Probabilistic Graph Models: Factor Graphs, Belief Propagation

ECE/CS 498 DS U/G

Lecture 22: Factor Graphs, Belief Propagation

Ravi K. Iyer

Dept. of Electrical and Computer Engineering

University of Illinois at Urbana Champaign

Announcements

- Course Timeline:
  - Mon 4/13: Factor graphs and belief propagation
  - Wed 4/15: ICA 5; HW4
- MP 3 Checkpoint 1 due **tonight @ 11:59 PM** on Compass 2G
  - .ipynb with Tasks 0-1 completed
  - PDF of slides with answers to Tasks 0-1 (using provided template)
- Final Project
  - Progress report 2 due **Friday April 17 @ 11:59 PM** on Compass2G
    - There should be *substantial* progress with projects by this point (i.e. meaningful results, ML/AI models)
- Discussion section on Friday 4/17
  - Practice with factor graphs and belief propagation

Introduction to Factor Graphs

Hidden Markov Models

Model

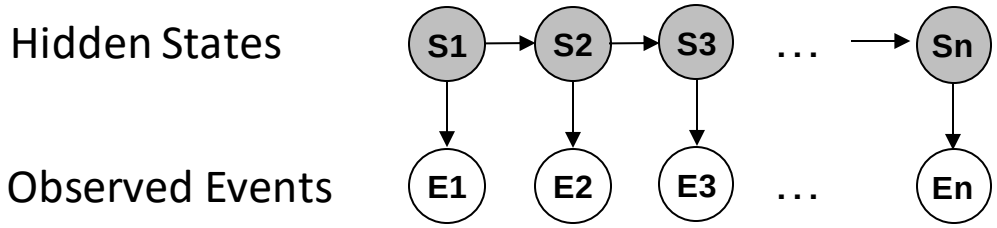
- Set of hidden states $\boldsymbol{S} = \{\boldsymbol{\sigma_1}, \ldots, \boldsymbol{\sigma_N}\}$
- Set of observable events $\boldsymbol{E} = \{\boldsymbol{\epsilon_1}, \ldots, \boldsymbol{\epsilon_M}\}$
- Transition probability matrix $\boldsymbol{A}$
- Observation matrix $\boldsymbol{B}$
- Initial distribution of hidden states $\boldsymbol{\pi}$

Model assumptions

- An observation depends on its hidden state
- A state variable only depends on the immediate previous state (Markov assumption)
- The future observations and the past observations are conditionally independent given the current hidden state

Advantages:

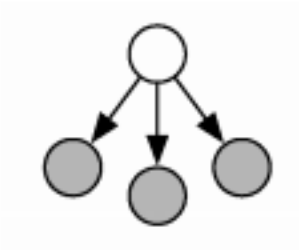
- HMM can model sequential nature of input data (future depends on the past)
- HMM has a linear-chain structure that clearly separates system state and observed events.



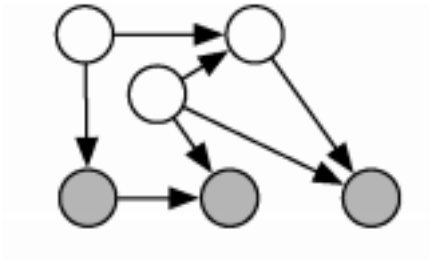
A Hidden Markov model on observed events and system states

$$P(S_1, \ldots, S_n, E_1, \ldots, E_n)$$
$$= P(S_1)P(E_1|S_1)\prod_{i=2}^{n} P(S_i|S_{i-1})P(E_i|S_i)$$

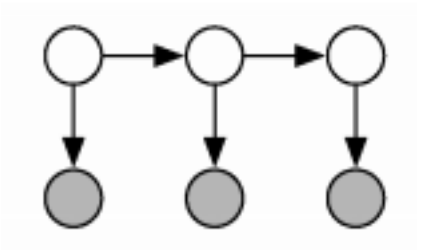
Taxonomy of graphical models



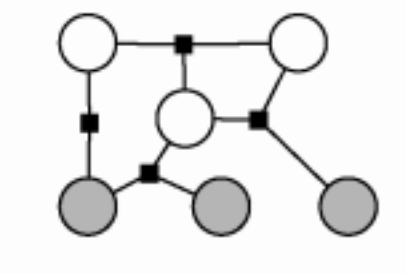
Naïve Bayes



Bayesian Network

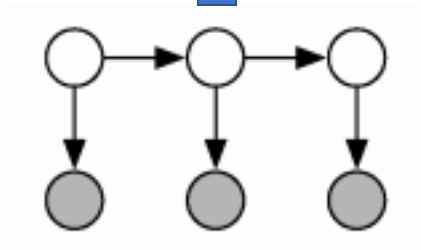


Dynamic Bayesian Network



Factor Graph

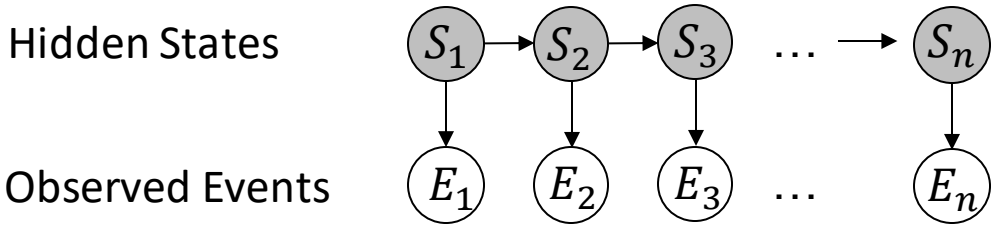
Conditional probabilities and statistical dependencies can be represented by a general type of graph: Factor Graph



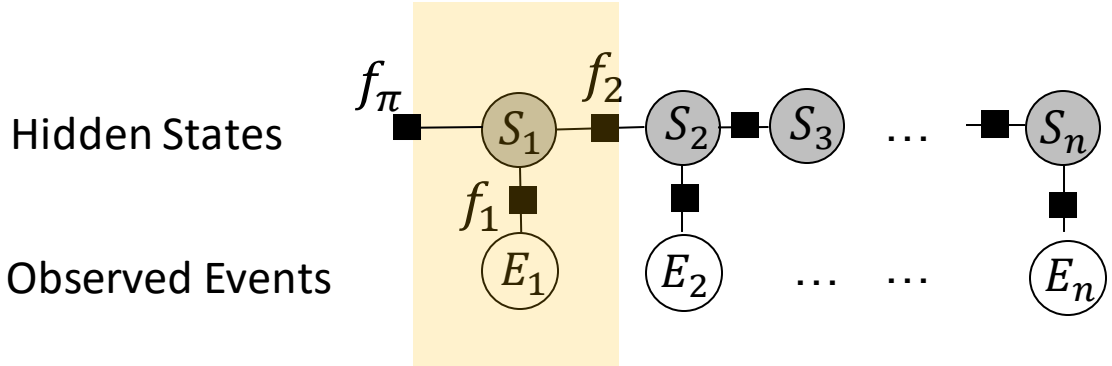
Hidden Markov Model

Conversion of a Hidden Markov Model to a Factor Graph

Hidden Markov Model

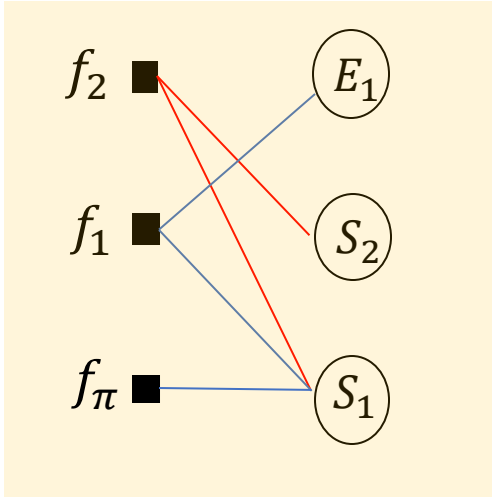


Factor Graph of the HMM



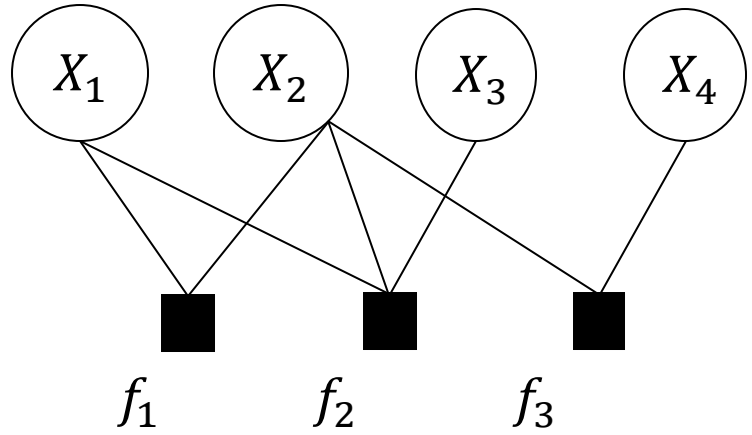
The above *Factor Graph* (FG) is a generalization of the Hidden Markov Model

- Boxes (f_π, f_1, f_2) represents factor function
- In the above case, it maintains the Markov assumption between states



Bipartite graph representation of the FG

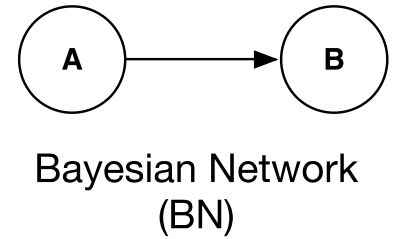
Definition of a Factor Graph



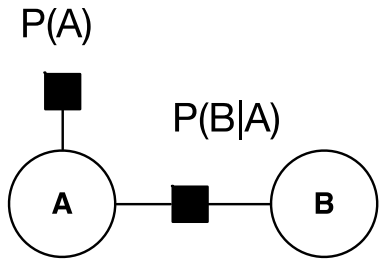
A factor graph is a **bipartite, undirected graph** of **random variables and factor functions. [Frey et. al. 01].**

G (graph) = (X,f,E); E denotes the edges

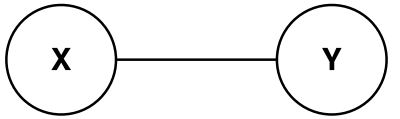
*FG can represent both **causal and non-causal relations**.*



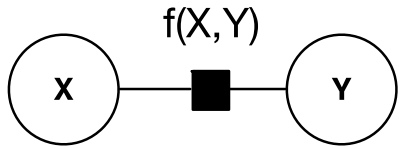
Bayesian Network (BN)



Factor Graph equivalent of BN



Undirected Graph



Factor Graph equivalent of UG

Example Factor function for HMMs

Assume that the state space and observation space are $S = \{\sigma_0, \sigma_1\}$, $E = \{\epsilon_1, \epsilon_2\}$. An example of factor functions is shown.

S	$f_\pi(S)$
σ_0	40
σ_1	25

S_t	E_t	$f_1(S_t, E_t)$
σ_0	ϵ_1	20
σ_0	ϵ_2	15
σ_1	ϵ_1	40
σ_1	ϵ_2	3

S_t	S_{t+1}	$f_2(S_t, S_{t+1})$
σ_0	σ_0	5
σ_0	σ_1	1
σ_1	σ_0	10
σ_1	σ_1	15

- Factor values represents the *affinities* between the related variables
  - E.g., $f_1(\sigma_1, \epsilon_1) > f_1(\sigma_0, \epsilon_1)$ implies that σ_1 and ϵ_1 are more compatible than σ_0 and ϵ_1
- Factor functions don't necessarily represent PDs or joint probability distributions
- How are these values found?
  1. Given by expert or from domain knowledge
  2. Derived from the data (priors)

Definition of Factor functions

Definition:

- Let $\boldsymbol{D}$ be a set of random variables. We define a factor f to be a function from $Val(D)$ to $\mathbb{R}$. A factor is non-negative if all its values are non-negative.
- The set of variables $\boldsymbol{D}$ is called the scope of the factor f and is denoted as $Scope(f)$.
- $Val(\boldsymbol{D})$ represents the set of values $\boldsymbol{D}$ can take.

Example:

A	B	$f(A,B)$
a_0	b_0	30
a_0	b_1	5
a_1	b_0	1
a_1	b_1	10

$\boldsymbol{D} = \{A, B\}$

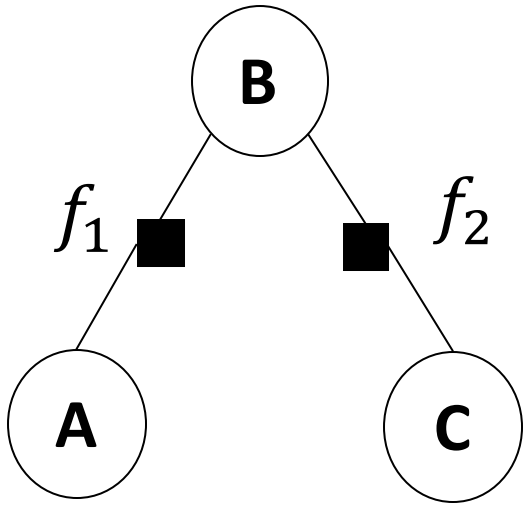
$Val(\boldsymbol{D}) = \{(a_0, b_0), (a_0, b_1), (a_1, b_0), (a_1, b_1)\}$

$A \in \{a_0, a_1\}$

$B \in \{b_0, b_1\}$

Product of Factor Functions in a Factor Graph

- In HMMs, we derived the joint distribution from the graph representation: $P(S_1, \dots, S_n, E_1, \dots, E_n) = P(S_1)P(E_1|S_1)\prod P(S_i|S_{i-1})P(E_i|S_i)$
- For a Factor Graph, the joint distribution can be derived from the product of factor functions (given that all factor functions are non-negative)



Example Factor Graph over variables A, B, C.

$$P(A, B, C) = \frac{1}{Z} f_1(A, B) f_2(B, C)$$

where, the normalization Z is given as

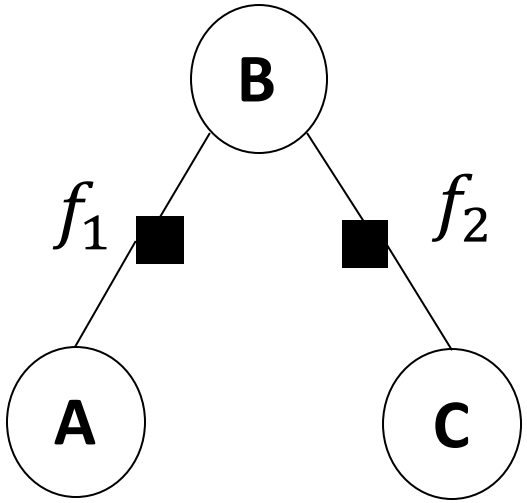
$$Z = \sum_{A,B,C} f(A, B, C) = \sum_{A,B,C} f_1(A, B) f_2(B, C)$$

Z is also referred to as the *partition function.*

Example of product of factor functions

Two factors f_1 and f_2 are multiplied in a way that "matches up" the common variables

$$f(A, B, C) = f_1(A, B) f_2(B, C)$$



f_1

a^1	b^1	0.5
a^1	b^2	0.8
a^2	b^1	0.1
a^2	b^2	0
a^3	b^1	0.3
a^3	b^2	0.9

f_2

b^1	c^1	0.5
b^1	c^2	0.7
b^2	c^1	0.1
b^2	c^2	0.2

f

a^1	b^1	c^1	$0.5 \cdot 0.5 = 0.25$
a^1	b^1	c^2	$0.5 \cdot 0.7 = 0.35$
a^1	b^2	c^1	$0.8 \cdot 0.1 = 0.08$
a^1	b^2	c^2	$0.8 \cdot 0.2 = 0.16$
a^2	b^1	c^1	$0.1 \cdot 0.5 = 0.05$
a^2	b^1	c^2	$0.1 \cdot 0.7 = 0.07$
a^2	b^2	c^1	$0 \cdot 0.1 = 0$
a^2	b^2	c^2	$0 \cdot 0.2 = 0$
a^3	b^1	c^1	$0.3 \cdot 0.5 = 0.15$
a^3	b^1	c^2	$0.3 \cdot 0.7 = 0.21$
a^3	b^2	c^1	$0.9 \cdot 0.1 = 0.09$
a^3	b^2	c^2	$0.9 \cdot 0.2 = 0.18$

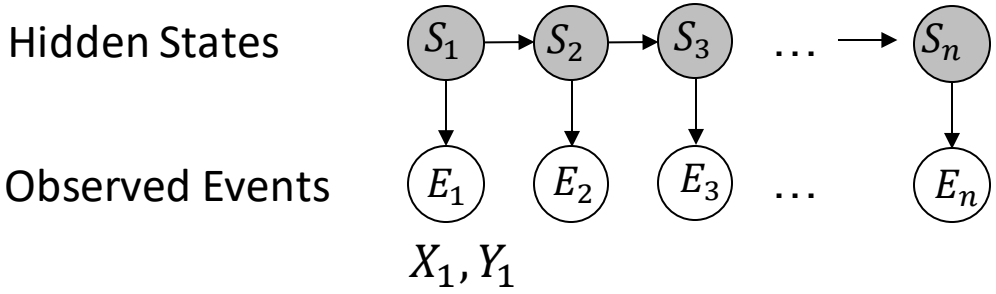
For example, $f(a^2, b^1, c^1) = f_1(a^2, b^1) f_2(b^1, c^1)$

Conversion of a Hidden Markov Model to a Factor Graph– Two dimension

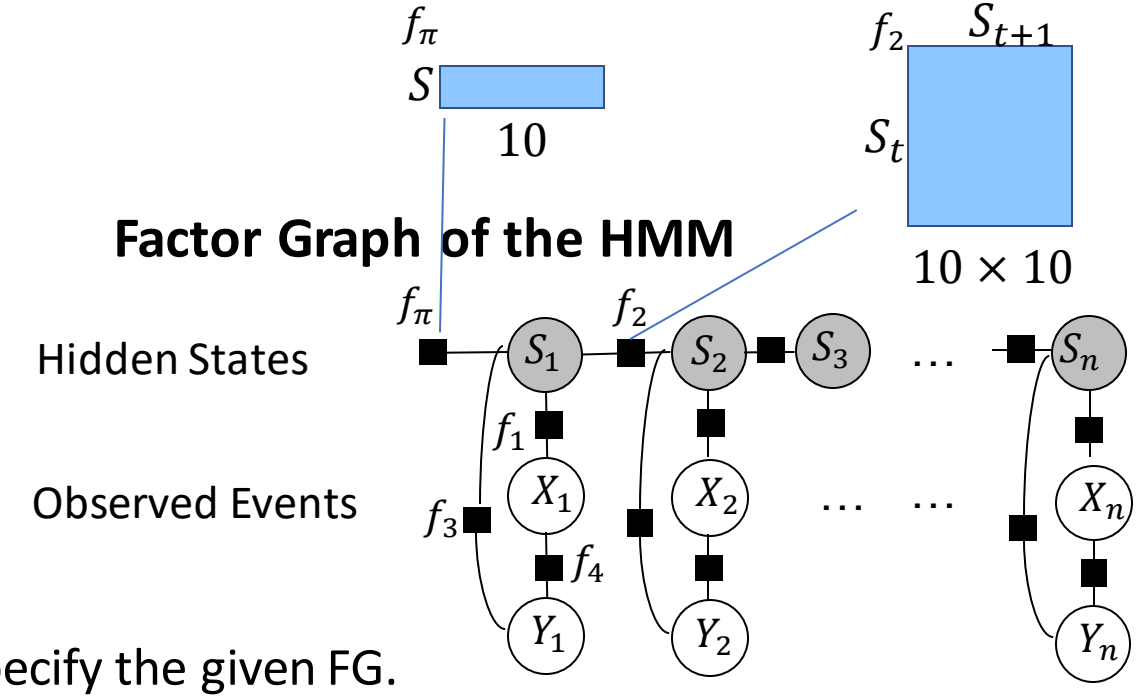
Assume that at each time point, two observations are made corresponding to random variables X and Y.

Example: Let $|\boldsymbol{S}| = \mathbf{10}, |\boldsymbol{X}| = \mathbf{10}, |\boldsymbol{Y}| = \mathbf{10}$

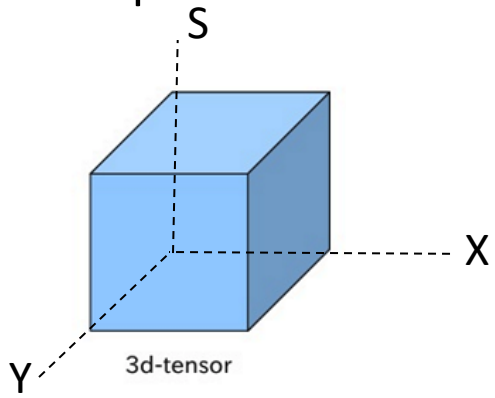
Hidden Markov Model



Factor Graph of the HMM

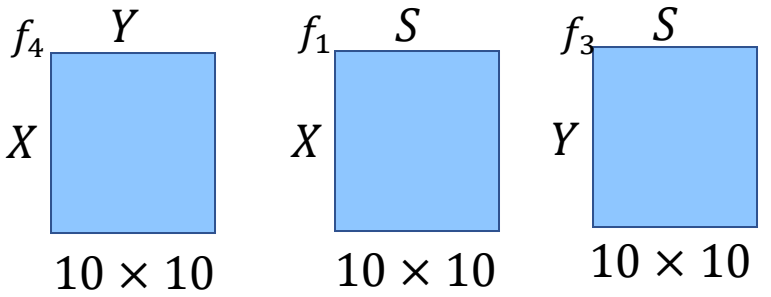


Fewer number of parameters are required are required to specify the given FG.



size of tensor is exponential

$10 \times 10 \times 10 = \mathbf{1000}$



size of five matrices

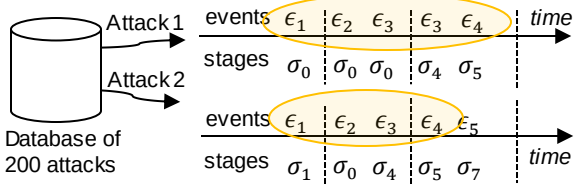
$10 + 10 \times 10 + 10 \times 10 + 10 \times 10 + 10 \times 10 = \mathbf{410}$

Modeling the credential stealing attack using Factor Graphs - Data

State space of variables
Attack stage: $X = \{\sigma_0, \sigma_1, \dots, \sigma_7\}$
(Observed) Events: $E = \{\epsilon_1, \dots, \epsilon_5\}$

OFFLINE ANNOTATION ON PAST ATTACKS

a) Annotated events and attack stages in a pair of attacks



b) Event-stage annotation table for the attack pair (Attack 1 and Attack 2)

Event	Attack stage
$\{\epsilon_1\}$	$\{\sigma_0 \mid \sigma_1\}$
$\{\epsilon_2\}$	$\{\sigma_0\}$
$\{\epsilon_3\}$	$\{\sigma_4\}$
$\{\epsilon_4\}$	$\{\sigma_5\}$
$\{\epsilon_5\}$	$\{\sigma_7\}$

ϵ_1	vulnerability scan	σ_0	benign
ϵ_2	login	σ_1	discovery
ϵ_3	sensitive_uri	σ_4	privilege escalation
ϵ_4	new _library	σ_5	persistence

- **Attack Information**
  - Multi-stage credential stealing attack
  - Attack stage $\sigma \in X$ is not observed; however an attack happens in a chain of exploits, thus we have a sequence of events
  - Each security event is a known variable ϵ, each takes value from a discrete set of events E
- **Problem statement.** Given a set of security events, infer whether an attack is in progress?
  - Goal is to detect and pre-empt the attack
- **Model assumptions**
  - There are multivariate relationships among the events
  - There is no restriction on order of the relationships (can be non-causal or correlation based)
- Markov Model and Bayesian Networks may not reflect the full range in this scenarios
- Factor graphs can be used for modeling highly complex attacks, where the causal

Modeling the credential stealing attack using Factor Graphs

OFFLINE LEARNING OF FACTOR FUNCTIONS

Example patterns, stages, probabilities, and significance learned from the attack pair

Pattern	Attack stages	Probability in past attacks	Significance (p-value)
$[\epsilon_1, \epsilon_3, \epsilon_4]$	$[\sigma_1, \sigma_4, \sigma_5]$	q_a	p_a
$[\epsilon_1]$	$[\sigma_0 \mid \sigma_1]$	q_b	p_b

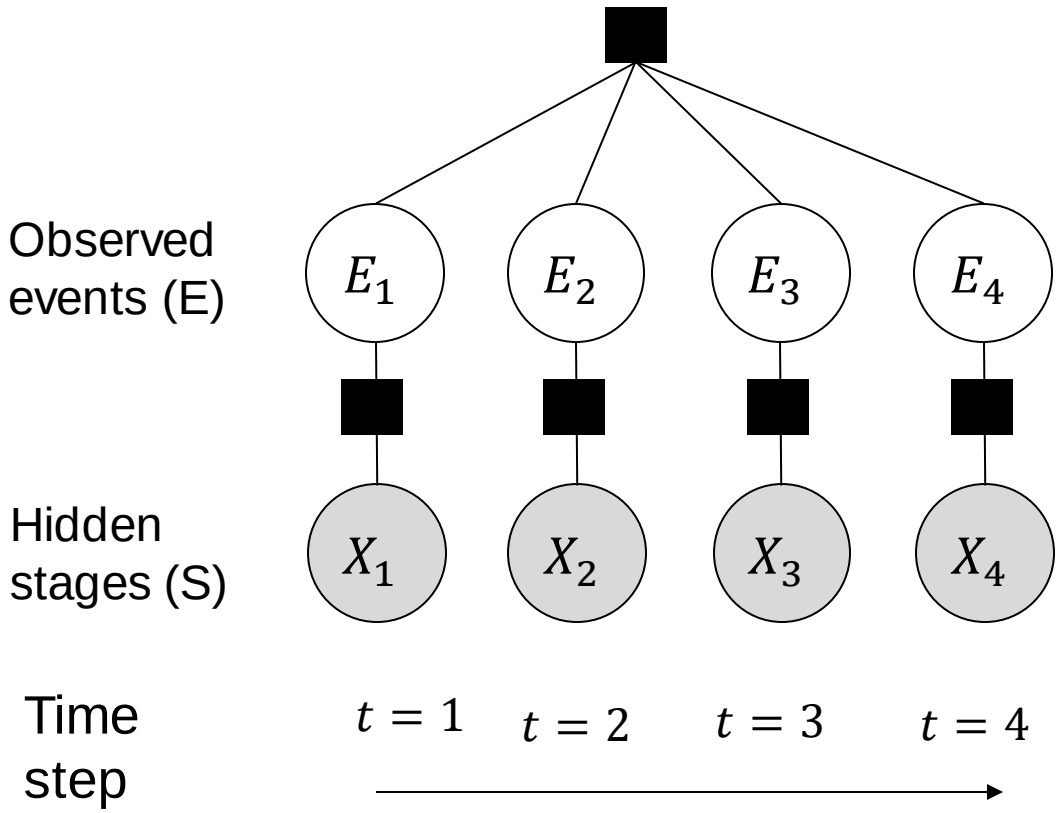
...

$$f(E) = \exp\{q_E(1 - p_E)\}$$

A factor function defined on the learned pattern, stages, and its significance

DETECTION OF UNSEEN ATTACKS

Factor Graph



Advantages and Disadvantages of Factor Graph

Advantage

- Factor graph subsumes HMMs, Markov Random Fields, Bayesian Networks etc.

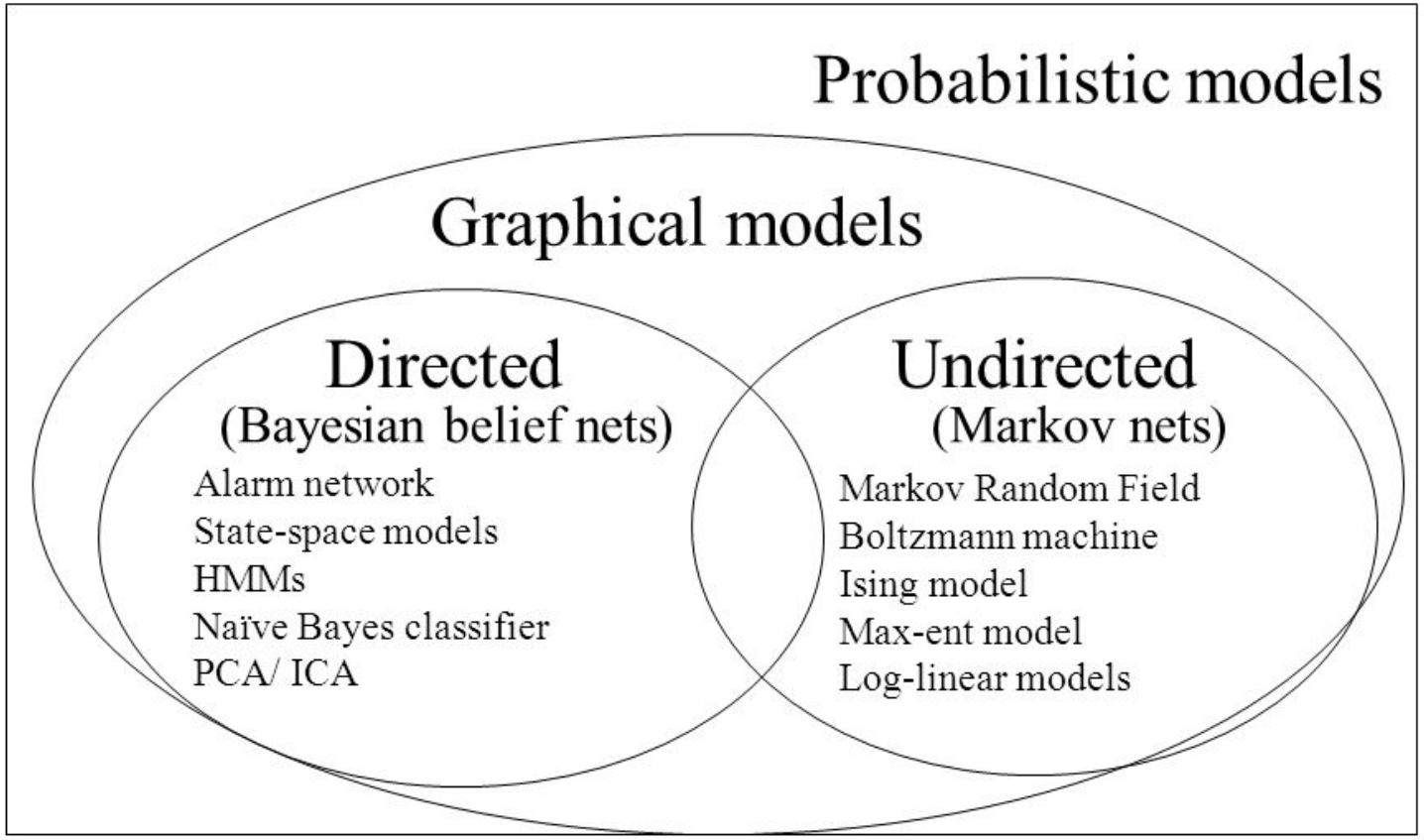
Disadvantage

- Limitations of probabilistic graphs in general

Comment

- If the problem is well represented by specific models such as Bayesian Networks, HMMs, Naïve Bayes or other graphical models then there is no need to generalize your problem as a factor graphs

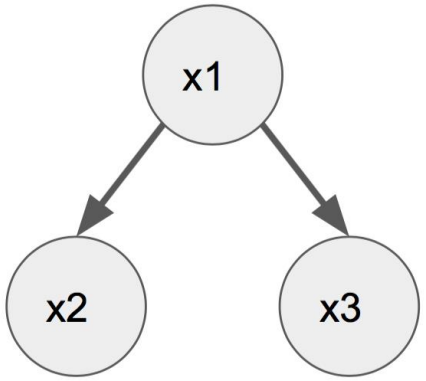
Taxonomy of Graphical Models



Machine Learning, A Probabilistic Perspective, Kevin Murphy, MIT Press

Bayesian Networks vs. Hidden Markov Models vs. Factor Graphs

Bayesian Network

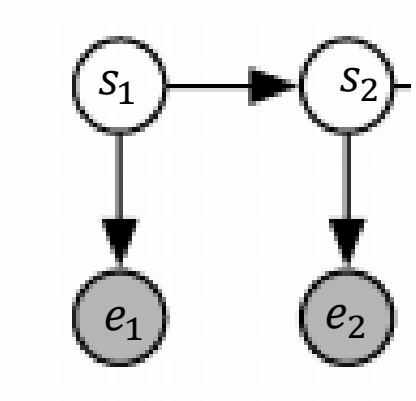


$$p(x_1)p(x_2|x_1)p(x_3|x_1)$$

Product of conditional probabilities

Causal relationships

Hidden Markov Model

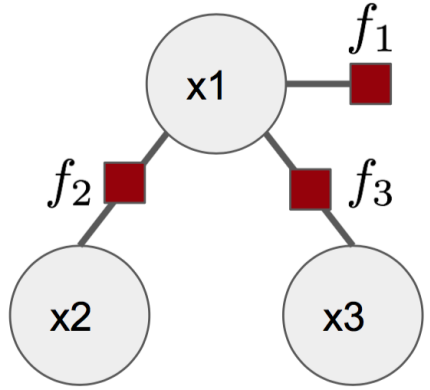


$$p(s_1)p(e_1|s_1)p(s_2|s_1)p(e_2|s_2)$$

Product of Temporal dependencies among variable

Temporal and statistical dependencies

Factor Graph



$$\frac{1}{Z} f_1(x_1) f_2(x_2, x_1) f_3(x_1, x_3)$$

Product of dependencies using univariate, bivariate, or multivariate functions

Both types of relations (including prior on a variable)

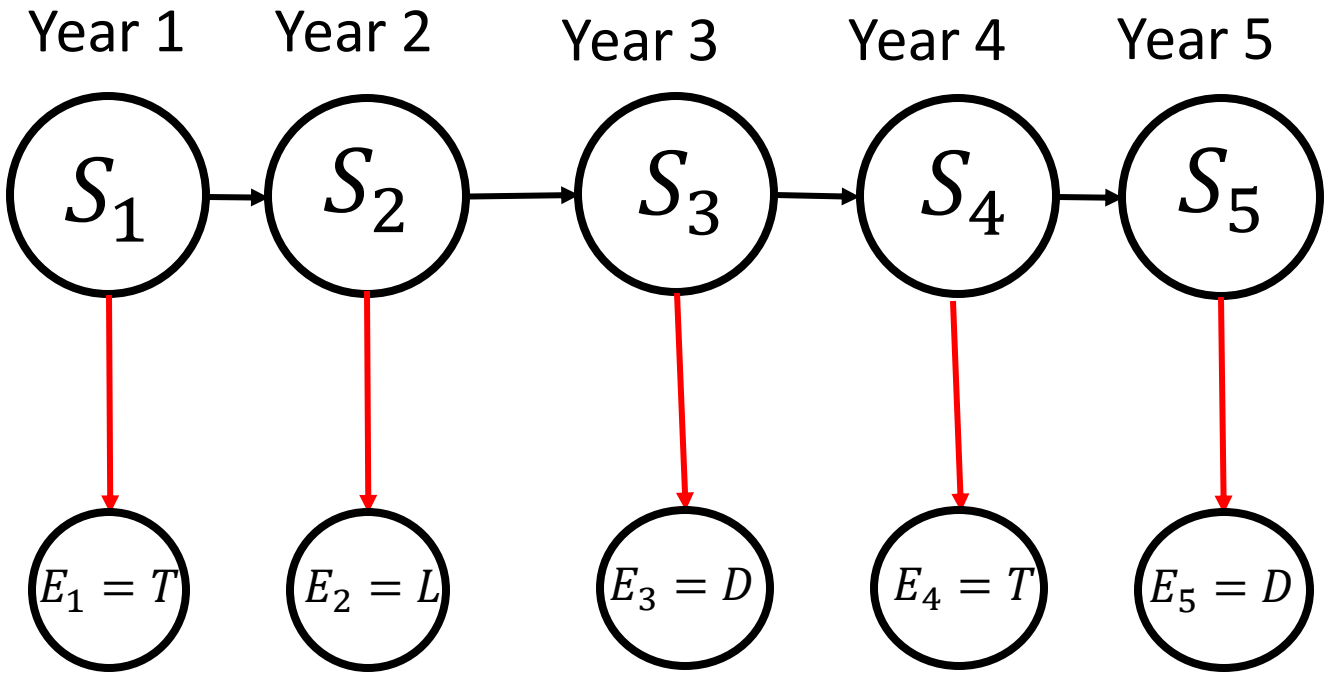
Practice with Factor Graph

HMM Example - Paleontological Temperature Model

- State space of hidden states: $S = \{H, C\}$
- State space of observations: $E = \{T, D, L\}$

- Transition probability matrix: A
- Observation Matrix: B
- Initial distribution for the hidden states: π

Given by an oracle

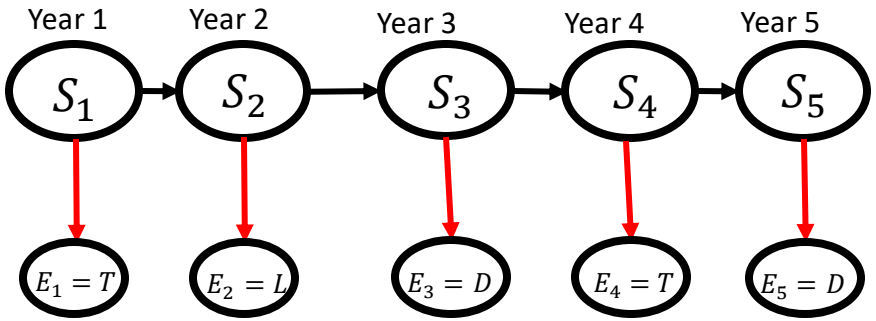


$$A = \begin{array}{c} \\ H \\ C \end{array} \begin{array}{c} \begin{array}{cc} H & C \end{array} \\ \begin{bmatrix} 0.7 & 0.3 \\ 0.4 & 0.6 \end{bmatrix} \end{array}$$

$$B = \begin{array}{c} \\ H \\ C \end{array} \begin{array}{c} \begin{array}{ccc} T & D & L \end{array} \\ \begin{bmatrix} 0.1 & 0.4 & 0.5 \\ 0.7 & 0.2 & 0.1 \end{bmatrix} \end{array}$$

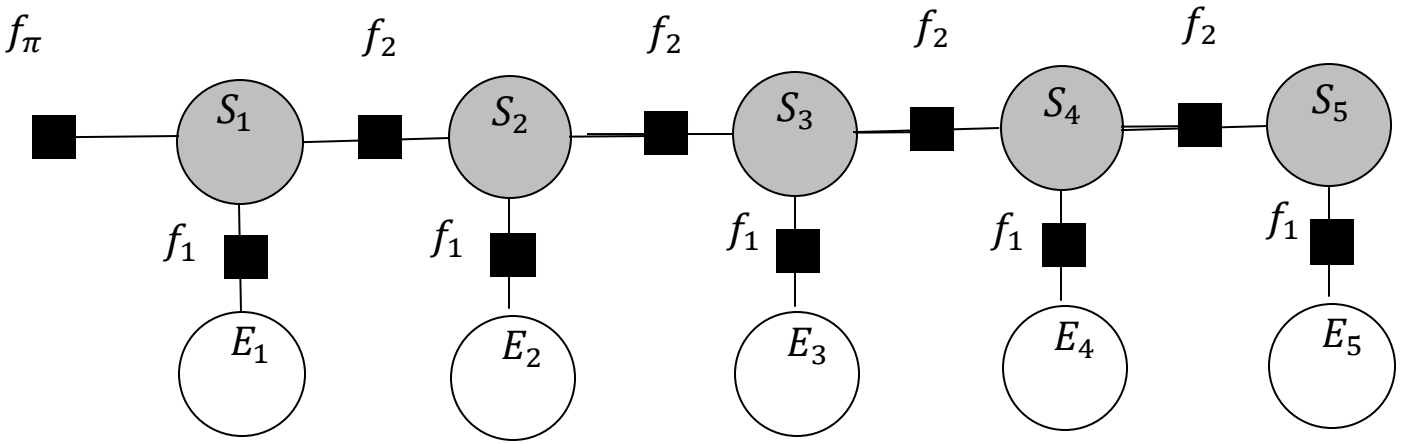
$$\pi = \begin{array}{c} \begin{array}{cc} H & C \end{array} \\ \begin{bmatrix} 0.5 & 0.5 \end{bmatrix} \end{array}$$

First Step – Drawing Factor Graph from HMM



Hidden States

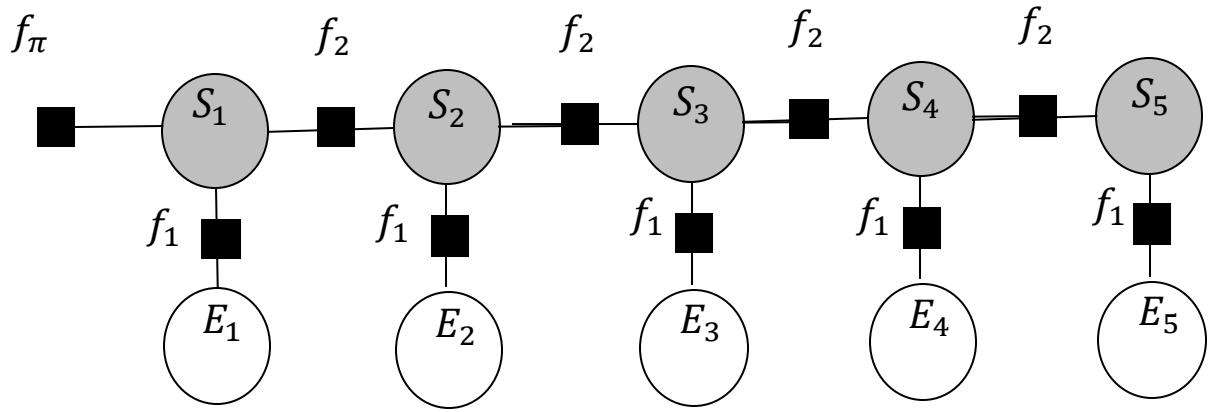
Observed Events



Why are the factor functions…

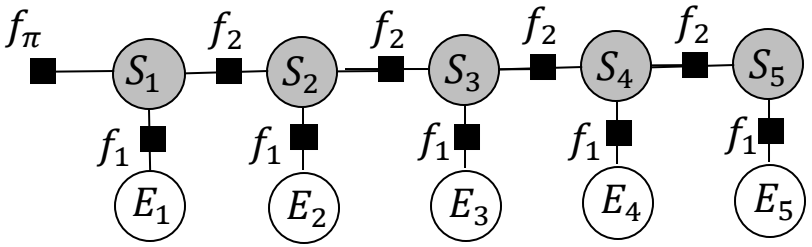
- **Between every pair of states the same?**
- **Between every pair of state and observation the same?**

Next – Figuring out the Factor Functions



- f_π needs to capture the prior probabilities for the states
- f_1 needs to capture the affinity between observations and states
- f_2 needs to capture the affinity between consecutive states

Next – Figuring out the Factor Functions

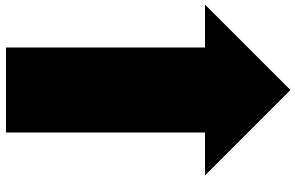


- f_π needs to capture the prior probabilities for the states

H C

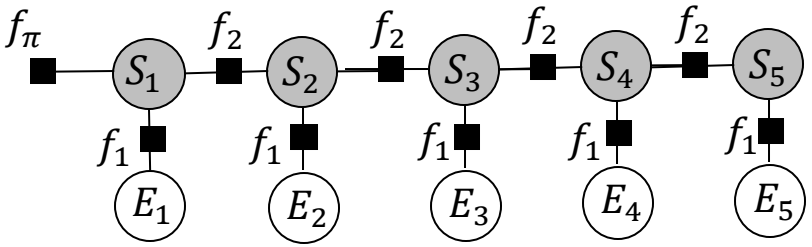
$[0.5\ 0.5]$

π



S_1	$f_\pi(S_1)$
H	0.5
C	0.5

Next – Figuring out the Factor Functions



- f_1 needs to capture the affinity between observations and states. (i.e., $P(E_i|S_i)$)

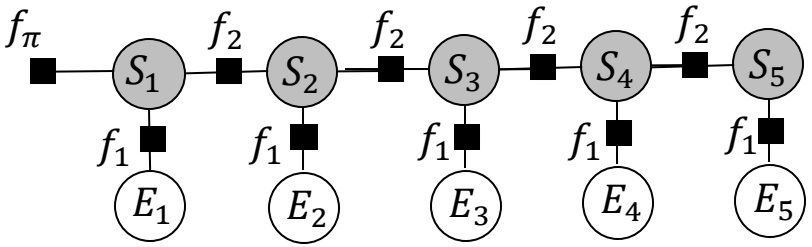
	T	D	L
H	0.1	0.4	0.5
C	0.7	0.2	0.1

B

S_i	E_i	$f_1(S_i, E_i)$
H	T	0.1
	D	0.4
	L	0.5
C	T	0.7
	D	0.2
	L	0.1

Note that the values in this table don't sum to 1 => f_1 is not a joint probability but a conditional probability!

Next – Figuring out the Factor Functions



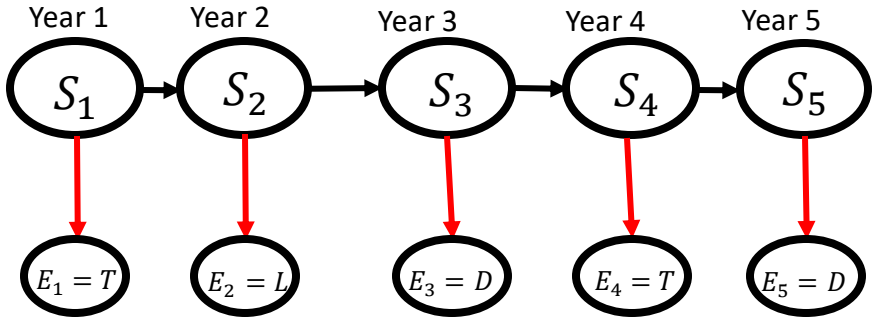
- f_2 needs to capture the affinity between consecutive states. (i.e., $P(S_{i+1}|S_i)$)

$$A = \begin{array}{c} \\ H \\ C \end{array} \begin{array}{c} \begin{array}{cc} H & C \end{array} \\ \begin{bmatrix} 0.7 & 0.3 \\ 0.4 & 0.6 \end{bmatrix} \end{array}$$

S_i	S_{i+1}	$f_2(S_i, S_{i+1})$
H	H	0.7
	C	0.3
C	H	0.4
	C	0.6

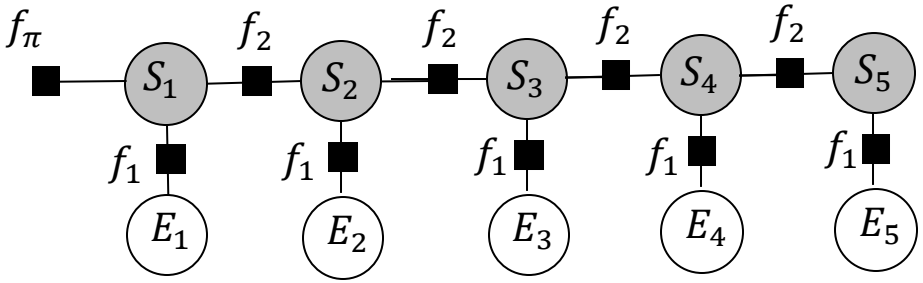
Note that the values in this table don't sum to 1 => f_2 is not a joint probability but a conditional probability!

After That – Calculating the Joint



$$P(S_1, \dots, S_5, E_1, \dots, E_5) =$$

$$P(S_1)P(E_1|S_1)\prod_{i=2}^{5} P(S_i|S_{i-1})P(E_i|S_i)$$



$$P(S_1, \dots, S_5, E_1, \dots, E_5) =$$

$$\frac{1}{Z} f_\pi(S_1) f_1(S_1, E_1) \prod_{i=2}^{5} f_2(S_{i-1}, S_i) f_1(S_i, E_i)$$

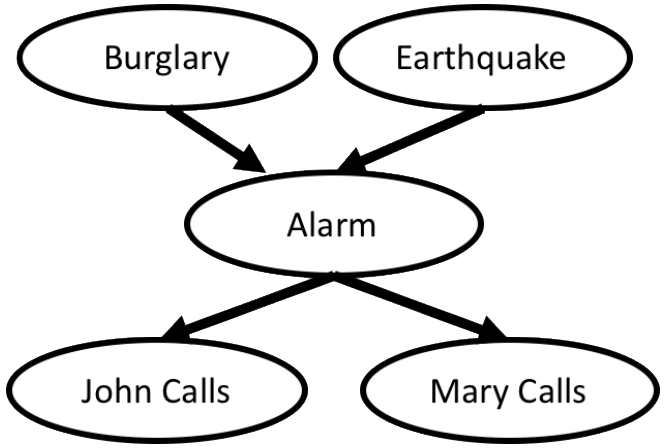
$$Z = \sum_{S_i \in \{H,C\}, E_i \in \{T,D,L\}} f_\pi(S_1) f_1(S_1, E_1) \prod_{i=2}^{5} f_2(S_{i-1}, S_i) f_1(S_i, E_i)$$

Belief Propagation

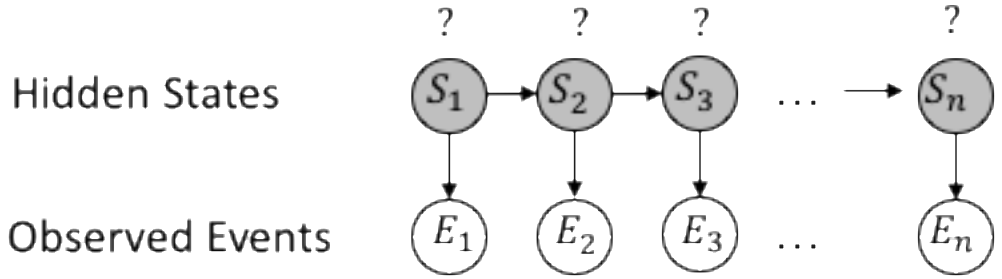
Inference on Graphical Models

Problems we have already looked at:

Bayes Network



Hidden Markov Model



Calculate the joint probability $P(B, J, A, E, M)$
$= P(J|A)\ P(M|A)P(A|B, E)P(B)P(E)$

Calculate the state probability $P(B) = \sum_{J,A,E,M} P(B, J, A, E, M)$

Calculate the conditional distribution $P(S_t|E_1, \dots, E_n)$

Factorize (Bayes Theorem)

$\propto P(S_t|E_1, \dots, E_t) * P(E_{t+1}, \dots, E_n|S_t, E_1, \dots, E_t)$

Use the Markov Property

$= P(S_t|E_1, \dots, E_t) * P(E_{t+1}, \dots, E_n|S_t)$

Forward Backward Algorithm

$= \alpha_t \odot \beta_t$

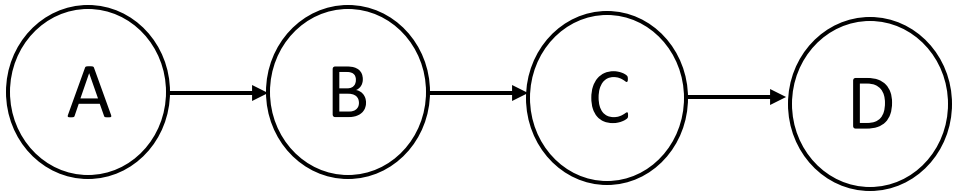
Just involves computation of joint distributions and its marginalization

Example of inference on a Bayesian Network

Consider the following Bayesian Network

- $A \in \{a^1, a^2\}, B \in \{b^1, b^2\}, C \in \{c^1, c^2\}, D \in \{d^1, d^2\}$

Inference task: Compute P(D)



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

- Simple way would be to generate each possible sequence (A,B,C,D) and sum over them
  - Exponential in the number of variables

Example of inference on Bayesian Network

- Enumerating all combinations
- Each term has 3 multiplications
- 8+8 = 16 terms
- Total multiplication ops = 16x3 = 48

- 7 additions for d^1 and 7 additions for d^2
- Total additions ops = 7+7=14

$$
\begin{aligned}
& P(a^1)\ P(b^1 \mid a^1)\ P(c^1 \mid b^1)\ P(d^1 \mid c^1) \\
+\ & P(a^2)\ P(b^1 \mid a^2)\ P(c^1 \mid b^1)\ P(d^1 \mid c^1) \\
+\ & P(a^1)\ P(b^2 \mid a^1)\ P(c^1 \mid b^2)\ P(d^1 \mid c^1) \\
+\ & P(a^2)\ P(b^2 \mid a^2)\ P(c^1 \mid b^2)\ P(d^1 \mid c^1) \\
+\ & P(a^1)\ P(b^1 \mid a^1)\ P(c^2 \mid b^1)\ P(d^1 \mid c^2) \\
+\ & P(a^2)\ P(b^1 \mid a^2)\ P(c^2 \mid b^1)\ P(d^1 \mid c^2) \\
+\ & P(a^1)\ P(b^2 \mid a^1)\ P(c^2 \mid b^2)\ P(d^1 \mid c^2) \\
+\ & P(a^2)\ P(b^2 \mid a^2)\ P(c^2 \mid b^2)\ P(d^1 \mid c^2)
\end{aligned}
$$

$$
\begin{aligned}
& P(a^1)\ P(b^1 \mid a^1)\ P(c^1 \mid b^1)\ P(d^2 \mid c^1) \\
+\ & P(a^2)\ P(b^1 \mid a^2)\ P(c^1 \mid b^1)\ P(d^2 \mid c^1) \\
+\ & P(a^1)\ P(b^2 \mid a^1)\ P(c^1 \mid b^2)\ P(d^2 \mid c^1) \\
+\ & P(a^2)\ P(b^2 \mid a^2)\ P(c^1 \mid b^2)\ P(d^2 \mid c^1) \\
+\ & P(a^1)\ P(b^1 \mid a^1)\ P(c^2 \mid b^1)\ P(d^2 \mid c^2) \\
+\ & P(a^2)\ P(b^1 \mid a^2)\ P(c^2 \mid b^1)\ P(d^2 \mid c^2) \\
+\ & P(a^1)\ P(b^2 \mid a^1)\ P(c^2 \mid b^2)\ P(d^2 \mid c^2) \\
+\ & P(a^2)\ P(b^2 \mid a^2)\ P(c^2 \mid b^2)\ P(d^2 \mid c^2)
\end{aligned}
$$

All terms involved in computation of $P(d^1)$ and $P(d^2)$ respectively.

Example of inference on Bayesian Network

Can we reduce the number of computations?

- Many terms are common; they can be computed once and reused

Consider the orange highlighted box, $P(c^1|b^1)P(d^1|c^1)$ is common.

Compute: $P(a^1)P(b^1|a^1) + P(a^2)P(b^1|a^2)$

Consider the blue highlighted box, $P(c^1|b^2)P(d^1|c^1)$ is common.

Compute: $P(a^1)P(b^2|a^1) + P(a^2)P(b^2|a^2)$

Define: $\tau_1(B) = P(a^1)P(B|a^1) + P(a^2)P(B|a^2)$

where $B \in \{b^1, b^2\}$

$$
\begin{array}{llll}
 & P(a^1) & P(b^1 \mid a^1) & P(c^1 \mid b^1) & P(d^1 \mid c^1) \\
+ & P(a^2) & P(b^1 \mid a^2) & P(c^1 \mid b^1) & P(d^1 \mid c^1) \\
+ & P(a^1) & P(b^2 \mid a^1) & P(c^1 \mid b^2) & P(d^1 \mid c^1) \\
+ & P(a^2) & P(b^2 \mid a^2) & P(c^1 \mid b^2) & P(d^1 \mid c^1) \\
+ & P(a^1) & P(b^1 \mid a^1) & P(c^2 \mid b^1) & P(d^1 \mid c^2) \\
+ & P(a^2) & P(b^1 \mid a^2) & P(c^2 \mid b^1) & P(d^1 \mid c^2) \\
+ & P(a^1) & P(b^2 \mid a^1) & P(c^2 \mid b^2) & P(d^1 \mid c^2) \\
+ & P(a^2) & P(b^2 \mid a^2) & P(c^2 \mid b^2) & P(d^1 \mid c^2)
\end{array}
$$

$$
\begin{array}{llll}
 & P(a^1) & P(b^1 \mid a^1) & P(c^1 \mid b^1) & P(d^2 \mid c^1) \\
+ & P(a^2) & P(b^1 \mid a^2) & P(c^1 \mid b^1) & P(d^2 \mid c^1) \\
+ & P(a^1) & P(b^2 \mid a^1) & P(c^1 \mid b^2) & P(d^2 \mid c^1) \\
+ & P(a^2) & P(b^2 \mid a^2) & P(c^1 \mid b^2) & P(d^2 \mid c^1) \\
+ & P(a^1) & P(b^1 \mid a^1) & P(c^2 \mid b^1) & P(d^2 \mid c^2) \\
+ & P(a^2) & P(b^1 \mid a^2) & P(c^2 \mid b^1) & P(d^2 \mid c^2) \\
+ & P(a^1) & P(b^2 \mid a^1) & P(c^2 \mid b^2) & P(d^2 \mid c^2) \\
+ & P(a^2) & P(b^2 \mid a^2) & P(c^2 \mid b^2) & P(d^2 \mid c^2)
\end{array}
$$

All terms involved in computation of $P(d^1)$ and $P(d^2)$ respectively.

Example of inference on Bayesian Network

Consider the orange highlighted box, $P(d^1|c^1)$ is common.

Compute: $\tau_1(b^1)P(c^1|b^1) + \tau_1(b^2)P(c^1|b^2)$

Consider the blue highlighted box, $P(d^1|c^2)$ is common.

Compute: $\tau_1(b^1)P(c^2|b^1) + \tau_1(b^2)P(c^2|b^2)$

Define: $\tau_2(C) = \tau_1(b^1)P(C|b^1) + \tau_1(b^2)P(C|b^2)$

where $C \in \{c^1, c^2\}$

$$
\begin{array}{lll}
 & \tau_1(b^1) & P(c^1 \mid b^1) \quad P(d^1 \mid c^1) \\
+ & \tau_1(b^2) & P(c^1 \mid b^2) \quad P(d^1 \mid c^1) \\
+ & \tau_1(b^1) & P(c^2 \mid b^1) \quad P(d^1 \mid c^2) \\
+ & \tau_1(b^2) & P(c^2 \mid b^2) \quad P(d^1 \mid c^2)
\end{array}
$$

$$
\begin{array}{lll}
 & \tau_1(b^1) & P(c^1 \mid b^1) \quad P(d^2 \mid c^1) \\
+ & \tau_1(b^2) & P(c^1 \mid b^2) \quad P(d^2 \mid c^1) \\
+ & \tau_1(b^1) & P(c^2 \mid b^1) \quad P(d^2 \mid c^2) \\
+ & \tau_1(b^2) & P(c^2 \mid b^2) \quad P(d^2 \mid c^2)
\end{array}
$$

All terms involved in computation of $P(d^1)$ and $P(d^2)$ respectively. The sum is simplified because of use of $\tau_1(B)$.

Example of inference on Bayesian Network

- Computation shown alongside is easy and gives $P(D)$
- Previous steps are equivalent to **pushing summation inside**

$$P(D) = \sum_C \sum_B \sum_A P(A)P(B|A)P(C|B)P(D|C)$$

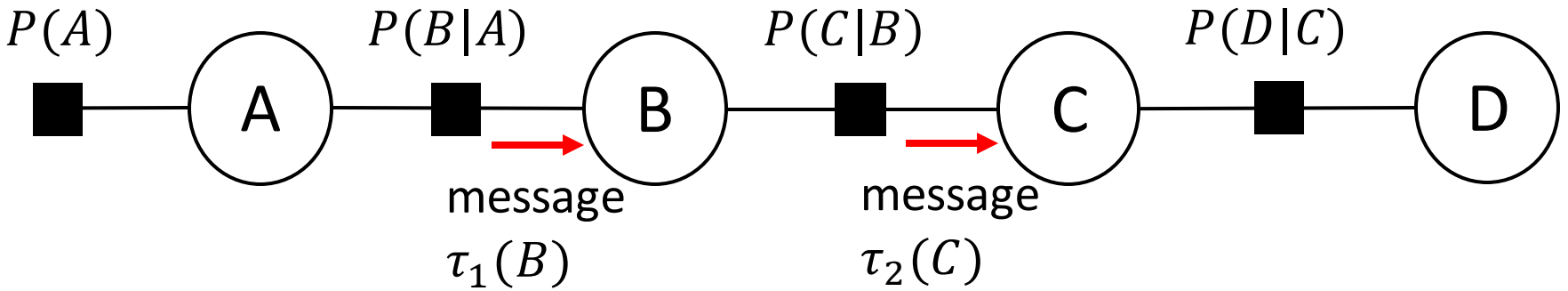
$$P(D) = \sum_C P(D|C) \overbrace{\sum_B P(C|B) \underbrace{\sum_A P(A)P(B|A)}_{\tau_1(B)}}^{\tau_2(C)}$$

$$\begin{array}{rll} & \tau_2(c^1) & P(d^1 \mid c^1) \\ + & \tau_2(c^2) & P(d^1 \mid c^2) \end{array}$$

$$\begin{array}{rll} & \tau_2(c^1) & P(d^2 \mid c^1) \\ + & \tau_2(c^2) & P(d^2 \mid c^2) \end{array}$$

Computation of $P(D)$ is simplified because of use of $\tau_1(B)$, $\tau_2(C)$.

Flow of computations (messages) in Factor Graph corresponding to the given BN



Sum-product algorithm reduces computations

- Pushing summations inside reduced the number of computations
  - Simple way: 48 multiplications + 14 additions
  - Pushing summations inside: 4x3 multiplications + 2x3 additions
  - Can be up to linear in number of variables (much better than exponential!)

- What helped in addressing the exponential blowup of marginalizing the joint distribution?
  - Graph structure – because of structure of Bayesian Network, some subexpression in the joint depend only on a small number of variables
  - Pushing summation inside – by computing these expressions once and caching the results, we can avoid generating them exponentially many times

- Referred to as **sum-product algorithm** or **Belief Propagation**

Inference problem on Factor Graphs

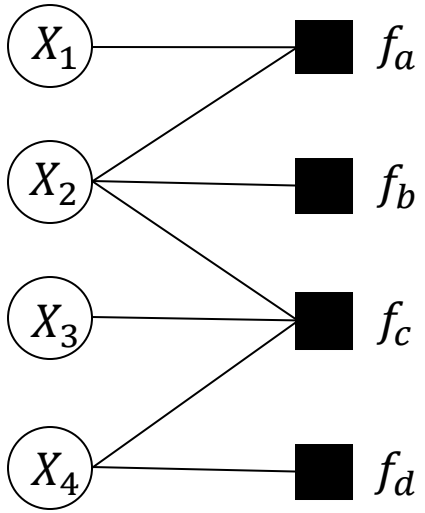
What is the problem we are trying to solve?

- Marginalization on Factor Graphs

Marginal probability

$$P(X_i) = \sum_{\boldsymbol{X} \setminus X_i} P(\boldsymbol{X})$$

All variables except X_i



Challenge:

- Computationally expensive because the sum is calculated on all variables except one

Approach:

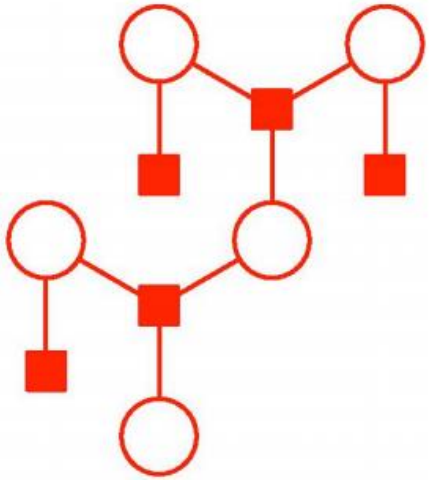
- Factorize the joint distribution according the structure
- Use belief propagation to reduce computations

$X_i \in \{0,1\}$ is a discrete variable
e.g., a Boolean variable

$f_c(X_c)$ is a tensor
on a set of variables X_c

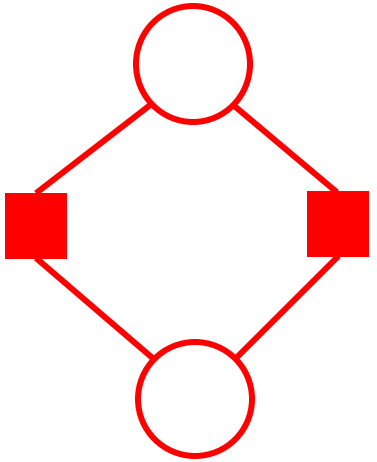
Belief propagation

- Also known as sum-product algorithm
- Computes marginal distributions by “pushing in summations”
- Exact inference for linear graphs and trees
- Approximate inference for graphs with loops; performs remarkably well



Tree Factor graph

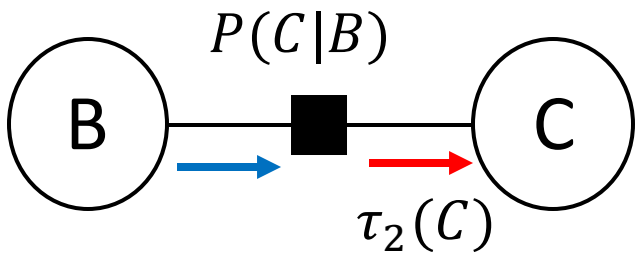
- In case of Factor Graphs, involves two types of **messages**
  - From factor to variables
  - From variables to factors



Factor graph with loop

Belief Propagation – Message from factor to variable

Recall from previous example:

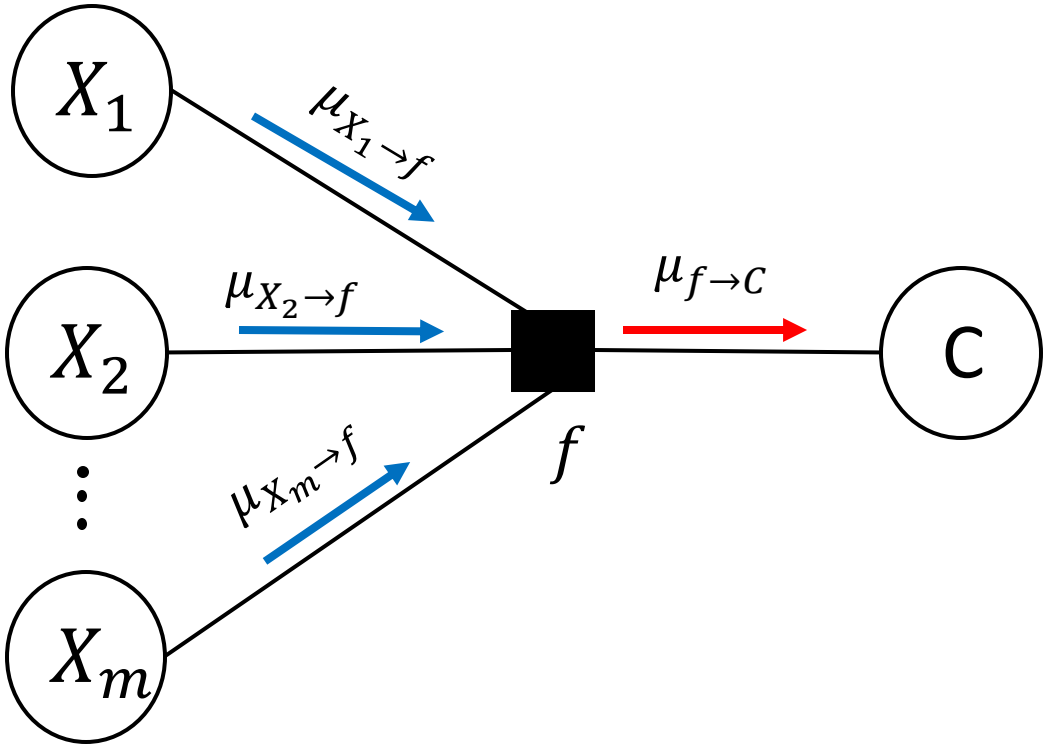


$$\tau_2(C) = \sum_B P(C|B)\tau_1(B)$$

To get the general expression, denote by:

$$f(B,C) = P(C|B)$$
$$\mu_{f\to C}(C) = \tau_2(C)$$
$$\mu_{B\to f}(B) = \tau_1(B)$$

In general:

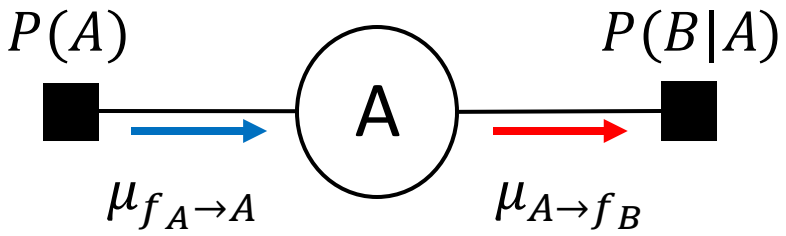


$$\mu_{f\to C}(C) = \sum_{X_1,X_2,\ldots,X_m} f(C,X_1,\ldots,X_m) \prod_{i=1}^{m} \mu_{X_i\to f}(X_i)$$

Message from factor to variable: Product of all incoming messages and factor, sum out previous variables

Belief Propagation – Message variable to factor

Recall from previous example:

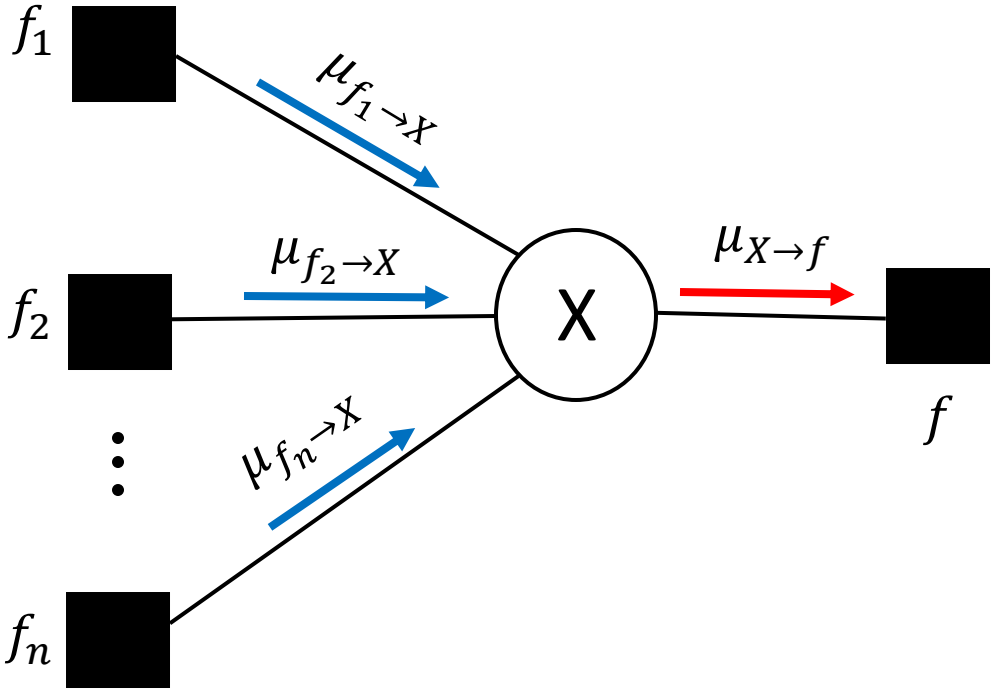


$$\mu_{A \to f_B}(A) = \mu_{f_A \to A}(A) = P(A)$$

Where,

$$f_A(A) = P(A)$$
$$f_B(A, B) = P(B|A)$$

In general:



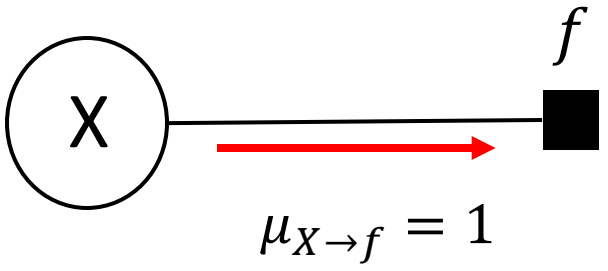
$$\mu_{X \to f}(X) = \prod_{i=1}^{n} \mu_{f_i \to X}(X)$$

Message from variable to factor: Product of all incoming messages

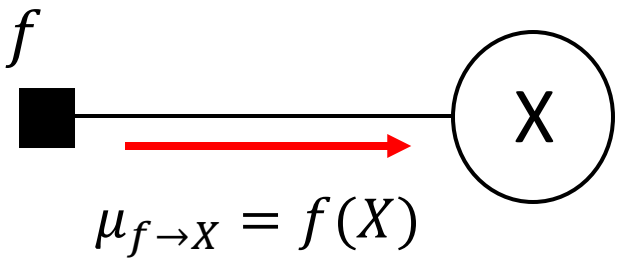
Belief Propagation: General Algorithm

Steps to compute marginal distribution for all variables

- How to start the algorithm
  - Choose a node in the factor graph as root node
  - Compute all the leaf-to-root messages
  - Compute all the root-to-leaf messages



- Initial Conditions
  - Starting from a factor leaf/root node, the initial factor-to-variable message is the factor itself
  - Starting from a variable leaf/root node, the initial variable-to-factor message is a vector of ones

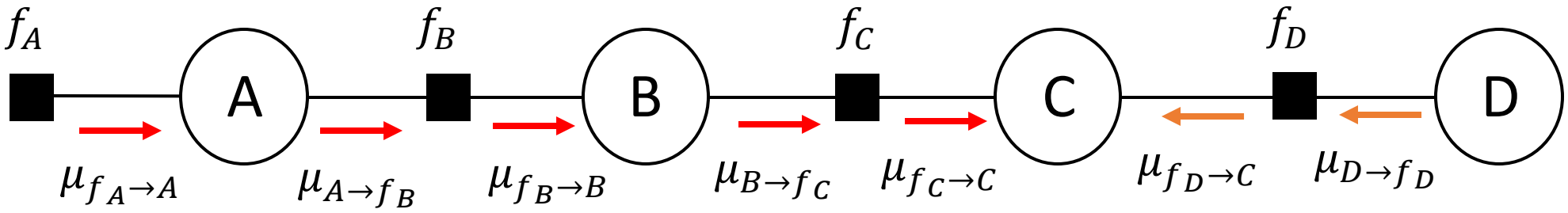


Initialization

- Computing marginals
  - Marginal is given by the product of all incoming messages; normalize if necessary

Example of belief propagation

Compute $P(C)$



$$\mu_{f_A \to A}(A) = f_A(A) = P(A)$$

$$\mu_{A \to f_B}(A) = \mu_{f_A \to A}(A) = P(A)$$

$$\mu_{f_B \to B}(B) = \sum_A f_B(A,B)\, \mu_{f_A \to A}(A)$$

$$= \sum_A P(B|A)\, P(A)$$

$$\mu_{B \to f_C}(B) = \mu_{f_B \to B}(B)$$

$$= \sum_A P(B|A)\, P(A)$$

$$\mu_{f_C \to C}(C) = \sum_B f_C(B,C)\, \mu_{B \to f_B}(B)$$

$$= \sum_B P(C|B) \sum_A P(B|A) P(A)$$

$$\mu_{D \to f_D}(D) = 1$$

$$\mu_{f_D \to C}(C)$$

$$= \sum_D f_D(C,D)\, \mu_{f_D \to C}(C)$$

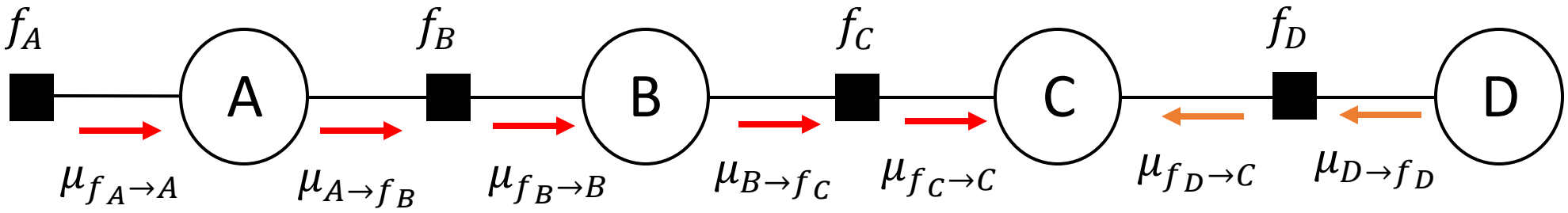
$$= \sum_D P(D|C)$$

Example of belief propagation

$$\mu_{f_D \to C}(C) = \sum_D P(D|C)$$

$$\mu_{f_C \to C}(C) = \sum_B P(C|B) \sum_A P(B|A)P(A)$$

Compute $P(C)$



$$P(C) = \mu_{f_C \to C}(C)\mu_{f_D \to C}(C)$$

Verifying that the above computation gives the marginal distribution

$$P(C) = \left(\sum_B P(C|B) \sum_A P(B|A)P(A)\right)\left(\sum_D P(D|C)\right) = \sum_B P(C|B) \sum_A P(B|A)P(A) \sum_D P(D|C)$$

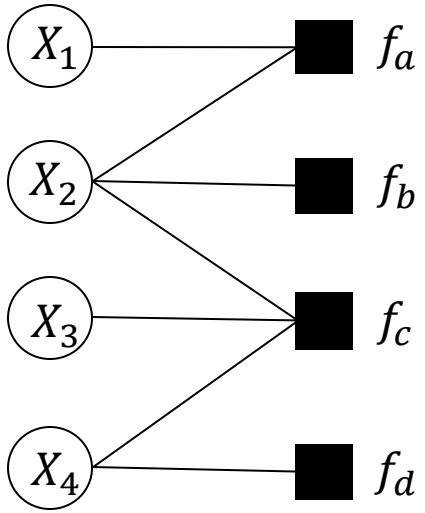
$$= \sum_A \sum_B \sum_D P(A)P(B|A)P(C|B)P(D|C) = \sum_{A,B,D} P(A,B,C,D)$$

Marginalization on Factor Graphs

Marginal probability

$$P(X_i) = \sum_{\boldsymbol{X} \setminus X_i} P(\boldsymbol{X})$$

All variables except X_i



Inference method	Description
Belief Propagation	Exact inference on non-loop FG
Sampling - Markov Chain Monte Carlo, Gibbs	Approximate inference
Variational Inference	Approximate inference

$X_i \in \{0,1\}$ is a discrete variable
e.g., a Boolean variable

$f_c(X_c)$ is a tensor
on a set of variables X_c

References

- Daphne Koller, Nir Friedman's textbook on Graphical Models
- https://www.doc.ic.ac.uk/~mpd37/teaching/ml_tutorials/2016-11-09-Svensson-BP.pdf