Factor graphs are a convenient detailed view of factorisation properties of a pdf. For us they represent mainly a very practical calculational tool for:

- Marginalisation, inference, sampling.

But they do not bring out clearly independence statements and should be viewed as complementing BN and MRF concepts. Also, when you do modelling you would typically not get the factor graph directly, but rather the BN or MRF.

**Definition.** A distribution is said to have a factor graph representation if it can be written in the form

\[
P(x) = \frac{1}{Z} \prod_{s \in \mathcal{V}} f_s(x_s)
\]

where \( Z = \sum_{x \in \mathcal{S} \times \mathcal{V}} \prod_{s \in \mathcal{V}} f_s(x_s) \), and the product runs over a certain number of subsets \( S \subseteq \mathcal{V} \) (not necessarily cliques here!) and \( x_s = \{x_i\}_{i \in S} \). Here \( f_s(x_s) > 0 \) are not normalised.
The factor graph is

\[ f_a(x_1), f_b(x_1, x_2), f_c(x_1, x_2, x_3), f_d(x_2). \]

\[ p(x) = \frac{1}{n^2} f_a(x_1) f_b(x_1, x_2) f_c(x_1, x_2, x_3) f_d(x_2). \]

\[ \sim \]

Factor graph is Bivalent: Variable node, \( f \) constraint on factor node.
II. BN & MRF conversion to Factor Graphs.

II.1: MRF → factor graph conversion.

By Hammersley-Clifford, a MRF can be written as:

\[ p(x) = \frac{1}{Z} \prod_{C \in \text{max cliques}} \psi_c(x_c) \]

associate 0 to variable \( x_1, \ldots, x_N \),
associate 1 to max cliques \( C \).

Example:

The factor graph is not unique.
Example of the Ising model on square lattice \( \mathbb{Z}^2 \).

\[
p(\sigma) = \prod_{ij \in E} \exp \left( h \cdot \sigma_i \sigma_j - J \sigma_i \sigma_j \right)
\]

\( \mathbf{G} \) MRF on the graph

\( \mathbf{G} \) factor graph

Max cliques are \((ij) \in E\).

(And \( i \in V \) are not max cliques.)
II, 2. BN → Faith graph conversion

\[ p(x) = \prod_{i=1}^{N} p(x_i | \text{pa}(x_i)) \quad \text{a BN} \]

associate \( \odot \) to \( x_1, \ldots, x_N \)

associate \( \Box \) to \( p(x_i | \text{pa}(x_i)) \)

Example.

\[ p(x) = p(x_1) \cdot p(x_2) \cdot p(x_3 | x_1, x_2) \]
Factor graphs are useful devices to perform marginalization. It turns out marginalization by message passing is exact on trees (see below). So it is interesting to look more precisely when is a factor graph a tree?

**Definition:** A factor graph is a tree if there are no loops in its bipartite graph of variable and factor nodes.

**Definition:** A MRF has a tree structure if its undirected graph has no loops (basically same definition as for factor graphs).

**Definition:** A BN is a polytree if the graph (without taking directions of edges into account) is a tree. If furthermore there is a unique node with no parents (the ancestor of everybody!) and all nodes have a unique parent then we call the BN a Tree.
Properties of conversions for trees

* A tree MRF converted to a factor graph (say by the canonical conversion above) yields a tree factor graph.

* A polytree BN converted to a factor graph (say by the canonical conversion above) yields a tree factor graph.

* Remark that by Moralization we converted BN to MRF, but a polytree BN does not become a tree MRF;

* A tree BN yields a tree MRF after Moralization.

So moralization does not introduce new edges.
Important tasks for learning are:

- Compute marginals $p(x_i)$ & $p(x_i, x_j)$ e.g. in order to fit them to empirical frequency to learn parameters.

- Marginalize over hidden variables e.g. in ABM in order to learn parameters from samples.

- Learn the underlying graph itself.

The main method for efficient marginalization is by "message passing" / "belief propagation" / "sum-product algorithm" (synonymous).

In general this method is approximate and there are no convergence guarantees. But on tree factor graphs it is exact. Here we derive the message passing algorithm for trees (and discuss the implementation on general factor graphs after).

[See typed notes on Web page →]
For high-dimensional pdf the elementary sampling methods (rejection sampling, importance sampling) fail or are not well suited or reliable. Our main aim is to introduce the Markov Chain Monte Carlo (MCMC) method and in particular the popular easily applicable sub-case of Gibbs sampling also known as heat bath dynamics or Glauber dynamics.

Before introducing MCMC we briefly discuss the easiest topic of sampling from Belief Networks with ancestral sampling.

### 4.1. Belief Networks and ancestral sampling

Recall $f \equiv BN$:

$$p(x) = \frac{1}{Z} p(x_c | (p_a_i)_{i=1})$$

We can order variables such that parent variables always come before children (causal generations). One can show that this is always possible for a Directed Acyclic Graph (DAG): $(pa_i \not\rightarrow c_i)$ (partial order).
For example:

\[ p(x) = p(x_1)p(x_2)p(x_3|x_2, x_1) \times p(x_4|x_3) \times p(x_5|x_4, x_3) \times p(x_6|x_5, x_5) \]

We sample in the order \( x_1, x_2, x_3, x_4, x_5, x_6 \) i.e. from ancestor to later generations.

**Forward Ancestral Sampling Algorithm**

\[
\hat{x}_1 \sim p(x_1) \\
\hat{x}_2 \sim p(x_2) \\
\hat{x}_3 \sim p(x_3 | \hat{x}_2, \hat{x}_1) \\
\hat{x}_4 \sim p(x_4 | \hat{x}_3) \\
\hat{x}_5 \sim p(x_5 | \hat{x}_4) \\
\hat{x}_6 \sim p(x_6 | \hat{x}_5, \hat{x}_5)
\]

This is an exact procedure, however, if we run algorithm \( m \) times we have \( m \) samples \( \hat{X}^{(1)}, \ldots, \hat{X}^{(m)} \) which are all independent.
Remarks:

1) If we have some variable which is evidential, say $X_6$ is observed to a value $\overline{X}_6$ and we want to sample from $p(X_1, X_2, X_3, X_4, X_5 | \overline{X}_6)$. Then we cannot use ancestral sampling, as such because

$$p(X_1, X_2, X_3, X_4, X_5 | \overline{X}_6) = \frac{p(X_1)p(X_2)p(X_3 | X_2, X_1)p(X_4 | X_5)p(X_6 | X_3, X_5)}{\sum_{X_1, X_2, X_3, X_4, X_5} (\text{Numerical})}.$$ 

So a priori one would have to compute the marginal $p(\overline{X}_6)$ in the denominator and ancestral sampling is not enough for that. One could still use ancestral sampling combined with rejections each time $\overline{X}_6 \neq \overline{X}_6$. But this in practice turns out to be very costly.

2) In nice cases it could be that the marginal above in the denominator equals 1 (for example this is the case for $p(X_5) = 1$; exercise!). Then one can use ancestral sampling again for $p(X_1, X_2, X_3, X_4, X_6 | \overline{X}_5)$.

3) In a nutshell if we have a denominator (Normalizing factors) which is non-trivial then it is difficult to use ancestral sampling; and MCMC is very useful even for BN.
As said before this is useful for sampling MRF, factor graph, and conditional probabilities of BN's. Because of the Markov property of MRF's it can be often implemented nicely (heat bath dynamics in next paragraph). It is however not an exact method contrary to ancestral sampling and the samples are not strictly iid.

The general idea is to construct a Markov Chain s.t. $p(x)$ is the stationary distribution of this MCMC. Then one runs the MC starting from an initial state: $X^{(0)} \rightarrow X^{(1)} \rightarrow \ldots \rightarrow X^{(t)} \rightarrow \ldots \rightarrow X^{(T)}$

We hope that for $T$ large enough $X^{(T)}$ is approximately distributed according to the stationary distribution $p(x)$. \[\text{This is strictly true for } T=\infty\]. In practice because $X^{(T)}$, $X^{(T+1)}$, $X^{(T+2)}$, \ldots are not independent we take samples every $M$-th iteration: $X^{(T)}$, $X^{(T+M)}$, $X^{(T+2M)}$, \ldots for $M$ large. Or we run the chain many times from many initial conditions.

This is costly \ldots
IV. 2. (a) Recapitulation on Markov Chains

Take a finite discrete state space \( S = \{1, 2, \ldots, 181\} \). A MC is a stochastic process \( \{X(t), t = 0, 1, \ldots\} \) with 
\( X(t) \in S \) satisfying

\[
q \left( X(t) \mid X(t-1) = x^{(t-1)} \right) = q \left( X(t) \mid X^{(t-1)} \right)
\]

We consider homogeneous (in time) chains for which

\[
q \left( X(t) = j \mid X(t-1) = i \right) = Q_{ij} = Q_{ij} \quad \text{independent of } t.
\]

\( Q_{ij} \) is a \( |S| \times |S| \) matrix called the transition matrix.

Note that \( \sum_{j=1}^{181} Q_{ij} = 1 \). [for \( i \) you go somewhere with probability one].

Let \( \Pi_t(\mathcal{X}^t) \) be the probability of the state at time \( t \).

It is easy to show (the Chapman-Kolmogorov equation)

\[
\Pi_t(\mathcal{X}^t) = \sum_{X^{t-1} \in S} q \left( \mathcal{X}^t \mid \mathcal{X}^{t-1} \right) \Pi_{t-1}(\mathcal{X}^{t-1})
\]

Definition: A stationary distribution is a distribution satisfying:

\[
\Pi_{stat}(\mathcal{X}) = \sum_{X' \in S} q \left( \mathcal{X} \mid \mathcal{X}' \right) \Pi_{stat}(X').
\]

It may not always exist. If it exists it may not be unique.
Definition: A Markov chain is called irreducible if the transition matrix \( Q \) connects any state \( i \) and \( j \) in finite time and in both directions, i.e., there exist \( m_i, m_j \) such that
\[
(Q^m_{m_i})_{i,j} \neq 0 \quad \text{and} \quad (Q^m_{m_j})_{j,i} \neq 0.
\]

Theorem: For an irreducible Markov chain in a finite state space, there always exist a unique stationary distribution.

Definition: We say that an irreducible Markov chain in a finite state space is ergodic if for all \( \pi \in \mathbb{S} \) we have
\[
\Pi_t^e(\pi) \to \Pi_{\text{stat}}^e(\pi) \quad \text{as} \quad t \to +\infty.
\]

Theorem: An irreducible and aperiodic Markov chain in a finite state space is ergodic. [A criterion for aperiodicity is that there exist no loops or \( Q \) has not the eigenvalue \(-1\) in spectrum.]

Definition: A Markov chain is said to satisfy the detailed balance condition if its stationary distribution satisfies
\[
q(x|x')\Pi_{\text{stat}}(x') = q(x'|x)\Pi_{\text{stat}}(x).
\]

[i.e., mass transfer \( x' \to x \) = mass transfer \( x \to x' \).]
Remark: any distribution that satisfies the detailed balance condition must be stationary. Indeed:

\[ q(x' | x') \pi(x') = q(x' | x) \pi(x) \]

\[ \Rightarrow \quad \sum_{x} q(x' | x') \pi(x') = \sum_{x} q(x' | x) \pi(x) \]

\[ \Rightarrow \quad \pi(x') = \sum_{x} q(x' | x) \pi(x) \]

which is the definition of stationarity.

### IV. 2.6 The Metropolis-Hastings MCMC method

Let \( \rho(x) \) a dirichlet from which we want to sample. We construct \( q(x' | x) \) s.t. the MC is irreducible and has dirichlet \( \rho(x) \).

**Metropolis-Hastings Method**

Take a "base chain" or "proposal chain" \( \tilde{q}(x' | x) \) over state space \( S \). We take \( \tilde{q} \) s.t. it is an irreducible chain.

- Start chain at \( x^{(0)} \) at time \( t=0 \).
- At time \( t \) generate \( x^{t+1} \sim \tilde{q}(x^{t+1} | x^t) \) "proposed".
- Accept \( x^{t+1} \) with probability \( A(x^{t+1}, x^t) \) where:
\[ A(x^{t+1}, x^t) = \min \left[ 1, \frac{q(x^{t+1} | x^t) p(x^{t+1})}{q(x^t | x^{t+1}) p(x^t)} \right] \]

- If \( x^{t+1} \) is accepted it forms the new state. If it is not accepted keep the state \( x^t \).
- Iterate.

Properties of the Metropolis-Hastings chain (not difficult to show):

(i) It is irreducible (because \( q \) is)

(ii) It is aperiodic (because rejection probability introduces self-loops)

(iii) On a finite state space, by the previous there is a unique stationary distribution and the chain is ergodic, i.e.

\[ \pi^*(x) \to \pi_{\text{stationary}}(x) = p(x) \quad \forall \, x \in S. \]

(iv) The chain satisfies detailed balance condition.
Special Case of Metropolis–Hastings (called Metropolis).

For a symmetric proposal chain \( q(x' | x) = q(x | x') \) we have
\[
A(x', x) = \min\left(1, \frac{p(x')}{p(x)}\right).
\]
The new sample \( x' \) is accepted with probability \( \frac{p(x')}{p(x)} \) if it is more probable: \( p(x') \geq p(x) \). It is accepted with lower probability \( p(x') \leq p(x) \) with probability \( p(x') < p(x) \).

Proof of detailed balance condition for the Metropolis–Hastings chain

We should show
\[
q(x' | x) p(x) = q(x | x') p(x').
\]
For \( x = x' \) this is trivial.

For \( x \neq x' \):
\[
q(x' | x) = A(x', x) \frac{q(x' | x)}{q(x | x')},
\]
acceptance, proposed prob.

So we should check if
\[
A(x', x) \tilde{q}(x' | x) p(x) \approx A(x, x') \tilde{q}(x | x') p(x'),
\]
\[\text{i.e. } \min\left(\tilde{q}(x' | x) p(x); \tilde{q}(x | x') p(x')\right) \approx \min\left(\tilde{q}(x | x') \frac{p(x)}{p(x')}; \tilde{q}(x' | x) \frac{p(x')}{p(x)}\right)\]
which is obviously true.
**Algorithm for Gibbs sampling**

- Take index $i$ at random in $\{1, \ldots, n\}$ and make the move $x \rightarrow x'$ where
  
  \[
  \begin{cases}
  x_i' \sim p(x_i' | \{x_j\}_{j \neq i}) \\
  x_j' = x_j \text{ for } j \neq i
  \end{cases}
  \]

One can see that this corresponds to a Metropolis-Hastings chain with proposal or ten chain

\[
\tilde{q}(x' | x) = \begin{cases} 
\frac{1}{N} p(x_i' | \{x_j\}_{j \neq i}) & \text{if } x'_i = x_i \text{ for all } j \neq i \\
0 & \text{otherwise}
\end{cases}
\]

It is also possible to see that $A(x', x) = 1$.
Let \( p(x) = \frac{1}{\mathcal{Z}} \prod_{c} \psi_c(x_c) \)

where \( c \) runs over Max-Cliques or factor nodes of a
factor graph. Note that \( \mathcal{Z} = \sum_{\mathbf{x}} \prod_{c} \psi_c(x_c) \)
is intractable. But note also that it simplifies in ratios \( \frac{p(x')}{p(x)} \). Therefore, we do not need to compute \( \mathcal{Z} \). Moreover, in
Gibbs sampling, we change one node at a time and
the HRF property becomes very handy:

\[ x = (x_1, \ldots, x_N) \rightarrow x' = (x'_1, \ldots, x'_N) = (x_1, \ldots, x_{i-1}, x'_i, x_{i+1}, \ldots, x_N) \]

where \( i \) is drawn at random in \( \{1, \ldots, N\} \) and

\[ p(x'_i | x_{-i}^{'}, x_c) = p(x'_i | \text{MB}(c)) \]

The r.h.s. can be written more explicitly as

\[ \prod_{c: i \in c} \psi_c(x'_c) \quad \xrightarrow{c: c \not= i} \quad \int dx'_c \prod_{c: c \not= i} \psi_c(x'_c) \]

for which we sample by
low-dim. methods.
Example of the Ising model (or Boltzmann machine)

\[ P_{\text{Ising}}(S) = \frac{\exp \left\{ \sum_{\langle ij \rangle} J_{ij} s_i s_j + \sum_i h_i s_i \right\}}{Z} \]

where \( s_i \in \{-1, +1\} \). The MC move the move

\[ S = (s_1, \ldots, s_N) \rightarrow S' = (s_1, \ldots, s_{i-1}, s_i', s_{i+1}, \ldots, s_N) \quad \text{for some} \quad i \in \{1, \ldots, N\} \quad \text{with probability} \]

\[ P(s_i' \mid s_i, \ldots, s_N) = P(s_i' \mid \text{flip}(i)) \]

\[ = \frac{\exp \left\{ \sum_{\langle ij \rangle} J_{ij} s_i' s_j + h_i s_i' \right\}}{\sum_{s_i'} \exp \left\{ \sum_{\langle ij \rangle} J_{ij} s_i' s_j + h_i s_i' \right\}} \]

\[ = \frac{\exp \left\{ s_i' \left( \sum_{\langle ij \rangle} J_{ij} s_j + h_i \right) \right\}}{2 \cosh \left( \sum_{\langle ij \rangle} J_{ij} s_j + h_i \right)} \]

\[ = \frac{1}{2} \left\{ 1 + s_i' \tanh \left( \frac{\sum_{\langle ij \rangle} J_{ij} s_j + h_i}{2} \right) \right\} \]

**Summary of Algorithm**

- Pick \( i \) uniformly at random in \( \{1, \ldots, N\} \)
- Set \( s_i' = \pm 1 \) with Prob = \( \frac{1}{2} \left\{ 1 \pm \tanh \left( \sum_{\langle ij \rangle} J_{ij} s_j + h_i \right) \right\} \)
- Iterate

This yields \( S^{(0)}, S^{(1)}, \ldots, S^{(T)} \). For \( T \to \infty \) we have \( S^{(T)} \sim P_{\text{Ising}}(S) \).