Probabilistic Graphical Models and Their Applications

Bernt Schiele

Max Planck Institute for Informatics

slides adapted from Peter Gehler

November 25, 2o2o

Today's Topics

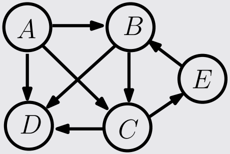
- Recap: Bayes Networks
- Markov Networks (slides from last time)
- Factor Graphs
- Inference
  - exact inference (trees)
  - sum-product algorithm

The story so far...

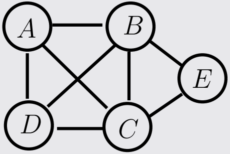
Graph Definitions

- A graph consists of *vertices* and *edges*

Graph

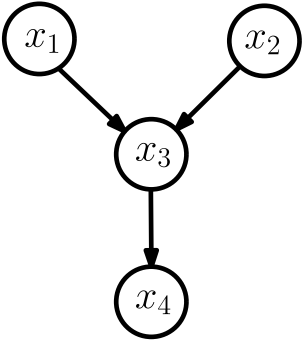


A directed graph – directed edges.
Bayesian Networks (or Belief Networks)



An undirected graph – undirected edges.
Markov random fields (or Markov Networks)

Belief Network: Example



$$p(x_1, x_2, x_3, x_4) = p(x_4|x_3)p(x_3|x_1, x_2)p(x_2)p(x_1)$$

Belief Networks Definition

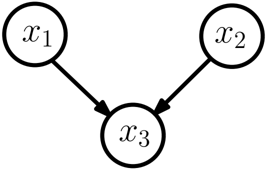
Belief network

A belief network is a distribution of the form

$$p(x_1, \ldots, x_D) = \prod_{i=1}^{D} p(x_i \mid pa(x_i)), \tag{1}$$

where $pa(x)$ denotes the parental variables of x

Collider and Conditional Independence



- x_3 a collider ? yes
- $x_1 \perp\!\!\!\perp x_2 \mid x_3$? no! (explaining away)

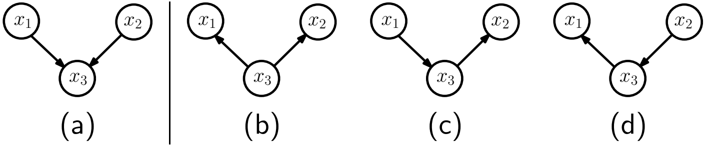
$$\begin{aligned} p(x_1, x_2 \mid x_3) &= p(x_1, x_2, x_3)/p(x_3) \\ &= p(x_1)p(x_2)\underbrace{p(x_3 \mid x_1, x_2)/p(x_3)}_{\neq 1 \text{ in general}} \end{aligned}$$

- $x_1 \perp\!\!\!\perp x_2$? yes

$$p(x_1, x_2) = \sum_{x_3} p(x_3 \mid x_1, x_2)p(x_1)p(x_2) = p(x_1)p(x_2)$$

Belief Networks

- Graphical Models specify a list of conditional independence statements
- We can use D-separation to test for conditional independence
- Some Networks look different but are Markov equivalent (b,c,d are Markov equivalent)



Markov Equivalence

Markov equivalence

Two graphs are Markov equivalent if they represent the same set of conditional independence statements. (holds for directed and undirected graphs)

skeleton

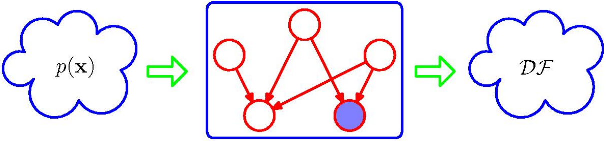
Graph resulting when removing all arrows of edges

immorality

Parents of a child with no connection

- Markov equivalent $\Leftrightarrow$ same skeleton and same set of immoralities

Filter View of a Graphical Model



- Graphical model implies a list of conditional independences
- Regard as filter:
  - only distributions that satisfy all conditional independences are allowed to pass
- One graph describes a whole family of probability distributions
- Extremes:
  - Fully connected, no constraints, all p pass
  - no connections, only product of marginals may pass

Markov Networks

Markov Networks

- So far, factorization with each factor a probability distribution
  - Normalization as a by-product
- Alternative:

$$p(a, b, c) = \frac{1}{Z}\phi(a, b)\phi(b, c) \tag{2}$$

- Here Z normalization constant or partition function

$$Z = \sum_{a,b,c} \phi(a, b)\phi(b, c) \tag{3}$$

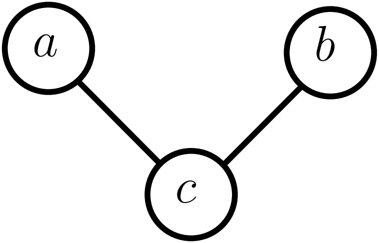
Definitions

Potential

A potential $\phi(x)$ is a non-negative function of the variable x. A joint potential $\phi(x_1, \ldots, x_D)$ is a non-negative function of the set of variables.

- Distribution (as in belief networks) is a special choice

Example



$$p(a,b,c) = \frac{1}{Z}\phi_{ac}(a,c)\phi_{bc}(b,c) \tag{4}$$

Markov Network

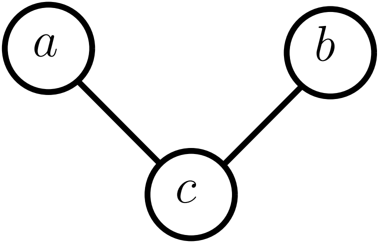
Markov Network

For a set of variables $\mathcal{X} = \{x_1, \ldots, x_D\}$ a Markov network is defined as a product of potentials over the maximal cliques $\mathcal{X}_c$ of the graph $\mathcal{G}$

$$p(x_1, \ldots, x_D) = \frac{1}{Z} \prod_{c=1}^{C} \phi_c(\mathcal{X}_c) \tag{5}$$

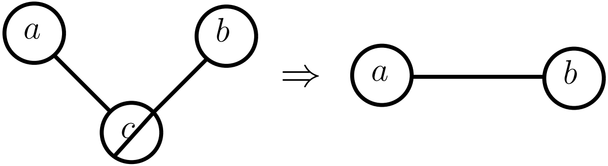
- Special case: cliques of size 2 – pairwise Markov network
- In case all potentials are strictly positive this is called a Gibbs distribution

Properties of Markov Networks



$$p(a, b, c) = \frac{1}{Z}\phi_{ac}(a, c)\phi_{bc}(b, c) \tag{6}$$

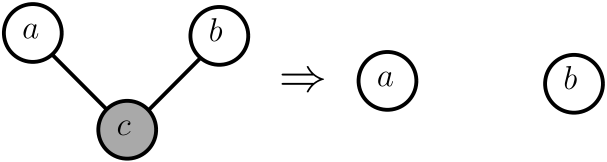
Properties of Markov Networks



- Marginalizing over c makes a and b "graphically" dependent

$$p(a,b) = \sum_c \frac{1}{Z}\phi_{ac}(a,c)\phi_{bc}(b,c) = \frac{1}{Z}\phi_{ab}(a,b) \quad (7)$$

Properties of Markov Networks



- Conditioning on c makes a and b independent

$$p(a, b \mid c) = p(a \mid c)p(b \mid c) \tag{8}$$

- This is opposite to the directed version $a \rightarrow c \leftarrow b$ where conditioning *introduced* dependency

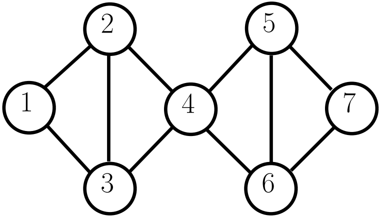
Local Markov Property

Local Markov Property

$$p(x \mid \mathcal{X} \setminus \{x\}) = p(x \mid ne(x)) \tag{9}$$

- Condition on neighbours independent on rest

Local Markov Property – Example



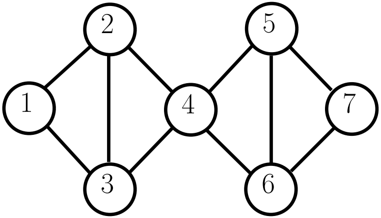
- $x_4 \perp\!\!\!\perp \{x_1, x_7\} \mid \{x_2, x_3, x_5, x_6\}$

Global Markov Property

Global Markov Property

For disjoint sets of variables $(\mathcal{A}, \mathcal{B}, \mathcal{S})$ where $\mathcal{S}$ separates $\mathcal{A}$ from $\mathcal{B}$, then $\mathcal{A} \perp\!\!\!\perp \mathcal{B} \mid \mathcal{S}$

Local Markov Property – Example

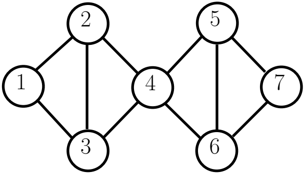


- $x_1 \perp\!\!\!\perp x_7 \mid \{x_4\}$
- and others

Hammersley-Clifford Theorem

- An undirected graph specifies a set of conditional independence statements
- Question: What is the most general factorization (of the joint distribution) that satisfies these independences?
- In other words: given the graph, what is the implied factorization?

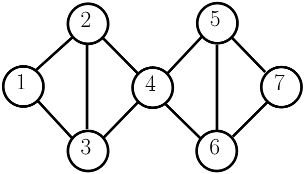
Finding the Factorization



- Eliminate variable one by one
- Let's start with x_1

$$p(x_1, \ldots, x_7) = p(x_1 \mid x_2, x_3)p(x_2, \ldots, x_7) \tag{10}$$

Finding the Factorization



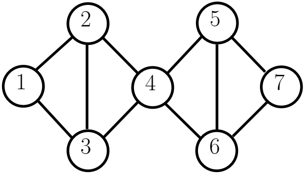
- Graph specifies:

$$
\begin{aligned}
& p(x_1, x_2, x_3 \mid x_4, \ldots, x_7) &= p(x_1, x_2, x_3 \mid x_4) \\
\Rightarrow \quad & p(x_2, x_3 \mid x_4, \ldots, x_7) &= p(x_2, x_3 \mid x_4)
\end{aligned}
$$

- Hence

$$
p(x_1, \ldots, x_7) = p(x_1 \mid x_2, x_3) p(x_2, x_3 \mid x_4) p(x_4, x_5, x_6, x_7)
$$

Finding the Factorization



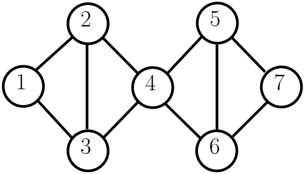
- We continue to find

$$p(x_1, \ldots, x_7) = p(x_1 \mid x_2, x_3)p(x_2, x_3 \mid x_4) \\ p(x_4 \mid x_5, x_6)p(x_5, x_6 \mid x_7)p(x_7)$$

- A factorization into clique potentials (maximal cliques)

$$p(x_1, \ldots, x_7) = \frac{1}{Z}\phi(x_1, x_2, x_3)\phi(x_2, x_3, x_4)\phi(x_4, x_5, x_6)\phi(x_5, x_6, x_7)$$

Finding the Factorization



- Markov conditions of graph $G \Rightarrow$ factorization F into clique potentials
- And conversely: $F \Rightarrow G$

Hammersley-Clifford Theorem

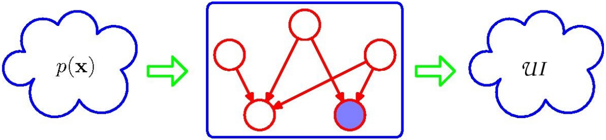
Hammersely-Clifford

This factorization property $G \Leftrightarrow F$ holds for any undirected graph provided that the potentials are positive

- Thus also loopy ones: $x_1 - x_2 - x_3 - x_4 - x_1$
- Theorem says, distribution is of the form

$$p(x_1, x_2, x_3, x_4) = \frac{1}{Z} \phi_{12}(x_1, x_2) \phi_{23}(x_2, x_3) \phi_{34}(x_3, x_4) \phi_{41}(x_4, x_1)$$

Filter View



- Let $\mathcal{UI}$ denote the distributions that can pass
  - those that satisfy all conditional independence statements
- Let $\mathcal{UF}$ denote the distributions with factorization over cliques
- Hammersley-Clifford says : $\mathcal{UI} = \mathcal{UF}$

Factor Graphs

Notation:

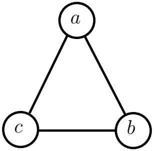
- for brevity in the following often $\phi_c(X_c) = \phi(X_c)$

Relationship Potentials to Graphs

- Consider

$$p(a,b,c) = \frac{1}{Z}\phi(a,b)\phi(b,c)\phi(c,a)$$

- What is the corresponding Markov network (graphical representation)?



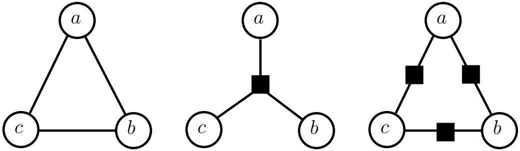
- and which other factorization is represented by this network?

$$p(a,b,c) = \frac{1}{Z}\phi(a,b,c)$$

- The factorization is not specified by the graph
- This is why we look at Factor Graphs

Relationship Potentials to Graphs

- Now consider we introduce an extra node (a square) for each factor

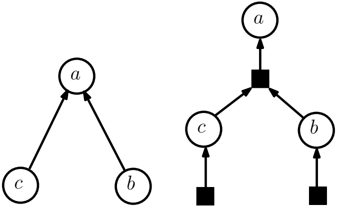


(1) (2) (3)

- (1): Markov Network
- (2): Factor graph representation of $\phi(a, b, c)$
- (3): Factor graph representation of $\phi(a, b)\phi(b, c)\phi(c, a)$
- Different factor graphs can have the same Markov network (2,3)⇒(1)

Similarly for Directed Graphs

- A directed factor graph also retains the structure of the factorization for a belief network



- But we skip those arrows usually

Factor Graph Definition

Factor Graph

Given a function

$$f(x_1, \dots, x_n) = \prod_i \psi_i(\mathcal{X}_i),$$

the factor graph (FG) has a node (represented by a square) for each factor $\psi_i(\mathcal{X}_i)$ and a variable node (represented by a circle) for each variable x_j.
When used to represent a distribution

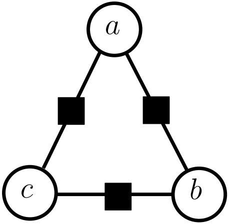
$$p(x_1, \dots, x_n) = \frac{1}{Z} \prod_i \psi_i(\mathcal{X}_i),$$

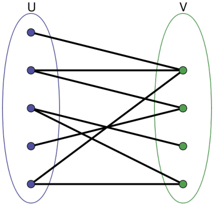
a normalization constant is assumed.

Bi-partite Graph

bipartite

A bipartite graph is a graph whose vertices can be divided into two disjoint sets U and V such that every edge connects a vertex in U to one in V

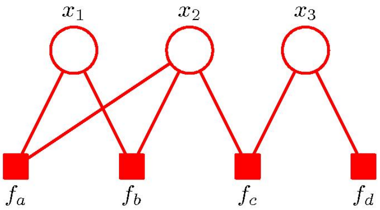




Factor graphs are bipartite graphs between variable nodes and factor nodes (see example next slide)

Factor Graph: Example 1

- Question: which distribution ?



- Answer:

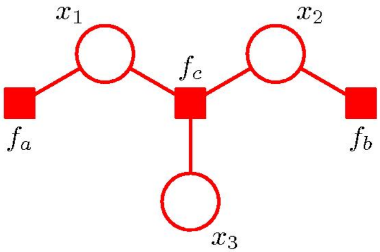
$$p(x) = \frac{1}{Z} f_a(x_1, x_2) f_b(x_1, x_2) f_c(x_2, x_3) f_d(x_3) \tag{11}$$

Factor Graph: Example 2

- Question: Which factor graph ?

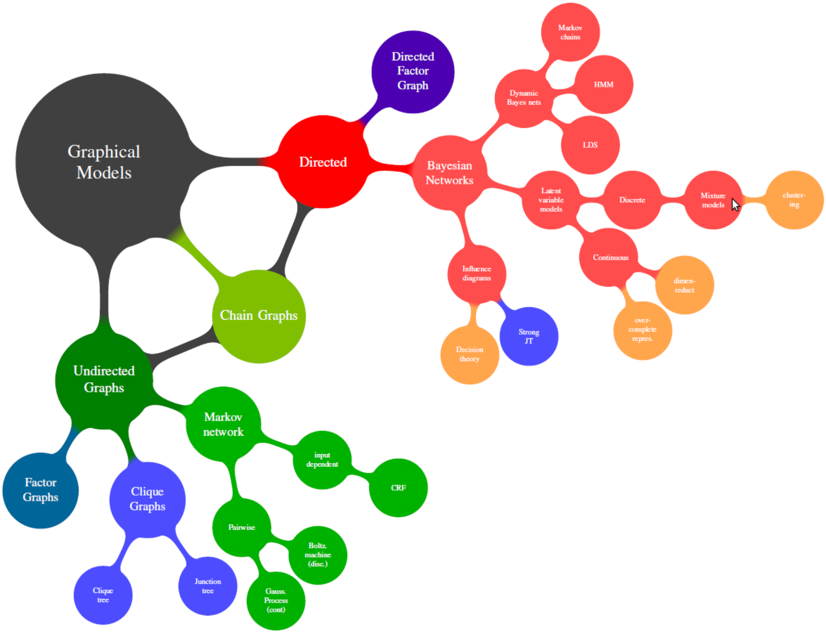
$$p(x_1, x_2, x_3) = p(x_1)p(x_2)p(x_3 \mid x_1, x_2) \tag{12}$$

- Answer:



Summary (so far)

- With graphical models we represent probability distributions graphically
- Belief networks: directed graphs, causal dependency
- Markov networks: undirected, local cliques of dependent variables
- Factor graphs
  - Making the factorization explicit
  - Not a larger class of distributions, "just" a different way of drawing the graph
- Always think in terms of factor graphs


Graphical Models
Directed
Directed Factor Graph
Bayesian Networks
Dynamic Bayes nets
Markov chains
HMM
LDS
Latent variable models
Discrete
Mixture models
cluster-ing
Continuous
dimen-reduct
over-complete repres.
Influence diagrams
Decision theory
Strong JT
Chain Graphs
Undirected Graphs
Markov network
input dependent
CRF
Pairwise
Boltz. machine (disc.)
Gauss. Process (cont)
Factor Graphs
Clique Graphs
Clique tree
Junction tree

Inference in Trees

Inference - what to infer?

- Given distribution

$$p(x) = p(x_1, \ldots, x_n) \tag{13}$$

- Inference: computing functions of the distribution, e.g.
  - mean
  - mode
  - marginal
  - conditionals

Inference - what to infer?

- Mean

$$\mathbb{E}_{p(x)}[x] = \sum_{x \in \mathcal{X}} x p(x)$$

- Mode (most likely state)

$$x^* = \operatorname*{argmax}_{x \in \mathcal{X}} p(x)$$

- Conditional Distributions

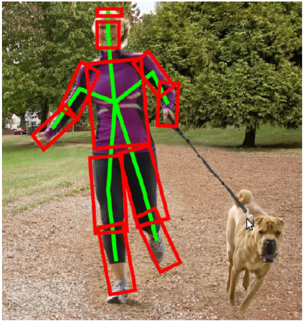
$$p(x_i, x_j \mid x_k, x_l) \quad \text{or} \quad p(x_i \mid x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$$

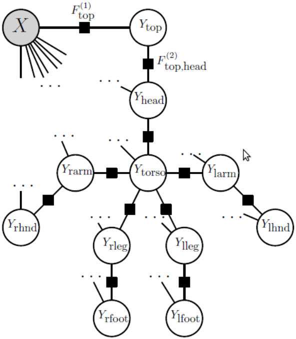
- Max-Marginals

$$x_i^* = \operatorname*{argmax}_{x_i \in \mathcal{X}_i} p(x_i) = \operatorname*{argmax}_{x_i \in \mathcal{X}_i} \sum_{(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)} p(x)$$

Example: Pictorial Structures

- Find body parts



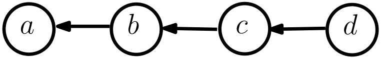


[Fischler& Elschlager, 1973],[Felsenzwalb& Huttenlocher, 2000]

Variable Elimination

In the following: marginal inference in singly-connected graphs (= trees):

- Consider Markov chain ($a, b, c, d \in \{0, 1\}$)

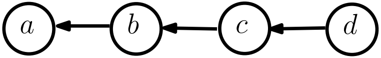


  with distribution

$$p(a, b, c, d) = p(a \mid b)p(b \mid c)p(c \mid d)p(d) \tag{14}$$

- Task: compute the marginal $p(a)$

Variable Elimination



$$p(a) = \sum_{b,c,d} p(a,b,c,d) \tag{15}$$
$$= \sum_{b,c,d} p(a \mid b)p(b \mid c)p(c \mid d)p(d) \tag{16}$$

- Naive: $2 \times 2 \times 2 = 8$ states to sum over (binary variables)
- Re-order summation:

$$p(a) = \sum_{b,c} p(a \mid b)p(b \mid c) \underbrace{\sum_{d} p(c \mid d)p(d)}_{\gamma_d(c)} \tag{17}$$

Variable Elimination

$$p(a) = \sum_{b,c} p(a \mid b) p(b \mid c) \underbrace{\sum_{d} p(c \mid d) p(d)}_{\gamma_d(c)}$$

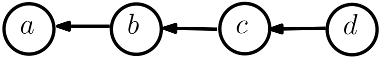
$$p(a) = \sum_{b} p(a \mid b) \underbrace{\sum_{c} p(b \mid c) \gamma_d(c)}_{\gamma_c(b)}$$

$$p(a) = \sum_{b} p(a \mid b) \gamma_c(b)$$

- We need $2 + 2 + 2 = 6$ calculations (binary variables)
- For a chain of length n scales linearly $n * 2$
  (cf naive approach 2^n)

Finding Conditional Marginals

- Again:

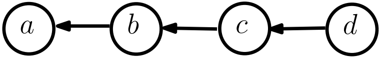


$$p(a,b,c,d) = p(a \mid b)p(b \mid c)p(c \mid d)p(d)$$

- Now find $p(d \mid a)$

$$\begin{aligned} p(d \mid a) = \frac{p(d,a)}{p(a)} \quad & \propto \quad \sum_{b,c} p(a \mid b)p(b \mid c)p(c \mid d)p(d) \\ & = \quad \sum_{c} \underbrace{\sum_{b} p(a \mid b)p(b \mid c)}_{\gamma_b(c)} p(c \mid d)p(d) \\ & \overset{def}{=} \quad \gamma_c(d) \text{ not a distribution} \end{aligned}$$

Finding Conditional Marginals – 2



- Found that

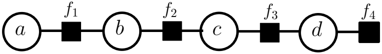
$$p(d \mid a) = k\gamma_c(d) \tag{18}$$

- and since $\sum_d p(d \mid a) = 1$

$$k = \frac{1}{\sum_d \gamma_c(d)} \tag{19}$$

- Again $\gamma_c(d)$ is not a distribution (but a message)

Again, now with factor graphs



$$p(a,b,c,d) = \frac{1}{Z} f_1(a,b) f_2(b,c) f_3(c,d) f_4(d) \tag{20}$$

$$\begin{aligned} p(a,b,c) &= \sum_d p(a,b,c,d) && (21) \\ &= \frac{1}{Z} f_1(a,b) f_2(b,c) \underbrace{\sum_d f_3(c,d) f_4(d)}_{\mu_{d\to c}(c)} && (22) \end{aligned}$$

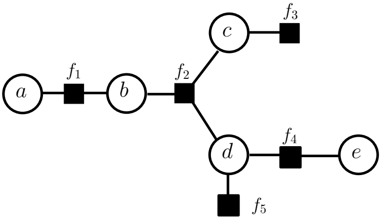
$$p(a,b) = \sum_c p(a,b,c) = \frac{1}{Z} f_1(a,b) \underbrace{\sum_c f_2(b,c) \mu_{d\to c}(c)}_{\mu_{c\to b}(b)} \tag{23}$$

Inference in Chain Structured Factor Graphs

- Simply recurse further
- $\gamma_{m \to n}(n)$ carries the information beyond m
- We did not need the factors – in general (next) we will see that making a distinction is helpful

General singly-connected factor graphs – 1

- Now consider a branching graph:



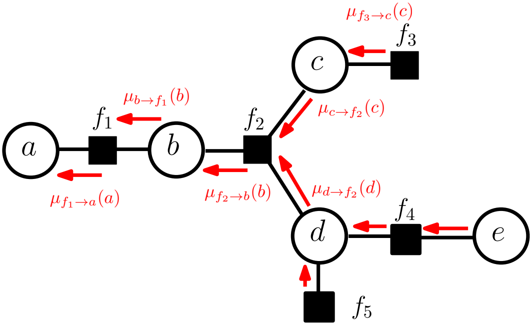
with factors

$$f_1(a,b)f_2(b,c,d)f_3(c)f_4(d,e)f_5(d) \tag{24}$$

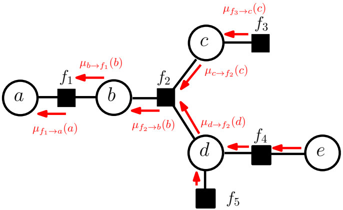
- For example: find marginal $p(a,b)$

General singly-connected factor graphs – 2

- Idea: compute messages



General singly-connected factor graphs – 3



$$p(a,b) = \frac{1}{Z} f_1(a,b) \underbrace{\sum_{c,d,e} f_2(b,c,d) f_3(c) f_5(d) f_4(d,e)}_{\mu_{f_2 \to b}(b)}$$

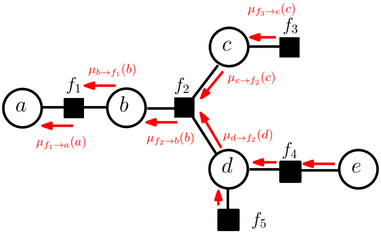
$$\mu_{f_2 \to b}(b) = \sum_{c,d} f_2(b,c,d) \underbrace{f_3(c)}_{\mu_{c \to f_2}(c)} \underbrace{f_5(d) \sum_{e} f_4(d,e)}_{\mu_{d \to f_2}(d)}$$

Factor-to-Variable Messages

$$\mu_{f_2 \to b}(b) = \sum_{c,d} f_2(b,c,d) \underbrace{f_3(c)}_{\mu_{c \to f_2}(c)} \underbrace{f_5(d) \sum_{e} f_4(d,e)}_{\mu_{d \to f_2}(d)}$$

$$\mu_{f_2 \to b}(b) = \sum_{c,d} f_2(b,c,d) \mu_{c \to f_2}(c) \mu_{d \to f_2}(d)$$

Factor-to-Variable Messages



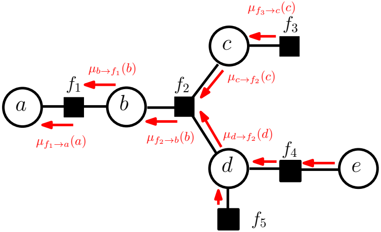
- Here (repeated from last slide):

$$\mu_{f_2\to b}(b) = \sum_{c,d} f_2(b,c,d)\mu_{c\to f_2}(c)\mu_{d\to f_2}(d) \tag{25}$$

- more general:

$$\mu_{f\to x}(x) = \sum_{y\in\mathcal{X}_f\setminus x} \phi_f(\mathcal{X}_f) \prod_{y\in\{\mathsf{ne}(f)\setminus x\}} \mu_{y\to f}(y) \tag{26}$$

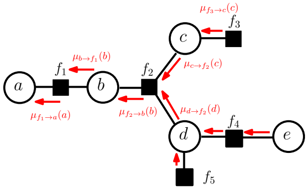
General singly-connected factor graphs – 4



$$\mu_{d \to f_2}(d) = \underbrace{f_5(d)}_{\mu_{f_5 \to d}(d)} \underbrace{\sum_e f_4(d, e)}_{\mu_{f_4 \to d}(d)}$$

$$\mu_{d \to f_2}(d) = \mu_{f_5 \to d}(d) \mu_{f_4 \to d}(d)$$

Variable-to-Factor Messages



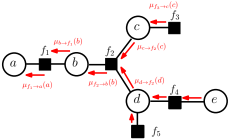
- Here (repeated from last slide):

$$\mu_{d\to f_2}(d) = \mu_{f_5\to d}(d)\mu_{f_4\to d}(d) \tag{27}$$

- General:

$$\mu_{x\to f}(x) = \prod_{g\in\{\mathsf{ne}(x)\setminus f\}} \mu_{g\to x}(x) \tag{28}$$

General singly-connected factor graphs – 5



- If we want to compute the marginal $p(a)$ (use factor-to-variable message):

$$p(a) = \frac{1}{Z}\mu_{f_1 \to a}(a) = \underbrace{\sum_b f_1(a,b)\mu_{b \to f_1}(b)}_{\mu_{f_1 \to a}(a)} \tag{29}$$

- which we could also view as

$$p(a) = \frac{1}{Z}\sum_b f_1(a,b)\underbrace{\mu_{b \to f_1}(b)}_{\mu_{f_2 \to b}(b)} \tag{30}$$

Comments

- Many subscripts :)
- Once computed, messages can be re-used
- All marginals $(p(c), p(d), p(c, d), ...)$ can be written as a function of messages
- The algorithm to compute all messages: Sum-Product algorithm

Sum-Product Algorithm – Overview

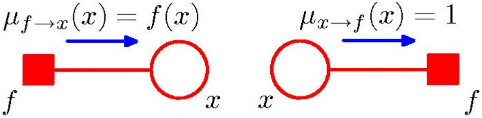
- Algorithm to compute all messages efficiently
- Assuming the graph is singly-connected (= tree)

1. Initialization
2. Variable to Factor message
3. Factor to Variable message

- Then compute any desired marginals
- Also known as belief propagation

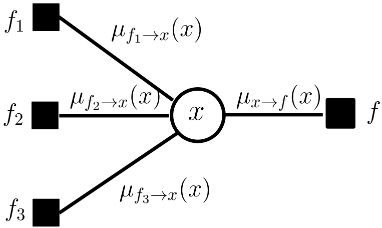
1. Initialization

- Messages from extremal (simplical) node factors are initialized to the factor (left)
- Messages from extremal (simplical) variable nodes are set to unity (right)



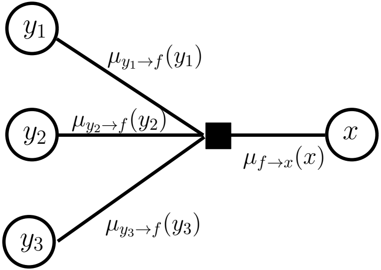
2. Variable to Factor Message

$$\mu_{x \to f}(x) = \prod_{g \in \{\mathsf{ne}(x) \setminus f\}} \mu_{g \to x}(x) \tag{31}$$



3. Factor to Variable Message

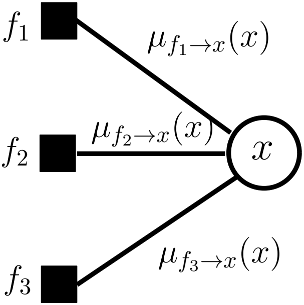
$$\mu_{f \to x}(x) = \sum_{y \in \mathcal{X}_f \setminus x} \phi_f(\mathcal{X}_f) \prod_{y \in \{\mathsf{ne}(f) \setminus x\}} \mu_{y \to f}(y) \tag{32}$$



- We sum over all states in the set of variables
- This explains the name for the algorithm (sum-product)

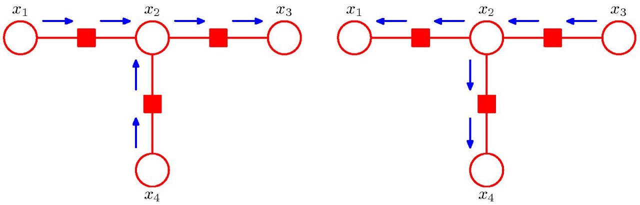
Marginal

$$p(x) \propto \prod_{f \in \mathsf{ne}(x)} \mu_{f \to x}(x) \tag{33}$$



Message ordering

- Messages depend on previously computed messages
- Only extremal nodes/factors do not depend on other messages
- To compute all messages in the graph
    1. leaf-to-root: (pick root node - here x_3 - compute messages pointing towards root)
    2. root-to-leave: (compute messages pointing away from root)



Computing the Partition Function

- The partition function ($p(x) = \frac{1}{Z}\prod_f \phi_f(\mathcal{X}_f)$) (normalization constant) Z can be computed after the leaf-to-root step (no need for the root-to-leaf step) (choose any $x \in \mathcal{X}$)

$$
\begin{aligned}
Z &= \sum_{\mathcal{X}} \prod_f \phi_f(\mathcal{X}_f) && (34) \\
&= \sum_x \sum_{\mathcal{X}\setminus\{x\}} \prod_{f \in \mathsf{ne}(x)} \prod_{f \notin \mathsf{ne}(x)} \phi_f(\mathcal{X}_f) && (35) \\
&= \sum_x \prod_{f \in \mathsf{ne}(x)} \sum_{\mathcal{X}\setminus\{x\}} \prod_{f \notin \mathsf{ne}(x)} \phi_f(\mathcal{X}_f) && (36) \\
&= \sum_x \prod_{f \in \mathsf{ne}(x)} \mu_{f\to x}(x) && (37)
\end{aligned}
$$

Log-Messages

- In large graphs, messages may become very small
- Work with log-messages instead $\lambda = \log \mu$
- Variable-to-factor messages

$$\mu_{x \to f}(x) = \prod_{g \in \{\mathsf{ne}(x) \setminus f\}} \mu_{g \to x}(x) \tag{38}$$

then becomes

$$\lambda_{x \to f}(x) = \sum_{g \in \{\mathsf{ne}(x) \setminus f\}} \lambda_{g \to x}(x) \tag{39}$$

Log-Messages

- Work with log-messages instead $\lambda = \log \mu$
- Factor-to-Variable messages

$$\mu_{f \to x}(x) = \sum_{y \in \mathcal{X}_f \setminus x} \Phi_f(\mathcal{X}_f) \prod_{y \in \{\mathsf{ne}(f) \setminus x\}} \mu_{y \to f}(y) \tag{40}$$

then becomes

$$\lambda_{f \to x}(x) = \log \left(\sum_{y \in \mathcal{X}_f \setminus x} \Phi(\mathcal{X}_f) \exp \left[\sum_{y \in \{\mathsf{ne}(f) \setminus x\}} \lambda_{y \to f}(y) \right] \right) \tag{41}$$

Trick

- Log-Factor-to-Variable Message:

$$\lambda_{f\to x}(x) = \log \sum_{y \in \mathcal{X}_f \setminus x} \Phi_f(\mathcal{X}_f) \exp \sum_{y \in \{\mathsf{ne}(f) \setminus x\}} \lambda_{y \to f}(y) \tag{42}$$

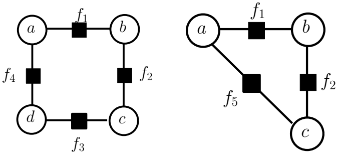
- large numbers lead to numerical instability
- Use the following equality

$$\log \sum_i \exp(v_i) = \alpha + \log \sum_i \exp(v_i - \alpha) \tag{43}$$

- With $\alpha = \max \lambda_{y\to f}(y)$

Problems with Loops

- Marginalizing over d introduces new link (changes graph structure – in contrast to singly connected graphs)



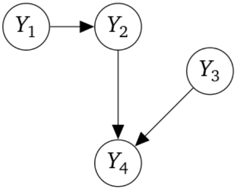
$$p(a,b,c,d) = \frac{1}{Z} f_1(a,b) f_2(b,c) f_3(c,d) f_4(d,a)$$

and marginal

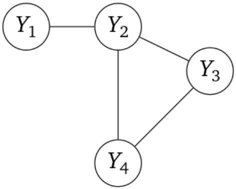
$$p(a,b,c) = \frac{1}{Z} f_1(a,b) f_2(b,c) \underbrace{\sum_d f_3(c,d) f_4(d,a)}_{f_5(a,c)}$$

Next Time ...

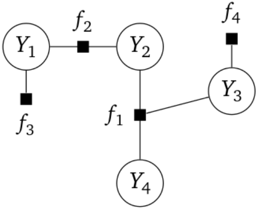
- ... inference when life is not so easy:



(a) Bayesian Network



(b) Markov Random Field



(c) Factor Graph

Relationship Directed – Undirected Models: Maps

D map

A graph is said to be a D map (dependency map) of a distribution if every conditional independence statement satisfied by the distribution is reflected in the graph

- A completely disconnected graph contains all possible independence statements for its variables
- $\Rightarrow$ it is a trivial D map for any distribution

Relationship Directed – Undirected Models: Maps

I map

A graph is said to be an I map (independence map) of a distribution if every conditional independence implied by the graph is satisfied by the distribution

- A fully connected graph implies no independence statements
- $\Rightarrow$ it is a trivial I map for any distribution

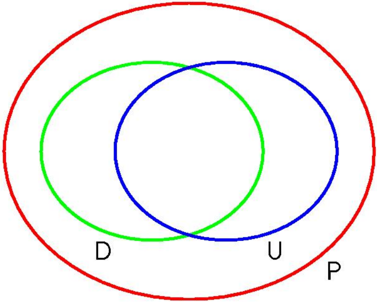
Relationship Directed – Undirected Models: Maps

perfect map

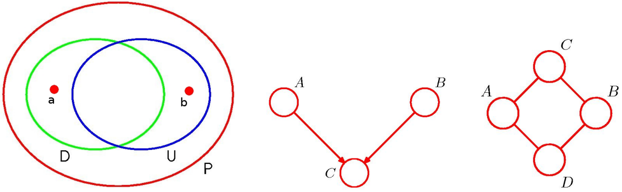
If every conditional independence property of the distribution is reflected in the graph, **and vice versa**, then the graph is said to be a perfect map for that distribution.

- A perfect map is therefore both I map and a D map of the distribution

Relationship Directed – Undirected GM



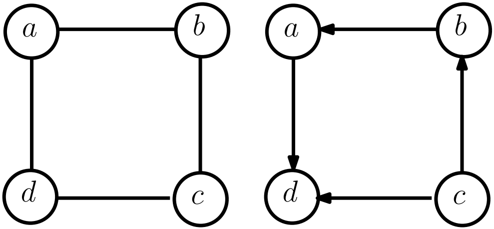
- P – set of all distributions for a given set of variables
- distributions that can be represented as a perfect map
  - using undirected graph – U
  - using a directed graph – D



(a) (b)

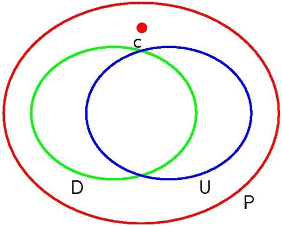
- Middle: conditional independence properties ($A \perp\!\!\!\perp B \mid \emptyset$ and $A \not\perp\!\!\!\perp B \mid C$) cannot be expressed using an undirected graph over the same three variables
- Right: conditional independence properties ($A \not\perp\!\!\!\perp B \mid \emptyset$, $A \perp\!\!\!\perp B \mid \{C, D\}$, and $C \perp\!\!\!\perp D \mid \{A, B\}$) cannot be expressed using a directed graph over the same four variables

Counter Example



- Any DAG on the four variables will have (at least) one collider, assume it is d
- Marginalizing out d will leave a DAG with no link between a and c
- Marginalizing in the undirected graph adds a link between a and c (immoral)

Chain Graphs



- What is "c"?
- Chain graphs contain both directed and undirected links
- Its class is broader than any single one alone