational complexity

Algorithm	Complexity
Fix-lag EP smoothing	$O(nLd^2)$
Stochastic mixture of Kalman filters	$O(MLd^2)$
Rao-blackwised particle smoothers	$O(MNLd^2)$

L : Length of fixed-lag smooth window

d : Dimension of the parameter vector

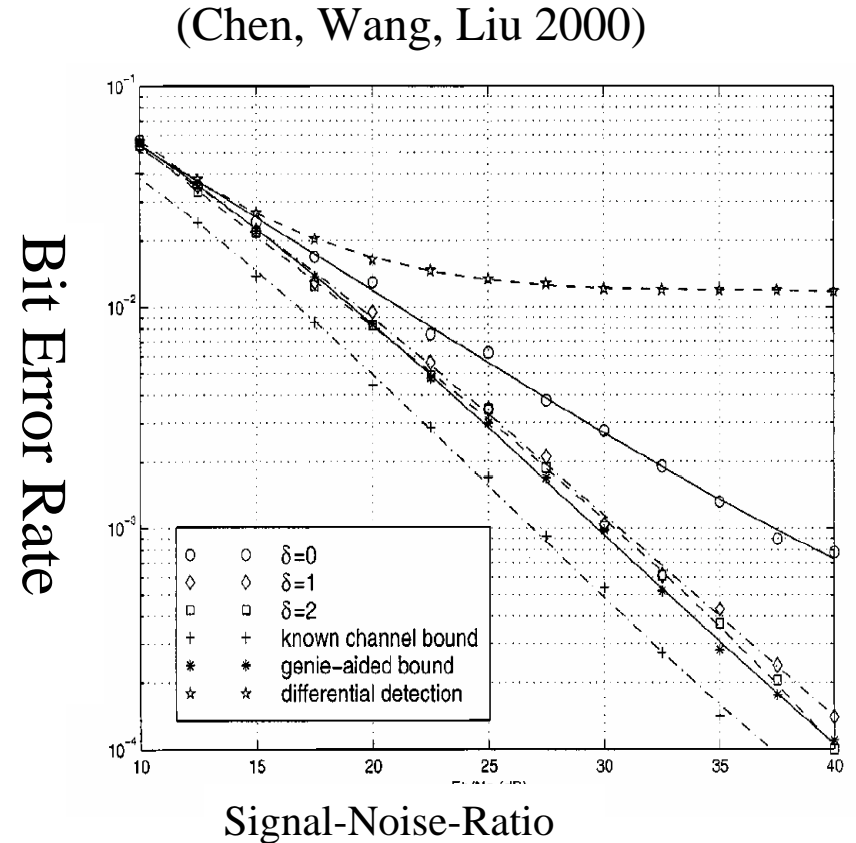
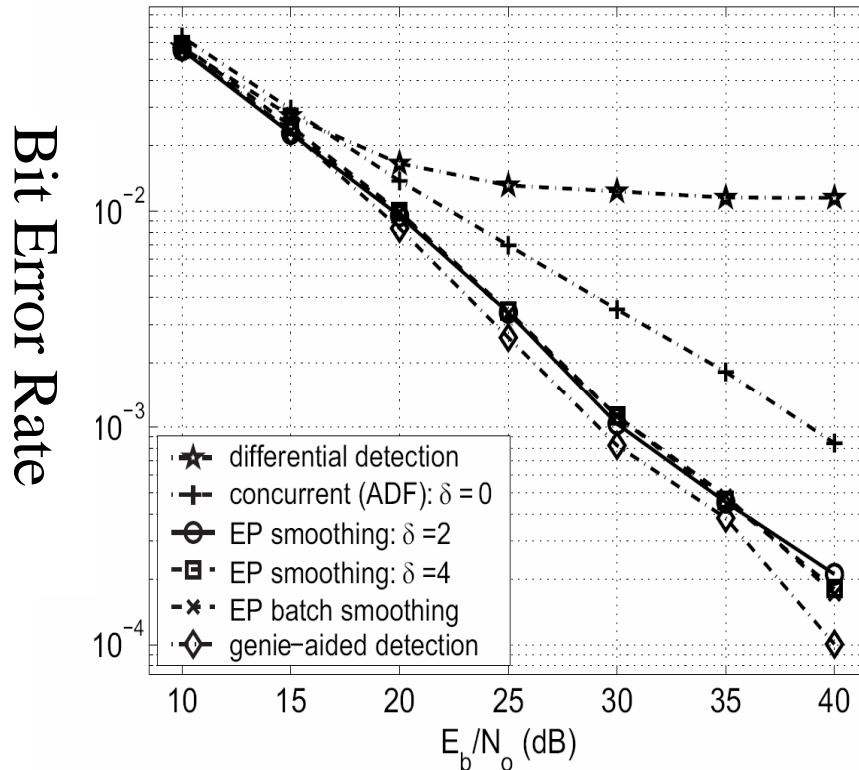
n : Number of EP iterations (Typically, 4 or 5)

M : Number of samples in filtering (Often larger than 500 or 100)

N : Number of samples in smoothing (Larger than 50)

Fixed-lag EP smoothing is 10 to 100 times faster than Rao-blackwellised particle smoothers!

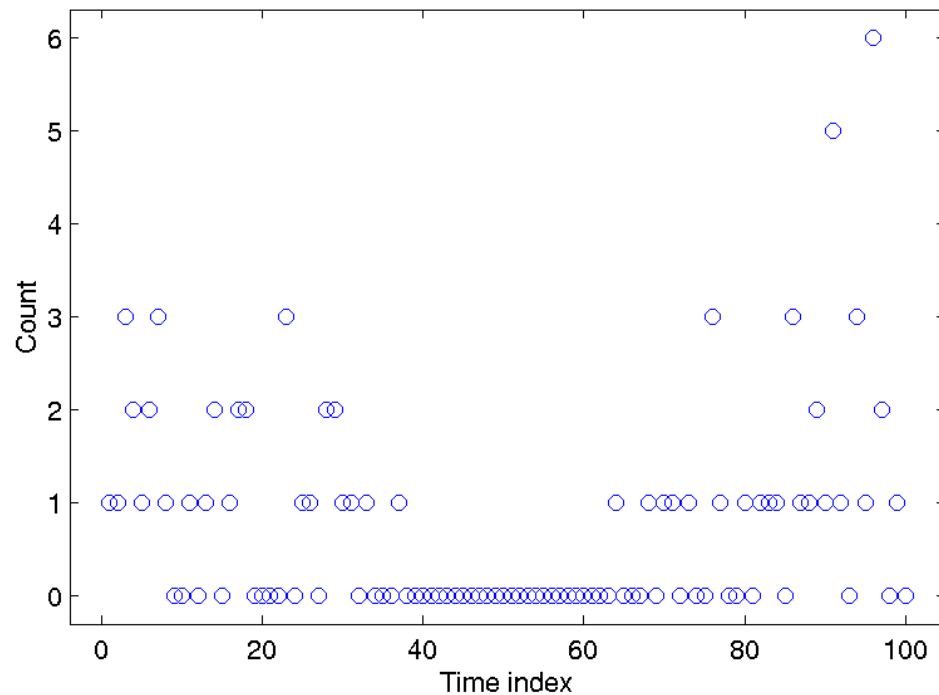
Estimation accuracy



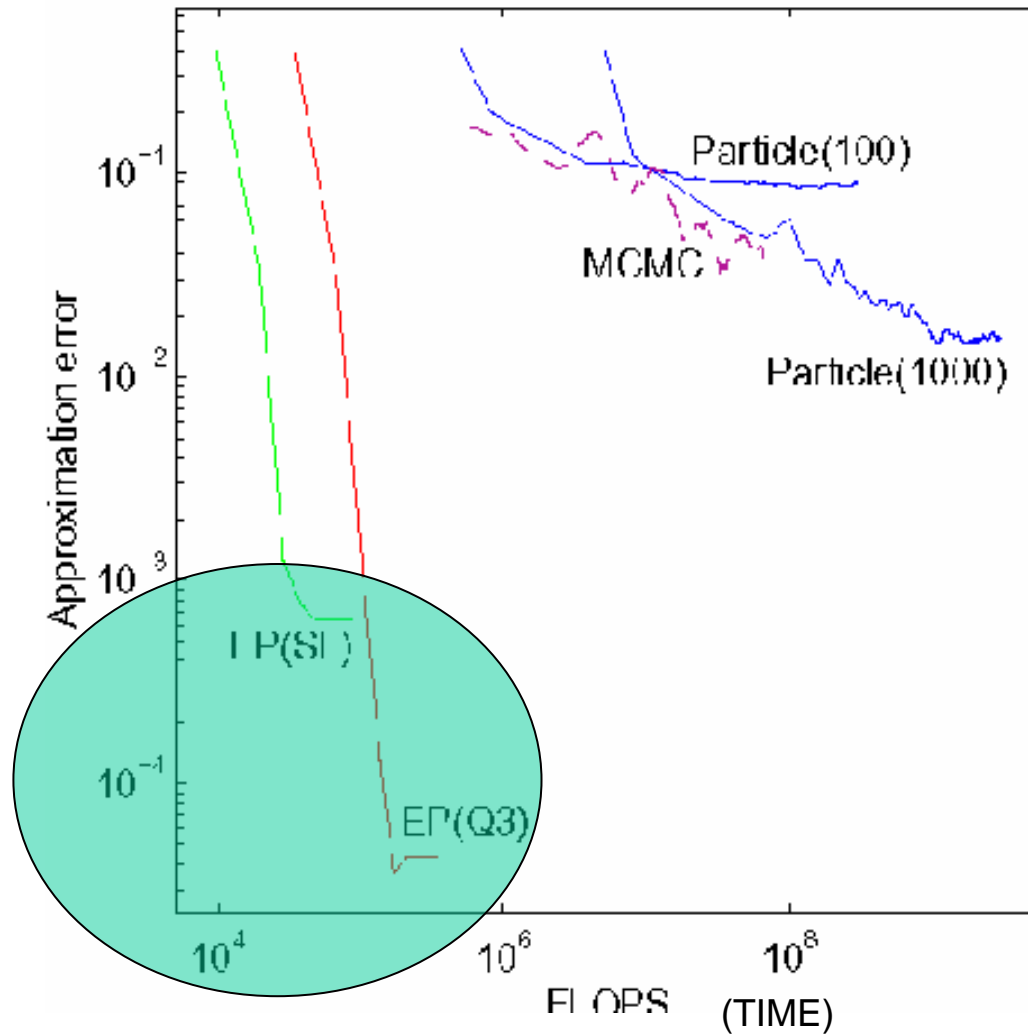
EP achieves accuracy comparable to particle smoothers, but at least times faster.

Another example: Poisson Tracking

y_t is an integer-valued Poisson variate with $\exp(x_t)$ mean where $\{x_t\}$ is a hidden Markovian process.



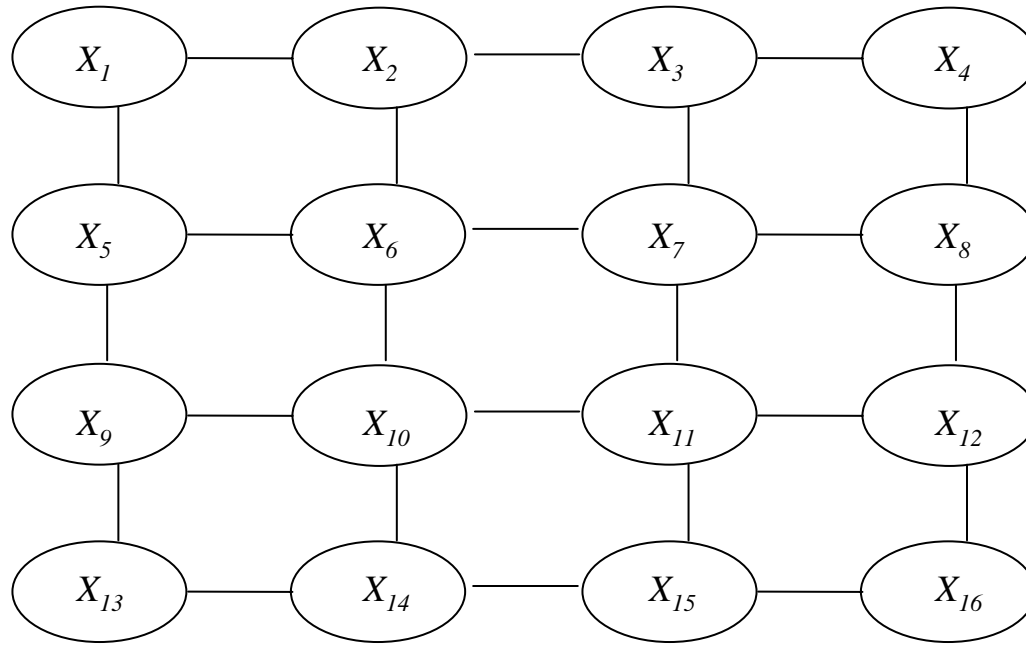
Accuracy/Efficiency Tradeoff



Outline: section 2

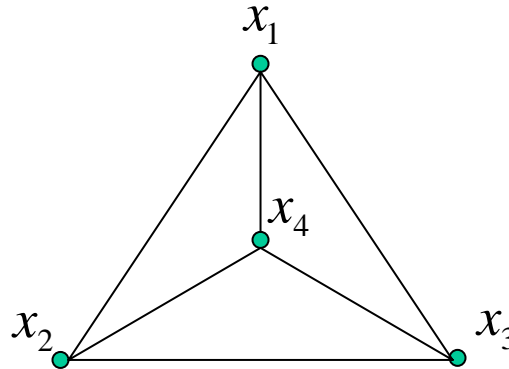
- Inference on graphical models
 - EP on hybrid dynamic networks for wireless signal detection
 - Tree-structured approximation for message passing on loopy graphs
 - Message approximation for detecting Protein-DNA binding sites
- Learning conditional graphical models
 - Bayesian conditional random fields for handwritten ink analysis and news group parsing

Inference on Markov random fields



Problem: estimate marginal distributions of the variables indexed by the nodes in a Markov random field (MRF), e.g., $p(\mathbf{x}_i)$, $i = 1, \dots, 16$.

4-node MRF

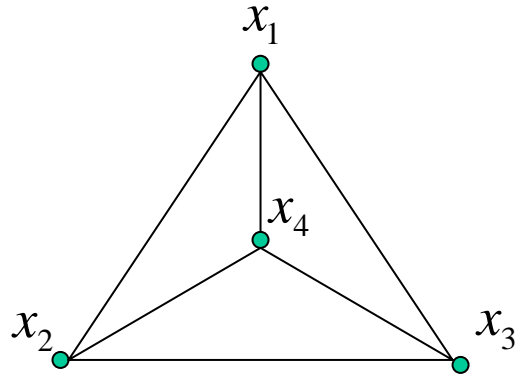


Joint distribution is product of pairwise potentials for all edges:

$$p(\mathbf{x}) = \prod_a f_a(\mathbf{x})$$

Want to approximate $p(\mathbf{x})$ by a simpler distribution

BP vs. TreeEP



projection

projection

BP

x_1

x_4

x_2

x_3

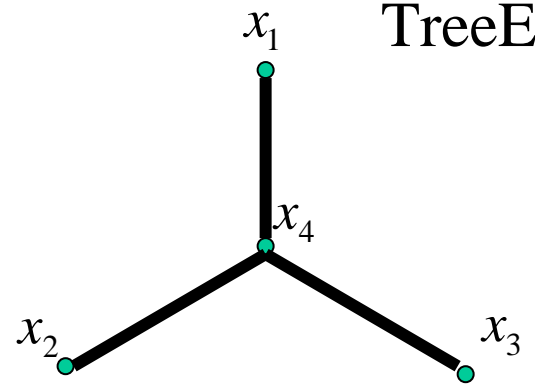
TreeEP

x_1

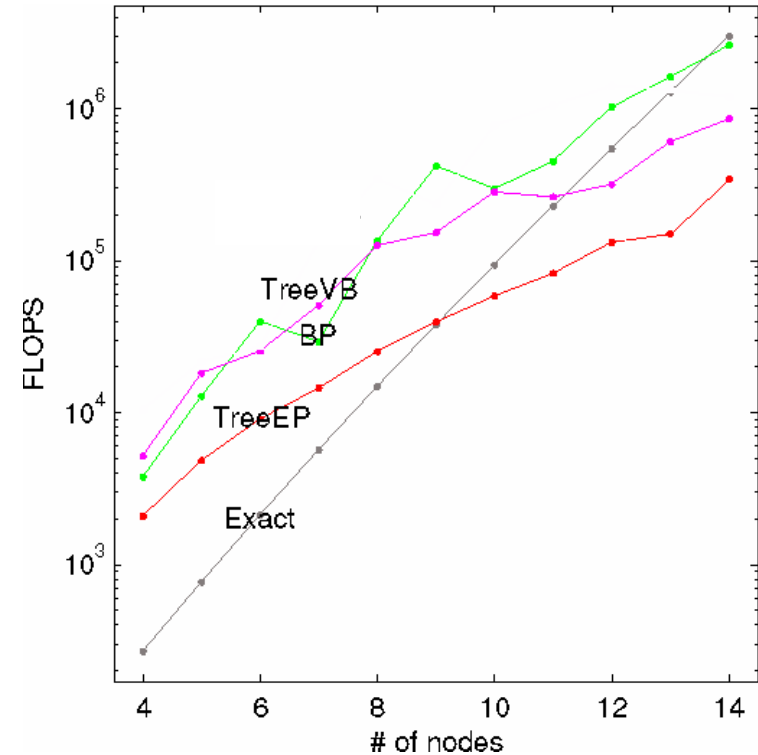
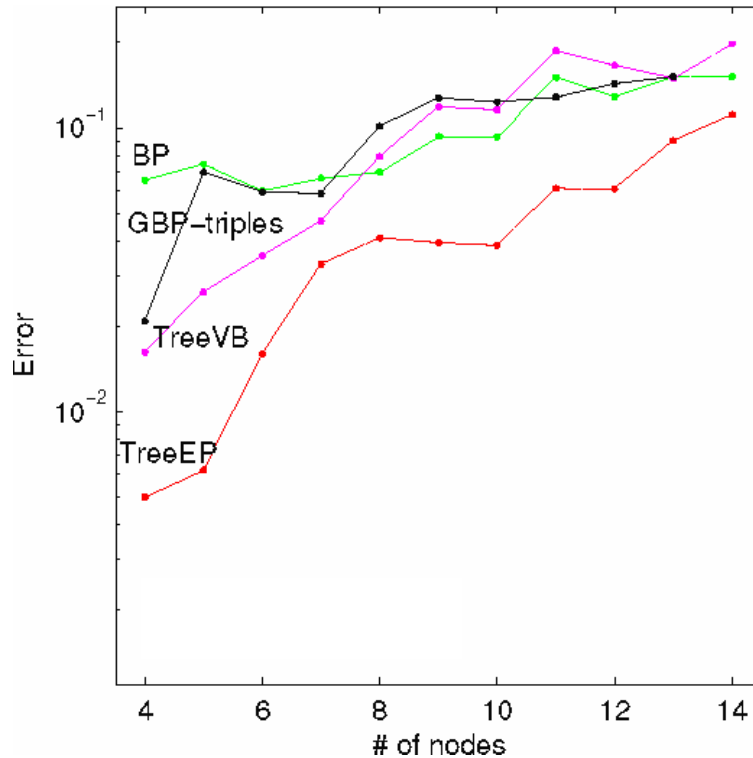
x_4

x_2

x_3



Fully-connected graphs



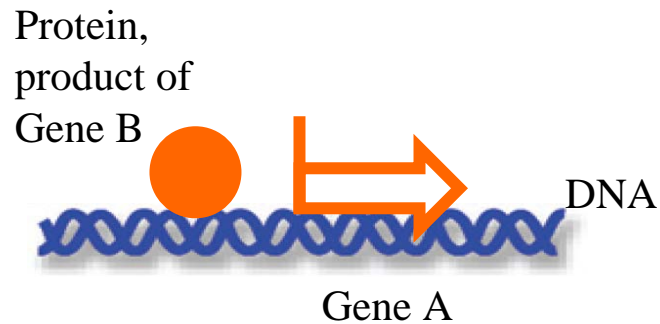
Results are averaged over 10 graphs with randomly generated potentials

TreeEP performs the same or better than all other methods in both accuracy and efficiency!

Outline: section 2

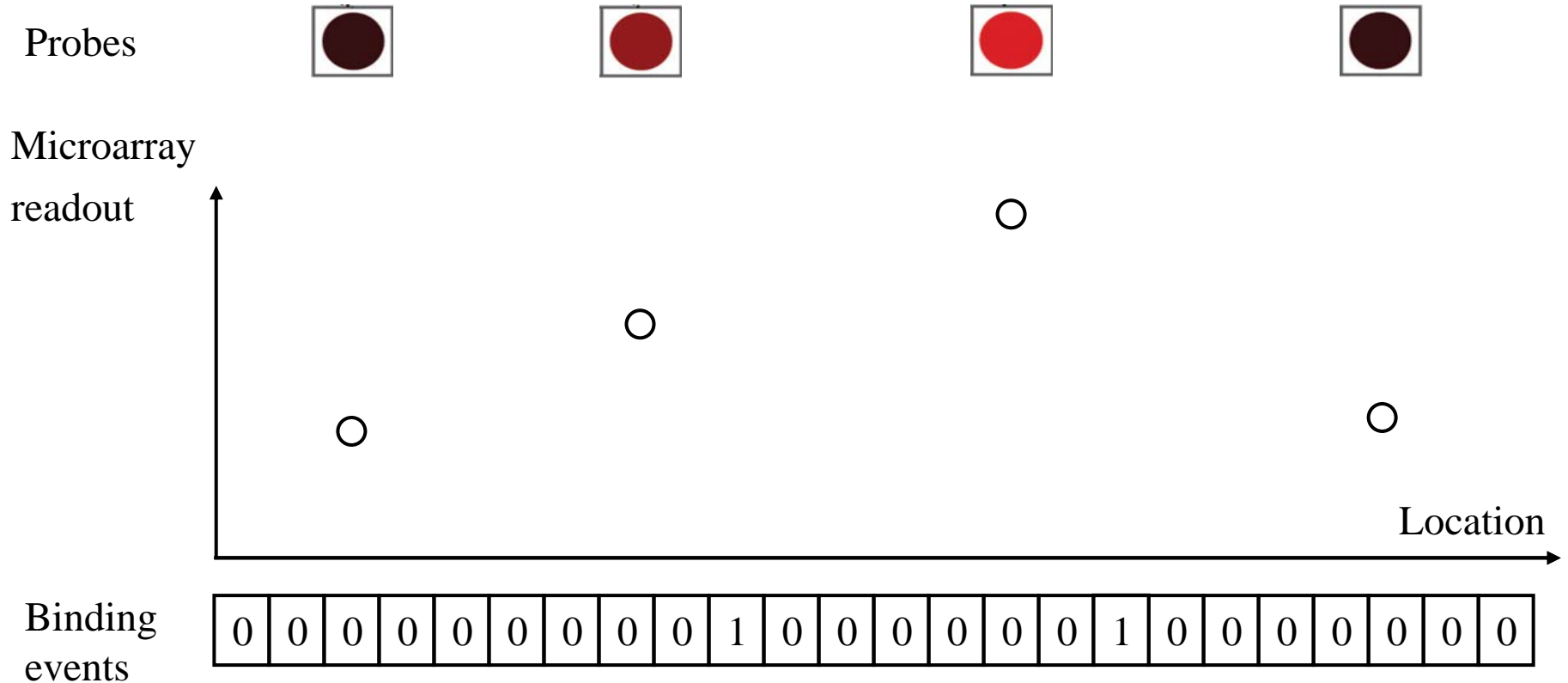
- Inference on graphical models
 - EP on hybrid dynamic networks for wireless signal detection
 - Tree-structured approximation for message passing on loopy graphs
 - Message approximation for detecting Protein-DNA binding sites
- Learning conditional graphical models
 - Bayesian conditional random fields for handwritten ink analysis and news group parsing

Inferring high-resolution protein-DNA binding locations



Identify whole-genome protein-DNA binding sites based on high-throughput biological data

Estimate binding sites from ChIP-chip data



1: Protein-DNA binding

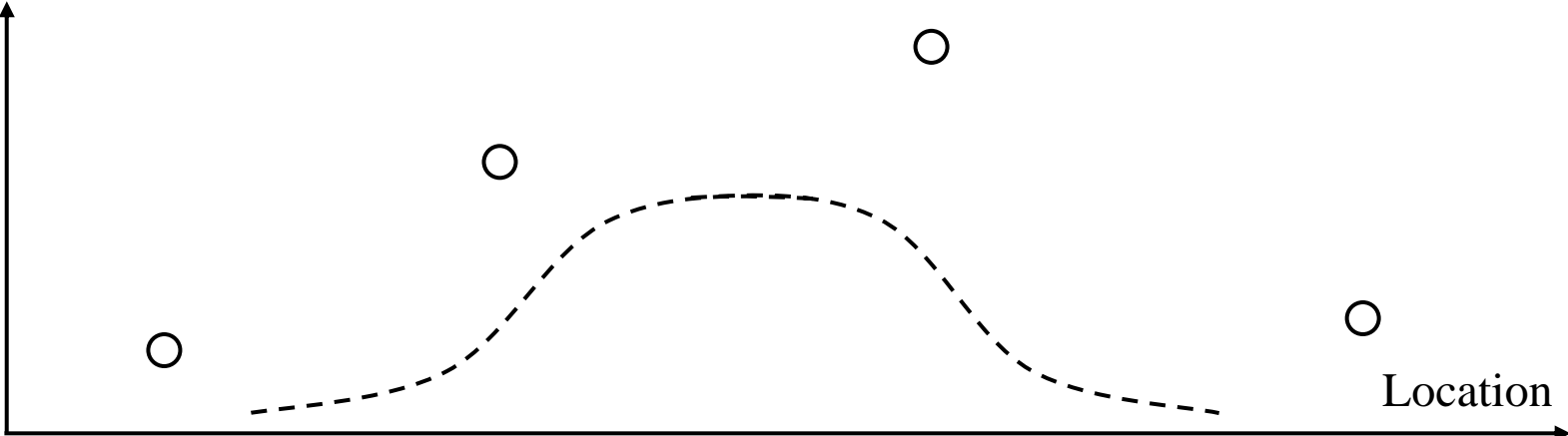
0: no binding

Impulse response of a single binding event

Probes



Microarray readout



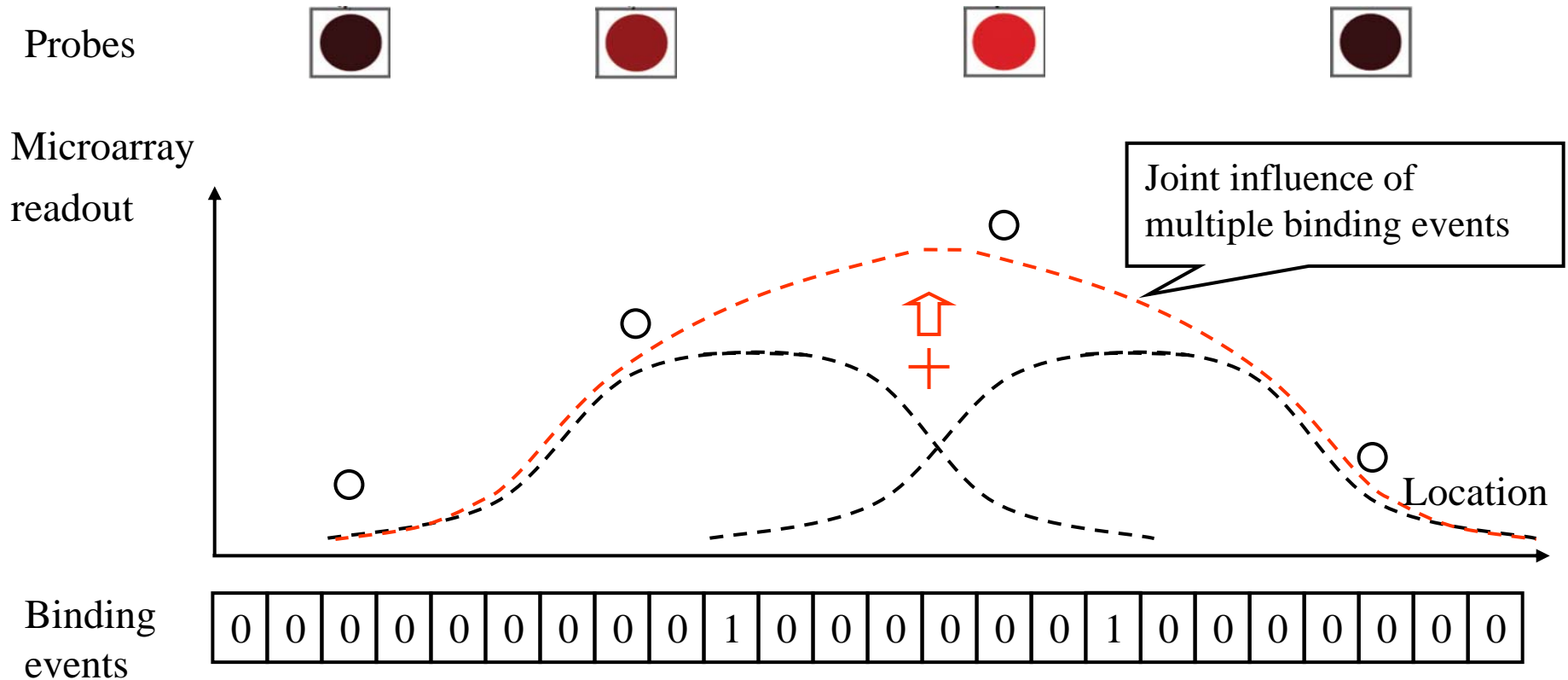
Binding events

0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

1: Protein-DNA binding

0: no binding

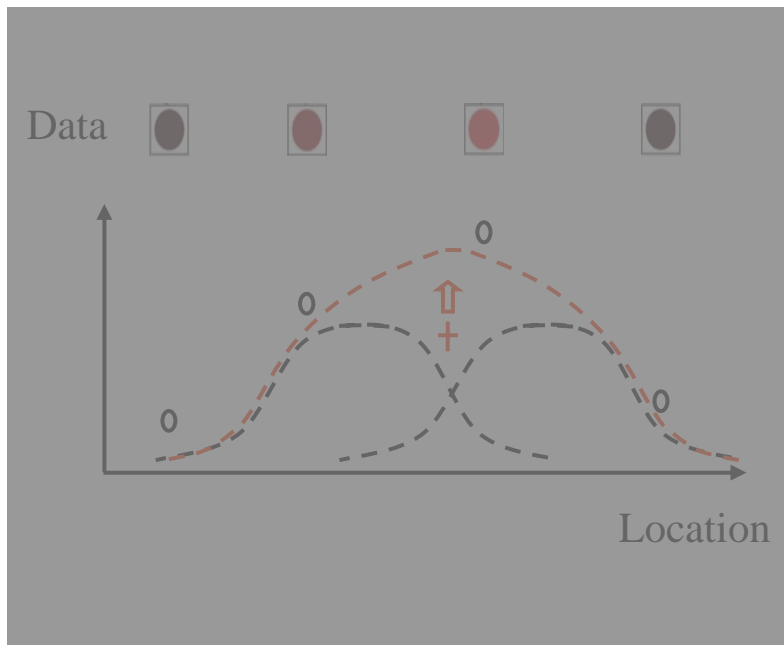
Unknown number of binding events



1: Protein-DNA binding

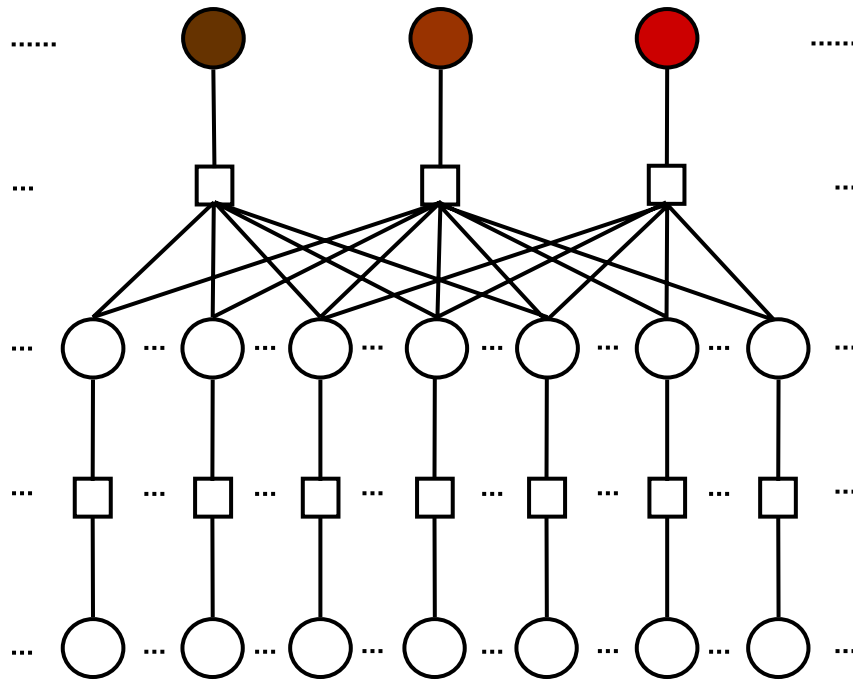
0: no binding

Previous approaches



- Ratio (thresholding microarray data)
Sensitive to noise
- Rosetta (Boyer, L.A. et al., *Cell* 2005)
Probabilistic, but ignoring impulse response function
- MPeak (Kim, T.H. et al., *Nature* 2005)
Ignoring joint influence of multiple impulse response functions

Joint Binding Deconvolution (JBD) models data generation process (Qi et al. Nature biotech. 2006)



Data: y_i

Impulse response

Likelihood $o_i(\mathbf{r})$: $N(y_i / \sum_j a_{|i-j|} r_j, \sigma)$

Binding events: r_j

Binding prior: $p(r_j / \pi_j)$

Prior parameter: π_j

Factor graph representation of JBD

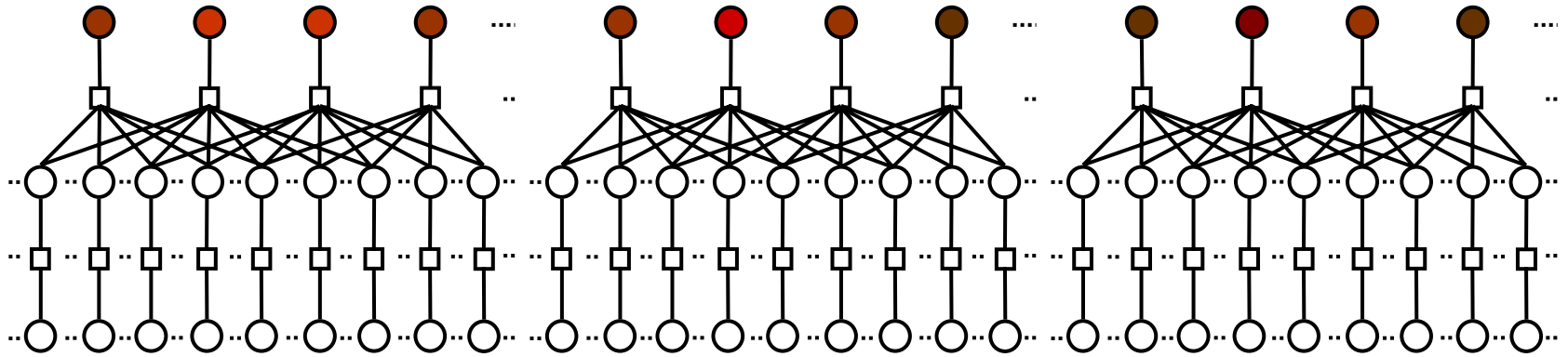
The joint distribution of variables (circles) is proportional to the product of factors (squares)

Exact posterior distribution:

$$p(\mathbf{r}, \boldsymbol{\pi} | \mathbf{y}) \propto \prod_i o_i(\mathbf{r}) \prod_j u_j(r_j, \pi_j) \prod_j p(\pi_j)$$

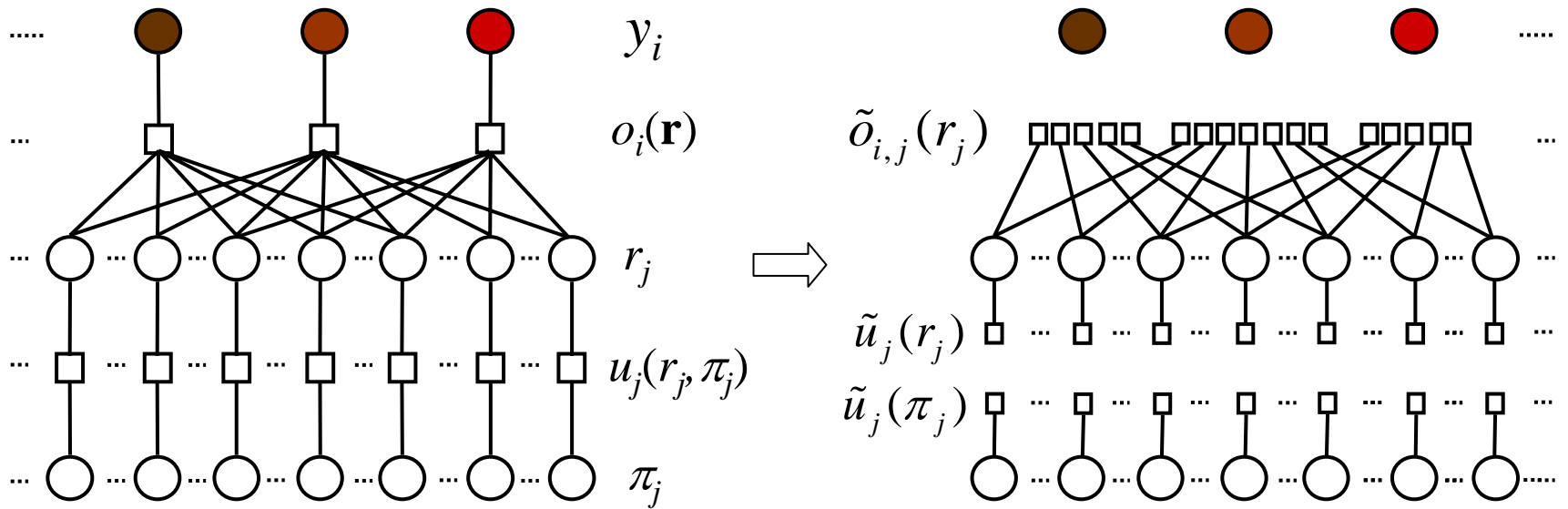
$p(\pi_j)$: not included in the graph

Inference on JBD graphical model is difficult



- Large scale (yeast: ~1.5M hidden variable nodes; human: ~150M hidden variable nodes)
- Strong coupling

EP approximation makes JBD efficient



Exact posterior distribution:

$$p(\mathbf{r}, \boldsymbol{\pi} \mid \mathbf{y}) \propto \prod_i o_i(\mathbf{r}) \prod_j u_j(r_j, \pi_j) \prod_j p(\pi_j)$$

$$o_i(\mathbf{r}) = \mathcal{N}(y_i / \sum_j a_{|i-j|} r_j, \sigma)$$

$$u_j(r_j, \pi_j) = p(r_j / \pi_j)$$

Approximate posterior distribution:

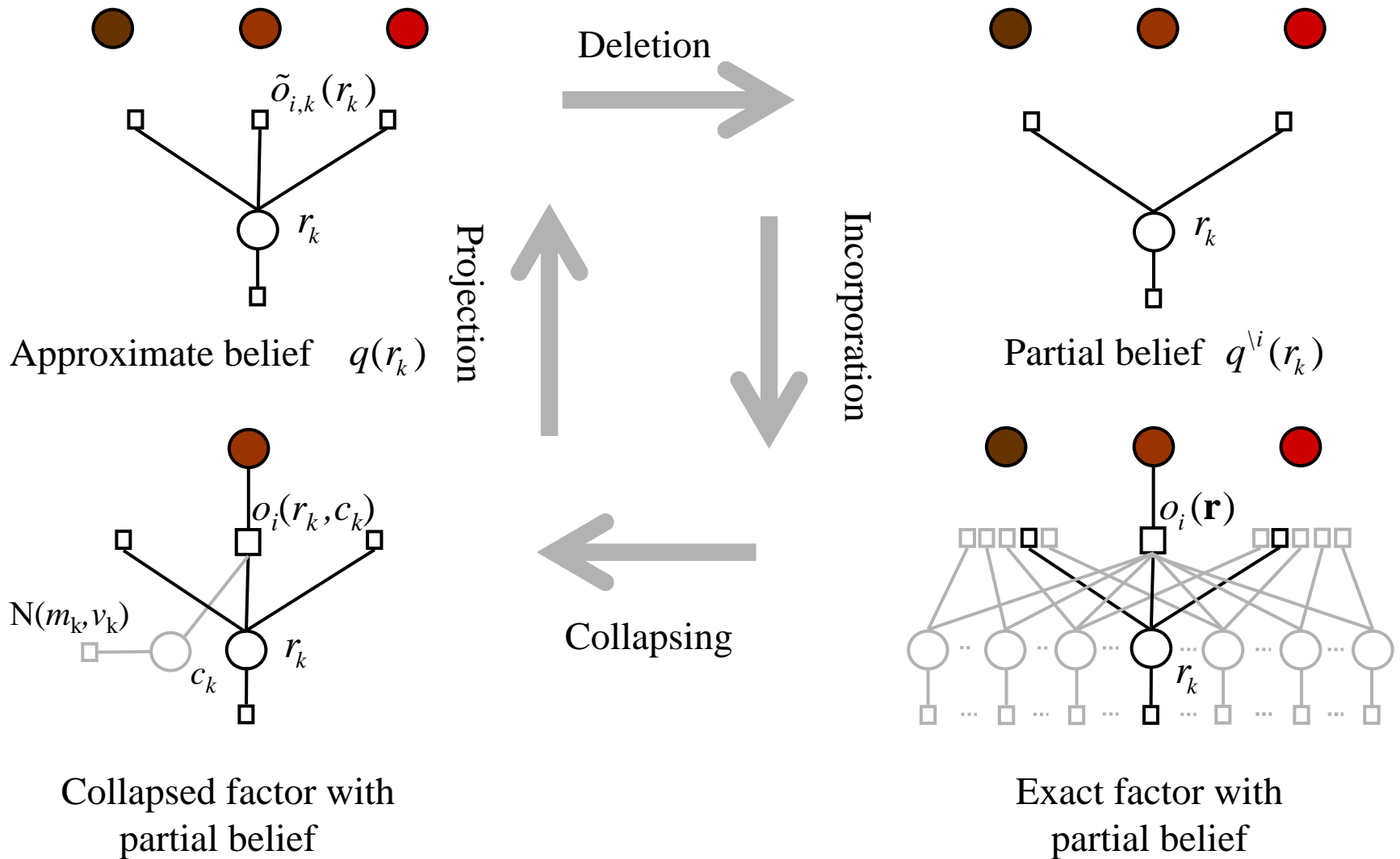
$$q(\mathbf{r}, \boldsymbol{\pi}) = q(\mathbf{r}) q(\boldsymbol{\pi}) = \{\prod_j q(r_j)\} \prod_j q(\pi_j)$$

$$q(r_j) \propto \prod_i \tilde{o}_{i,j}(r_j) \tilde{u}_j(r_j)$$

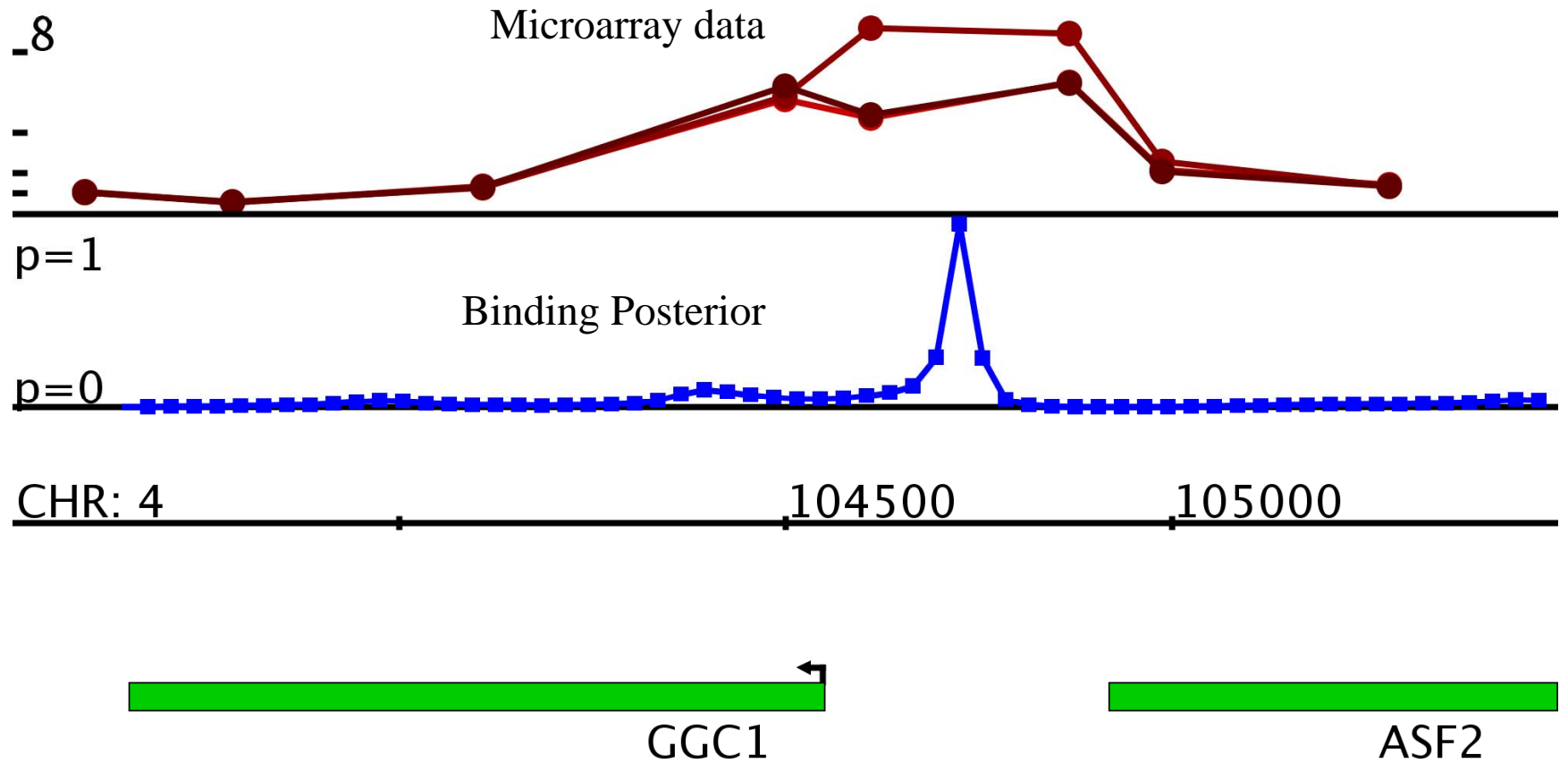
$$q(\pi_j) \propto \prod_i \tilde{u}_j(\pi_j) p(\pi_j)$$

(Qi et al. 2005)

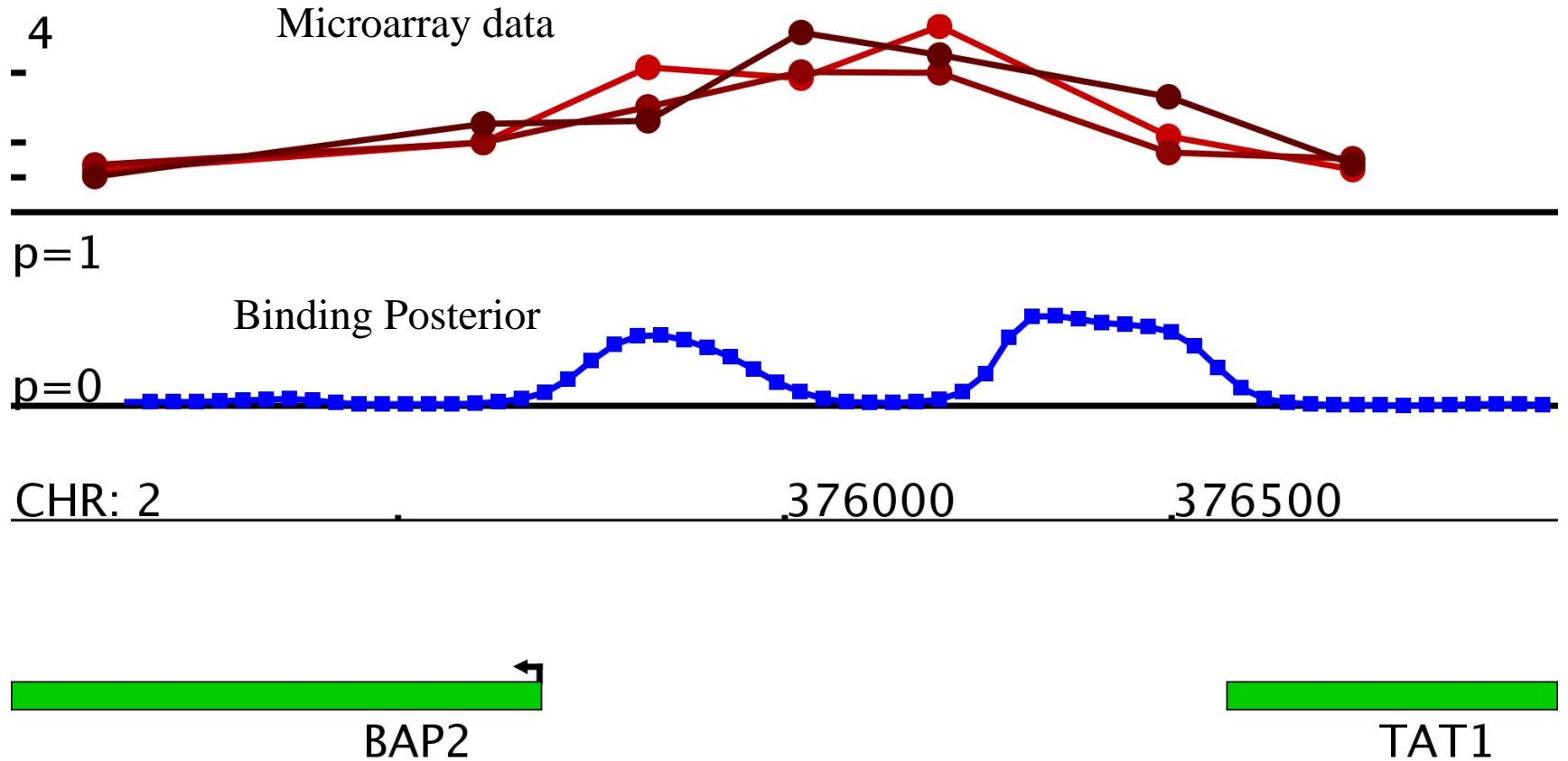
Update steps for processing one factor



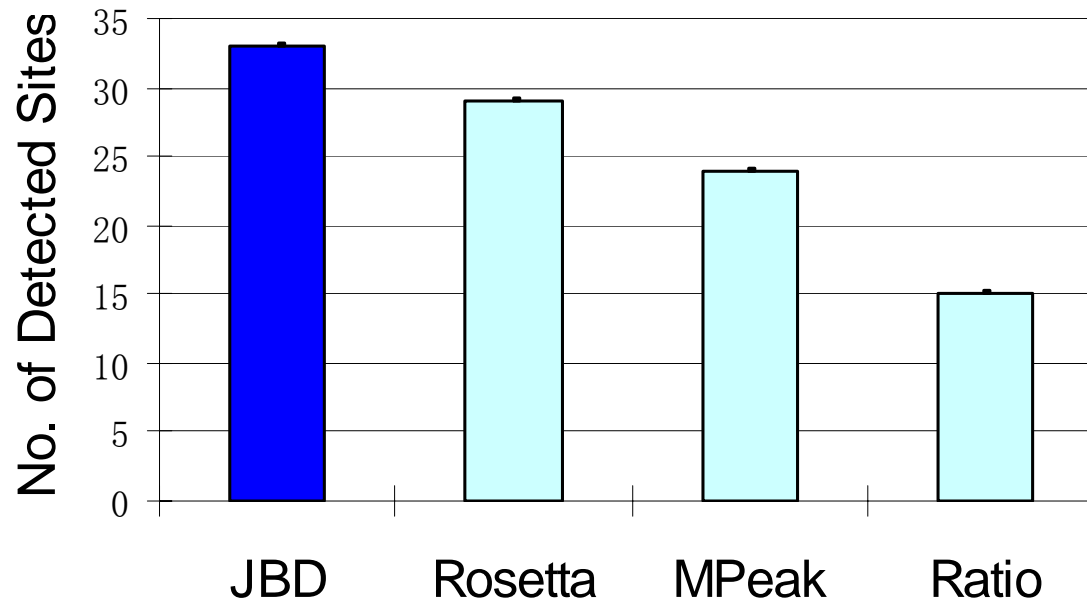
JBD finds bind event with high spatial resolution



JBD can discover two binding events under one peak



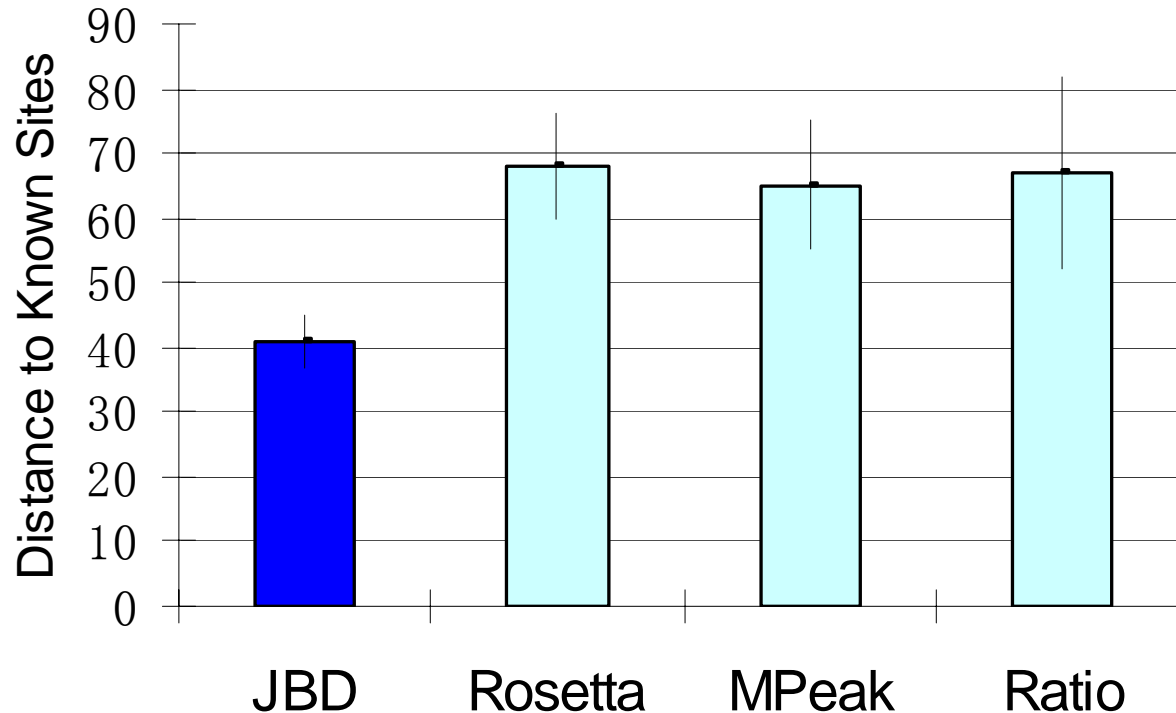
JBD finds more known binding events



Evaluated on GCN4 binding.

- JBD: Qi, Y et al., *Nature Biotech*, 2006
- MPEAK: Kim, T.H. et al., *Nature*, 2005
- Rosetta: Boyer, L.A. et al., *Cell* 2005
- Ratio: thresholding the enrichment ratio

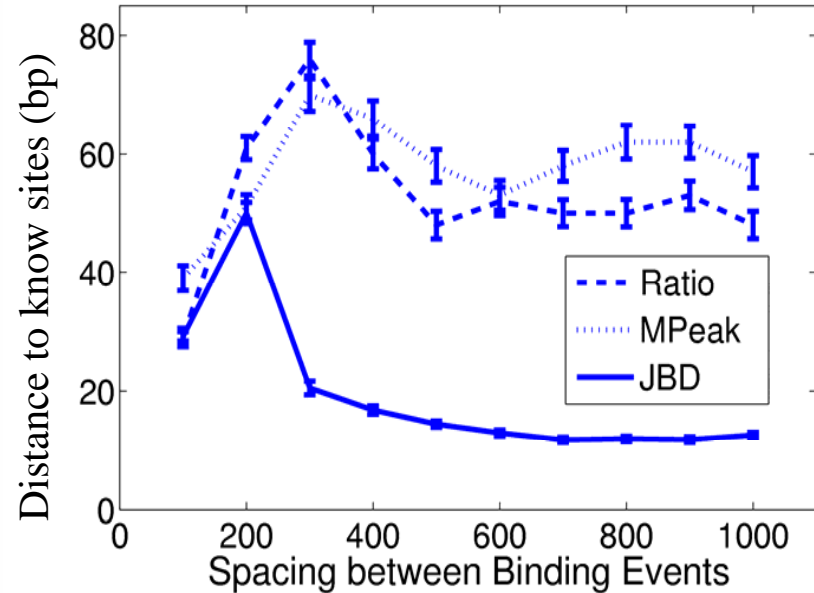
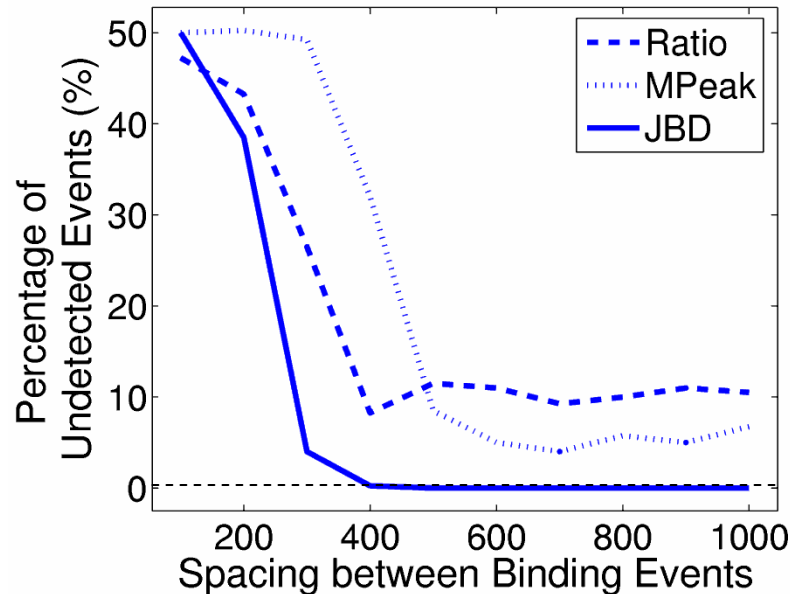
JBD achieves smaller distance to known sites



Evaluated on GCN4 binding.

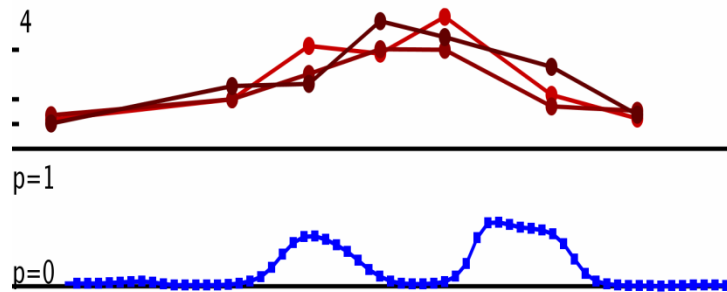
- JBD: Qi, Y et al., *Nature Biotech*, 2006
- MPEAK: Kim, T.H. et al., *Nature*, 2005
- Rosetta: Boyer, L.A. et al., *Cell* 2005
- Ratio

JBD better resolves neighboring binding events



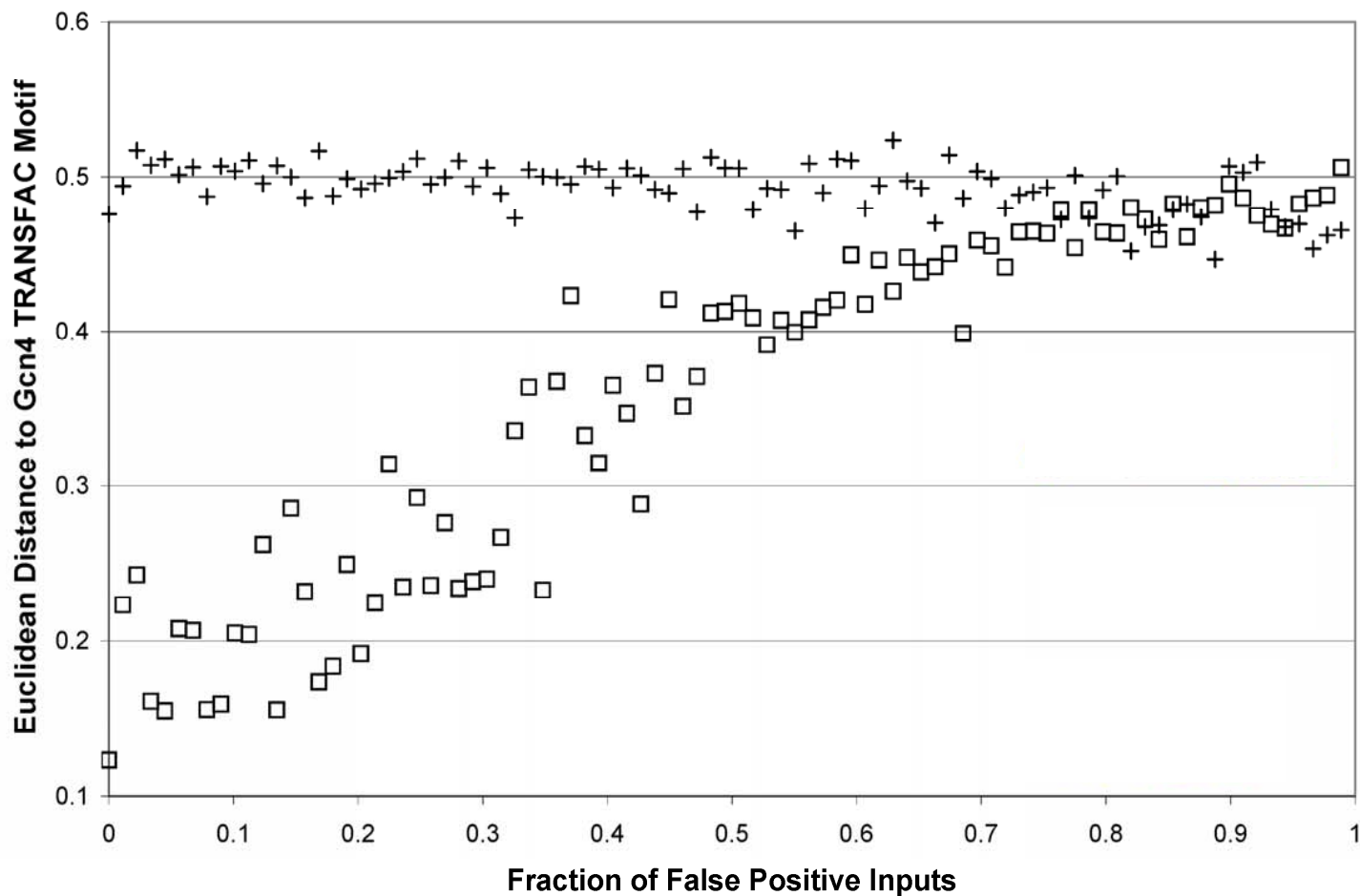
Comparing JBD, MPeak and Ratio methods on 200 simulated DNA regions, each containing two binding events.

Positional posterior derived from JBD led to the discovery of cryptic motif



- MEME + soft positional prior: found *Mig2* motif
- MEME (Bailey & Elkan 2003) or Gibbs sampling: failed to find *Mig2* motif
- MEME + Ratio: failed to find *Mig2* motif

Positional priors derived from JBD improve robustness of motif discovery to random input DNA sequences



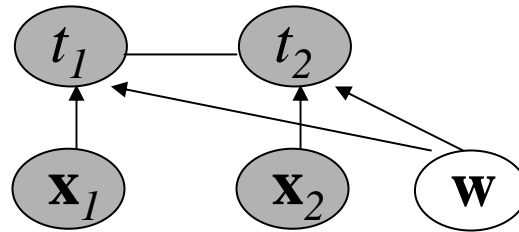
□ - Positional prior from JBD + MEME

+ - No positional prior (MEME: Bailey & Elkan 2003)

Outline: section 2

- Inference on graphical models
 - EP on hybrid dynamic networks for wireless signal detection
 - Tree-structured approximation for message passing on loopy graphs
 - Message approximation for detecting Protein-DNA binding sites
- Learning conditional graphical models
 - Bayesian conditional random fields for handwritten ink analysis and news group parsing

Conditional random fields (CRFs)



- Generalize traditional classification model by introducing correlation between labels.

Potential functions: $\Phi_{i,j}(t_i, t_j, \mathbf{x}; \mathbf{w})$

- Model data with independence and structure, e.g, web pages, natural languages, and multiple visual objects in a picture.
- Information at one location propagates to other locations.

Learning the parameter \mathbf{w} by ML/MAP

Maximum likelihood (ML) : Maximize the data likelihood

$$p(\mathbf{t}|\mathbf{x}, \mathbf{w}) = \frac{1}{Z(\mathbf{w})} \prod_{\{i,j\} \in \mathcal{E}} \Phi_{i,j}(t_i, t_j, \mathbf{x}; \mathbf{w})$$

where $Z(\mathbf{w}) = \sum_{\mathbf{t}} \prod_{\{i,j\} \in \mathcal{E}} \Phi_{i,j}(t_i, t_j, \mathbf{x}; \mathbf{w})$

Maximum a posterior (MAP): Gaussian prior on \mathbf{w}

$$\mathcal{N}(\mathbf{w}; \mathbf{0}, \text{diag}(\alpha))$$

ML/MAP problem: Overfitting to the noise in data.

Bayesian conditional random fields

- Bayesian training to avoid overfitting
- Need efficient training:
 - Exact posterior of \mathbf{w}

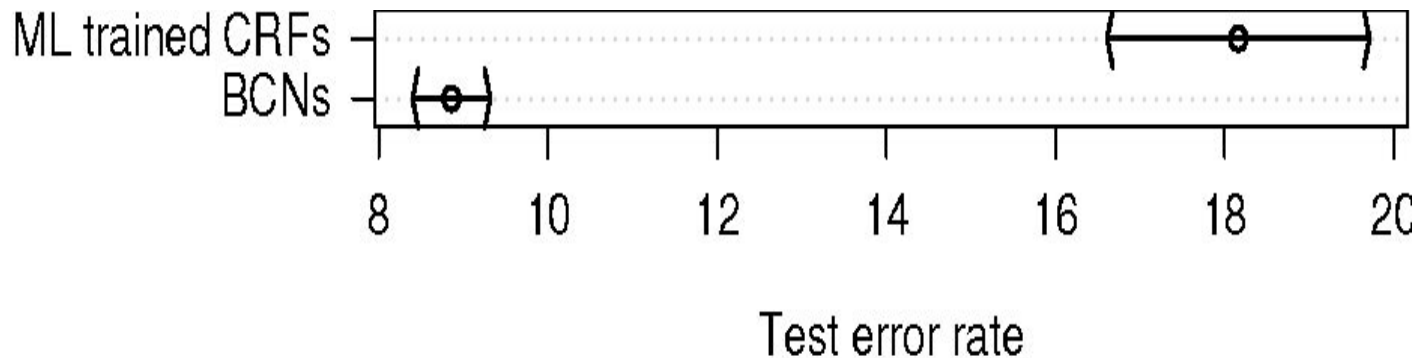
$$p(\mathbf{w}|\mathbf{t}, \mathbf{x}) \propto p(\mathbf{w})p(\mathbf{t}|\mathbf{w}, \mathbf{x}) = \mathcal{N}(\mathbf{w}; \mathbf{0}, \text{diag}(\boldsymbol{\alpha})) \frac{1}{Z(\mathbf{w})} \prod_{\{i,j\} \in \mathcal{E}} \Phi_{i,j}(t_i, t_j, \mathbf{x}; \mathbf{w})$$

- Gaussian approximate posterior by power-EP

$$q(\mathbf{w}) = \mathcal{N}(\mathbf{w}; \mathbf{0}, \text{diag}(\boldsymbol{\alpha})) \frac{1}{\tilde{Z}(\mathbf{w})} \prod_{\{i,j\} \in \mathcal{E}} \tilde{g}_{ij}(\mathbf{w})$$

Results on Synthetic Data

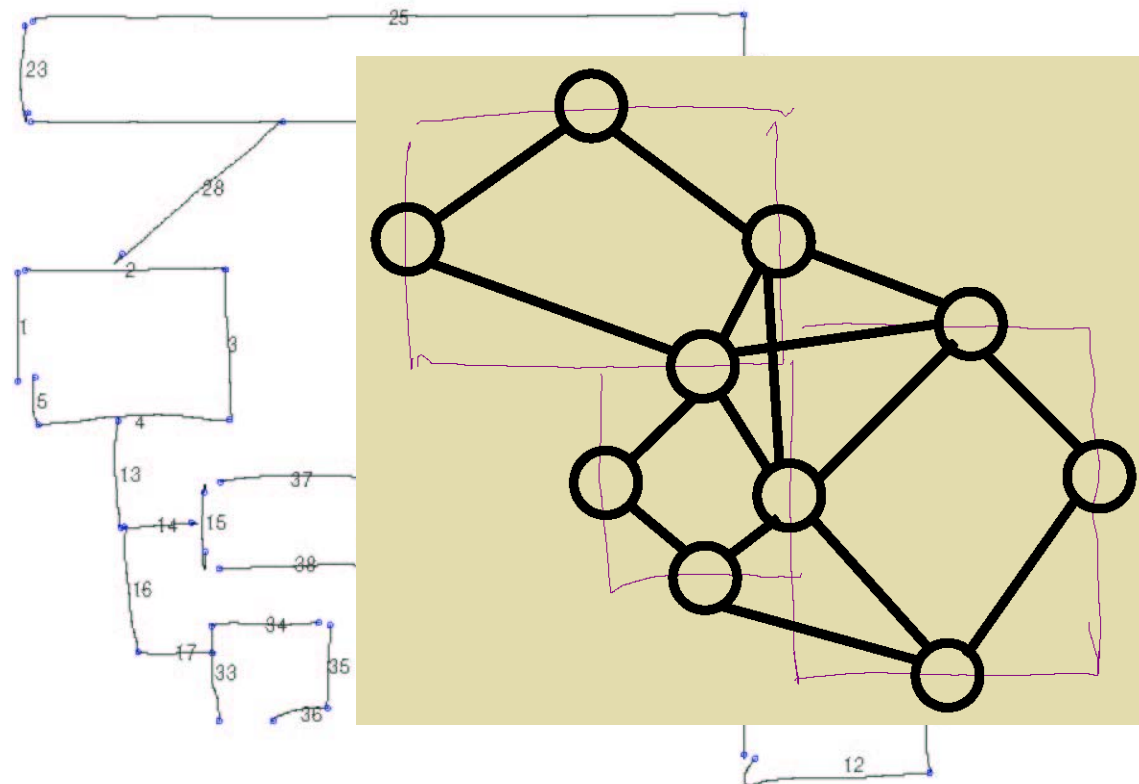
- Data generation: first, randomly sample input \mathbf{x} , fixed true parameters \mathbf{w} , and then sample the labels \mathbf{t}
- Graphical structure: Four nodes in a simple loop
- Comparing maximum likelihood trained CRF with Bayesian conditional networks: 10 Trials. 100 training examples and 1000 test examples.



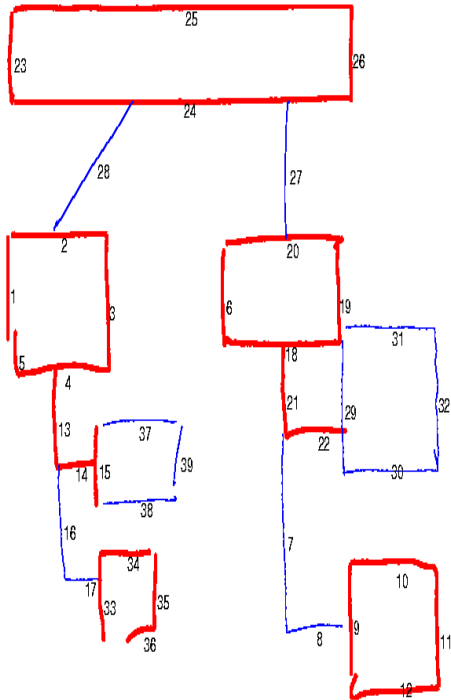
BCRFs significantly outperformed ML CRFs.

Analyzing handwritten organization charts

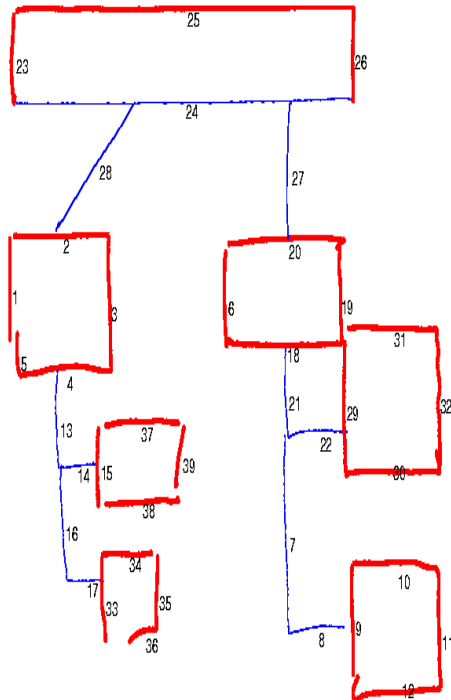
- Parsing a graph into different components: containers vs. connectors



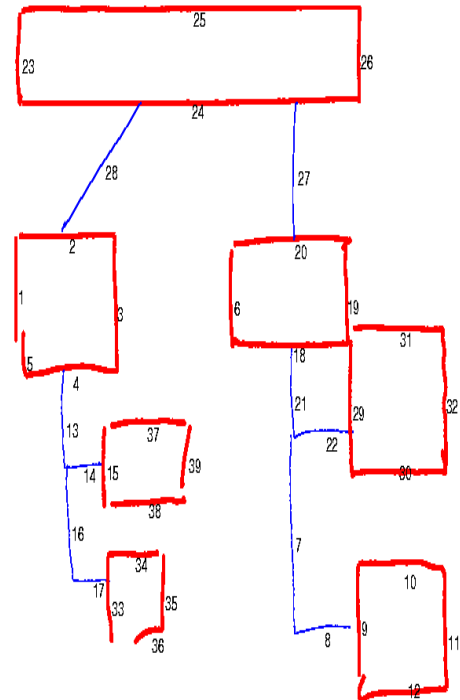
Comparing results



Results from Bayes
Point Machine



Results from MAP-
trained CRF



Results
from BCRF

Results

Algorithm	Test error rates
ML-Probit-CRF	10.7 \pm 1.21 \star
MAP-Probit-CRF	6.00 \pm 0.64 \star
ML-Exp-CRF	10.1 \pm 1.21 \star
MAP-Exp-CRF	5.20 \pm 0.79 \diamond
BCRF	4.39 \pm 0.62
BCRF-ARD	3.98 \pm 0.48

BCRF outperforms ML and MAP trained-CRFs. BCRF-ARD further improves test accuracy. The results are averaged over 20 runs.

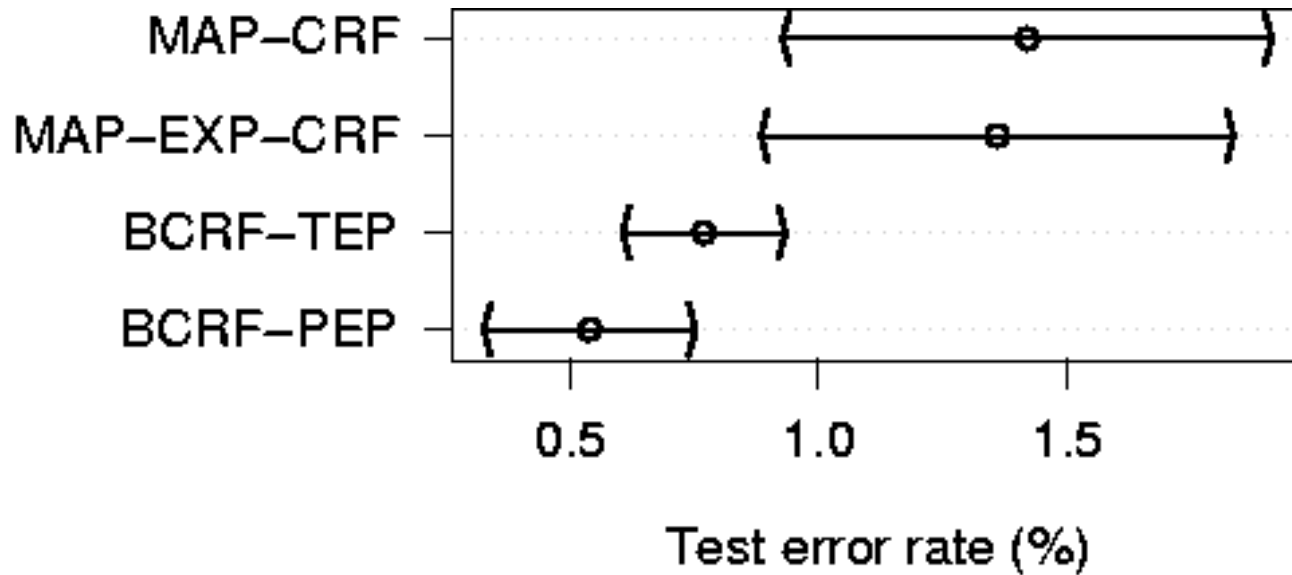
FAQs Labeling

- The dataset consists of 47 files, belonging to 7 Usenet newsgroup FAQs. Each file has multiple lines, which can be the header (H), a question (Q), an answer (A), or the tail (T).
- Task: label the lines that are questions or answers.

FAQs Features

begins-with-number	contains-question-mark
begins-with-ordinal	contains-question-word
begins-with-punctuation	ends-with-question-mark
begins-with-question-word	first-alpha-is-capitalized
begins-with-subject	indented
blank	indented-1-to-4
contains-alphanum	indented-5-to-10
contains-bracketed-number	more-than-one-third-space
contains-http	only-punctuation
contains-non-space	prev-is-blank
contains-number	prev-begins-with-ordinal
contains-pipe	shorter-than-30

Results



BCRFs outperform MAP-trained CRFs with a high statistical significance on FAQs labeling.

Summary

- Many random and deterministic (approximate) Bayesian inference methods for graphical models:
 - MCMC, Particle filters, VB, BP, EP, PEP, ...
- Ideas from statistics, optimization, and physics
- Applications in wireless communication, computational biology, computer vision, information extraction, etc.