Graphical Models

Muchang Bahng

Spring 2025

Contents

The concept of using latent variables to model some process will be used over and over again. We have seen simple examples of latent linear models, but what about nonlinear ones? It turns out that these can be seen as a specific instance of *graphical models*.

When computing high-dimensional distributions, the parameters needed to encode this density scales badly. We can see that a general Gaussian mixture model in $\mathbb{R}^n$ with k clusters requires $O(n^2k)$ parameters. If we wanted to sample from a distribution of portraits, then the dimension n would be the resolution of the image. For a 1024×1024 image, this requires $n = 3 \cdot 2^{20}$ dimensions, and modeling it with a GMM is hopeless. Fortunately, for complex distributions there is usually some dependencies (e.g. between neighboring pixels) that we can take advantage of. This is exactly what graphical models do. They factor complex distributions so that the scaling is much better. While there are graphical models that do not use latent variables, most interesting applications of graphical models require latent variables, and so we will focus on that. Additionally, we will introduce the EM algorithm, which will be used repeatedly and is particularly important in optimizing *variational autoencoders* in deep learning.

1 Bayesian Networks (Directed Graphical Models)

Note that the whole purpose of directed graphical models is to model some sort of *causal* relationship between two random variables. Note that while this is successful in practice, there is really no way to know for sure about any causality.

Definition 1.1 (Bayesian Network)

A **Bayesian network**, also known as a **directed probability model**, is a directed acyclic graph of M nodes representing a joint probability distribution of M scalar random variables. An edge pointing $A \to B$ means that the B is conditionally dependent on A, and that there is a very clear casual relationship coming from A to B. The **parents** of a node x_i is denoted pa_i, and the entire joint distribution can be broken up as such:

$$p(\mathbf{x}) = \prod_{m=1}^{M} p(x_m \mid x_{\text{pa}_m}) \tag{1}$$

which is unique due to it being a DAG. Not only is a Bayesian network easy to parameterize. We can also sample from the joint distribution by sequentially sampling starting from the parents to the final children, and discarding the ones (marginalizing) that we don't wish to sample. This is known as **ancestral sampling**.

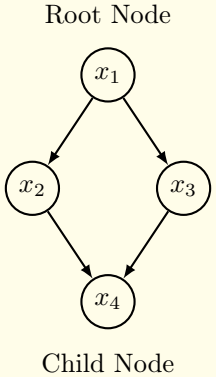


Figure 1

This following example cleared up any confusion when I learned Bayesian networks for the first time.

Example 1.1 (Relay Race)

Consider a 4×100m relay race where the final race time depends on multiple factors. We can model this as a Bayesian network where the total race time T depends on:

- Individual runner capabilities (R_1, R_2, R_3, R_4)
- Handoff success between runners (H_1, H_2, H_3)
- Individual leg performances (P_1, P_2, P_3, P_4)

The joint probability distribution factorizes as:

$$p(T, R_1, R_2, R_3, R_4, H_1, H_2, H_3, P_1, P_2, P_3, P_4) =$$
$$p(T|P_1, P_2, P_3, P_4) \prod_{i=1}^{4} p(R_i) \prod_{i=1}^{3} p(H_i|R_i, R_{i+1}) \prod_{i=1}^{4} p(P_i|R_i, H_{i-1})$$

where H_0 is undefined for P_1, and each runner's performance depends on their capability and the success of the previous handoff (except for the first runner). This network captures both the individual contributions and the critical dependencies between runners during baton exchanges.

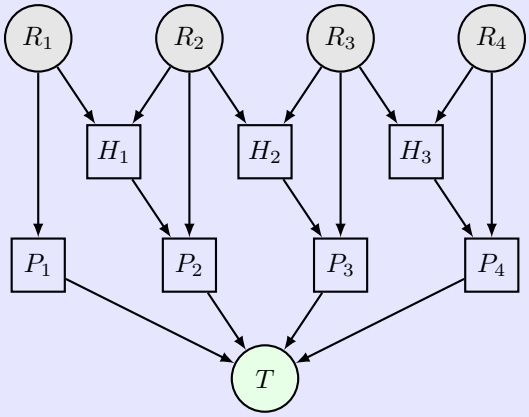


Figure 2: Bayesian Network for a 4x100m Relay Race. The graphical representation is much more compact and intuitive than simply writing out all the products.

Bayesian modelling with hierarchical priors.

Example 1.2 (Multinomial)

We first provide some motivation from a computational complexity perspective. Given a joint distribution of 2 random variables $\mathbf{x}_1, \mathbf{x}_2$, say which are multinomial with K classes, their joint distribution $p(\mathbf{x}_1, \mathbf{x}_2)$ is captured by $K^2 - 1$ parameters. For a general M random variables, then we have to keep a total of $K^M - 1$ parameters, and this increases exponentially. By building a directed graph with say r maximum number of variables appearing on either side of the conditioning bar in a single probability distribution, then the computational complexity scales as $O(K^r)$, which may save a lot of time if $r << M$.

Extending upon this example, we can see that we want to balance two things:

1. Fully conncted graphs have completely general distributions and have $O(K^M - 1)$ number of parameters (too complex).
2. If there are no links, the joint distribution fully factorizes into the product of its marginals and has $M(K-1)$ parameters (too simple) .

Graphs that have an intermediate level of connectivity allow for more general distributions compared to the

fully factorized one, while requiring fewer parameters than the general joint distribution. One model that balances this out is the hidden markov model.

Example 1.3 (Chain Graph)

Consider an M-node Markov chain. The marginal distribution $p(\mathbf{x}_1)$ requires $K-1$ parameters, and the remaining conditional distributions $p(\mathbf{x}_i \mid \mathbf{x}_{i-1})$ requires $K(K-1)$ parameters. Therefore, the total number of parameters is

$$K - 1 + (M-1)(K-1)K \in O(MK^2) \tag{2}$$

which scales relatively well, and we have

$$p(\{\mathbf{x}_m\}) = p(\mathbf{x}_1) \prod_{m=2}^{M} p(\mathbf{x}_m \mid \mathbf{x}_{m-1}) \tag{3}$$

TBD

We can turn this same graph into a Bayesian model by introducing priors for the paramters. Therefore, each node requires an additional parent representing the distribution over parameters (e.g. prior can be Dirichlet)

$$p(\{\mathbf{x}_m, \mu_m\}) = p(\mathbf{x}_1 \mid \mu_1)p(\mu_1) \prod_{m=2}^{M} p(\mathbf{x}_m \mid \mathbf{x}_{m-1}, \mu_m)p(\mu_m) \tag{4}$$

with $p(\mu_m) = \text{Dir}(\mu_m \mid \alpha_m)$ for some predetermined fixed hyperparameter α_m.

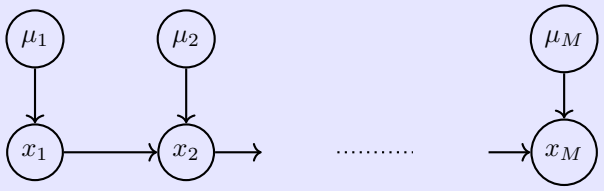


Figure 3

We could also choose to share a common prior over the parameters, trading flexibility for computational feasibility.

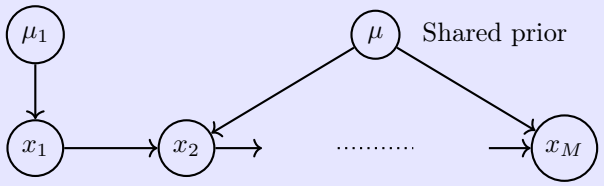


Figure 4

Another way to make more compact representations is through parameterized models. For example, if we have to compute $p(y = 1 \mid \mathbf{x}_1, \dots, \mathbf{x}_M)$, this in general has $O(K^M)$ parameters. However, we can obtain a more parsimonious form by using a logistic function acting on a linear combination of the parent variables

$$p(y = 1 \mid \mathbf{x}_1, \dots, \mathbf{x}_m) = \sigma\left(w_0 + \sum_{i=1}^{M} w_i x_i\right) = \sigma(\mathbf{w}^T \mathbf{x}) \tag{5}$$

We can look at an example how this is applied to sampling from high-dimensional Gaussian with **linear Gaussian models**.

Example 1.4 (Multivariate Gaussian)

Consider an arbitrary acyclic graph over D random variables, in which eachnode represents a single continuous Gaussian distribution with its mean given by a linear function of its parents.

$$p(x_i \mid \mathbf{pa}_i) = N\left(x_i \middle| w_{ij}x_j + b_j, v_i\right)$$

Given a multivariate Gaussian, let us try to decompose it into a directed graph. The log of the joint distribution takes form

$$\ln p(\mathbf{x}) = \sum_{i=1}^{D} \ln p(x_i \mid \text{pa}_i) = -\sum_{i=1}^{D} \frac{1}{2v_i}\left(x_i - \sum_{j \in \text{pa}_i} w_{ij}x_j - b_i\right)^2 + \text{const}$$

To compute the mean, we can see that by construction, every x_i is dependent on its ancestors, so

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

so by linearity of expectation, we have

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

So again, we can start at the top of the graph and compute the expectation. To compute covariance, we can obtain the i,jth element of $\mathbf{\Sigma}$ with a recurrence relation:

$$\begin{aligned}
\Sigma_{ij} &= \mathbb{E}[(x_i - \mathbb{E}[x_i])(x_j - \mathbb{E}[x_j])] \\
&= \mathbb{E}\left[(x_i - \mathbb{E}[x_i])\left(\sum_{k \in \text{pa}_j} w_{jk}(x_k - \mathbb{E}[x_k]) + \sqrt{v_i}\epsilon_j\right)\right] \\
&= \sum_{k \in \text{pa}_j} w_{jk}\Sigma_{ik} + I_{ij}v_j
\end{aligned}$$

If there were no links in the graphs, then the w_{ij}'s are 0, and so $\mathbb{E}[\mathbf{x}] = [b_1, \dots, b_D]$, making the covariance diagonal.If the graph is fully connected, then the total number of parameters is $D + D(D-1)/2$, which corresponds to a general symmetric covariance matrix.

Example 1.5 (Bilinear Gaussian Model)

Consider the following model

$$\begin{aligned}
u &\sim N(0,1) \\
v &\sim N(0,1) \\
r &\sim N(uv, 1)
\end{aligned}$$

where the mean of r is a product of 2 Gaussians. This is also a parameterized model.

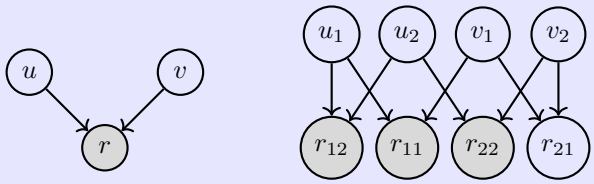


Figure 5

Definition 1.2 (Conditional Independence in Directed Graphs)

We say that a is independent of b given c if

$$p(a \mid b, c) = p(a \mid c)$$

or equivalently,

$$p(a, b \mid c) = p(a \mid b, c)\, p(b \mid c) = p(a \mid c)\, p(b \mid c)$$

Conveniently, we can directly read conditional independence properties of the joint distribution from the graph without any analytical measurements.

Example 1.6 (Conditional Independence on Dataset)

We can demonstrate conditional independence with iid data. Consider the problem of density estimation of some dataset $\mathcal{D} = \{x_i\}$ with some parameterized distribution of μ. Originally, the observations are not independent since they depend on μ.

$$p(\mathcal{D}) = \int_\mu p(\mathcal{D} \mid \mu)\, p(\mu)\, d\mu \tag{6}$$

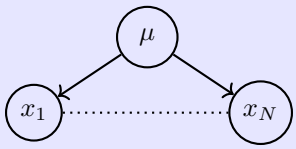


Figure 6

If we condition on μ and considered the joint over the observed variables, the variables are independent.

$$p(\mathcal{D} \mid \mu) = \prod_{n=1}^{N} p(x_n \mid \mu) \tag{7}$$

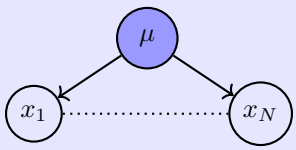


Figure 7

The example above identifies a node (the parent μ) where, if observed, causes the rest of the nodes to become independent. We can extend on this idea by taking an arbitrary x_i and finding a set of nodes such that if

they are observed, then x_i is indepedent from every other node.

Definition 1.3 (Markov Blanket in Directed Graphs)

The **Markov blanket** of a node is the minimal set of nodes that must be observed to make this node independent of all other nodes. It turns out that the parents, children, and coparents are all in the Markov blanket.

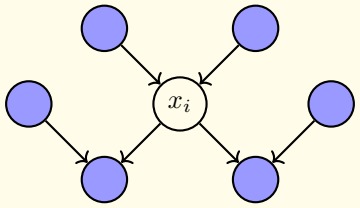


Figure 8

Note that

$$p(x_i \mid x_{j\neq i}) = \frac{p(x_1, \dots, x_M)}{\int p(x_1, \dots, x_M)\, dx} = \frac{\prod_k p(x_k \mid \text{pa}_k)}{\int \prod_k p(x_k \mid \text{pa}_k)\, dx_i} \tag{8}$$

One final interpretation is that we can view directed graphs as **distribution filters**. We take the joint probability distribution, will starts off as fully connected, and the directed graphs “filters" away the edges that are not needed. Therefore, the joint probability distribution $p(\mathbf{x})$ is only allows through the filter if and only if it satisfies the factorization property.

2 Markov Random Field (Undirected Graphical Models)

As the name implies, undirected models use undirected graphs, which are used to model relationships that go both ways rather than just one. Unlike directed graphs, which are useful for expressing casual relationships between random variables, undirected graphs are useful for expressing soft constraints between random variables.

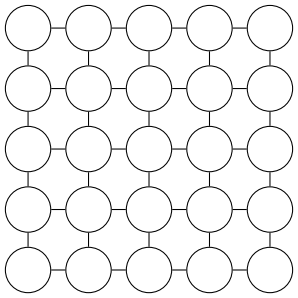


Figure 9: An MRF can be represented with this graph.

Definition 2.1 (Conditional Independence in Undirected Graphs)

Fortunately, conditional independence is easier compared to directed models. We can say A is conditionally independent to B given C if C blocks all paths between any node in A and any node in B.

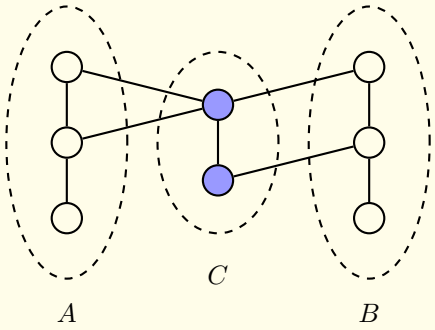


Figure 10: A is conditionally independent given C, denoted $A \perp\!\!\!\perp B|C$.

Definition 2.2 (Markov Blanket in Undirected Graphs)

The Markov blanket of a node, which is the minimal set of nodes that must be observered to make this node independent of the rest of the nodes, is simply the nodes that are directly connected to that node.

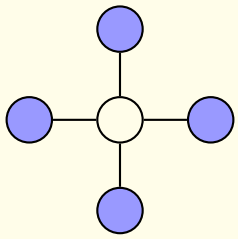


Figure 11: Once the neighbors of a node are realized, the node is independent of the rest of the nodes.

Therefore, the conditional distribution of x_i conditioned on all the variables in the graph is dependent only on the variables in the Markov blanket.

Now, let us talk about how we can actually define a probability distribution with this graph.

Definition 2.3 (Clique)

In an undirected graph, a **clique** is a set of nodes such that there exists a link between all pairs of nodes in that subset. A **maximal clique** is a clique such that it is not possible to include any other nodes in the set without it ceasing it to be a clique.

Given a joint random variable $\mathbf{x}$ represented by an undirected graph, the joint distribution is given by the product of non-negative potential functions over the maximal cliques

$$p(\mathbf{x}) = \frac{1}{Z} \prod_C \phi_C(x_C) \tag{9}$$

where

$$Z = \int p(\mathbf{x})\, d\mathbf{x} \tag{10}$$

is the normalizing constant, called the **partition function**. That is, each x_C is a maximal clique and ϕ_C is the nonnegative potential function of that clique.

This assignment looks pretty arbitrary. How do we know that any arbitrary joint distribution of $\mathbf{x}$, which has a undirected graphical representation, can be represented as the product of a bunch of functions over

the maximum cliques? Fortunately, there is a mathematical result that proves this.

Theorem 2.1 (Hammersley-Clifford)

The joint probability distribution of any undirected graph can be written as the product of potential functions on the maximal cliques of the graph. Furthermore, for any factorization of these potential functions, there exists an undirected graph for which is the joint.

Example 2.1 ()

For example, the joint distribution of the graph below

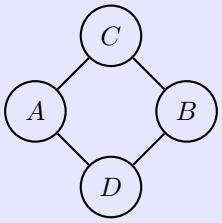


Figure 12

factorizes into

$$p(A,B,C,D) = \frac{1}{Z}\phi(A,C)\,\phi(C,B)\,\phi(B,D)\,\phi(A,D) \tag{11}$$

Note that each potential function ϕ is a mapping from the joint configuration of random variables in a clique to non-negative real numbers. The choice of potential functions is not restricted to having specific probabilistic interpretations, but since they must be nonnegative, we can just represent them as an exponential. The negative sign is not needed, but is a remnant of physics notation.

$$p(\mathbf{x}) = \frac{1}{Z}\prod_C \phi_C(x_C) = \frac{1}{Z}\exp\left\{-\sum_C E(x_C)\right\} = \frac{1}{Z}\underbrace{\exp\{-E(\mathbf{x})\}}_{\substack{\text{Boltzmann}\\ \text{distribution}}} \tag{12}$$

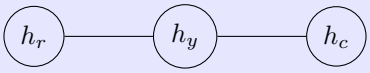
Any distribution that can be represented as the form above is called a **Boltzmann distribution**. So far, all we stated is that the joint probability distribution can be expressed as the product of a bunch of potential functions, but besides the fact that it is nonnegative, there is no probabilistic interpretation of these potentials (or equivalently, the energy functions). While this does give us greater flexibility in choosing potential functions, we must be careful in choosing them (e.g. choosing something like x^2 may cause the integral to diverge, making the joint not well-defined).

Clearly, these potential functions over the cliques should express which configuration of the local variables are preferred to others. It should assign higher values to configurations that are deemed (either by assumption or through training data) to be more probable. That is, each potential is like an "expert" that provides some opinion (the value) on a configuration, and the product of the values of all the potential represents the total opinion of all the experts. Therefore, global configurations with relatively high probabilities are those that find a good balance in satisfying the (possibly conflicting) influences of the clique potentials.

Example 2.2 (Transmission of Colds)

Say that you want to model a distribution over three binary variables: whether you or not you, your coworker, and your roommate is sick (0 represents sick and 1 represents healthy). Then, you can make simplifying assumptions that your roommate and your coworker do not know each other, so it is very unlikely that one of thme will give the other an infection such as a cold directly. Therefore,

we can model the indirect transmission of a cold from your coworker to your roommate by modeling the transmission of the cold from your coworker to you and then you to your roommate. Therefore, we have a model of form



One max clique contains h_y and h_c. The factor for this clique can be defined by a table and might have values resembling these.

	$h_y = 0$	$h_y = 1$
$h_c = 0$	2	1
$h_c = 1$	1	10

Table 1: States and Values of h_y and h_c

This table completely describes the potential function of this clique. Both of you are usually healthy, so the state $(1, 1)$ gets the maximum value of 1. If one of you are sick, then it is likely that the other is sick as well, so we have a value of 2 for $(0, 0)$. Finally, it is most unlikely that one of you is sick and the other healthy, which has a value of 1.

3 Hidden Markov Models