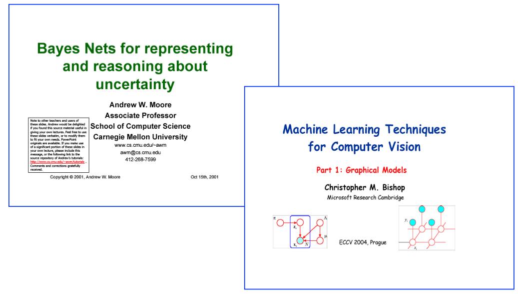
Robert Collins
CSE586

Introduction to Graphical Models

Readings in Prince textbook: Chapters 10 and 11
but mainly only on directed graphs at this time

Credits: Several slides are from:



Review: Probability Theory

- Sum rule (marginal distributions)

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

- Product rule

$$p(\mathbf{x}, \mathbf{y}) = p(\mathbf{x}|\mathbf{y})p(\mathbf{y})$$

- From these we have Bayes' theorem

$$p(\mathbf{y}|\mathbf{x}) = \frac{p(\mathbf{x}|\mathbf{y})p(\mathbf{y})}{p(\mathbf{x})}$$

  – with normalization factor

$$p(\mathbf{x}) = \sum_{\mathbf{y}} p(\mathbf{x}|\mathbf{y})p(\mathbf{y})$$

Christopher Bishop, MSR

Review: Conditional Probabilty

- Conditional Probability (rewriting product rule)

  P(A | B) = P (A, B) / P(B)

- Chain Rule

$$P(A,B,C,D) = P(A)\ \frac{P(A,B)}{P(A)}\ \frac{P(A,B,C)}{P(A,B)}\ \frac{P(A,B,C,D)}{P(A,B,C)}$$

$$= P(A)\ \ P(B \mid A)\ \ P(C \mid A,B)\ \ P(D \mid A,B,C)$$

- Conditional Independence

  P(A, B | C) = P(A | C) P(B | C)

  – statistical independence

  P(A, B) = P(A) P(B)

Overview of Graphical Models

- Graphical Models model conditional dependence/ independence
- Graph structure specifies how joint probability factors
- Directed graphs

$$p(x_1, \ldots, x_D) = \prod_{i=1}^{D} p(x_i|\mathrm{pa}_i)$$ Example:HMM

- Undirected graphs

$$p(\mathbf{x}) = \frac{1}{Z}\prod_C \psi_C(\mathbf{x}_C)$$ Example:MRF

- Inference by message passing: belief propagation
  – Sum-product algorithm
  – Max-product (Min-sum if using logs)

We will focus mainly on directed graphs right now.

The Joint Distribution

Example: Boolean variables A, B, C

Recipe for making a joint distribution of M variables:

Andrew Moore, CMU

The Joint Distribution

Example: Boolean variables A, B, C

Recipe for making a joint distribution of M variables:

1. Make a truth table listing all combinations of values of your variables (if there are M Boolean variables then the table will have 2^M rows).

A	B	C
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

The Joint Distribution

Example: Boolean variables A, B, C

Recipe for making a joint distribution of M variables:

1. Make a truth table listing all combinations of values of your variables (if there are M Boolean variables then the table will have 2^M rows).
2. For each combination of values, say how probable it is.

A	B	C	Prob
0	0	0	0.30
0	0	1	0.05
0	1	0	0.10
0	1	1	0.05
1	0	0	0.05
1	0	1	0.10
1	1	0	0.25
1	1	1	0.10

The Joint Distribution

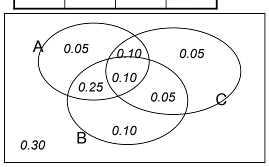
Example: Boolean variables A, B, C

Recipe for making a joint distribution of M variables:

1. Make a **truth table** listing all combinations of values of your variables (if there are M Boolean variables then the table will have 2^M rows).
2. For each combination of values, say how probable it is.
3. If you subscribe to the axioms of probability, those numbers must sum to 1.

truth table

A	B	C	Prob
0	0	0	0.30
0	0	1	0.05
0	1	0	0.10
0	1	1	0.05
1	0	0	0.05
1	0	1	0.10
1	1	0	0.25
1	1	1	0.10



Joint distributions

- Good news

  Once you have a joint distribution, you can answer all sorts of probabilistic questions involving combinations of attributes

Using the Joint

gender	hours_worked	wealth	
Female	v0:40.5-	poor	0.253122
		rich	0.0245895
	v1:40.5+	poor	0.0421768
		rich	0.0116293
Male	v0:40.5-	poor	0.331313
		rich	0.0971295
	v1:40.5+	poor	0.134106
		rich	0.105933

P(Poor Male) = 0.4654

$$P(E) = \sum_{\text{rows matching } E} P(\text{row})$$

Using the Joint

gender	hours_worked	wealth	
Female	v0:40.5-	poor	0.253122
		rich	0.0245895
	v1:40.5+	poor	0.0421768
		rich	0.0116293
Male	v0:40.5-	poor	0.331313
		rich	0.0971295
	v1:40.5+	poor	0.134106
		rich	0.105933

P(Poor) = 0.7604

$$P(E) = \sum_{\text{rows matching } E} P(\text{row})$$

Inference with the Joint

computing conditional probabilities

gender	hours_worked	wealth	
Female	v0:40.5-	poor	0.253122
		rich	0.0245895
	v1:40.5+	poor	0.0421768
		rich	0.0116293
Male	v0:40.5-	poor	0.331313
		rich	0.0971295
	v1:40.5+	poor	0.134106
		rich	0.105933

$$P(E_1 \mid E_2) = \frac{P(E_1 \wedge E_2)}{P(E_2)} = \frac{\sum_{\text{rows matching } E_1 \text{ and } E_2} P(\text{row})}{\sum_{\text{rows matching } E_2} P(\text{row})}$$

Andrew Moore, CMU

Inference with the Joint

gender	hours_worked	wealth	
Female	v0:40.5-	poor	0.253122
		rich	0.0245895
	v1:40.5+	poor	0.0421768
		rich	0.0116293
Male	v0:40.5-	poor	0.331313
		rich	0.0971295
	v1:40.5+	poor	0.134106
		rich	0.105933

$$P(E_1 \mid E_2) = \frac{P(E_1 \wedge E_2)}{P(E_2)} = \frac{\sum_{\text{rows matching } E_1 \text{ and } E_2} P(\text{row})}{\sum_{\text{rows matching } E_2} P(\text{row})}$$

P(Male | Poor) = 0.4654 / 0.7604 = 0.612

Andrew Moore, CMU

Joint distributions

- Good news

  Once you have a joint distribution, you can answer all sorts of probabilistic questions involving combinations of attributes

- Bad news

  Impossible to create JD for more than about ten attributes because there are so many numbers needed when you build the thing.

For 10 binary variables you need to specify 2^{10}-1 numbers = 1023.

(question for class: why the -1?)

How to use Fewer Numbers

- Factor the joint distribution into a product of distributions over subsets of variables
- Identify (or just assume) independence between some subsets of variables
- Use that independence to simplify some of the distributions
- Graphical models provide a principled way of doing this.

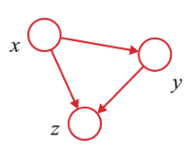
Factoring

- Consider an arbitrary joint distribution

$$p(x, y, z)$$

- We can always factor it, by application of the chain rule

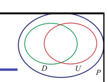
$$\begin{aligned} p(x,y,z) &= p(x)p(y,z|x) \\ &= p(x)p(y|x)p(z|x,y) \end{aligned}$$

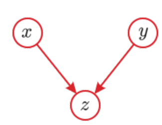


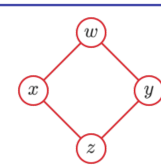
what this factored form looks like as a graphical model

Christopher Bishop, MSR

Directed versus Undirected Graphs







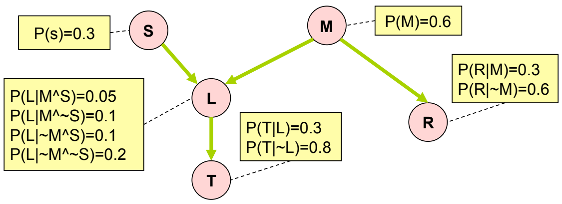
Directed Graph Examples:
- •Bayes nets
- •HMMs

Undirected Graph Examples
- • MRFS

Note: The word "graphical" denotes the graph structure underlying the model, not the fact that you can draw a pretty picture of it (although that helps).

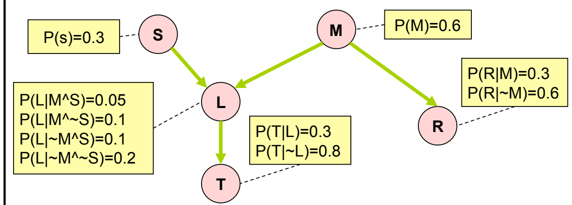
Christopher Bishop, MSR

Graphical Model Concepts



- Nodes represent random variables.
- Edges (or lack of edges) represent conditional dependence (or independence).
- Each node is annotated with a table of conditional probabilities wrt parents.

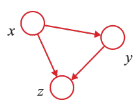
Graphical Model Concepts



Note: The word "graphical" denotes the graph structure underlying the model, not the fact that you can draw a pretty picture of it using graphics.

Directed Acyclic Graphs

- Directed acyclic means we can't follow arrows around in a cycle.
- Examples: chains; trees
- Also, things that look like this:



- We can "read" the factored form of the joint distribution immediately from a directed graph

$$p(x_1, \ldots, x_D) = \prod_{i=1}^{D} p(x_i|\text{pa}_i)$$

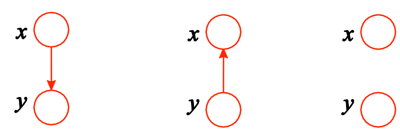
where pa_i denotes the parents of i

Factoring Examples

- Joint distribution

$$p(x_1, \ldots, x_D) = \prod_{i=1}^{D} p(x_i|\text{pa}_i)$$

where pa_i denotes the parents of i



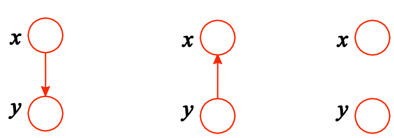
P(x| parents of x) P(y| parents of y)

Factoring Examples

- Joint distribution

$$p(x_1, \ldots, x_D) = \prod_{i=1}^{D} p(x_i|\text{pa}_i)$$

where pa_i denotes the parents of i



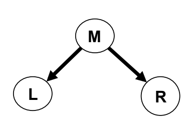
p(x,y)=p(x)p(y|x) p(x,y)=p(y)p(x|y) p(x,y)=p(x)p(y)

Factoring Examples

- We can "read" the form of the joint distribution directly from the directed graph

$$p(x_1, \ldots, x_D) = \prod_{i=1}^{D} p(x_i|\text{pa}_i)$$

where pa_i denotes the parents of i



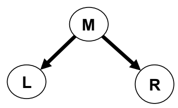
P(L| parents of L) P(M| parents of M) P(R| parents of R)

Factoring Examples

- We can "read" the form of the joint distribution directly from the directed graph

$$p(x_1, \ldots, x_D) = \prod_{i=1}^{D} p(x_i|\mathrm{pa}_i)$$

where pa_i denotes the parents of i



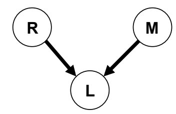
P(L,R,M) = P(M) P(L | M) P(R | M)

Factoring Examples

- We can "read" the form of the joint distribution directly from a directed graph

$$p(x_1, \ldots, x_D) = \prod_{i=1}^{D} p(x_i|\mathrm{pa}_i)$$

where pa_i denotes the parents of i



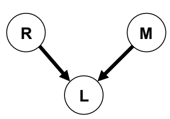
P(L| parents of L) P(M| parents of M) P(R| parents of R)

Factoring Examples

- We can "read" the form of the joint distribution directly from a directed graph

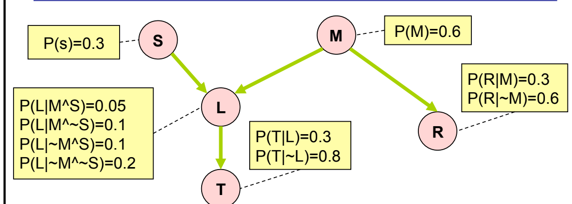
$$p(x_1, \ldots, x_D) = \prod_{i=1}^{D} p(x_i|\mathrm{pa}_i)$$

where pa_i denotes the parents of i



Note: P(L,R,M) = P(L|R,M)P(R)P(M)

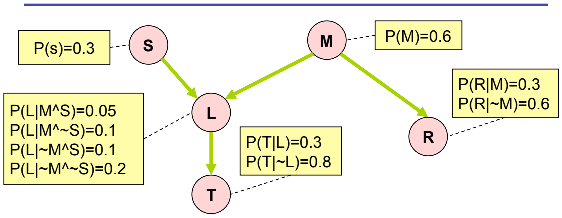
Graphical Model Concepts



- How about this one?

P(L,M,R,S,T) =

Graphical Model Concepts



- How about this one?

P(L,M,R,S,T) = P(S)P(M)P(L|S,M)P(R|M)P(T|L)

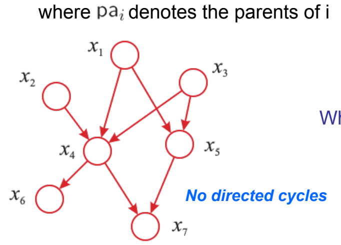
Note to mention P(T|L)+P(T|~L)

Factoring Examples

- Joint distribution

$$p(x_1, \ldots, x_D) = \prod_{i=1}^{D} p(x_i|\mathrm{pa}_i)$$

where pa_i denotes the parents of i



What about this one?

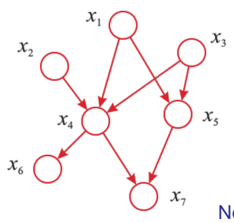
Christopher Bishop, MSR

Factoring Examples

- How many probabilities do we have to specify/learn (assuming each x_i is a binary variable)?

if fully connected, we would need 2^7-1 = 127

but, for this connectivity, we need 1+1+1+8+4+2+4 = 21



$$p(x_1 \ldots x_7) = \overset{2^0}{p(x_1)}\,\overset{2^0}{p(x_2)}\,\overset{2^0}{p(x_3)}\,\overset{2^3}{p(x_4|x_1,x_2,x_3)}\,\overset{2^2}{p(x_5|x_1,x_3)}\,\overset{2^1}{p(x_6|x_4)}\,\overset{2^2}{p(x_7|x_4,x_5)}$$

Note: If all nodes were independent, we would only need 7!

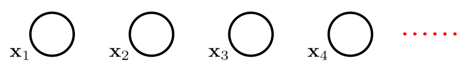
Important Case: Time Series

Consider modeling a time series of sequential data
x1, x2, ... , xN

These could represent
- locations of a tracked object over time
- observations of the weather each day
- spectral coefficients of a speech signal
- joint angles during human motion

Modeling Time Series

Simplest model of a time series is that all observations are independent.

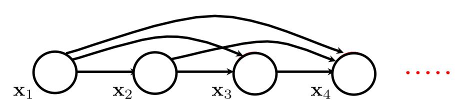


This would be appropriate for modeling successive tosses {heads,tails} of an unbiased coin.

However, it doesn't really treat the series as a sequence. That is, we could permute the ordering of the observations and not change a thing.

Modeling Time Series

In the most general case, we could use chain rule to state that any node is dependent on all previous nodes...



P(x1,x2,x3,x4,...) = P(x1)P(x2|x1)P(x3|x1,x2)P(x4|x1,x2,x3)...

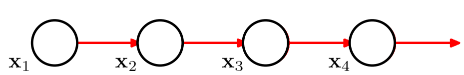
Look for an intermediate model between these two extremes.

Modeling Time Series

Markov assumption:
P(xn | x1,x2,...,xn-1) = P(xn | xn-1)

that is, assume all conditional distributions depend only on the most recent previous observation.

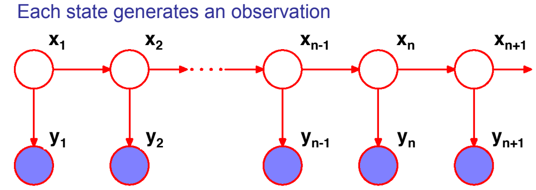
The result is a first-order Markov Chain



P(x1,x2,x3,x4,...) = P(x1)P(x2|x1)P(x3|x2)P(x4|x3)...

Modeling Time Series

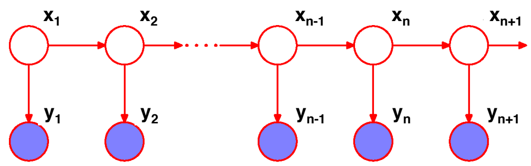
Generalization: State-Space Models
You have a Markov chain of latent (unobserved) states
Each state generates an observation



Goal: Given a sequence of observations, predict the sequence of unobserved states that maximizes the joint probability.

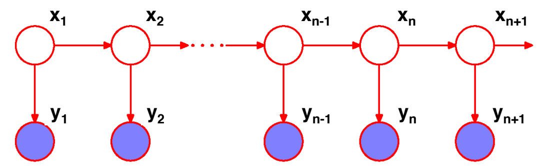
Modeling Time Series

Examples of State Space models
- Hidden Markov model
- Kalman filter

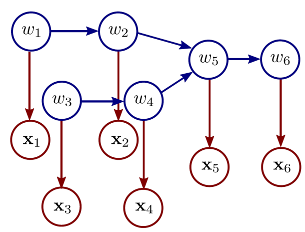


Modeling Time Series

P(x1,x2,x3,x4,....,y1,y2,y3,y4,...) =
P(x1)P(y1|x1)P(x2|x1)P(y2|x2)P(x3|x2)P(y3|x3)P(x4|x3)P(y4|x4)......



Example of a Tree-structured Model

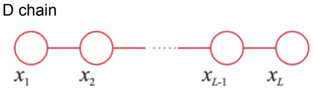


Confusion alert: Our textbook uses "w" to denote a world state variable and "x" to denote a measurement. (we have been using "x" to denote world state and "y" as the measurement).

Message Passing

Message Passing : Belief Propagation

- Example: 1D chain



- Find marginal for a particular node

$$p(x_i) = \sum_{x_1}\cdots\sum_{x_{i-1}}\sum_{x_{i+1}}\cdots\sum_{x_L} p(x_1,\ldots,x_L)$$

- for M-state nodes, cost is $O(M^L)$
- exponential in length of chain
- but, we can exploit the graphical structure (conditional independences)

M is number of discrete values a variable can take
L is number of variables

Applicable to both directed and undirected graphs.

Key Idea of Message Passing

multiplication distributes over addition

a * b + a * c = a * (b + c)

as a consequence:

$$\sum_i\sum_j\sum_k a_i b_j c_k = \sum_i\sum_j a_i b_j \left(\sum_k c_k\right)$$
$$= \sum_i a_i \left[\sum_j b_j \left(\sum_k c_k\right)\right]$$

Example

$$\sum_{i=1}^{2}\sum_{j=1}^{3}\sum_{k=1}^{4} a_i b_j c_k =$$

$$a_1b_1c_1+a_1b_1c_2+a_1b_1c_3+a_1b_1c_4+a_1b_2c_1+a_1b_2c_2+a_1b_2c_3+a_1b_2c_4$$
$$+a_1b_3c_1+a_1b_3c_2+a_1b_3c_3+a_1b_3c_4+a_2b_1c_1+a_2b_1c_2+a_2b_1c_3+a_2b_1c_4$$
$$+a_2b_2c_1+a_2b_2c_2+a_2b_2c_3+a_2b_2c_4+a_2b_3c_1+a_2b_3c_2+a_2b_3c_3+a_2b_3c_4$$

48 multiplications + 23 additions

$$\sum_{i=1}^{2} a_i \left[\sum_{j=1}^{3} b_j \left(\sum_{k=1}^{4} c_k\right)\right] =$$

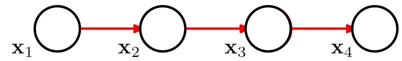
$$a_1[b_1(c_1+c_2+c_3+c_4)+b_2(c_1+c_2+c_3+c_4)+b_3(c_1+c_2+c_3+c_4)]$$
$$+a_2[b_1(c_1+c_2+c_3+c_4)+b_2(c_1+c_2+c_3+c_4)+b_3(c_1+c_2+c_3+c_4)]$$

5 multiplications + 6 additions

For message passing, this principle is applied to functions of random variables, rather than the variables as done here.

Message Passing

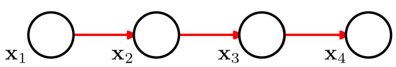
In the next several slides, we will consider an example of a simple, four-variable Markov chain.



$$P(x1,x2,x3,x4) = P(x1)\ P(x2|x1)\ P(x3|x2)\ P(x4|x3)$$

Message Passing

Now consider computing the marginal distribution of variable x3

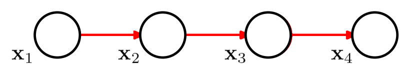


$$P(x1,x2,x3,x4) = P(x1)\ P(x2|x1)\ P(x3|x2)\ P(x4|x3)$$

$$\begin{aligned} P(x3) &= \sum_{x1}\sum_{x2}\sum_{x4} P(x1,x2,x3,x4) \\ &= \sum_{x1}\sum_{x2}\sum_{x4} P(x1)\ P(x2|x1)\ P(x3|x2)\ P(x4|x3) \end{aligned}$$

Message Passing

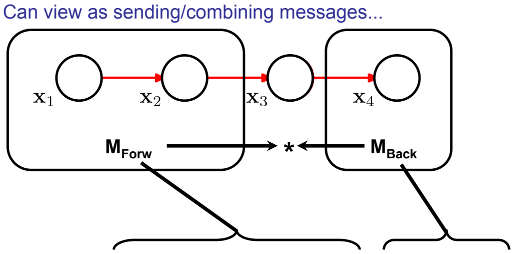
Multiplication distributes over addition...



$$\begin{aligned} P(x3) &= \sum_{x1}\sum_{x2}\sum_{x4} P(x1)\ P(x2|x1)\ P(x3|x2)\ P(x4|x3) \\ &= \sum_{x1}\sum_{x2} P(x1)\ P(x2|x1)\ P(x3|x2) \left(\sum_{x4} P(x4|x3)\right) \\ &= \left(\sum_{x2}\left(\sum_{x1} P(x1)\ P(x2|x1)\right) P(x3|x2)\right)\left(\sum_{x4} P(x4|x3)\right) \end{aligned}$$

Message Passing, aka Forward-Backward Algorithm

Can view as sending/combining messages...

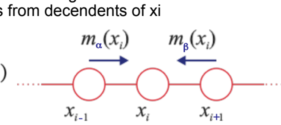


$$= \left(\sum_{x2}\left(\sum_{x1} P(x1)\ P(x2|x1)\right) P(x3|x2)\right)\left(\sum_{x4} P(x4|x3)\right)$$

Forward-Backward Algorithm

- Express marginals as product of messages evaluated forward from ancesters of xi and backwards from decendents of xi

$$p(x_i) = \frac{1}{Z} m_\alpha(x_i) m_\beta(x_i)$$



- Recursive evaluation of messages

$$\begin{aligned} m_\alpha(x_i) &= \sum_{x_{i-1}} \psi(x_{i-1}, x_i) m_\alpha(x_{i-1}) \\ m_\beta(x_i) &= \sum_{x_{i+1}} \psi(x_i, x_{i+1}) m_\beta(x_{i+1}) \end{aligned}$$

- Find Z by normalizing $p(x_i)$

Works in both directed and undirected graphs

Christopher Bishop, MSR

Confusion Alert!

This standard notation for defining message passing **heavily** overloads the notion of multiplication, e.g. the messages are not scalars – it is more appropriate to think of them as vectors, matrices, or even tensors depending on how many variables are involved, with "multiplication" defined accordingly.

$$p(x_i) = \frac{1}{Z} m_\alpha(x_i) m_\beta(x_i)$$

$$m_\alpha(x_i) = \sum_{x_{i-1}} \psi(x_{i-1}, x_i) m_\alpha(x_{i-1})$$

$$m_\beta(x_i) = \sum_{x_{i+1}} \psi(x_i, x_{i+1}) m_\beta(x_{i+1})$$

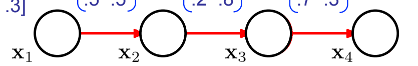
Not scalar multiplication!

Specific numerical example

$P(x1)$ $[.7\ \ .3]$

$P(x2|x1)$ $\begin{bmatrix} .4 & .6 \\ .5 & .5 \end{bmatrix}$ $P(x3|x2)$ $\begin{bmatrix} .8 & .2 \\ .2 & .8 \end{bmatrix}$ $P(x4|x3)$ $\begin{bmatrix} .5 & .5 \\ .7 & .3 \end{bmatrix}$

Note: these are conditional probability tables, so values in each row must sum to one



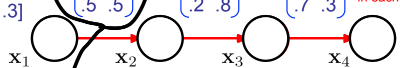
$$P(x1,x2,x3,x4) = P(x1)\ P(x2|x1)\ P(x3|x2)\ P(x4|x3)$$

Specific numerical example

$P(x1)$ $[.7\ \ .3]$

$P(x2|x1)$ $\begin{bmatrix} .4 & .6 \\ .5 & .5 \end{bmatrix}$ $P(x3|x2)$ $\begin{bmatrix} .8 & .2 \\ .2 & .8 \end{bmatrix}$ $P(x4|x3)$ $\begin{bmatrix} .5 & .5 \\ .7 & .3 \end{bmatrix}$

Note: these are conditional probability tables, so values in each row must sum to one



$$P(x1,x2,x3,x4) = P(x1)\ P(x2|x1)\ P(x3|x2)\ P(x4|x3)$$

Interpretation

$P(x2|x1)$

$$\begin{array}{c|cc} & P(x2=1) & P(x2=2) \\ \hline P(x1=1) & .4 & .6 \\ P(x1=2) & .5 & .5 \end{array}$$

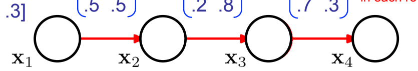
$.6 \rightarrow$ P(x2=2|X1=1) = 0.6

Specific numerical example

$P(x1)$ $[.7\ \ .3]$

$P(x2|x1)$ $\begin{bmatrix} .4 & .6 \\ .5 & .5 \end{bmatrix}$ $P(x3|x2)$ $\begin{bmatrix} .8 & .2 \\ .2 & .8 \end{bmatrix}$ $P(x4|x3)$ $\begin{bmatrix} .5 & .5 \\ .7 & .3 \end{bmatrix}$

Note: these are conditional probability tables, so values in each row must sum to one



$$P(x1,x2,x3,x4) = P(x1)\ P(x2|x1)\ P(x3|x2)\ P(x4|x3)$$

Sample computations:

P(x1=1, x2=1, x3=1, x4=1) = (.7)(.4)(.8)(.5) = .112

P(x1=2, x2=1, x3=2, x4=1) = (.3)(.5)(.2)(.7) = .021

Specific numerical example

Joint Probability, represented in a truth table

x1	x2	x3	x4	P(x1,x2,x3,x4)
1.0000	1.0000	1.0000	1.0000	0.1120
1.0000	1.0000	1.0000	2.0000	0.1120
1.0000	1.0000	2.0000	1.0000	0.0392
1.0000	1.0000	2.0000	2.0000	0.0168
1.0000	2.0000	1.0000	1.0000	0.0420
1.0000	2.0000	1.0000	2.0000	0.0420
1.0000	2.0000	2.0000	1.0000	0.2352
1.0000	2.0000	2.0000	2.0000	0.1008
2.0000	1.0000	1.0000	1.0000	0.0600
2.0000	1.0000	1.0000	2.0000	0.0600
2.0000	1.0000	2.0000	1.0000	0.0210
2.0000	1.0000	2.0000	2.0000	0.0090
2.0000	2.0000	1.0000	1.0000	0.0150
2.0000	2.0000	1.0000	2.0000	0.0150
2.0000	2.0000	2.0000	1.0000	0.0840
2.0000	2.0000	2.0000	2.0000	0.0360

Specific numerical example

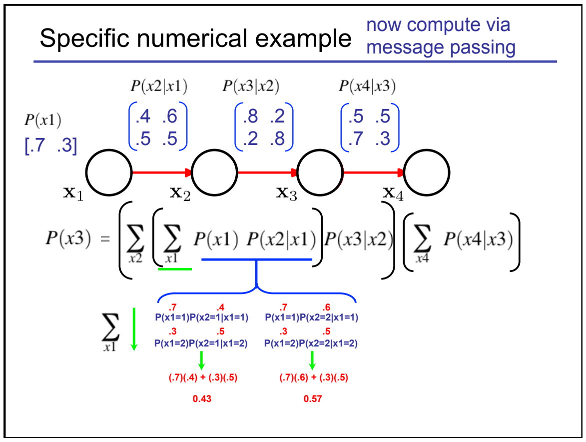
Joint Probability

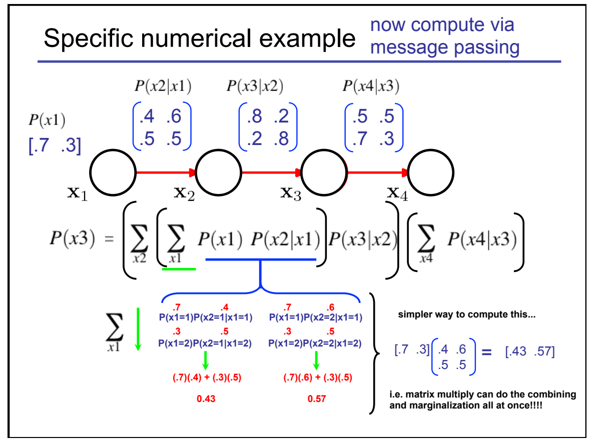
x1	x2	x3	x4	P(x1,x2,x3,x4)
1.0000	1.0000	1.0000	1.0000	0.1120
1.0000	1.0000	1.0000	2.0000	0.1120
1.0000	1.0000	2.0000	1.0000	0.0392
1.0000	1.0000	2.0000	2.0000	0.0168
1.0000	2.0000	1.0000	1.0000	0.0420
1.0000	2.0000	1.0000	2.0000	0.0420
1.0000	2.0000	2.0000	1.0000	0.2352
1.0000	2.0000	2.0000	2.0000	0.1008
2.0000	1.0000	1.0000	1.0000	0.0600
2.0000	1.0000	1.0000	2.0000	0.0600
2.0000	1.0000	2.0000	1.0000	0.0210
2.0000	1.0000	2.0000	2.0000	0.0090
2.0000	2.0000	1.0000	1.0000	0.0150
2.0000	2.0000	1.0000	2.0000	0.0150
2.0000	2.0000	2.0000	1.0000	0.0840
2.0000	2.0000	2.0000	2.0000	0.0360

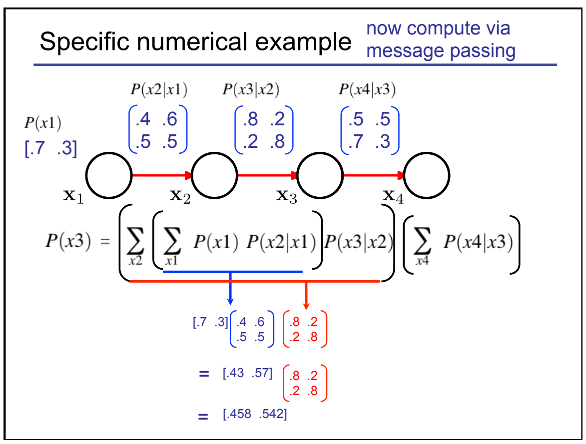
Compute marginal of x3:

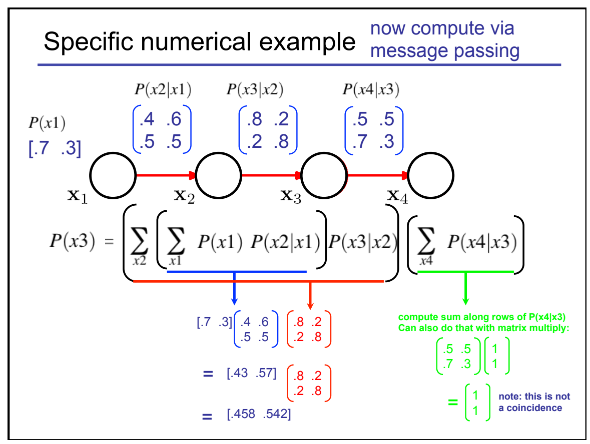
P(x3=1) = 0.458

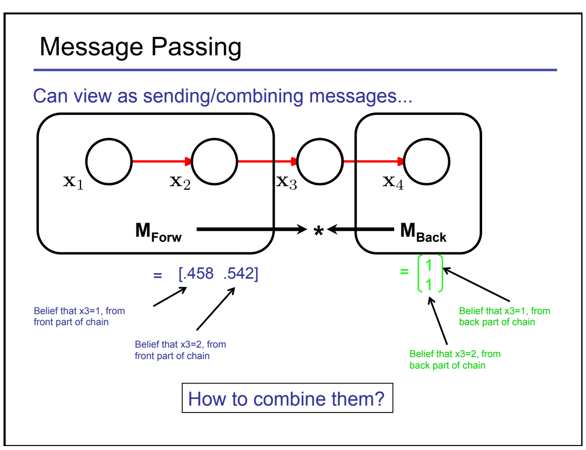
P(x3=2) = 0.542

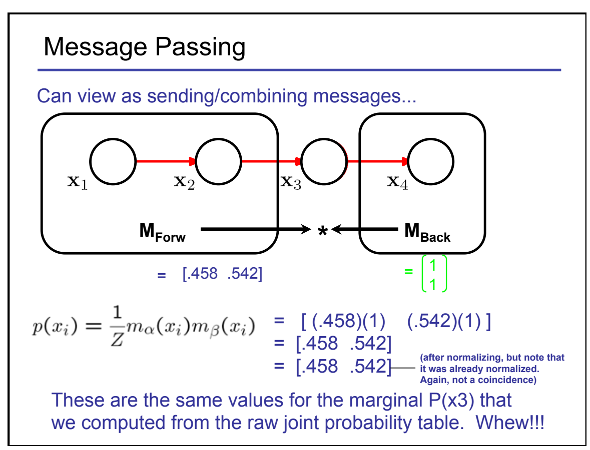

Specific numerical example
now compute via message passing
P(x2|x1)
P(x3|x2)
P(x4|x3)
P(x1)
[.7 .3]
.4 .6
.5 .5
.8 .2
.2 .8
.5 .5
.7 .3
x1
x2
x3
x4
P(x3) = Σx2 (Σx1 P(x1) P(x2|x1)) P(x3|x2) (Σx4 P(x4|x3))
P(x1=1)P(x2=1|x1=1)
P(x1=2)P(x2=1|x1=2)
P(x1=1)P(x2=2|x1=1)
P(x1=2)P(x2=2|x1=2)
(.7)(.4) + (.3)(.5)
(.7)(.6) + (.3)(.5)
0.43
0.57


Specific numerical example
now compute via message passing
P(x2|x1)
P(x3|x2)
P(x4|x3)
P(x1)
[.7 .3]
x1
x2
x3
x4
P(x3) = Σx2 (Σx1 P(x1) P(x2|x1)) P(x3|x2) (Σx4 P(x4|x3))
P(x1=1)P(x2=1|x1=1)
P(x1=2)P(x2=1|x1=2)
P(x1=1)P(x2=2|x1=1)
P(x1=2)P(x2=2|x1=2)
(.7)(.4) + (.3)(.5)
(.7)(.6) + (.3)(.5)
0.43
0.57
simpler way to compute this...
[.7 .3] (.4 .6 / .5 .5) = [.43 .57]
i.e. matrix multiply can do the combining and marginalization all at once!!!!

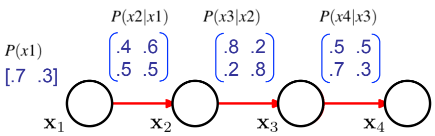

Specific numerical example
now compute via message passing
P(x2|x1)
P(x3|x2)
P(x4|x3)
P(x1)
[.7 .3]
x1
x2
x3
x4
P(x3) = Σx2 (Σx1 P(x1) P(x2|x1)) P(x3|x2) (Σx4 P(x4|x3))
[.7 .3] (.4 .6 / .5 .5) (.8 .2 / .2 .8)
= [.43 .57] (.8 .2 / .2 .8)
= [.458 .542]


Specific numerical example
now compute via message passing
P(x2|x1)
P(x3|x2)
P(x4|x3)
P(x1)
[.7 .3]
x1
x2
x3
x4
P(x3) = Σx2 (Σx1 P(x1) P(x2|x1)) P(x3|x2) (Σx4 P(x4|x3))
[.7 .3] (.4 .6 / .5 .5) (.8 .2 / .2 .8)
= [.43 .57] (.8 .2 / .2 .8)
= [.458 .542]
compute sum along rows of P(x4|x3)
Can also do that with matrix multiply:
(.5 .5 / .7 .3) (1 / 1)
= (1 / 1)
note: this is not a coincidence


Message Passing
Can view as sending/combining messages...
x1
x2
x3
x4
MForw
MBack
*
= [.458 .542]
= (1 / 1)
Belief that x3=1, from front part of chain
Belief that x3=2, from front part of chain
Belief that x3=1, from back part of chain
Belief that x3=2, from back part of chain
How to combine them?


Message Passing
Can view as sending/combining messages...
x1
x2
x3
x4
MForw
MBack
*
= [.458 .542]
= (1 / 1)
p(x_i) = 1/Z m_α(x_i) m_β(x_i)
= [(.458)(1) (.542)(1)]
= [.458 .542]
= [.458 .542]
(after normalizing, but note that it was already normalized. Again, not a coincidence)
These are the same values for the marginal P(x3) that we computed from the raw joint probability table. Whew!!!

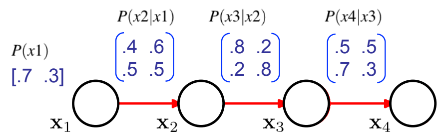
Specific numerical example



If we want to compute all marginals, we can do it in one shot by cascading, for a big computational savings.

We need one cascaded forward pass, one separate cascaded backward pass, then a combination and normalization at each node.

Specific numerical example



forward pass

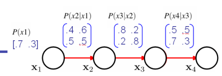
$$[.7\ \ .3]\begin{pmatrix}.4 & .6\\ .5 & .5\end{pmatrix}\begin{pmatrix}.8 & .2\\ .2 & .8\end{pmatrix}\begin{pmatrix}.5 & .5\\ .7 & .3\end{pmatrix}$$

backward pass

$$\begin{pmatrix}.4 & .6\\ .5 & .5\end{pmatrix}\begin{pmatrix}.8 & .2\\ .2 & .8\end{pmatrix}\begin{pmatrix}.5 & .5\\ .7 & .3\end{pmatrix}\begin{pmatrix}1\\ 1\end{pmatrix}$$

Then combine each by elementwise multiply and normalize

Specific numerical example



forward pass

$$[.7\ \ .3]\begin{pmatrix}.4 & .6\\ .5 & .5\end{pmatrix}\begin{pmatrix}.8 & .2\\ .2 & .8\end{pmatrix}\begin{pmatrix}.5 & .5\\ .7 & .3\end{pmatrix}$$

backward pass

$$\begin{pmatrix}.4 & .6\\ .5 & .5\end{pmatrix}\begin{pmatrix}.8 & .2\\ .2 & .8\end{pmatrix}\begin{pmatrix}.5 & .5\\ .7 & .3\end{pmatrix}\begin{pmatrix}1\\ 1\end{pmatrix}$$

Then combine each by elementwise multiply and normalize

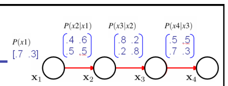
Forward:	[.7 .3]	[.43 .57]	[.458 .542]	[.6084 .3916]
Backward:	[1 1]	[1 1]	[1 1]	[1 1]
combined+ normalized	[.7 .3]	[.43 .57]	[.458 .542]	[.6084 .3916]

Specific numerical example

1.0000	1.0000	1.0000	1.0000	0.1120
1.0000	1.0000	1.0000	2.0000	0.1120
1.0000	1.0000	2.0000	1.0000	0.0392
1.0000	1.0000	2.0000	2.0000	0.0168
1.0000	2.0000	1.0000	1.0000	0.0420
1.0000	2.0000	1.0000	2.0000	0.0420
1.0000	2.0000	2.0000	1.0000	0.2352
1.0000	2.0000	2.0000	2.0000	0.1008
2.0000	1.0000	1.0000	1.0000	0.0600
2.0000	1.0000	1.0000	2.0000	0.0600
2.0000	1.0000	2.0000	1.0000	0.0210
2.0000	1.0000	2.0000	2.0000	0.0090
2.0000	2.0000	1.0000	1.0000	0.0150
2.0000	2.0000	1.0000	2.0000	0.0150
2.0000	2.0000	2.0000	1.0000	0.0840
2.0000	2.0000	2.0000	2.0000	0.0360

```
num truth table entries = 16
Computation using joint prob table took 0.062000 sec
marginal x1:    0.700000    0.300000
marginal x2:    0.430000    0.570000
marginal x3:    0.458000    0.542000
marginal x4:    0.608400    0.391600
Computation using BP sum-product took 0.000000 sec
marginal x1:    0.700000    0.300000
marginal x2:    0.430000    0.570000
marginal x3:    0.458000    0.542000
marginal x4:    0.608400    0.391600
```

Specific numerical example



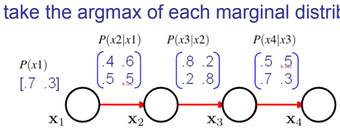
Note: In this example, a directed Markov chain using true conditional probabilities (rows sum to one), only the forward pass is needed. This is true because the backward pass sums along rows, and always produces [1 1]' .

We didn't really need forward AND backward in this example.

Forward:	[.7 .3]	[.43 .57]	[.458 .542]	[.6084 .3916]	**these are already the marginals for this example**
Backward:	[1 1]	[1 1]	[1 1]	[1 1]	**Didn't need to do these steps**
combined+ normalized	[.7 .3]	[.43 .57]	[.458 .542]	[.6084 .3916]	

Max Marginals

What if we want to know the most probably state (mode of the distribution)? Since the marginal distributions can tell us which value of each variable yields the highest marginal probability (that is, which value is most likely), we might try to just take the argmax of each marginal distribution.

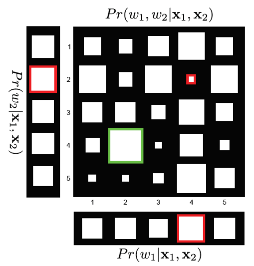


marginals computed by belief propagation	[.7 .3]	[.43 .57]	[.458 .542]	[.6084 .3916]
max marginals	↑	↑	↑	↑
argmax	$\hat{x}_1 = 1$	$\hat{x}_2 = 2$	$\hat{x}_3 = 2$	$\hat{x}_4 = 1$

Although that's correct in this example, it isn't always the case

Max Marginals can Fail to Find the MAP

However, the max marginals find most likely values of each variable treated individually, which may not be the combination of values that jointly maximize the distribution.



max marginals:
w1=4, w2=2

actual MAP solution:
w1=2, w2=4

Max-product Algorithm

- Goal: find

$$\mathbf{x}^{\text{MAP}} = \arg\max_{\mathbf{x}} p(\mathbf{x})$$

  – define the "max marginal"

$$\boldsymbol{M}(x_i) = \max_{x_1} \cdots \max_{x_{i-1}} \max_{x_{i+1}} \cdots \max_{x_L} p(x_1, \ldots, x_L)$$

  – then

$$x_i^{\text{MAP}} = \arg\max_{x_i} \phi(x_i)$$

- Message passing algorithm with "sum" replaced by "max"
- Generalizes to any two operations forming a semiring

Christopher Bishop, MSR

Computing MAP Value

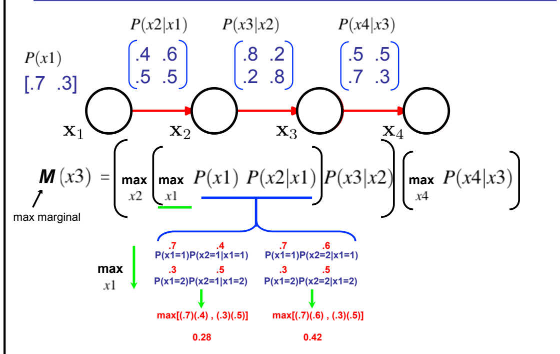
$$\boldsymbol{M}(x_i) = \max_{x_1} \cdots \max_{x_{i-1}} \max_{x_{i+1}} \cdots \max_{x_L} p(x_1, \ldots, x_L)$$

Can solve using message passing algorithm with "sum" replaced by "max".

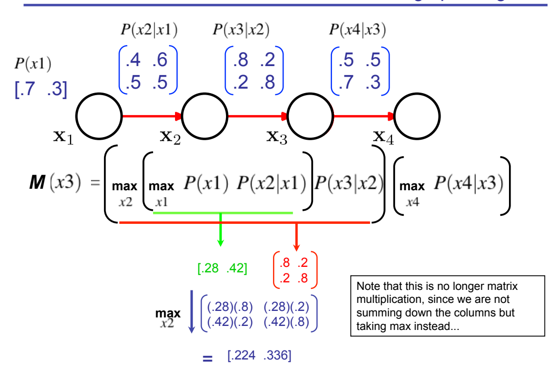
In our chain, we start at the end and work our way back to the root (x1) using the max-product algorithm, keeping track of the max value as we go.

$$\underbrace{\max_i(a_i \underbrace{\max_j(b_j \underbrace{\max_k(c_k)}_{\text{stage1}})}_{\text{stage2}})}_{\text{stage3}}$$

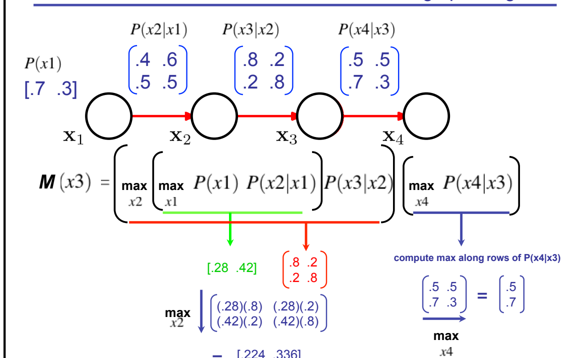
Specific numerical example **max product** message passing



Specific numerical example **max product** message passing

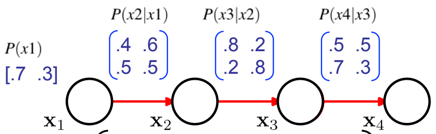


Specific numerical example **max product** message passing



Specific numerical example

max product
message passing



$$\boldsymbol{M}(x3) = \left(\max_{x2} \left(\max_{x1} P(x1)\,P(x2|x1) \right) P(x3|x2) \right) \left(\max_{x4} P(x4|x3) \right)$$

[.224 .336]

[.5 .7]

max [(.224)(.5) , (.336)(.7)]

.112 .2352

= .2352 argmax = 2
x3

interpretation:
the mode of the joint distribution is .2352, and the value of variable x3, in the configuration that yields the mode value, is 2.

Specific numerical example

Joint Probability, represented in a truth table

x1	x2	x3	x4	P(x1,x2,x3,x4)
1.0000	1.0000	1.0000	1.0000	0.1120
1.0000	1.0000	1.0000	2.0000	0.1120
1.0000	1.0000	2.0000	1.0000	0.0392
1.0000	1.0000	2.0000	2.0000	0.0168
1.0000	2.0000	1.0000	1.0000	0.0420
1.0000	2.0000	1.0000	2.0000	0.0420
1.0000	2.0000	2.0000	1.0000	0.2352
1.0000	2.0000	2.0000	2.0000	0.1008
2.0000	1.0000	1.0000	1.0000	0.0600
2.0000	1.0000	1.0000	2.0000	0.0600
2.0000	1.0000	2.0000	1.0000	0.0210
2.0000	1.0000	2.0000	2.0000	0.0090
2.0000	2.0000	1.0000	1.0000	0.0150
2.0000	2.0000	1.0000	2.0000	0.0150
2.0000	2.0000	2.0000	1.0000	0.0840
2.0000	2.0000	2.0000	2.0000	0.0360

Largest value of joint prob = mode = MAP

interpretation:
the mode of the joint distribution is .2352, and the value of variable x3, in the configuration that yields the mode value, is 2.

max [(.224)(.5) , (.336)(.7)]

.112 .2352

= .2352 argmax = 2
x3

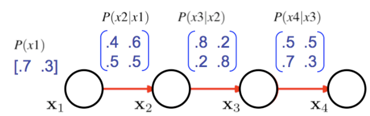
Computing Arg-Max of MAP Value

$$x_i^{\text{MAP}} = \arg\max_{x_i} \phi(x_i)$$

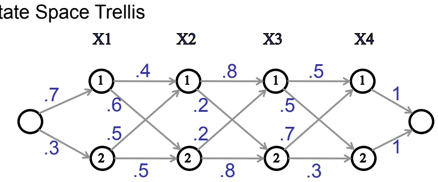
Chris Bishop, PRML:
"At this point, we might be tempted simply to continue with the message passing algorithm [sending forward-backward messages and combining to compute argmax for each variable node]. However, because we are now maximizing rather than summing, it is possible that there may be multiple configurations of x all of which give rise to the maximum value for p(x). In such cases, this strategy can fail because it is possible for the individual variable values obtained by maximizing the product of messages at each node to belong to different maximizing configurations, giving an overall configuration that no longer corresponds to a maximum. The problem can be resolved by adopting a rather different kind of message passing..."

Essentially, the solution is to write a dynamic programming algorithm based on max-product.

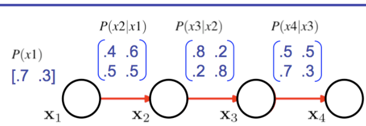
Specific numerical example: MAP Estimate



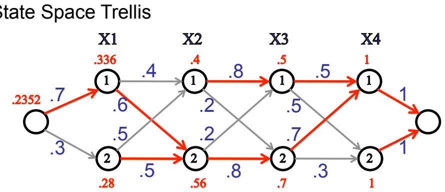
DP State Space Trellis



Specific numerical example: MAP Estimate



DP State Space Trellis



Specific numerical example

Joint Probability, represented in a truth table

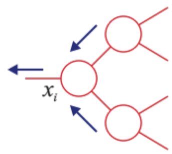
x1	x2	x3	x4	P(x1,x2,x3,x4)
1.0000	1.0000	1.0000	1.0000	0.1120
1.0000	1.0000	1.0000	2.0000	0.1120
1.0000	1.0000	2.0000	1.0000	0.0392
1.0000	1.0000	2.0000	2.0000	0.0168
1.0000	2.0000	1.0000	1.0000	0.0420
1.0000	2.0000	1.0000	2.0000	0.0420
1.0000	2.0000	2.0000	1.0000	0.2352
1.0000	2.0000	2.0000	2.0000	0.1008
2.0000	1.0000	1.0000	1.0000	0.0600
2.0000	1.0000	1.0000	2.0000	0.0600
2.0000	1.0000	2.0000	1.0000	0.0210
2.0000	1.0000	2.0000	2.0000	0.0090
2.0000	2.0000	1.0000	1.0000	0.0150
2.0000	2.0000	1.0000	2.0000	0.0150
2.0000	2.0000	2.0000	1.0000	0.0840
2.0000	2.0000	2.0000	2.0000	0.0360

Largest value of joint prob = mode = MAP

achieved for
x1=1, x2=2, x3=2, x4=1

Belief Propagation Summary

- Definition can be extended to general tree-structured graphs
- Works for both directed AND undirected graphs
- Efficiently computes marginals and MAP configurations
- At each node:
  - form product of *incoming* messages and local evidence
  - marginalize to give *outgoing* message
  - one message in each direction across every link



- Gives exact answer in any acyclic graph (no loops).

Loopy Belief Propagation

- BP applied to graph that contains loops
  - needs a propagation “schedule”
  - needs multiple iterations
  - might not converge
- Typically works well, even though it isn’t supposed to
- State-of-the-art performance in error-correcting codes