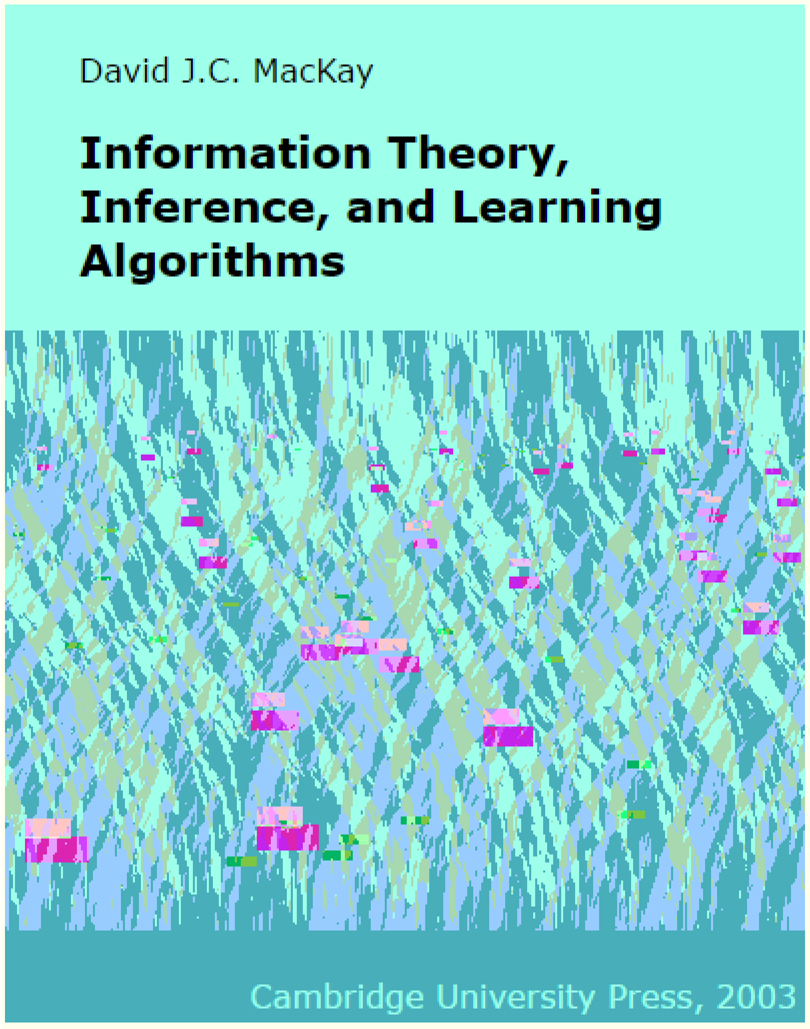
ECE521 Lecture 22
3 April 2017

The max-sum algorithm

Mark Ebden

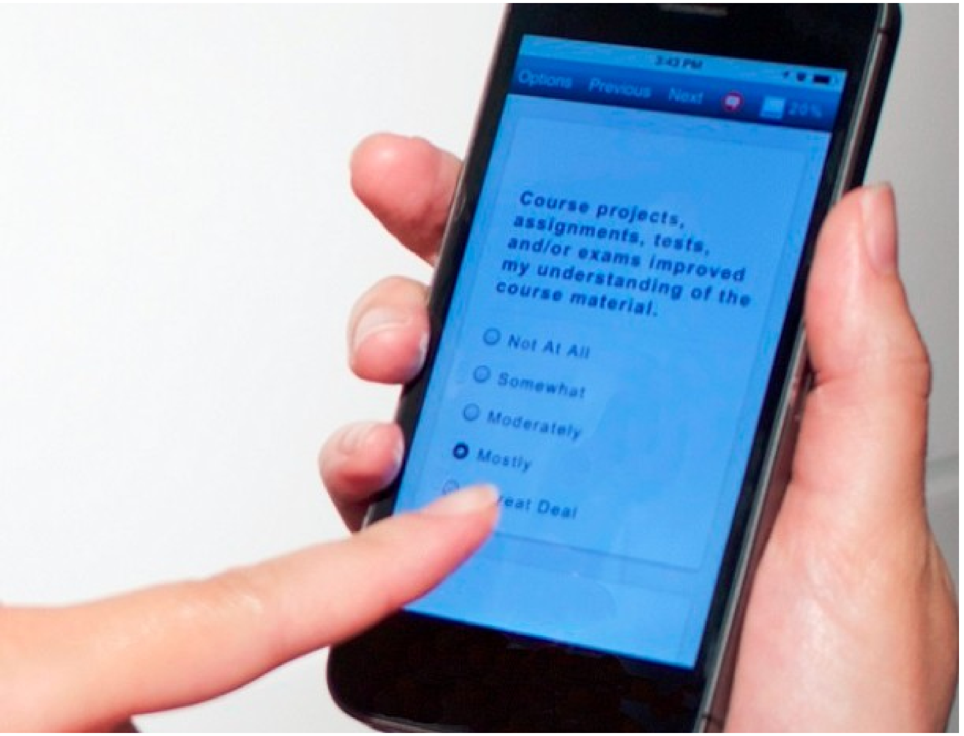
Examples of optional readings

- Murphy 17.4.4 & 20.2
- Bishop 8.4.5 & 13.2.5
- MacKay 26.3

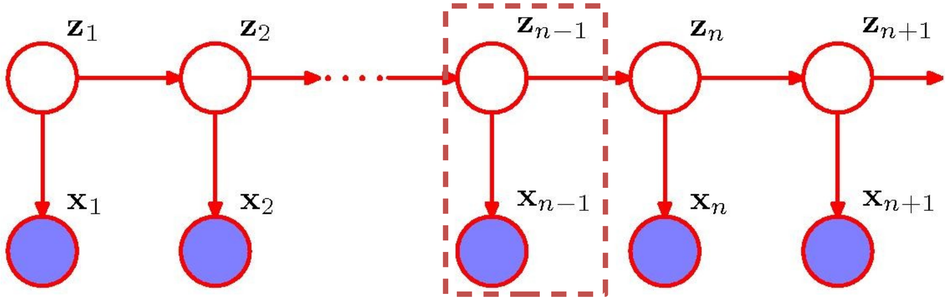


Outline

1. HMMs continued
2. Graphical models continued:
   - Max-sum algorithm
   - A few points moving forward
3. Course evaluations (15 min)



1. HMMs continued



Three problems and three solutions:

1. Computing probabilities of observed sequences: *Forward-backward algorithm*
2. Learning of parameters: *Baum-Welch algorithm*
3. Inference of hidden state sequences: *Viterbi algorithm* ←

(Notes from Lecture 20)

2. Graphical models cont'd from Lecture 21

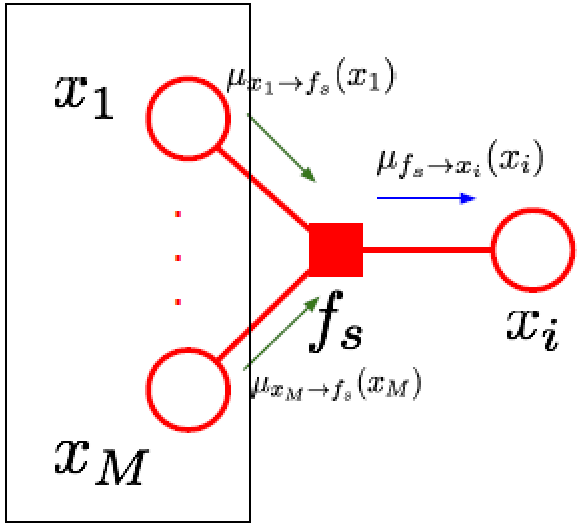
Recall the sum-product algorithm's messages:

Factor-to-variable messages:

$$\mu_{f_s \to x_i}(x_i) = \sum_{Ne(f_s) \setminus x_i} \left[f_s(x_i, x_1, \ldots, x_M) \prod_{x_m \in Ne(f_s) \setminus x_i} \mu_{x_m \to f_s}(x_m) \right]$$

Variable-to-factor messages:

$$\mu_{x_i \to f_s}(x_i) = \prod_{f_n \in Ne(x_i) \setminus f_s} \mu_{f_n \to x_i}(x_i)$$



From sum-product to max-product

The sum-product algorithm computes probabilities for a **subset** of the variables of a **factor graph** , e.g. P(a, b, c, d)

- Marginal distributions , e.g. P(b)
- Joint distributions of a subset of variables , e.g. P(a,b)
- Conditional distributions (often the posterior distributions of our interest) , e.g. P(a,c | d) = P(a,c,d) / P(d)

Whereas, the goal of the *max-product algorithm* is to find

$$p(\mathbf{x}^{\max}) = \max_{\mathbf{x}} p(\mathbf{x})$$ and especially $$\mathbf{x}^{\max} = \arg\max_{\mathbf{x}} p(\mathbf{x})$$

From sum-product to max-product

Going from the forward-backward algorithm to the Viterbi algorithm was a matter of **replacing summations with maxes**.

This is what happens going from sum-product to max-product.

Baum-Welch = HMM EM
Viterbi = HMM max-product (akin to max-sum...)
Forward-backward = HMM sum-product

From max-product to max-sum

- It's often convenient to work with the logarithm of the joint distribution
- It's very easy to introduce this in our max-product work, because the max operator and logarithm function can be interchanged:

$$\ln\left(\max_{\mathbf{x}} p(\mathbf{x})\right) = \max_{\mathbf{x}} \ln p(\mathbf{x})$$

- Some authors use the terms 'max-product' and 'max-sum' almost interchangeably because the only difference is taking the logarithm
- Replacing maxes with mins gives the *min-sum* algorithm

Max-sum algorithm

The new messages are:

$$\mu_{f\to x}(x) = \max_{x_1,\ldots,x_M}\left[\ln f(x,x_1,\ldots,x_M) + \sum_{m\in\mathrm{ne}(f_s)\setminus x} \mu_{x_m\to f}(x_m)\right]$$

$$\mu_{x\to f}(x) = \sum_{l\in\mathrm{ne}(x)\setminus f} \mu_{f_l\to x}(x).$$

When finished:

$$p^{\max} = \max_x \left[\sum_{s\in\mathrm{ne}(x)} \mu_{f_s\to x}(x)\right]$$

and

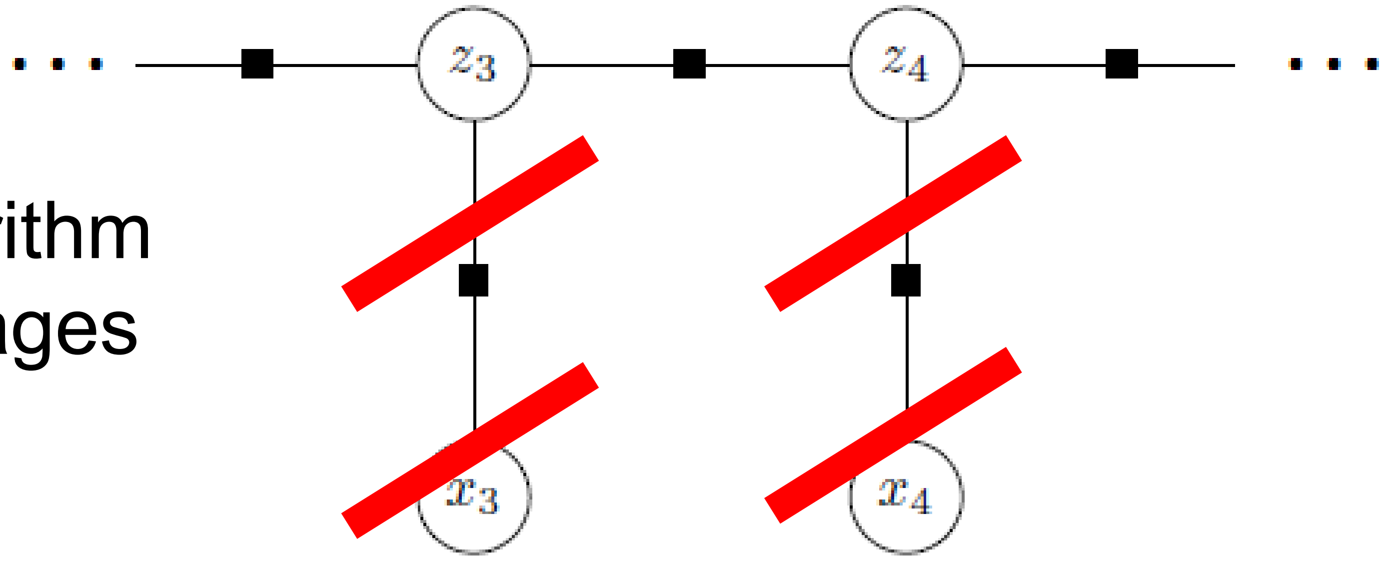
$$x^{\max} = \arg\max_x \left[\sum_{s\in\mathrm{ne}(x)} \mu_{f_s\to x}(x)\right]$$

The Viterbi algorithm

The max-sum messages in an HMM:

$$\mu_{\mathbf{z}_n \to f_{n+1}}(\mathbf{z}_n) = \mu_{f_n \to \mathbf{z}_n}(\mathbf{z}_n)$$

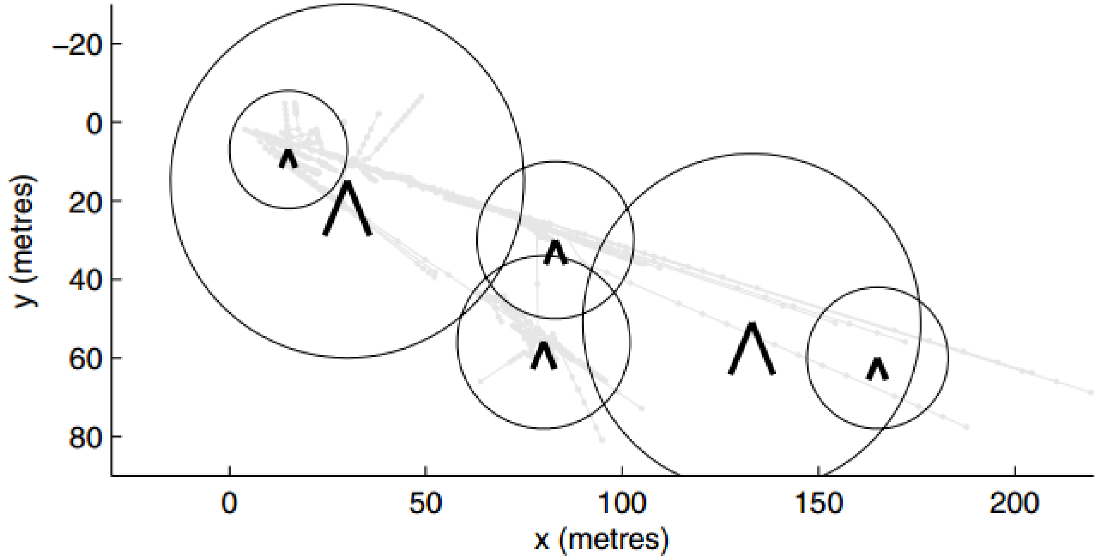
$$\mu_{f_{n+1} \to \mathbf{z}_{n+1}}(\mathbf{z}_{n+1}) = \max_{\mathbf{z}_n} \left\{ \ln f_{n+1}(\mathbf{z}_n, \mathbf{z}_{n+1}) + \mu_{\mathbf{z}_n \to f_{n+1}}(\mathbf{z}_n) \right\}$$

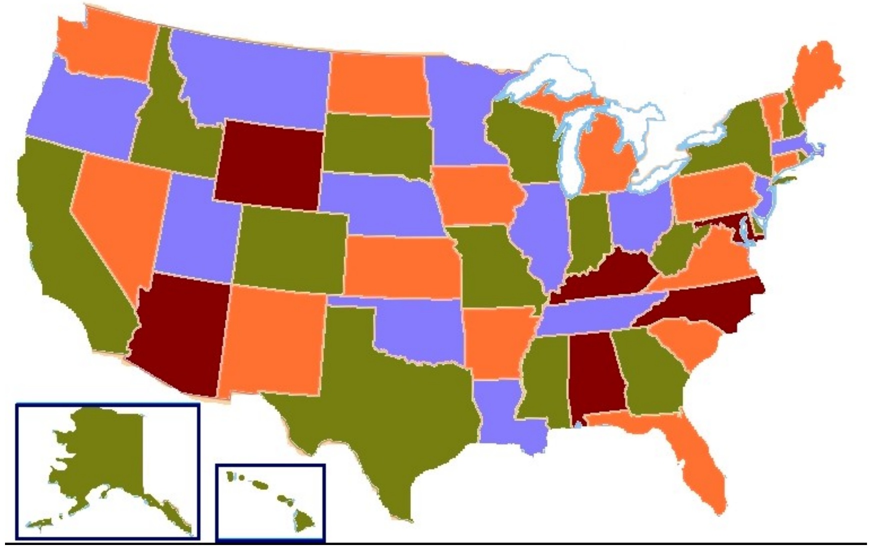


As with the sum-product algorithm from last week, run the messages up and down the factor graph once, and you are finished.

Handling ties in max-sum

Is one moving object
more worthwhile to observe
than another?



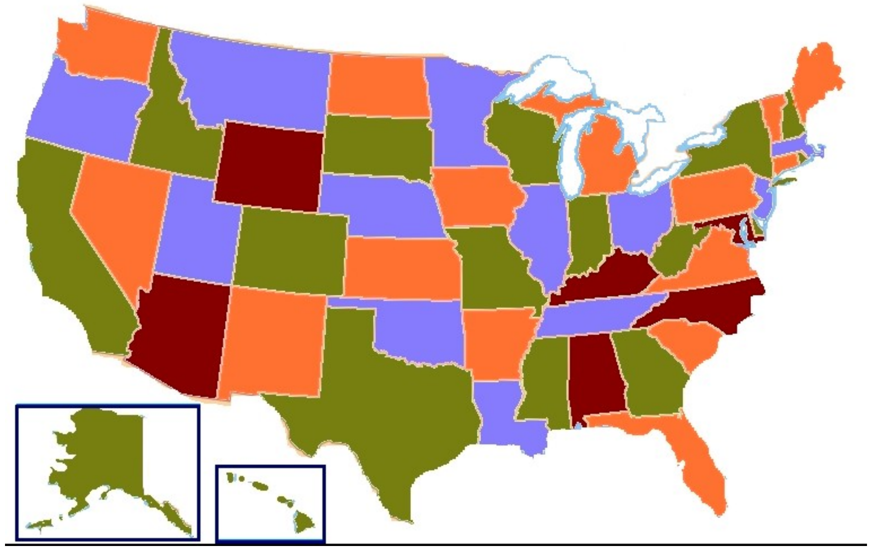


Is one colour better than another?

Handling ties in max-sum

Introduce preferences γ_{mj} many orders of magnitude lower than the function of interest, via a random number generator.

The *m*th variable node will have preference γ_{mj} for the *j*th state.



Comparing algorithms: max-sum vs Ising model

In Lecture 17, to solve the Ising model we had used something called *iterated conditional modes* (ICM), consisting of a very primitive form of message: the new state of a node.

Whereas, max-sum is more communicative: “If you select state *x*, then the highest score for me and others is...”

ICM isn’t guaranteed to find a global maxima on trees.

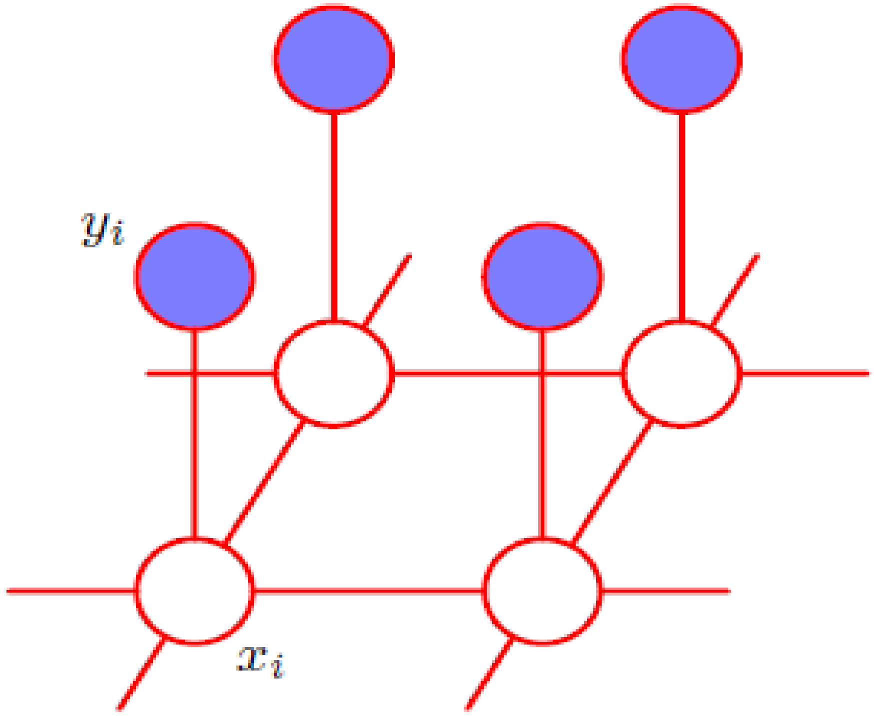
Recall the Ising model

- x_i : original image
- y_i : noisy image

$$p(\mathbf{x}, \mathbf{y}) = \frac{1}{Z} \exp\{-E(\mathbf{x}, \mathbf{y})\}$$

where

$$E(\mathbf{x}, \mathbf{y}) = h \sum_i x_i - \beta \sum_{\{i,j\}} x_i x_j - \eta \sum_i x_i y_i$$



Energy function

$$E(\mathbf{x}, \mathbf{y}) = h \sum_i x_i - \beta \sum_{\{i,j\}} x_i x_j - \eta \sum_i x_i y_i$$

bias   neighbours   observed

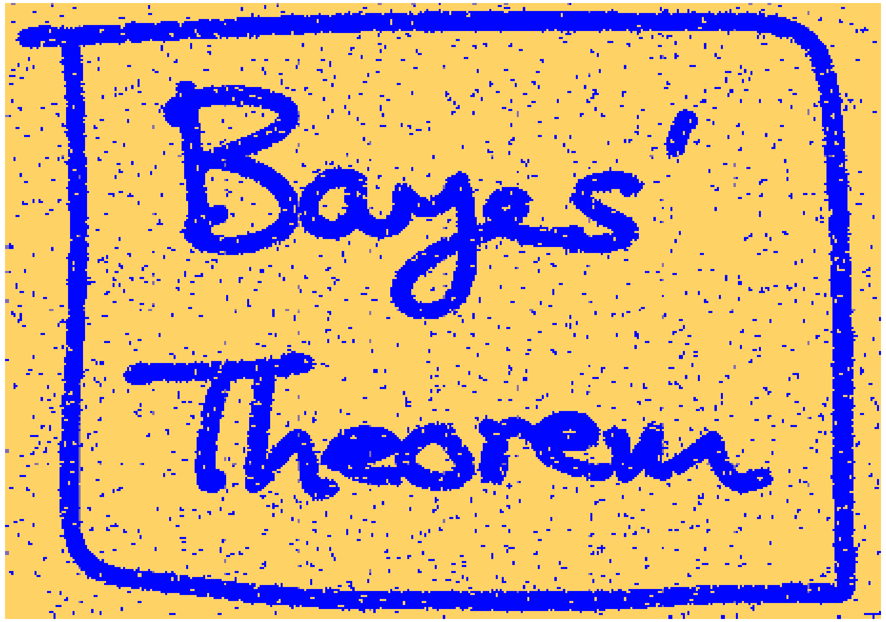
- The relative values of h, β, and η control these three effects
- What are the maximal cliques in an Ising model?

Solving the Ising model

- Select $\beta = 1.0, \eta = 2.1$ and $h = 0$
- Initialize **x** to **y**
- Until convergence:

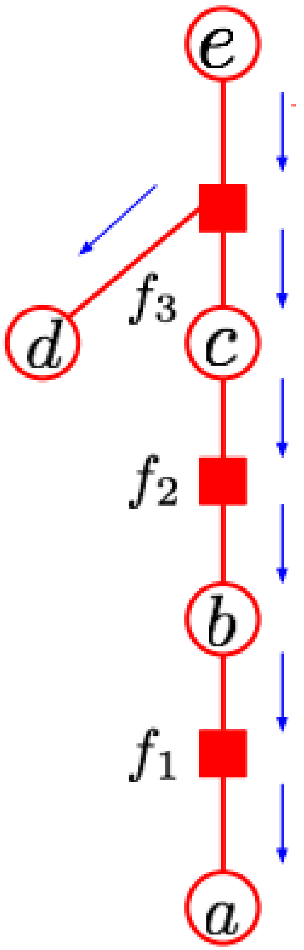
  for each x_i :

  $x_i \leftarrow \text{argmin } E(x_i, y_i)$



Tutorial 10 and max-product

Example of the sum product algorithm



$$\mu_{a\to f_1}(a) = [1,1]^T$$
$$\mu_{d\to f_3}(d) = [1,1]^T$$
$$\mu_{f_1\to b}(b) = [3,3]^T$$
$$\mu_{b\to f2}(b) = [3,3]^T$$
$$\mu_{f_2\to c}(c) = [15,4.5]^T$$
$$\mu_{c\to f_3}(c) = [15,4.5]^T$$
$$\mu_{f_3\to e}(e) = [58.5,58.5]^T$$
$$\mu_{e\to f_3}(e) = [1,1]^T$$
$$\mu_{f_3\to d}(e) = [58.5,58.5]^T$$
$$\mu_{f_3\to c}(c) = [6,6]^T$$
$$\mu_{c\to f_2}(c) = [6,6]^T$$
$$\mu_{f_2\to b}(b) = [24,15]^T$$
$$\mu_{b\to f_1}(b) = [24,15]^T$$
$$\mu_{f_1\to a}(a) = [39,78]^T$$

$$f_1(a,b) = \begin{bmatrix} f_1(0,0), f_1(1,0) \\ f_1(0,1), f_1(1,1) \end{bmatrix} = \begin{bmatrix} 1,2 \\ 1,2 \end{bmatrix}$$

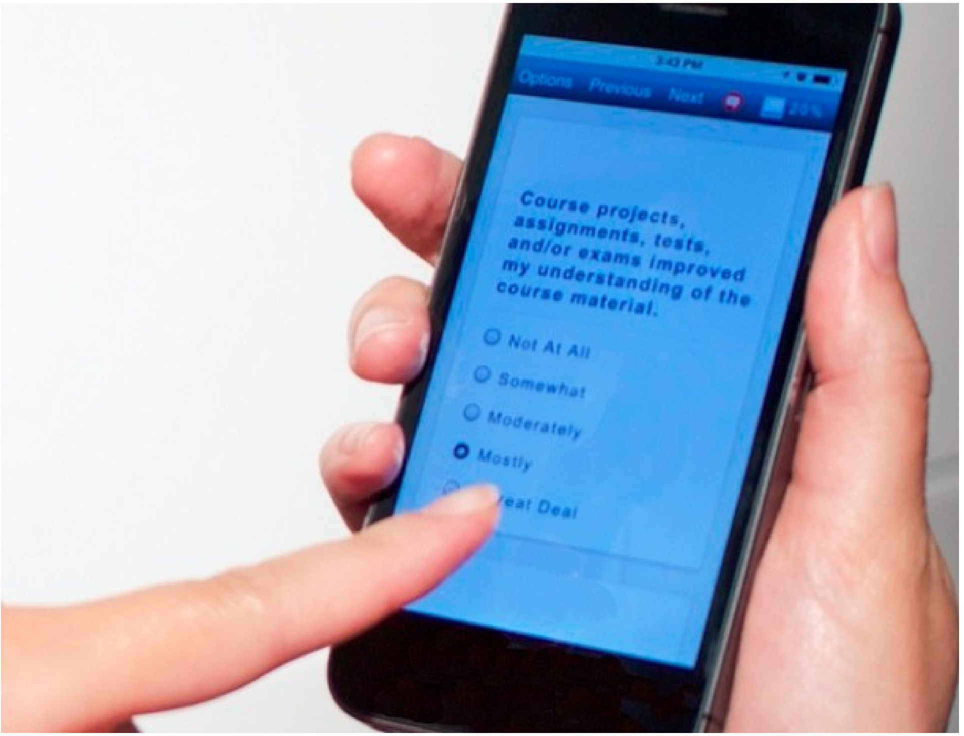
$$f_2(b,c) = \begin{bmatrix} 3,2 \\ 1,0.5 \end{bmatrix}$$

$$f_3(c=0,d,e) = \begin{bmatrix} 1,2 \\ 2,1 \end{bmatrix}$$

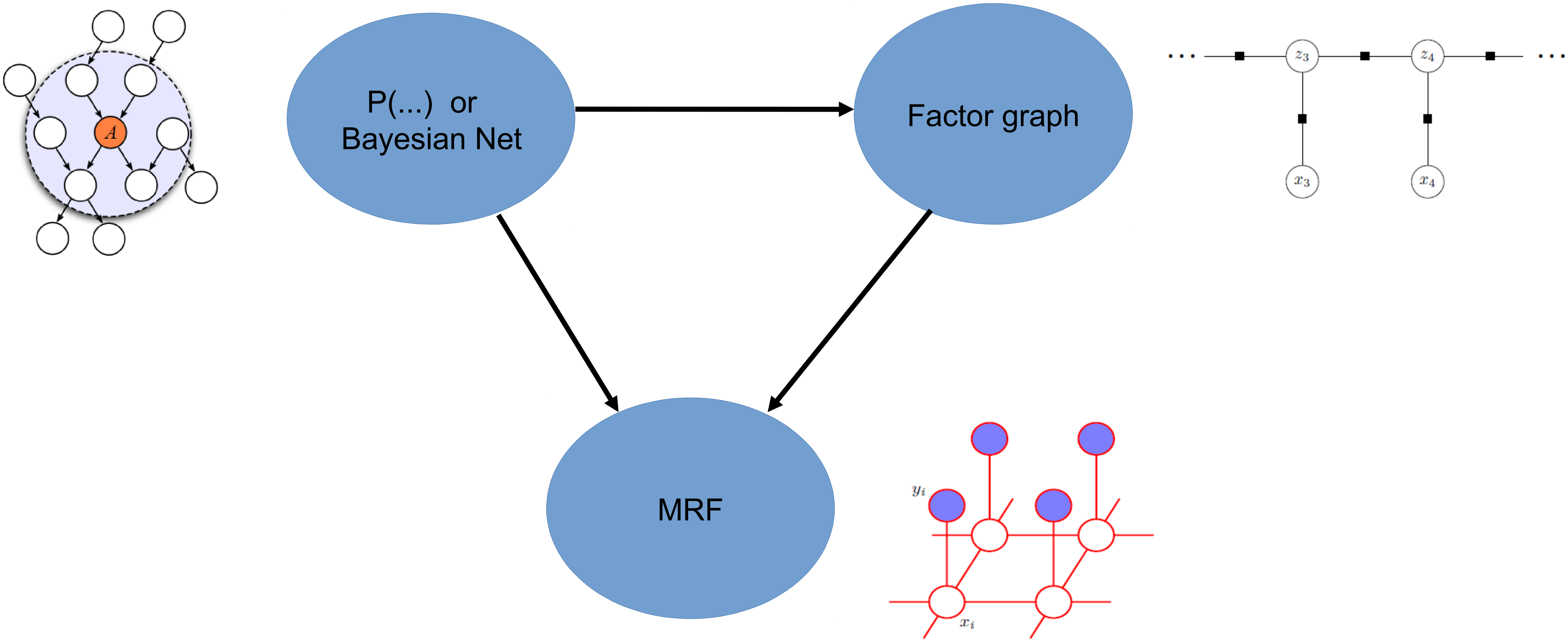
$$f_3(c=1,d,e) = \begin{bmatrix} 1,2 \\ 2,1 \end{bmatrix}$$

Outline

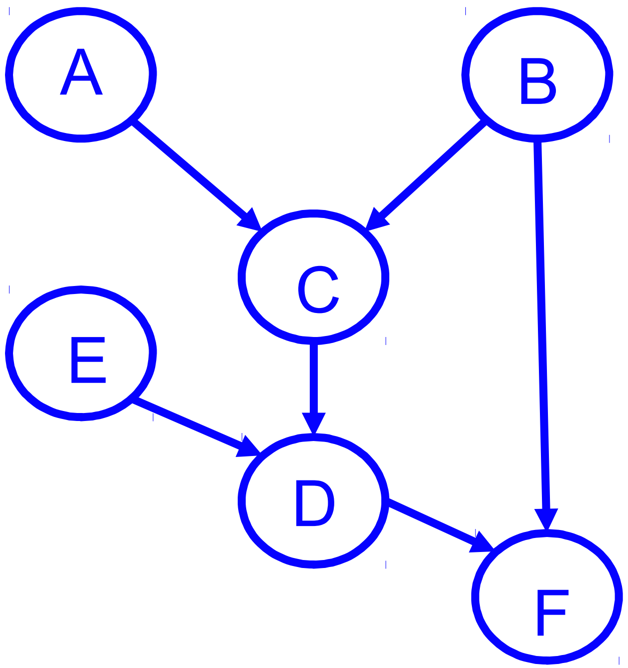
1. HMMs continued
2. Graphical models continued:
   - Max-sum algorithm
   - A few points moving forward
3. Course evaluations (15 min)

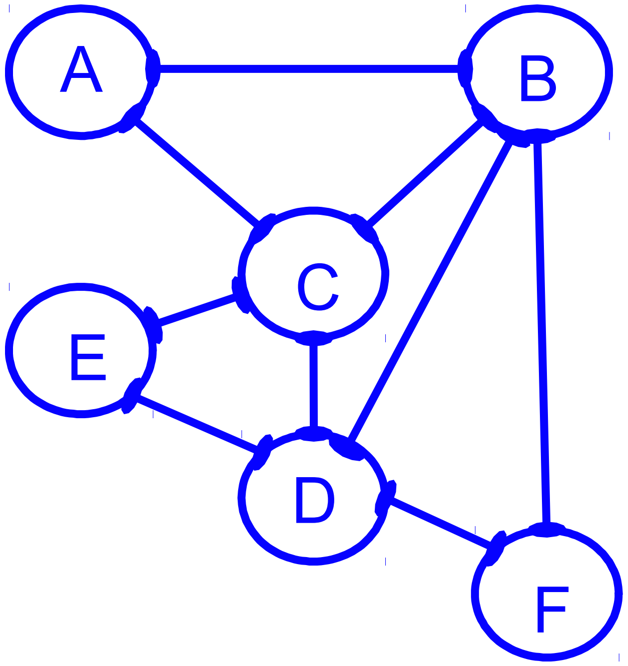


Converting among types of graphical models



Converting BNs to MRFs





All co-parents must be married (nodes with a child in common must be connected to one another)

Two protocols for message passing

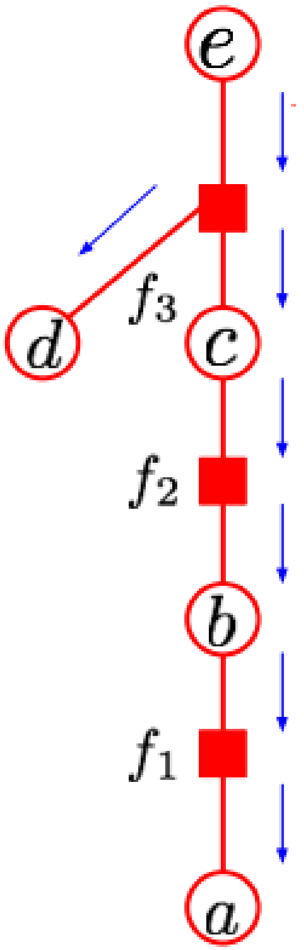
- Serial (our topic so far):
  - Messages travel from the leaves to the root and back

- Parallel (new topic):
  1. Initialize all messages to a vector of all 1’s
  2. In parallel, each node computes its message
  3. Repeating #2 for d iterations leads to convergence

     (d is the diameter of the graph, i.e. the maximum distance between any two nodes)

Tutorial 10: the serial protocol

Example of the sum product algorithm



$\mu_{a\to f_1}(a) = [1,1]^T$

$\mu_{d\to f_3}(d) = [1,1]^T$

$\mu_{f_1\to b}(b) = [3,3]^T$

$\mu_{b\to f2}(b) = [3,3]^T$

$\mu_{f_2\to c}(c) = [15,4.5]^T$

$\mu_{c\to f_3}(c) = [15,4.5]^T$

$\mu_{f_3\to e}(e) = [58.5,58.5]^T$

$\mu_{e\to f_3}(e) = [1,1]^T$

$\mu_{f_3\to d}(e) = [58.5,58.5]^T$

$\mu_{f_3\to c}(c) = [6,6]^T$

$\mu_{c\to f_2}(c) = [6,6]^T$

$\mu_{f_2\to b}(b) = [24,15]^T$

$\mu_{b\to f_1}(b) = [24,15]^T$

$\mu_{f_1\to a}(a) = [39,78]^T$

$$f_1(a,b) = \begin{bmatrix} f_1(0,0), f_1(1,0) \\ f_1(0,1), f_1(1,1) \end{bmatrix} = \begin{bmatrix} 1,2 \\ 1,2 \end{bmatrix}$$

$$f_2(b,c) = \begin{bmatrix} 3,2 \\ 1,0.5 \end{bmatrix}$$

$$f_3(c=0,d,e) = \begin{bmatrix} 1,2 \\ 2,1 \end{bmatrix}$$

$$f_3(c=1,d,e) = \begin{bmatrix} 1,2 \\ 2,1 \end{bmatrix}$$

Example of the parallel protocol

- Homework: try it for a small graph and confirm it matches

Outline

1. HMMs continued
2. Graphical models continued:
   - Max-sum algorithm
   - A few points moving forward
3. Course evaluations (15 min)