Chapter 8

Basics of Graphical Models

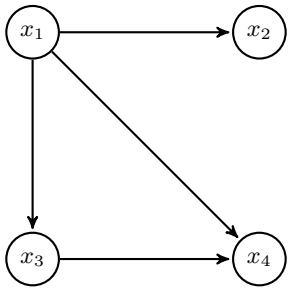
8.1 Introduction

Graphical models (GMs) are used to represent distributions on graphs. They enable us to *represent conditional independencies* and *factorization of distributions* facilitate probabilistic inference through message passing algorithms. There are different types of GMs:

- Bayesian Networks (BN, aka Directed Graphical Models): Natural for representing causal relationships
- Markov Random Fields (MRF, aka Undirected Graphical Models): Suitable for representing co-influence or non-causal relationships among a subsets of variables, e.g., friendship in social networks and pixels in an image (adjacent pixels are likely to have similar colors).
- Factor Graphs: A flexible type of GM that can represent distributions represented by BNs and MRFs.

8.2 Bayesian Networks

A Bayesian network is a **directed acyclic graph** (DAG) with some additional attributes. A DAG is a graph whose edges have direction and in which there is no cycle if one follows the edges based on their direction. In a DAG, a **parent** of a node y is a node x such that there is an edge from x to y. A **child** of y is a node z such that y is the parent of z. An **ancestor** is a parent, parent of a parent, etc., and a **descendant** is a child, child of a child, etc. A **complete DAG** is a DAG such that with an edge between each pair of vertices. An example of a DAG with four nodes is shown below.



In a Bayesian network represented by a DAG G:

- Nodes $x_1, \ldots, x_m$ represent variables or quantities (can be scalar or vector)
- Edges represent causal relationships

- The probability distribution over $x_1^m = x_1, \ldots, x_m$ can be expressed as:

$$p(x_1^m) = \prod_{i=1}^{m} p(x_i | \, \mathrm{pa}(x_i))$$

  where $\mathrm{pa}(x_i)$ are the parents of x_i in G, i.e., nodes with an edge *to* x_i.

We then say that the distribution p **factorizes** with respect to G. For example, for a distribution p that factorizes with respect to the graph shown above, we have

$$p(x_1, x_2, x_3, x_4) = p(x_1)p(x_2|x_1)p(x_3|x_1)p(x_4|x_1, x_3). \tag{8.1}$$

What does (8.1) tell us about the distribution? Recall that based on the chain rule of probability, we always have

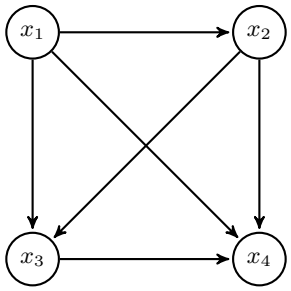
$$p(x_1, x_2, x_3, x_4) = p(x_1)p(x_2|x_1)p(x_3|x_1, x_2)p(x_4|x_1, x_2, x_3). \tag{8.2}$$

It is straightforward to show that (8.1) is equivalent to

$$\begin{aligned} p(x_3|x_1, x_2) &= p(x_3|x_1), \\ p(x_4|x_1, x_2, x_3) &= p(x_4|x_1, x_3). \end{aligned} \tag{8.3}$$

These two expressions are conditional independence statements, which we can restate as $x_3 \perp\!\!\!\perp x_2 \,|\, x_1$ and $x_4 \perp\!\!\!\perp x_2 \,|\, x_1, x_3$. Thus saying that p factorizes with respect to the graph above is equivalent to assuming (8.3). This is in general true. The set of missing incoming edges for each node in the graph represents a conditional independence assumption.

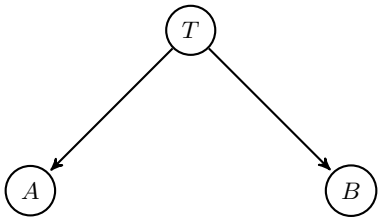
The complete graph, shown for four nodes below, represents the factorization given in (8.2), which holds for any distribution and thus the graph can represent any distribution. But such a graph is not particularly useful since the power of graphical models results from the independence assumptions that they encode.



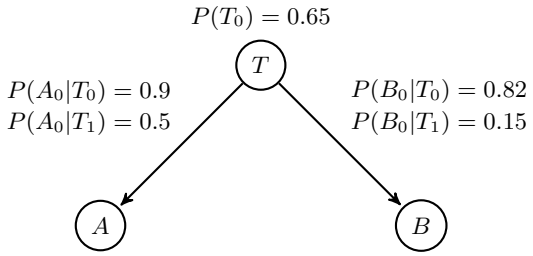
Note that the complete graph is acyclic as it imposes an ordering over the nodes (in this case, x_1, x_2, x_3, x_4). *We can view any Bayesian network as being obtained from a complete DAG by removing edges.* So every Bayesian network is also acyclic.

Example 8.1. Alice and Bob are employees of a business in Charlottesville, both of whom take 29S to get to work. We are interested in whether they arrive on time or late. We assume their arrival time is affected by traffic, which leads to dependence, but there aren't any other factors that can affect both of them. Let $A = 0$ and $A = 1$ denote Alice being on time and being late A_1, respectively and similarly for Bob ($B = 0$ and B_1). Traffic is either normal ($T = 0$) or heavy ($T = 1$). We use X_0 and X_1 as shorthand for $X = 0$ and $X = 1$ for our random variables.

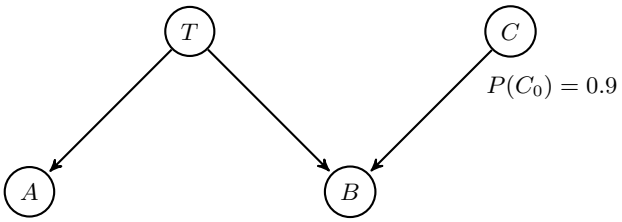
The Bayesian Network that models the probability distribution is shown below.



This graph implies that $A \perp\!\!\!\perp B|T$ and that $p(ABT) = p(T)p(A|T)p(B|T)$. We now have the **structure** of our model. But we still need the **conditional probability distributions** to complete the model. Suppose these distributions are as below:



Taking the example a step further, suppose that Bob has a son, Charlie (C_0 and C_1) who has to be dropped off at school. Charlie being late has an effect on Bob being late. We will adjust the Bayesian Network below and use the joint probability distribution in the following table.



CT	$P(B_0\|CT)$	$P(B_1\|CT)$
C_0T_0	0.9	0.1
C_0T_1	1/6	5/6
C_1T_0	0.1	0.9
C_1T_1	0	1

Note that this new conditional distribution does not change any previously calculated probabilities involving Traffic, Alice, and Bob, but the numbers were chosen specifically to achieve this—this is not always the case.

Based on this graph, the joint probability distribution is:

$$p(ABTC) = p(T)p(C)p(A|T)p(B|CT).$$

It is easy to show that $T \perp\!\!\!\perp C$ but as we will see below $T \not\perp\!\!\!\perp C|B$.

Bayesian networks facilitate certain kinds of reasoning. In **causal reasoning**, we draw conclusions about unobserved effects base on observed causes. For example, if we know there was heavy traffic, then it is more likely that Bob was late, $p(B_1|T_1) = 0.85 > p(B_1) = 0.41$. **Evidential reasoning** allows us to say something about the cause by observing the effects. For example,

$$p(T_1|B_1) = \frac{p(B_1|T_1)p(T_1)}{p(B_1)} = 0.7177 > p(T_1) = 0.35,$$

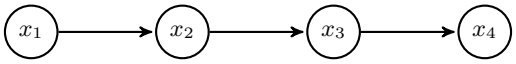
tells us that heavy traffic is more likely when Bob is late, even though we have no direct information about the traffic.

We also have $p(T_1|B_1C_1) < P(T_1|B_1)$, which makes intuitive sense. Bob being late provides evidence for traffic being heavy. But if we know Charlie is late, then we have an alternative explanation for Bob being late, lessening the need for traffic being heavy as a reason for Bob's tardiness. This type of reasoning, where given an effect, occurrence of one cause lessens the probability of another cause, is called **explaining away**. △

8.2.1 Markov Model

A **Markov Model** or a **Markov chain** is a Bayesian network whose graph consists of a single path. Such a model can, for example, represent the total winning of a gambler as a function of time, where each game is independent. The main assumption is that *given the present, the future is independent of the past*: how much money you'll have after the next game is independent of past games, if your current worth is known. Another, idealized example is weather forecast: Given that we know today's weather, past weather is irrelevant for the purpose of forecasting tomorrow's weather.

A Markov chain with four nodes is given below

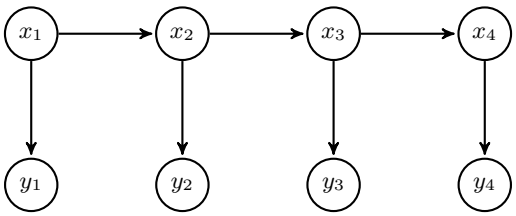


with an associated joint probability distribution that factorizes as

$$p(x_1^4) = p(x_1)p(x_2|x_1)p(x_3|x_2)p(x_4|x_3).$$

Consider a set of n random variables each of which can take on k different values. The most general probability distribution over these variables will have $k^n - 1$ parameters (the -1 comes from the fact that we know the probabilities must sum to one). In practice, this is such a huge number even for $k = 2$ and relatively small n, e.g., $n = 100$, that we can't even store the distribution, let alone learn it from data. The Markov model, however, has $(k-1) + (n-1)k(k-1)$ parameters, which is much more manageable. This is an example of graphical models making modeling more feasible.

A closely related model is the **hidden Markov model (HMM)**:



An HMM is used when the true state of the system cannot be directly observed but we can observe some function of the state. For example, x_i can represent if cancer is in remission or not and y_i can represent observations from medical tests.

Like Markov random fields, Markov and hidden Markov models are named after Russian mathematician Andrey Markov, but Markov models are Bayesian Networks and not Markov Random Fields.

8.2.2 Why graphical models?

Graphical models, such as Bayesian networks are useful for several reasons.

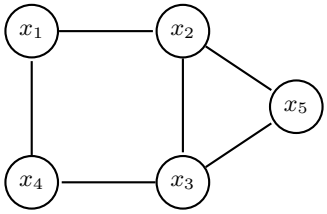
- They provide a simple but flexible way to encode conditional independencies, enabling us to answer questions about independence based on graphs.
- GMs help constructing tractable models. As an example, see the number of parameters for a Markov chain versus an unrestricted model described above.
- Restriction to GMs has computational benefits, allowing us to draw conclusions about hidden quantities based on observations efficiently using algorithms such as belief propagation.

8.3 Markov Random Fields

Definition 8.2 (Clique and maximal clique)**.** The following definitions from graph theory will be used in this section. In an undirected graph, a *clique* is a subset of nodes such that there is an edge between any

two of them. A *maximal clique* is a clique such that there are no nodes not in the clique that connected to all the nodes already in the clique.

Suppose that we are interested in developing a political party affiliation model for a group of 5 people (or millions of people if we have social network data). Let's assume their friendships are given by the following graph



in which each node x_i represents the party of person i and an edge between x_i and x_j means that i and j are friends. How can we develop a probability distribution that can help us in this task?

We would like to encode the following observations in our distribution. We know that if two people are friends (e.g., 1 and 2), then it is more likely for them to have a common political alignment. Furthermore, for three people who are all friends (2,3,5), it is perhaps even more likely that they share the same political views. Let party affiliation be denoted by 0 or 1. We define

$$\psi_{ij}(x_i, x_j) = \begin{cases} 1, & x_i = x_j \\ 1/2, & x_i \neq x_j \end{cases} \tag{8.4}$$

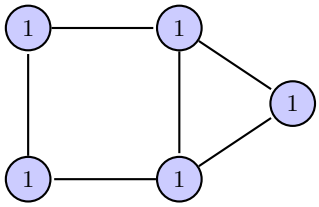
and

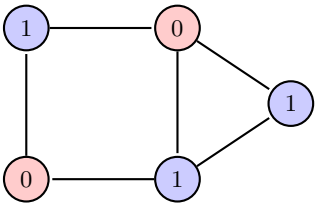
$$\psi_{ijk}(x_i, x_j, x_k) = \begin{cases} 1, & x_i = x_j = x_k \\ 1/2, & \text{if two of the three are equal} \end{cases} \tag{8.5}$$

So agreements are assigned a higher value. Now we can define a probability distribution as

$$p(x_1, \ldots, x_5) \propto \psi_{12}(x_1, x_2)\psi_{14}(x_1, x_4)\psi_{34}(x_3, x_4)\psi_{235}(x_2, x_3, x_5), \tag{8.6}$$

which assigns higher probability to configurations in which cliques of friends are in the same parties, as we wanted. For example, the probability of the left configuration is 16 times as likely to occur as the one on the right.





Note that there is no guarantee that the right side of (8.6) sums to 1 when going over all possible configurations so we need a normalization factor, which in this context is called the partition function,

$$Z = \sum_{x_1^5} \psi_{12}(x_1, x_2)\psi_{14}(x_1, x_4)\psi_{34}(x_3, x_4)\psi_{235}(x_2, x_3, x_5).$$

We can then write

$$p(x_1, \ldots, x_5) = \frac{1}{Z}\psi_{12}(x_1, x_2)\psi_{14}(x_1, x_4)\psi_{34}(x_3, x_4)\psi_{235}(x_2, x_3, x_5).$$

In our example, it turns out that $Z = 8.5$, and thus $p(1, 1, 1, 1, 1) = 0.11765$ while $p(1, 0, 1, 0, 1) = 0.0073529$.

Finally, we note while we chose the potential function for each pair and triple to be the same regardless of the identity of the nodes, this is not a necessity; for example, we could have chosen different functions for ψ_{12} and ψ_{34}.

We can now consider the general case. A **Markov random field (MRF)** or an undirected graphical model consists of an undirected graph G with nodes $x_1^m = x_1, \ldots, x_m$, and a probability distribution p that *factorizes with respect to* G, i.e.,

$$p(x_1^m) = \frac{1}{Z} \prod_{C \text{ is a clique in } G} \psi_C(x_C), \tag{8.7}$$

where for each clique C in G, x_C is the set of nodes in that clique, ψ_C is a *potential function*, which assigns non-negative values to all configurations of x_C, and Z is the *partition function*, which ensures that the right side is a proper distribution. Without loss of generality, we may assume the cliques are maximal by absorbing the potential functions for smaller cliques into the maximal clique. For our political party example above, for the clique with nodes x_2, x_3, x_5, we can either have 4 potential functions over all the sub-cliques,

$$\psi'(x_2, x_3)\psi'(x_3, x_5)\psi'(x_2, x_5)\psi'(x_2, x_3, x_5)$$

or a single potential function

$$\psi(x_2, x_3, x_5).$$

Both are valid and equally powerful in terms of representation.

When designing an MRF we incorporate local information into the potential functions, but the final result is that we learn about the global view of the entire system. Also, in an MRF, the relationships between nodes are symmetric rather than causal or directed.

8.3.1 Energy-based models

When for all configurations $\boldsymbol{x} = x_1^m$, the probability $p(\boldsymbol{x})$ is positive, it is helpful to represent the distribution as

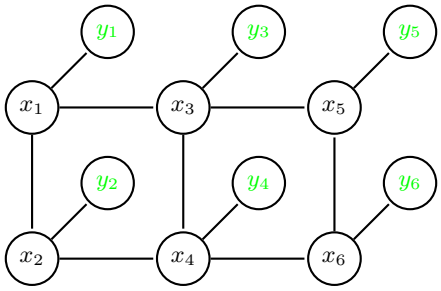
$$p(\boldsymbol{x}) \propto e^{-E(\boldsymbol{x})},$$

where $E(\cdot)$ is called the **energy function**. Such a distribution is also called a **Boltzmann distribution**. The terminology comes from statistical physics. In that context, lower energy corresponds to higher stability and thus higher probability for a system. For a graphical model, the energy function can be written as the sum of terms each of which correspond to a clique in the graph,

$$E(\boldsymbol{x}) = \sum_{C \text{ is a clique in } G} -\phi_C(\boldsymbol{x}_C) \quad \Rightarrow \quad p(\boldsymbol{x}) \propto \prod e^{\phi_C(\boldsymbol{x}_C)}$$

A **Boltzmann machine** is such a graphical model, typically with both nodes that can be observed and nodes that are hidden (latent).

Example 8.3 (**An MRF for denoising Images**). The figure below shows an MRF for a noisy black and white image. Here $x_1, x_2, \cdots, x_6$ represent the true B/W status of the pixels and $y_1, y_2, \cdots, y_6$ the noisy values (e.g., due to noise of a camera). We denote 'Black'=-1 and 'White' = 1.



The energy function can be written as

$$E(\boldsymbol{x}, \boldsymbol{y}) = -\sum_i^m \alpha_i x_i - \sum_{(i,j) \in \mathcal{E}(G)} \beta_{i,j} x_i x_j - \sum_i^m \zeta_i x_i y_i,$$

where $\mathcal{E}(G)$ is the set of edges between neighboring pixels and $\beta_{i,j} > 0$ and $\zeta_i > 0$. The α_i control how likely a pixel is to be white without considering other pixels. The interaction between neighboring pixels is controlled by β_{ij}; since each is positive, it is more likely for adjacent pixels to have the same status. We assume that it is more likely for the noisy pixel to match the true pixel and so $\zeta_i > 0$ as well.

In a denoising task, we are given $\boldsymbol{y}$ and our goal is to recover $\boldsymbol{x}$. A reasonable solution is

$$\arg\max_{\boldsymbol{x}} p(\boldsymbol{x}, \boldsymbol{y}).$$

If we can output fractional values (if the denoised image can be grayscale), another possible solution is

$$\mathbb{E}[\boldsymbol{X}|\boldsymbol{y}].$$

△

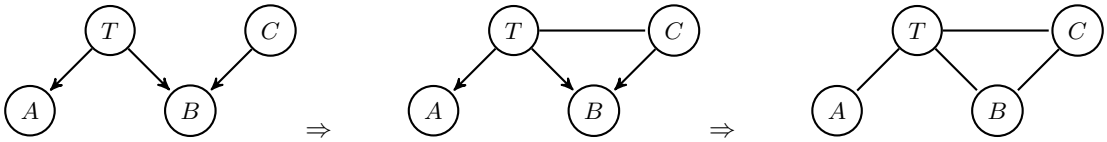
8.4 Moralization: Converting BNs to MRFs

In a BN, there is a term for each node x_i of the form

$$p(x_i|\operatorname{pa}(x_i)).$$

To be able to have the same term in an MRF, we need to have a clique containing x_i and its parents. So to design an MRF that can represent the same distribution as the BN, we first connect all the parents of each nodes with each other and then remove all directions from the edges.

Example 8.4. As an example, consider:



△

We have

$$p(A,B,T,C) = p(T)p(C)p(A|T)p(B|T,C) \qquad \Rightarrow \qquad p(A,B,T,C) = \psi(T)\psi(C)\psi(A,T)\psi(B,T,C),$$

where, for example, $\psi(B,T,C) = p(B|T,C)$.