9. Probabilistic Graphical Models
Directed Models
The Bayes Filter (Rep.)

\[ \text{Bel}(x_t) = p(x_t \mid u_1, z_1, \ldots, u_t, z_t) \]

(Bayes)

\[ = \eta \ p(z_t \mid x_t, u_1, z_1, \ldots, u_t) p(x_t \mid u_1, z_1, \ldots, u_t) \]

(Markov)

\[ = \eta \ p(z_t \mid x_t) \int p(x_t \mid u_1, z_1, \ldots, u_t, x_{t-1}) \]

\[ \quad \times p(x_{t-1} \mid u_1, z_1, \ldots, u_t) \, dx_{t-1} \]

(Tot. prob.)

(Markov)

\[ = \eta \ p(z_t \mid x_t) \int p(x_t \mid u_t, x_{t-1}) p(x_{t-1} \mid u_1, z_1, \ldots, u_t) \, dx_{t-1} \]

(Markov)

\[ = \eta \ p(z_t \mid x_t) \int p(x_t \mid u_t, x_{t-1}) p(x_{t-1} \mid u_1, z_1, \ldots, z_{t-1}) \, dx_{t-1} \]

(Markov)

\[ = \eta \ p(z_t \mid x_t) \int p(x_t \mid u_t, x_{t-1}) \text{Bel}(x_{t-1}) \, dx_{t-1} \]
Graphical Representation (Rep.)

We can describe the overall process using a **Dynamic Bayes Network**:

![Dynamic Bayes Network Diagram](image)

- This incorporates the following Markov assumptions:
  
  $$p(z_t \mid x_{0:t}, u_{1:t}, z_{1:t}) = p(z_t \mid x_t) \text{ (measurement)}$$
  
  $$p(x_t \mid x_{0:t-1}, u_{1:t}, z_{1:t}) = p(x_t \mid x_{t-1}, u_t) \text{ (state)}$$
Definition

A Probabilistic Graphical Model is a diagrammatic representation of a probability distribution.

- In a Graphical Model, random variables are represented as nodes, and statistical dependencies are represented using edges between the nodes.
- The resulting graph can have the following properties:
  - Cyclic / acyclic
  - Directed / undirected
- The simplest graphs are Directed Acyclic Graphs (DAG).
Simple Example

- Given: 3 random variables \( a, b, \) and \( c \)
- Joint prob: \( p(a, b, c) = p(c \mid a, b)p(a, b) = p(c \mid a, b)p(b \mid a)p(a) \)

A Graphical Model based on a DAG is called a **Bayesian Network**

Random variables can be discrete or continuous.
Simple Example

• In general: $K$ random variables $x_1, x_2, \ldots, x_K$
• Joint prob:
  
  $$p(x_1, \ldots, x_K) = p(x_K|x_1, \ldots, x_{K-1}) \cdots p(x_2|x_1)p(x_1)$$
• This leads to a fully connected graph.
• Note: The ordering of the nodes in such a fully connected graph is arbitrary. They all represent the joint probability distribution:

  $$p(a, b, c) = p(a|b, c)p(b|c)p(c)$$
  $$p(a, b, c) = p(b|a, c)p(a|c)p(c)$$
  $$\vdots$$
Bayesian Networks

Statistical independence can be represented by the absence of edges. This makes the computation efficient.

\[
p(x_1, \ldots, x_7) = p(x_1)p(x_2)p(x_3)p(x_4|x_1, x_2, x_3) \\
p(x_5|x_1, x_3)p(x_6|x_4)p(x_7|x_4, x_5)
\]

Intuitively: only \(x_1\) and \(x_3\) have an influence on \(x_5\)
Bayesian Networks

We can now define a one-to-one mapping from graphical models to probabilistic formulations:

General Factorization:

\[ p(x) = \prod_{k=1}^{K} p(x_k | pa_k) \]

where

\[ pa_k \triangleq \text{ancestors of } p_k \]

and

\[ p(x) = p(x_1, \ldots, x_K) \]
Elements of Graphical Models

In case of a series of random variables with equal dependencies, we can subsume them using a \textbf{plate}:

\[ p(t, w) = p(w) \prod_{n=1}^{N} p(t_n | w) \]
Elements of Graphical Models (2)

We distinguish between **input** variables and explicit hyper-parameters:

\[ p(t, w | x, \alpha, \sigma^2) = p(w | \alpha) \prod_{n=1}^{N} p(t_n | w, x_n, \sigma^2). \]
Elements of Graphical Models (3)

We distinguish between **observed** variables and **hidden** variables:

\[
p(w|t) \propto p(w) \prod_{n=1}^{N} p(t_n|w)
\]

(deterministic parameters omitted in formula)
Regression as a Graphical Model

Regression: Prediction of a new target value $\hat{t}$

$$p(\hat{t}, t, w \mid \hat{x}, x, \alpha, \sigma^2) = \prod_{n=1}^{N} p(t_n \mid x_n, w, \sigma^2) p(w \mid \alpha) p(\hat{t} \mid \hat{x}, w, \sigma^2)$$

Here: conditioning on all deterministic parameters

Using this, we can obtain the predictive distribution:

$$p(\hat{t} \mid \hat{x}, x, t, \alpha, \sigma^2) \propto \int p(\hat{t}, t, w \mid \hat{x}, x, \alpha, \sigma^2) \, dw$$
Two Special Cases

- We consider two special cases:
- All random variables are discrete; i.e. Each $x_i$ is represented by values $\mu_1, \ldots, \mu_K$ where

$$p(x | \mu) = \prod_{k=1}^{K} \mu_k^{x_k} \sum_{j=1}^{K} \mu_j = 1$$

- All random variables are Gaussian

$$x_i \sim \mathcal{N}(.; \mu_i, \sigma_i^2)$$
Discrete Variables: Example

- Two dependent variables: \( K^2 - 1 \) parameters

<table>
<thead>
<tr>
<th>( x_1 )</th>
<th>( p(x_1) )</th>
<th>( 1 )</th>
<th>( 0.2 )</th>
<th>( 2 )</th>
<th>( 0.8 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( K - 1 )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Independent joint distribution: \( 2(K - 1) \) parameters

\[
K - 1 + K(K - 1) = K^2 - 1
\]

\[
K - 1 + K - 1 = 2(K - 1)
\]
Discrete Variables: General Case

In a general joint distribution with $M$ variables we need to store $K^M - 1$ parameters.

If the distribution can be described by this graph:

then we have only $K - 1 + (M - 1) K(K - 1)$ parameters. This graph is called a Markov chain with $M$ nodes. The number of parameters grows only linearly with the number of variables.
Gaussian Variables

Assume all random variables are Gaussian and we define

\[ p(x_i | \text{pa}_i) = \mathcal{N}\left( x_i; \sum_{j \in \text{pa}_i} w_{ij} x_j + b_i, v_i \right) \]

Then one can show that the joint probability \( p(x) \) is a multivariate Gaussian. Furthermore:

\[ x_i = \sum_{j \in \text{pa}_i} w_{ij} x_j + b_j + \sqrt{v_i} \epsilon_i \quad \epsilon_i \sim \mathcal{N}(0, 1) \]

Thus:

\[ E[x_i] = \sum_{j \in \text{pa}_i} w_{ij} E[x_j] + b_i \]

i.e., we can compute the mean values recursively.
Gaussian Variables

Assume all random variables are Gaussian and we define

\[ p(x_i \mid pa_i) = \mathcal{N}(x_i; \sum_{j \in pa_i} w_{ij}x_j + b_i, \nu_i) \]

The same can be shown for the covariance. Thus:

- Mean and covariance can be calculated \textbf{recursively}

Furthermore it can be shown that:

- The \textbf{fully connected} graph corresponds to a Gaussian with a \textbf{general symmetric} covariance matrix
- The \textbf{non-connected} graph corresponds to a \textbf{diagonal} covariance matrix
Independence (Rep.)

**Definition 1.4:** Two random variables $X$ and $Y$ are **independent** iff:

$$p(x, y) = p(x)p(y)$$

For independent random variables $X$ and $Y$ we have:

$$p(x \mid y) = \frac{p(x, y)}{p(y)} = \frac{p(x)p(y)}{p(y)} = p(x)$$

**Notation:** $x \perp y \mid \emptyset$

Independence does **not** imply conditional independence! The same is true for the opposite case.
Conditional Independence (Rep.)

**Definition 1.5:** Two random variables $X$ and $Y$ are *conditional independent* given a third random variable $Z$ iff:

$$p(x, y \mid z) = p(x \mid z)p(y \mid z)$$

This is equivalent to:

$$p(x \mid z) = p(x \mid y, z) \quad \text{and} \quad p(y \mid z) = p(y \mid x, z)$$

Notation: $x \perp\!
\!
\!
\!
\perp y \mid z$
Conditional Independence: Example 1

This graph represents the probability distribution:

\[ p(a, b, c) = p(a|c)p(b|c)p(c) \]

Marginalizing out \( c \) on both sides gives

\[ p(a, b) = \sum_c p(a|c)p(b|c)p(c) \]

This is in general not equal to \( p(a)p(b) \).

**Thus:** \( a \) and \( b \) are not independent: \( a \perp \!\!\!\!\!\!\!\perp b \mid \emptyset \)
Conditional Independence: Example 1

Now, we condition on $c$ (it is assumed to be known):

$$p(a, b|c) = \frac{p(a, b, c)}{p(c)} = p(a|c)p(b|c)$$

Thus: $a$ and $b$ are conditionally independent given $c$: $a \perp b | c$

We say that the node at $c$ is a tail-to-tail node on the path between $a$ and $b$. 

Machine Learning for Computer Vision

PD Dr. Rudolph Triebel
Computer Vision Group
Conditional Independence: Example 2

This graph represents the distribution:

\[ p(a, b, c) = p(a)p(c | a)p(b | c) \]

Again, we marginalize over \( c \):

\[
p(a, b) = p(a) \sum_c p(c | a)p(b | c) = p(a) \sum_c p(c | a)p(b | c, a)
\]

\[
= p(a) \sum_c \frac{p(c, a)p(b, c, a)}{p(a)p(c, a)} = p(a) \sum_c p(b, c | a)
\]

\[
= p(a)p(b | a)
\]

And we obtain:

\[ a \perp b \mid \emptyset \]
Conditional Independence: Example 2

As before, now we condition on $c$:

\[ p(a, b | c) = \frac{p(a, b, c)}{p(c)} = \frac{p(a)p(c|a)p(b|c)}{p(c)} = p(a|c)p(b|c) \]

And we obtain: $a \perp b | c$

We say that the node at $c$ is a **head-to-tail node** on the path between $a$ and $b$. 
Conditional Independence: Example 3

Now consider this graph:

\[ p(a, b, c) = p(a)p(b)p(c | a, b) \]

using:

\[ \sum_c p(a, b, c) = p(a)p(b) \sum_c p(c | a, b) \]

we obtain:

\[ p(a, b) = p(a)p(b) \]

And the result is: \( a \perp b \mid \emptyset \)
Conditional Independence: Example 3

Again, we condition on $c$

\[
p(a, b | c) = \frac{p(a, b, c)}{p(c)} = \frac{p(a)p(b)p(c|a, b)}{p(c)}
\]

This results in:

\[a \not\independent b \mid c\]

We say that the node at $c$ is a **head-to-head node** on the path between $a$ and $b$. 
To Summarize

• When does the graph represent (conditional) independence?

Tail-to-tail case: if we condition on the tail-to-tail node
Head-to-tail case: if we cond. on the head-to-tail node
Head-to-head case: if we do not condition on the head-to-head node (and neither on any of its descendants)

In general, this leads to the notion of **D-separation** for directed graphical models.
D-Separation

Say: A, B, and C are non-intersecting subsets of nodes in a directed graph. A path from A to B is **blocked** by C if it contains a node such that either

a) the arrows on the path meet either **head-to-tail** or **tail-to-tail** at the node, and the node is in the set C, or

b) the arrows meet **head-to-head** at the node, and neither the node, nor any of its descendants, are in the set C.

If all paths from A to B are blocked, A is said to be **d-separated** from B by C.

**Notation:** \( \text{dsep}(A, B|C) \)
D-Separation

Say: A, B, and C are non-intersecting subsets of nodes in a directed graph.

• A path from A to B contains a node such that either:
  a) the arrows meet head-to-tail or tail-to-tail at the node,
  b) the arrows meet head-to-head and neither the node nor any of its descendants are in the set C.

• If all paths from A to B are blocked, A is said to be d-separated from B by C.

Notation: \( \text{dsep}(A, B|C) \)
D-Separation: Example

We condition on a descendant of $e$, i.e. it does not block the path from $a$ to $b$.  

$\neg \text{dsep}(a, b|c)$

We condition on a tail-to-tail node on the only path from $a$ to $b$, i.e $f$ blocks the path.  

$\text{dsep}(a, b|f)$
I-Map

**Definition 4.1:** A graph $G$ is called an **I-map** for a distribution $p$ if every D-separation of $G$ corresponds to a conditional independence relation satisfied by $p$:

\[
\forall A, B, C : \text{dsep}(A, B, C) \implies A \perp\!
\!
\!
\!
\perp B \mid C
\]

**Example:** The fully connected graph is an I-map for any distribution, as there are no D-separations in that graph.
Definition 4.2: A graph $G$ is called an \textbf{D-map} for a distribution $p$ if for every conditional independence relation satisfied by $p$ there is a D-separation in $G$:

$$\forall A, B, C : A \perp B \mid C \Rightarrow \text{dsep}(A, B, C)$$

Example: The graph without any edges is a D-map for any distribution, as all pairs of subsets of nodes are D-separated in that graph.
Perfect Map

**Definition 4.3:** A graph $G$ is called a **perfect map** for a distribution $p$ if it is a D-map and an I-map of $p$.

$$\forall A, B, C : A \perp\!\!\!\!\!\!\perp B \mid C \iff \text{dsep}(A, B, C)$$

A perfect map uniquely defines a probability distribution.
Consider a distribution of a node $x_i$ conditioned on all other nodes:

$$p(x_i | x_{\{j \neq i\}}) = \frac{p(x_1, \ldots, x_M)}{\int p(x_1, \ldots, x_M) \, dx_i} \prod_{k} p(x_k | pa_k)$$

$$= \frac{\prod_{k} p(x_k | pa_k)}{\int \prod_{k} p(x_k | pa_k) \, dx_i}$$

$$= p(x_i | x_{M_i})$$

Markov blanket $M_i$ at $x_i$: all parents, children and co-parents of $x_i$. Factors independent of $x_i$ cancel between numerator and denominator.
Summary

• Graphical models represent joint probability distributions using nodes for the random variables and edges to express (conditional) (in)dependence.

• A probability distribution can always be represented using a fully connected graph, but this is inefficient.

• In a directed acyclic graph, conditional independence is determined using D-separation.

• A perfect map implies a one-to-one mapping between c.i. relations and D-separations.

• The Markov blanket is the minimal set of observed nodes to obtain conditional independence.
4. Probabilistic Graphical Models

Undirected Models
Repetition: Bayesian Networks

Directed graphical models can be used to represent probability distributions. This is useful to do inference and to generate samples from the distribution efficiently.

\[
p(x_1, \ldots, x_7) = p(x_1)p(x_2)p(x_3)p(x_4|x_1, x_2, x_3) \\
p(x_5|x_1, x_3)p(x_6|x_4)p(x_7|x_4, x_5)
\]
Repetition: D-Separation

• D-separation is a property of graphs that can be easily determined
• An I-map assigns every d-separation a c.i. rel
• A D-map assigns every c.i. rel a d-separation
• Every Bayes net determines a unique prob. dist.
In-depth: The Head-to-Head Node

Example:

a: Battery charged (0 or 1)
b: Fuel tank full (0 or 1)
c: Fuel gauge says full (0 or 1)

We can compute \( p(\neg c) = 0.315 \)

and \( p(\neg c \mid \neg b) = 0.81 \)

and obtain \( p(\neg b \mid \neg c) \approx 0.257 \)

similarly: \( p(\neg b \mid \neg c, \neg a) \approx 0.111 \)

"a explains c away"
Repetition: D-Separation

\[ \neg \text{dsep}(a, b|c) \quad \text{dsep}(a, b|f) \]
Directed vs. Undirected Graphs

Using D-separation we can identify conditional independencies in directed graphical models, but:

- Is there a simpler, more intuitive way to express conditional independence in a graph?
- Can we find a representation for cases where an „ordering“ of the random variables is inappropriate (e.g. the pixels in a camera image)?

**Yes, we can:** by removing the directions of the edges we obtain an Undirected Graphical Model, also known as a Markov Random Field.
Example: Camera Image

- directions are counter-intuitive for images
- Markov blanket is not just the direct neighbors when using a directed model
All paths from $A$ to $B$ go through $C$, i.e. $C$ blocks all paths.

$A \perp B | C$

We only need to condition on the \textbf{direct neighbors} of $x$ to get c.i., because these already block every path from $x$ to any other node.
Factorization of MRFs

Any two nodes $x_i$ and $x_j$ that are not connected in an MRF are conditionally independent given all other nodes:

$$p(x_i, x_j \mid \mathbf{x}\setminus\{i,j\}) = p(x_i \mid \mathbf{x}\setminus\{i,j\})p(x_j \mid \mathbf{x}\setminus\{i,j\})$$

In turn: each factor contains only nodes that are connected

This motivates the consideration of cliques in the graph:

- A **clique** is a fully connected subgraph.
- A **maximal** clique can not be extended with another node without loosing the property of full connectivity.
Factorization of MRFs

In general, a Markov Random Field is factorized as

\[ p(x) = \frac{\prod_C \phi_C(x_C)}{\sum_{x'} \prod_C \phi_C(x'_C)} = \frac{1}{Z} \prod_C \phi_C(x_C) \quad (4.1) \]

where \( C \) is the set of all (maximal) cliques and \( \Phi_C \) is a positive function of a given clique \( x_C \) of nodes, called the **clique potential**. \( Z \) is called the **partition function**.

**Theorem (Hammersley/Clifford):** Any undirected model with associated clique potentials \( \Phi_C \) is a perfect map for the probability distribution defined by Equation (4.1).

As a conclusion, all probability distributions that can be factorized as in (4.1), can be represented as an MRF.
Converting Directed to Undirected Graphs (1)

In this case: \( Z=1 \)
Converting Directed to Undirected Graphs (2)

In general: conditional distributions in the directed graph are mapped to cliques in the undirected graph

However: the variables are not conditionally independent given the head-to-head node

Therefore: Connect all parents of head-to-head nodes with each other (moralization)
Problem: This process can remove conditional independence relations (inefficient)

Generally: There is no one-to-one mapping between the distributions represented by directed and by undirected graphs.
Representability

- As for DAGs, we can define an I-map, a D-map and a perfect map for MRFs.
- The set of all distributions for which a DAG exists that is a perfect map is different from that for MRFs.
Using Graphical Models

We can use a graphical model to do **inference**:

- Some nodes in the graph are **observed**, for others we want to find the posterior distribution.
- Also, computing the local **marginal distribution** $p(x_n)$ at any node $x_n$ can be done using inference.

**Question:** How can inference be done with a graphical model?

We will see that when exploiting conditional independences we can do efficient inference.
Inference on a Chain

The joint probability is given by

\[ p(x) = \frac{1}{Z} \psi_{1,2}(x_1, x_2) \psi_{2,3}(x_2, x_3) \psi_{3,4}(x_3, x_4) \psi_{4,5}(x_4, x_5) \]

The marginal at \( x_3 \) is

\[ p(x_3) = \sum_{x_1} \sum_{x_2} \sum_{x_4} \sum_{x_5} p(x) \]

In the general case with \( N \) nodes we have

\[ p(x) = \frac{1}{Z} \psi_{1,2}(x_1, x_2) \psi_{2,3}(x_2, x_3) \cdots \psi_{N-1,N}(x_{N-1}, x_N) \]

and

\[ p(x_n) = \sum_{x_1} \cdots \sum_{x_{n-1}} \sum_{x_{n+1}} \cdots \sum_{x_N} p(x) \]
Inference on a Chain

\[
p(x_3) = \sum_{x_1} \sum_{x_2} \sum_{x_4} \sum_{x_5} p(x)
\]

- This would mean \(K^N\) computations! A more efficient way is obtained by rearranging:

\[
p(x_3) = \frac{1}{Z} \sum_{x_1} \sum_{x_2} \sum_{x_4} \sum_{x_5} \psi_{1,2}(x_1, x_2) \psi_{2,3}(x_2, x_3) \psi_{3,4}(x_3, x_4) \psi_{4,5}(x_4, x_5)
\]

\[
= \frac{1}{Z} \sum_{x_2} \sum_{x_1} \sum_{x_4} \sum_{x_5} \psi_{1,2}(x_1, x_2) \psi_{2,3}(x_2, x_3) \psi_{3,4}(x_3, x_4) \psi_{4,5}(x_4, x_5)
\]

\[
= \frac{1}{Z} \sum_{x_2} \psi_{2,3}(x_2, x_3) \sum_{x_1} \psi_{1,2}(x_1, x_2) \sum_{x_4} \psi_{3,4}(x_3, x_4) \sum_{x_5} \psi_{4,5}(x_4, x_5)
\]

\[
\mu_\alpha(x_3) \leftarrow \text{Vectors of size K} \rightarrow \mu_\beta(x_3)
\]
Inference on a Chain

In general, we have

\[ p(x_n) = \frac{1}{Z} \left[ \sum_{x_{n-1}} \psi_{n-1,n}(x_{n-1}, x_n) \right] \cdots \left[ \sum_{x_1} \psi_{1,2}(x_1, x_2) \right] \cdots \]

\[ = \mu_\alpha(x_n) \left[ \sum_{x_{n+1}} \psi_{n,n+1}(x_n, x_{n+1}) \right] \cdots \left[ \sum_{x_N} \psi_{N-1,N}(x_{N-1}, x_N) \right] \cdots \]

\[ = \mu_\beta(x_n) \]
Inference on a Chain

The messages $\mu_\alpha$ and $\mu_\beta$ can be computed recursively:

$$\mu_\alpha(x_n) = \sum_{x_{n-1}} \psi_{n-1,n}(x_{n-1}, x_n) \left[ \sum_{x_{n-2}} \cdots \right]$$

$$= \sum_{x_{n-1}} \psi_{n-1,n}(x_{n-1}, x_n) \mu_\alpha(x_{n-1}).$$

$$\mu_\beta(x_n) = \sum_{x_{n+1}} \psi_{n,n+1}(x_n, x_{n+1}) \left[ \sum_{x_{n+2}} \cdots \right]$$

$$= \sum_{x_{n+1}} \psi_{n,n+1}(x_n, x_{n+1}) \mu_\beta(x_{n+1}).$$

Computation of $\mu_\alpha$ starts at the first node and computation of $\mu_\beta$ starts at the last node.
Inference on a Chain

- The first values of $\mu_\alpha$ and $\mu_\beta$ are:

$$
\mu_\alpha(x_2) = \sum_{x_1} \psi_{1,2}(x_1, x_2) \quad \mu_\beta(x_{N-1}) = \sum_{x_N} \psi_{N-1,N}(x_{N-1}, x_N)
$$

- The partition function can be computed at any node:

$$
Z = \sum_{x_n} \mu_\alpha(x_n) \mu_\beta(x_n)
$$

- Overall, we have $O(NK^2)$ operations to compute the marginal $p(x_n)$
Inference on a Chain

To compute local marginals:

- Compute and store all forward messages, \( \mu_\alpha(x_n) \).
- Compute and store all backward messages, \( \mu_\beta(x_n) \).
- Compute \( Z \) \textbf{once} at a node \( x_m \):
  \[
  Z = \sum_{x_m} \mu_\alpha(x_m) \mu_\beta(x_m)
  \]
- Compute
  \[
  p(x_n) = \frac{1}{Z} \mu_\alpha(x_n) \mu_\beta(x_n)
  \]

for all variables required.
More General Graphs

The message-passing algorithm can be extended to more general graphs:

- Undirected Tree
- Directed Tree
- Polytree

It is then known as the **sum-product algorithm**. A special case of this is **belief propagation**.
Factor Graphs

- The Sum-product algorithm can be used to do inference on undirected and directed graphs.
- A representation that generalizes directed and undirected models is the factor graph.

\[
p(x) = p(x_1)p(x_2)p(x_3|x_1, x_2) \quad f(x_1, x_2, x_3) = p(x_1)p(x_2)p(x_3|x_1, x_2)
\]

Directed graph

Factor graph
Factor Graphs

- The Sum-product algorithm can be used to do inference on undirected and directed graphs.
- A representation that generalizes directed and undirected models is the factor graph.

Undirected graph

\[ \psi(x_1, x_2, x_3) \]

Factor graph

\[ f(x_1, x_2, x_3) = \psi(x_1, x_2, x_3) \]
Sum-Product Inference in General Graphical Models

1. Convert graph (directed or undirected) into a factor graph (there are no cycles)

2. If the goal is to marginalize at node $x$, then consider $x$ as a root node

3. Initialize the recursion at the leaf nodes as:
   \[
   \mu_{f \rightarrow x}(x) = 1 \quad \text{(var)} \quad \text{or} \quad \mu_{x \rightarrow f}(x) = f(x) \quad \text{(fac)}
   \]

4. Propagate messages from the leaves to $x$

5. Propagate messages from $x$ to the leaves

6. Obtain marginals at every node by multiplying all incoming messages
Further Topics on Graphical Models

Other inference algorithms:

- Max-Sum algorithm: used to maximize the joint probability of all variables (no marginalization)
- Junction Tree algorithm: exact inference for general graphs (even with loops)
- Loopy belief propagation: approximate inference on general graphs (more efficient)

Special kind of undirected GM:

- Conditional Random fields (e.g.: classification)

More details: see class of Dr. Domokos
http://vision.in.tum.de/teaching/ss2016/lecture_graphical_models
Summary

- Undirected models (aka Markov random fields) provide an intuitive representation of conditional independence.

- An MRF is defined as a factorization over clique potentials and normalized globally.

- Directed and undirected models have different representative power (no simple "containment").

- Inference on undirected Markov chains is efficient using message passing.

- Factor graphs are more general; exact inference can be done efficiently using sum-product.