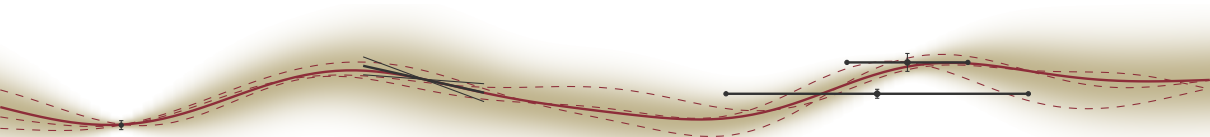
Probabilistic Inference and Learning
Lecture 16
Factor Graphs

Philipp Hennig

12 December 2018

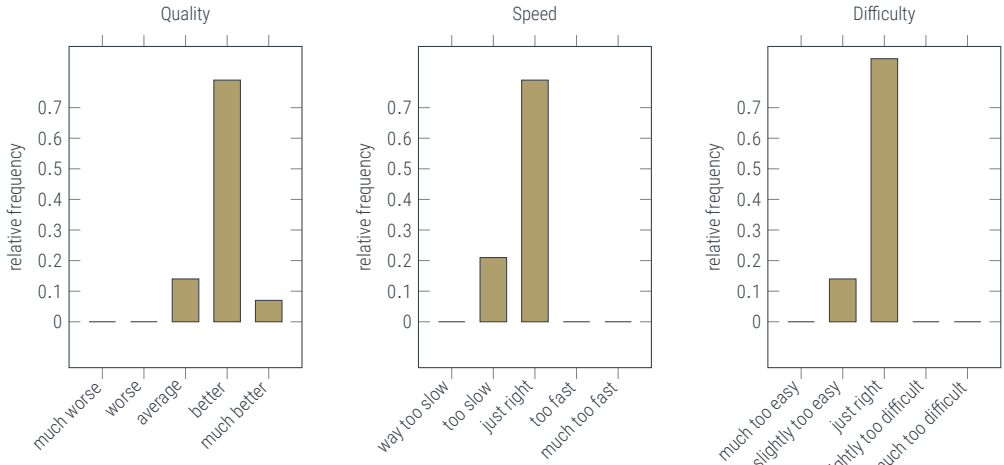
EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Faculty of Science
Department of Computer Science
Chair for the Methods of Machine Learning



Last Lecture: Debrief

Feedback dashboard



Last Lecture: Debrief

Detailed Feedback

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Things you did not like:

- exercises too hard
- too much text on slides
- missing link between DAGs and
- second part of lecture was too fast

Things you did not understand:

- why are DAGs not a subset of MRFs?
- MRFs
- why are you drinking “coffee” from the machine? The Clubhaus is across the street!
- how does the MRF for the burglar example look like?

Things you enjoyed:

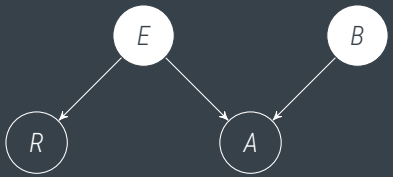
- concrete examples of use-cases

Overview of Lectures so far:

0. Introduction to Reasoning under Uncertainty
1. Probabilistic Reasoning
2. Probabilities over Continuous Variables
3. Gaussian Probability Distributions
4. Gaussian Parametric Regression
5. More on Parametric Regression
6. Gaussian Processes
7. More on Kernels & GPs
8. A practical GP example
9. Markov Chains, Time Series, Filtering
10. Classification
11. Empirical Example of Classification
12. Bayesianism and Frequentism
13. Stochastic Differential Equations
14. Exponential Families
15. Graphical Models
16. Factor Graphs

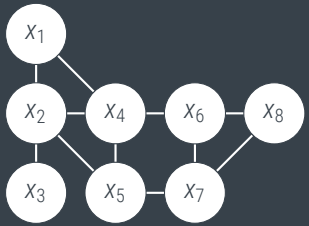
Today: Mapping between Frameworks & efficient Inference on Graphs

Directed Graphical Models / Bayesian Networks



- directly encode a facorization of the joint (it can be read off by parsing the graph from the children to the parents)
- however, reading off conditional independence structure is tricky (it requires considering *d*-separation)
- directed graphs are for encoding **generative knowledge** (think: scientific modelling)

Undirected Graphical Models / Markov Random Fields (MRFs)



- directly encode conditional independence structure (by definition)
- however, reading off the joint from the graph is tricky (it requires finding all maximal cliques, normalization constant is intractable)
- MRFs are for encoding **computational constraints** (think: computer vision)

From Directed to Undirected Graphs

Example: Markov Chain

[from Bishop, PRML, 2006]

$$p(\boldsymbol{x}) = p(x_1) \cdot p(x_2 \mid x_1) \cdot p(x_3 \mid x_2) \cdots p(x_n \mid x_{n-1})$$

From Directed to Undirected Graphs

Example: Markov Chain

[from Bishop, PRML, 2006]

$$
\begin{aligned}
p(\boldsymbol{x}) &= p(x_1) \cdot p(x_2 \mid x_1) \cdot p(x_3 \mid x_2) \cdots p(x_n \mid x_{n-1}) \\
&= \frac{1}{Z} \psi_{1,2}(x_1, x_2) \cdot \psi_{2,3}(x_2, x_3) \cdots \psi_{n-1,n}(x_{n-1}, x_n)
\end{aligned}
$$

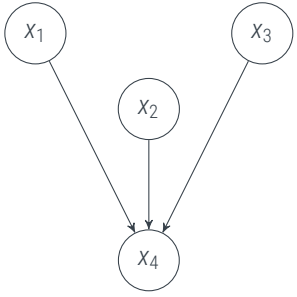
The MRF for a directed chain graph is a **Markov Chain**.

From Directed to Undirected Graphs

Moralization

[from Bishop, PRML, 2006]

- we need to ensure that each conditional term in the directed graph are captured in at least one clique of the undirected graph
- for nodes with only one parent, we can thus simply drop the arrow, and get $p(x_c \mid x_p) = \phi_{c,p}(x_c, x_p)$
- but for nodes with several parents, we have to connect ("marry") all the parents. This process is known as **moralization**.
- note that this **moralization** frequently frequently leads to densely connected graphs, losing all value of the graph.



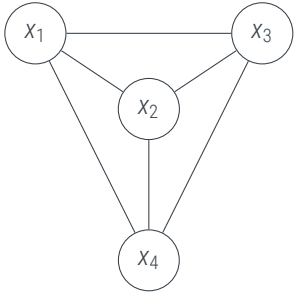
$$p(\boldsymbol{x}) = p(x_1) \cdot p(x_2) \cdot p(x_3) \cdot p(x_4 \mid x_1, x_2, x_3)$$

From Directed to Undirected Graphs

Moralization

[from Bishop, PRML, 2006]

- we need to ensure that each conditional term in the directed graph are captured in at least one clique of the undirected graph
- for nodes with only one parent, we can thus simply drop the arrow, and get $p(x_c \mid x_p) = \phi_{c,p}(x_c, x_p)$
- but for nodes with several parents, we have to connect ("marry") all the parents. This process is known as **moralization**.
- note that this **moralization** frequently frequently leads to densely connected graphs, losing all value of the graph.



$$p(\boldsymbol{x}) = p(x_1) \cdot p(x_2) \cdot p(x_3) \cdot p(x_4 \mid x_1, x_2, x_3)$$

Limits of Both Families

see also Lecture 1

Recall "two coins and a bell":

$p(A=1)=0.5$

$p(B=1)=0.5$

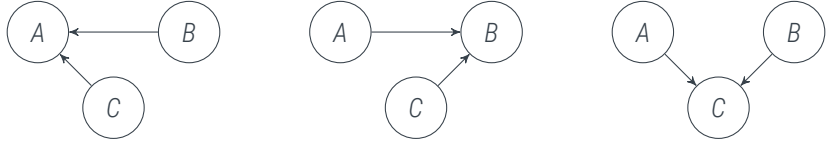
$p(C=1 \mid A=1,B=1)=1$	$p(C=1 \mid A=1,B=0)=0$
$p(C=1 \mid A=0,B=1)=0$	$p(C=1 \mid A=0,B=0)=1$

These CPTs imply $p(A|B)=p(A), p(B|C)=p(B)$ and $p(C|A)=p(C)$ and $P(C \mid B)=P(C)$.

We thus have three factorizations:

1. $p(A,B,C)=p(C|A,B)\cdot p(A|B)\cdot p(B)=p(C|A,B)\cdot p(A)\cdot p(B)$
2. $p(A,B,C)=p(A|B,C)\cdot p(B|C)\cdot p(C)=p(A|B,C)\cdot p(B)\cdot p(C)$
3. $p(A,B,C)=p(B|C,A)\cdot p(C|A)\cdot p(A)=p(B|C,A)\cdot p(C)\cdot p(A)$

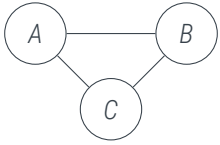
Each corresponds to a graph. Note that each can only express some of the independencies:



Limits of Both Families

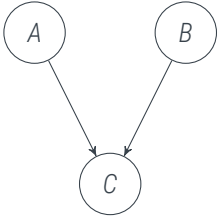
undirected case

The MRF for “two coins and a bell”, however, is totally useless.
It does not capture any of the conditional independencies.

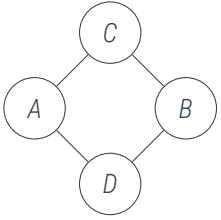


Limits of Both Families

[from Bishop, PRML, 2006]



$A \perp\!\!\!\perp B \mid \varnothing$ and $A \not\perp\!\!\!\perp B \mid C$



$x \not\perp\!\!\!\perp y \mid \varnothing \; \forall x, y$ and $C \perp\!\!\!\perp D \mid A \cup B$ and $A \perp\!\!\!\perp B \mid C \cup D$

The conditional independence properties of the directed graph on the left can not be represented by any MRF over the same three variables; and the conditional independence properties of the MRF on the right can not be represented by any directed graph on the same four variables.

Directed and Undirected Graphs fit different problems

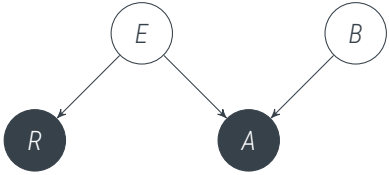
[cf. Bishop, 2006]

- Consider a distribution $p(\boldsymbol{x})$ and a graph $G = (V_{\boldsymbol{x}}, E)$.
- If every conditional *in*dependence statement satisfied by the distribution can be read off from the graph, then G is called an D-**map** of p. (The fully disconnected Graph is a trivial D-map for every p)
- If every conditional independence statement implied by G is also satisfied by p, then G is called a I-**map** of p. (The fully connected graph is a trivial I-map for every p).
- A G that is both an I-map and a D-map of p is called a **perfect map** of p.
- The set of distributions p for which there exists a directed graph that is a perfect map is distinct from the set of p for which there exists a perfect MRF map. (see two examples on previous slide. Markov Chains are an example where both MRF and directed graph are perfect). And there exist p for which there exists neither a directed nor an undirected perfect map (e.g. two coins and bell)

Uses of Directed Graphs

Direct mapping of generative processes

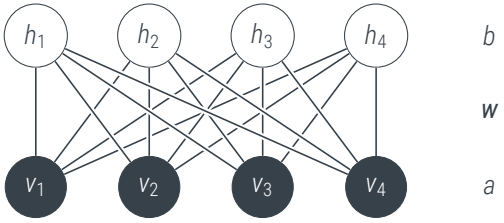
$$p(A, E, B, R) = p(A \mid R, E, B) \cdot p(R \mid E, B) \cdot p(E \mid B) \cdot p(B)$$
$$= p(A \mid E, B) \cdot p(R \mid E) \cdot p(E) \cdot p(B)$$



Directed graphical models tend to be useful in highly structured problems with mixed data types; physical, biological, chemical or social processes; where causal structure is known. When you want to model a process for which you have a "scientific" theory or just a mental model, you might want to start by writing down the directed model.

Example: Restricted Boltzmann machine

$$p(\boldsymbol{x}) = e^{-E(\boldsymbol{x})} = \exp\left(\sum_i a_i v_i + \sum_j b_j h_j + \sum_{ij} v_i w_{ij} h_j\right)$$



MRFs tend to be useful in highly regular but high-dimensional problems with unclear generative model, such as those encountered in computer vision and statistical physics. When your model has millions of parameters and you are more worried about computational complexity than interpretability, the conditional independence structure of MRFs can help keep things tractable.

Summary so far:

- directed and undirected graphs offer tools to graphically represent and inspect properties of joint probability distributions. Both are primarily a **design tool**
- each framework has its strengths and weaknesses. Strong simplification:

  Bayes Nets for encoding **structured generative knowledge** over heterogeneous variable sets, e.g. in scientific modelling

  MRFs for encoding **computational constraints** over large sets of similar variables, e.g. in computer vision (pixels)

next goal:

- a third type of graphical model, particularly well-suited for *automated* inference
- a general-purpose algorithm for *automated* inference
- a variant for efficient MAP inference

Factor Graphs

An explicit representation of functional relationships

[B. Frey, 1998]

- Both directed and undirected graphs provide a **factorization** of a distribution into **functions** over sets of **variables**

$$p(\boldsymbol{x}) = \prod_s f_s(\boldsymbol{x}_s)$$

directed: $f_s(\boldsymbol{x}_s)$ – conditional distribution
undirected: $f_s(\boldsymbol{x}_s)$ – potential function ($Z = f_z(\varnothing)$)

Definition

A **factor graph** is a *bipartite* graph $G = (V, F, E)$ of *variables* $v_i \in V$, *factors* $f_i \in F$ and *edges*, such that each edge connects a factor to a variable.

Brendan J. Frey

image: Toronto Star

Factor Graphs

An explicit representation of functional relationships

[B. Frey, 1998]

- Both directed and undirected graphs provide a **factorization** of a distribution into **functions** over sets of **variables**

$$p(\boldsymbol{x}) = \prod_S f_S(\boldsymbol{x}_S)$$

directed: $f_S(\boldsymbol{x}_S)$ – conditional distribution
undirected: $f_S(\boldsymbol{x}_S)$ – potential function ($Z = f_Z(\varnothing)$)

Definition

A **factor graph** is a *bipartite* graph $G = (V, F, E)$ of *variables* $v_i \in V$, *factors* $f_i \in F$ and *edges*, such that each edge connects a factor to a variable.

Frank Kschischang

image: U of Toronto

Factor Graphs

An explicit representation of functional relationships

[B. Frey, 1998]

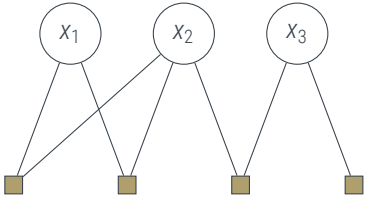
- Both directed and undirected graphs provide a **factorization** of a distribution into **functions** over sets of **variables**

$$p(\boldsymbol{x}) = \prod_s f_s(\boldsymbol{x}_s)$$

directed: $f_s(\boldsymbol{x}_s)$ – conditional distribution
undirected: $f_s(\boldsymbol{x}_s)$ – potential function ($Z = f_z(\varnothing)$)

Definition

A **factor graph** is a *bipartite* graph $G = (V, F, E)$ of *variables* $v_i \in V$, *factors* $f_i \in F$ and *edges*, such that each edge connects a factor to a variable.

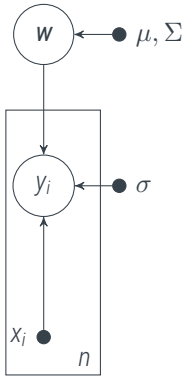


$$p(\boldsymbol{y}, \boldsymbol{w}) = \prod_{i=1}^{n} \mathcal{N}(y_i; \phi(x_i)^\intercal \boldsymbol{w}, \sigma^2) \mathcal{N}(\boldsymbol{w}; \mu, \Sigma)$$

To construct a factor graph from a directed graph

$$p(\boldsymbol{x}) = \prod_{c \in C} p_c(\boldsymbol{x}_c \mid x_{\mathrm{pa}(c)})$$

- draw a circle for each variable x_i
- draw a box for each conditional p_c
- connect each p_c to the variables in it

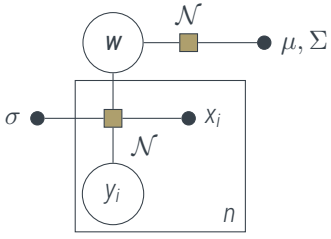


$$p(\boldsymbol{y}, \boldsymbol{w}) = \prod_{i=1}^{n} \mathcal{N}(y_i; \phi(x_i)^\intercal \boldsymbol{w}, \sigma^2) \mathcal{N}(\boldsymbol{w}; \mu, \Sigma)$$

To construct a factor graph from a directed graph

$$p(\boldsymbol{x}) = \prod_{c \in C} p_c(\boldsymbol{x}_c \mid x_{\mathrm{pa}(c)})$$

- draw a circle for each variable x_i
- draw a box for each conditional p_c
- connect each p_c to the variables in it

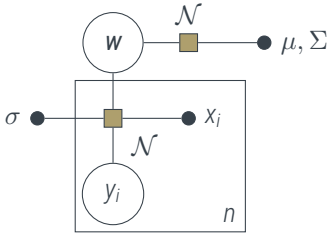


$$p(\boldsymbol{y}, \boldsymbol{w}) = \prod_{i=1}^{n} \mathcal{N}(y_i; \phi(x_i)^\intercal \boldsymbol{w}, \sigma^2) \mathcal{N}(\boldsymbol{w}; \mu, \Sigma)$$

To construct a factor graph from a MRF

$$p(\boldsymbol{x}) = \frac{1}{Z} \prod_{c \in C} \psi_c(\boldsymbol{x}_c)$$

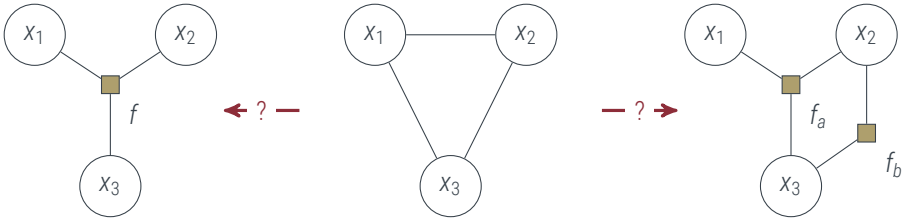
- draw a circle for each variable x_i
- draw a box for each factor (potential) ψ
- connect each ψ to the variables used in the factor



Explicit Functional Relationships Reveal Structure

Factor Graphs can express structure not visible in Undirected Graphs

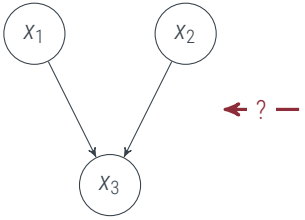
$$p(x_1, x_2, x_3) = f_a(x_1, x_2, x_3) \cdot f_b(x_2, x_3)$$

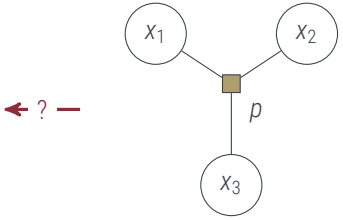


Explicit Functional Relationships Have No Direction!

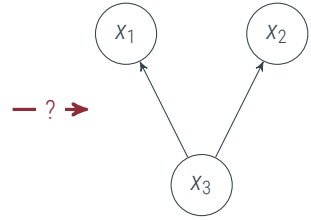
Factor graphs can mask conditional independence

$$p(x_1, x_2, x_3) = p(x_3 \mid x_1, x_2)p(x_1)p(x_2)$$





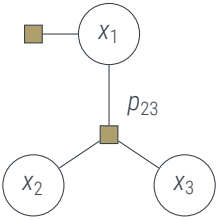
$$p(x_1, x_2, x_3) = p(x_1, x_2 \mid x_3)p(x_3)$$



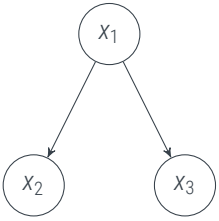
Arrows Have No Explicit Form!

Factor graphs can also reveal functional relationships

$$p(\boldsymbol{x}) = p_{23}(x_2, x_3 \mid x_1)p(x_1)$$

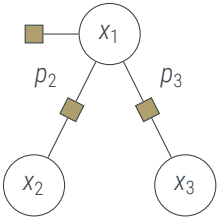


?



?

$$p(\boldsymbol{x}) = p_2(x_2 \mid x_1)p_3(x_3 \mid x_1)p(x_1)$$



The graphical view itself does not always capture all structure. But when factor graphs are encoded with the explicit functional form, some interesting structure can be automatically deduced.

What do we have to do, in general, to compute a **marginal** distribution

$$p(x_i) = \int p(x_1, \ldots, x_i, \ldots, x_n)\, dx_{j \neq i}$$

If the joint $p(x_1, \ldots, x_n)$ is given by a factor graph?

- J. Pearl. Probabilistic Reasoning in Intelligent Systems. Morgan Kaufmann, 1988.
- S.L. Lauritzen and D.J. Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. J. R. Stat. Soc., 50:157–224, 1988.

- F.R. Kschischang, B.J. Frey, and H.-A. Loeliger. Factor graphs and the sum- product algorithm. IEEE Transactions on Information Theory, 47(2):498–519, February 2001.

Base Case: Markov Chains

Filtering and Smoothing are special cases of the sum-product algorithm

[exposition based on Bishop, PRML, 2006]

Assume discrete $x_i \in [1, \dots, k]$ for the moment. What is the **marginal** $p(x_i)$?

$$p(\boldsymbol{x}) = \frac{1}{Z}\psi_{0,1}(x_0, x_1) \cdots \psi_{i-1,i}(x_{i-1}, x_i) \cdot \psi_{i,i+1}(x_i, x_{i+1}) \cdot \psi_{n-1,n}(x_{n-1}, x_n)$$

$$p(x_i) = \sum_{x_{\neq i}} p(\boldsymbol{x}) = \frac{1}{Z} \underbrace{\left(\sum_{x_{i-1}} \psi_{i-1,i}(x_{i-1}, x_i) \cdots \left(\sum_{x_1} \psi_{1,2}(x_1, x_2) \left(\sum_{x_0} \psi(x_0, x_1)\right)\right)\right)}_{=:\mu_{\rightarrow}(x_i)}$$

$$\cdot \underbrace{\left(\sum_{x_{i+1}} \psi_{i,i+1}(x_i, x_{i+1}) \cdots \left(\sum_{x_n} \psi_{n-1,n}(x_{n-1}, x_n)\right)\right)}_{=:\mu_{\leftarrow}(x_i)} = \frac{1}{Z}\mu_{\rightarrow}(x_i) \cdot \mu_{\leftarrow}(x_i).$$

$$p(x_i) = \frac{1}{Z} \underbrace{\left(\sum_{x_{i-1}} \psi_{i-1,i}(x_{i-1}, x_i) \cdots \left(\sum_{x_1} \psi_{1,2}(x_1, x_2) \left(\sum_{x_0} \psi(x_0, x_1) \right) \right) \right)}_{=: \mu_{\rightarrow}(x_i)} \cdot \underbrace{\left(\sum_{x_{i+1}} \psi_{i,i+1}(x_i, x_{i+1}) \cdots \left(\sum_{x_n} \psi_{n-1,n}(x_{n-1}, x_n) \right) \right)}_{=: \mu_{\leftarrow}(x_i)}.$$

- Marginal can be computed **locally**

$$p(x_i) = \frac{1}{Z} \mu_{\rightarrow}(x_i) \cdot \mu_{\leftarrow}(x_i) \quad \text{with} \quad Z = \sum_{x_i} \mu_{\rightarrow}(x_i) \cdot \mu_{\leftarrow}(x_i)$$

$$p(x_i) = \frac{1}{Z} \underbrace{\left(\sum_{x_{i-1}} \psi_{i-1,i}(x_{i-1}, x_i) \cdots \left(\sum_{x_1} \psi_{1,2}(x_1, x_2) \left(\sum_{x_0} \psi(x_0, x_1) \right) \right) \right)}_{=: \mu_{\rightarrow}(x_i)} \cdot \underbrace{\left(\sum_{x_{i+1}} \psi_{i,i+1}(x_i, x_{i+1}) \cdots \left(\sum_{x_n} \psi_{n-1,n}(x_{n-1}, x_n) \right) \right)}_{=: \mu_{\leftarrow}(x_i)}.$$

- Messages are recursive, thus computational complexity is $\mathcal{O}(n \cdot k^2)$

$$\mu_{\rightarrow}(x_i) = \sum_{i-1} \psi_{i-1,i}(x_{i-1}, x_i) \mu_{\rightarrow}(x_{i-1}) \qquad \mu_{\leftarrow}(x_i) = \sum_{x_{i+1}} \psi_{i,i+1}(x_i, x_{i+1}) \mu_{\leftarrow}(x_{i+1}).$$

- By storing local messages, **all marginals** can be computed in $\mathcal{O}(n \cdot k^2)$ (cf. filtering and smoothing)
- To compute one message from the preceding one, take the **sum** of the **product** of local factor and incoming message. To compute a local marginal, take the **sum** of the **product** of the incoming messages.

Directed and Undirected Graphs

- are tools to design models and algorithms. They each have different strengths and weaknesses:
  - **generative models** can be best represented as a **directed graph**
  - **computational separation** can be best captured by **Markov Random Fields**
- mapping from directed to undirected graphs often creates denser graphs that do not capture all conditional independence

Factor Graphs

- are a tool to directly represent an entire computation in a formal language (which also includes the functions in question themselves)
- both directed and undirected graphical models can be mapped onto factor graphs.

Inference on Chains

- separates into **local messages** being sent forwards and backwards along the factor graph
- both the local marginals and the *most-probable state* can be inferred in this way