Introduction to Graphical Models I

Hennie de Villiers

Computer Science Division, Stellenbosch University

These slides are adapted from those accompanying the book Bayesian Reasoning and Machine Learning. The book and demos can be downloaded from www.cs.ucl.ac.uk/staff/D.Barber/brml. We acknowledge David Barber for providing the original slides.
Graphical Models

GMs are graph based representations of various factorisation assumptions of distributions. These factorisations are typically equivalent to independence statements amongst (sets of) variables in the distribution.

**Belief Network**  Each factor is a conditional distribution. Generative models, AI, statistics. Corresponds to a DAG.

**Markov Network**  Each factor corresponds to a potential (non negative function). Related to the strength of relationship between variables, but not directly related to dependence. Useful for collective phenomena such as image processing. Corresponds to an undirected graph.

**Chain Graph**  A marriage of BNs and MNs. Contains both directed and undirected links.

**Factor Graph**  A barebones representation of the factorisation of a distribution. Often used for efficient computation and deriving message passing algorithms.

**The GM zoo**  There are many more kinds of GMs, each useful in its own right. We’ll touch on some more when we consider inference.
Undirected Graph

All the edges are undirected:

A clique is a fully connected subset of nodes. \((X_1, X_2, X_4)\) forms a (non-maximal) clique.

Maximal Clique
Clique which is not a subset of a larger clique. \((X_1, X_2, X_3, X_4)\) and \((X_2, X_3, X_5)\) are both maximal cliques.
Markov Network

Clique: Fully connected subset of nodes.
Maximal Clique: Clique which is not a subset of a larger clique.

A Markov Network is an undirected graph in which there is a potential (non-negative function) $\psi$ defined on each maximal clique.

The joint distribution is proportional to the product of all clique potentials.

$$ p(A, B, C, D, E) = \frac{1}{Z} \psi(A, C)\psi(C, D)\psi(B, C, E) $$

$$ Z = \sum_{A,B,C,D,E} \psi(A, C)\psi(C, D)\psi(B, C, E) $$
Belief Networks are Markov Networks

Belief networks can be represented as Markov networks. This example has the joint distribution

\[ p(A)p(B)p(C|A, B)p(D|C)p(E|B, C) \]

which is just a special case of

\[ \frac{1}{Z} \psi(A)\psi(B)\psi(C, A, B)\psi(D, C)\psi(E, B, C) \]

We can redraw the belief network as a Markov network by **moralization** (connecting common parents together with undirected links) and making all links undirected.
Example: The Ising model

\[ x_i \in \{+1, -1\}: \]

\[ p(x_1, \ldots, x_9) = \frac{1}{Z} \prod_{i \sim j} \phi_{ij}(x_i, x_j) \]

\[ \phi_{ij}(x_i, x_j) = e^{-\frac{1}{2T}(x_i - x_j)^2} \]

\( i \sim j \) denotes the set of indices where \( i \) and \( j \) are neighbours in the graph. The potential encourages neighbours to be in the same state.

---

Spontaneous global behaviour

\[ M = \left| \sum_{i=1}^{N} x_i \right| / N. \] As the temperature \( T \) decreases towards the critical temperature \( T_c \) a phase transition occurs in which a large fraction of the variables become aligned in the same state. Even though we only ‘softly’ encourage neighbours to be in the same state, for a low but finite \( T \), the variables are all in the same state. Paradigm for ‘emergent behaviour’.
Example Application of Markov Network – Part I

Problem: We want to recover a binary image from the observation of a corrupted version of it.

\[ X = \{X_i, i = 1, \ldots, D\} \quad X_i \in \{-1, 1\}: \text{clean pixel} \]

\[ Y = \{Y_i, i = 1, \ldots, D\} \quad Y_i \in \{-1, 1\}: \text{corrupted pixel} \]

\[ \phi(Y_i, X_i) = e^{\gamma X_i Y_i} \quad \text{encourage } Y_i \text{ and } X_i \text{ to be similar} \]

\[ \psi(X_i, X_j) = e^{\beta X_i X_j} \quad \text{encourage the image to be smooth} \]

\[ p(X, Y) \propto \left[ \prod_{i=1}^{D} \phi(Y_i, X_i) \right] \left[ \prod_{i \sim j} \psi(X_i, X_j) \right] \]

Finding the most likely \( X \) given \( Y \) is not easy (since the graph is not singly-connected), but approximate algorithms often work well.
Example Application of Markov Network – Part II

left  Original clean image
middle Observed (corrupted) image
right Most likely clean image \( \arg\max_X p(X|Y) \)
Properties of Markov Networks

\[ p(A, B, C) = \phi_{AC}(A, C)\phi_{BC}(B, C)/Z \]

Marginalising over \( C \) makes \( A \) and \( B \) (graphically) dependent. In general
\[ p(A, B) \neq p(A)p(B). \]

Conditioning on \( C \) makes \( A \) and \( B \) independent:
\[ p(A, B|C) = p(A|C)p(B|C). \]
General Rule for Independence in Markov Networks

- Remove all links neighbouring the variables in the conditioning set $Z$.
- If there is no path from any member of $X$ to any member of $Y$, then $X$ and $Y$ are conditionally independent given $Z$. 
Alternative Rule for Independence in Belief Networks

\( \mathcal{X} \Perp \mathcal{Y} | \mathcal{Z} \) ?

- **Ancestral Graph**: Remove any node which is neither in \( \mathcal{X} \cup \mathcal{Y} \cup \mathcal{Z} \) nor an ancestor of a node in this set, together with any edges in or out of such nodes.

- **Moralisation**: Add a line between any two nodes which have a common child. Remove arrowheads.

- **Separation**: Remove all links from \( \mathcal{Z} \).

- **Independence**: If there are no paths from any node in \( \mathcal{X} \) to one in \( \mathcal{Y} \) then \( \mathcal{X} \Perp \mathcal{Y} | \mathcal{Z} \).

![Diagram](image)
Alternative Rule for Independence in Belief Networks

$\mathcal{X} \perp \mathcal{Y} | \mathcal{Z}$?

- **Ancestral Graph:** Remove any node which is neither in $\mathcal{X} \cup \mathcal{Y} \cup \mathcal{Z}$ nor an ancestor of a node in this set, together with any edges in or out of such nodes.

- **Moralisation:** Add a line between any two nodes which have a common child. Remove arrowheads.

- **Separation:** Remove all links from $\mathcal{Z}$.

- **Independence:** If there are no paths from any node in $\mathcal{X}$ to one in $\mathcal{Y}$ then $\mathcal{X} \perp \mathcal{Y} | \mathcal{Z}$. 

Alternative Rule for Independence in Belief Networks

$\mathcal{X} \perp \mathcal{Y} | \mathcal{Z}$?

- **Ancestral Graph:** Remove any node which is neither in $\mathcal{X} \cup \mathcal{Y} \cup \mathcal{Z}$ nor an ancestor of a node in this set, together with any edges in or out of such nodes.

- **Moralisation:** Add a line between any two nodes which have a common child. Remove arrowheads.

- **Separation:** Remove all links from $\mathcal{Z}$.

- **Independence:** If there are no paths from any node in $\mathcal{X}$ to one in $\mathcal{Y}$ then $\mathcal{X} \perp \mathcal{Y} | \mathcal{Z}$. 

Alternative Rule for Independence in Belief Networks

$\mathcal{X} \perp \mathcal{Y} | \mathcal{Z}$?

- **Ancestral Graph**: Remove any node which is neither in $\mathcal{X} \cup \mathcal{Y} \cup \mathcal{Z}$ nor an ancestor of a node in this set, together with any edges in or out of such nodes.

- **Moralisation**: Add a line between any two nodes which have a common child. Remove arrowheads.

- **Separation**: Remove all links from $\mathcal{Z}$.

- **Independence**: If there are no paths from any node in $\mathcal{X}$ to one in $\mathcal{Y}$ then $\mathcal{X} \perp \mathcal{Y} | \mathcal{Z}$.
The Boltzmann machine

A MN on binary variables \( \text{dom}(x_i) = \{0, 1\} \) of the form

\[
p(x|w, b) = \frac{1}{Z(w, b)} e^{\sum_{i<j} w_{ij} x_i x_j + \sum_i b_i x_i}
\]

where the interactions \( w_{ij} \) are the ‘weights’ and the \( b_i \) the biases.

- This model has been studied in the machine learning community as a basic model of distributed memory and computation. The \( x_i = 1 \) represents a neuron ‘firing’, and \( x_i = 0 \) not firing. The matrix \( w \) describes which neurons are connected to each other. The conditional

\[
p(x_i = 1|x_{\setminus i}) = \sigma \left( b_i + \sum_{j \neq i} w_{ij} x_j \right), \quad \sigma(x) = e^x/(1 + e^x).
\]

- The graphical model of the BM is an undirected graph with a link between nodes \( i \) and \( j \) for \( w_{ij} \neq 0 \). For all but specially constrained \( w \) inference will be typically intractable.

- Given a set of data \( x^1, \ldots, x^n \), one can set the parameters \( w, b \) by maximum likelihood (though this is computationally difficult).
Factor Graphs

A square node represents a factor (non-negative function) of its neighbouring variables.

The joint function is the product of all factors:

\[ f(A, B, C, D, E) = f_1(A, B)f_2(B, C, D)f_3(C, E)f_4(D, E) \]

Factor graphs are useful for performing efficient computations (not just for probability).
Variable Elimination

Factor graphs allow us to precisely represent the elimination of variables.

\[ f(A, B, C, D, E) = f_1(A, B)f_2(B, C, D)f_3(C, E)f_4(D, E) \]
Variable Elimination

Factor graphs allow us to precisely represent the elimination of variables.

\[ f(B, C, D, E) = \sum_{A} f_1(A, B) f_2(B, C, D) f_3(C, E) f_4(D, E) \]

\[ = \sigma_{A \rightarrow B}(B) f_2(B, C, D) f_3(C, E) f_4(D, E) \]
Factor Graphs versus Markov Networks

(a) $\phi(a, b, c)$
(b) $\phi(a, b)\phi(b, c)\phi(c, a)$
(c) $\phi(a, b, c)$

- Both (a) and (b) have the same Markov network (c).
- Whilst (b) contains the same (lack of) independence statements as (a), it expresses more constraints on the form of the potential.
Expressiveness of Belief and Markov Networks

Cannot represent independence information in certain belief networks with a Markov network.

A Belief network

Markov representation?
Since we have a term $p(C|A, B)$, the MN must have the clique $A, B, C$:
Expressiveness of Belief and Markov Networks

Cannot represent independence information in certain Markov networks with a Belief network.

A Markov network

\[ A \]
\[ B \quad C \quad D \]

\[ B \perp C \mid A, D \]

Belief Network representation?

Any DAG on \( A, B, C, D \) must have a collider.

\[ A \]
\[ B \quad C \quad D \]

\[ B \parallel C \mid A, D \]