Probabilistic Machine Learning

Derek Li

Contents

1 Introduction to Probabilistic Model

1.1 Overview of Probabilistic Model

We have random variables $X = (X_1, \cdots, X_N)$, and we want a model that captures the relationship between these variables. The approach of probabilistic generative models is to relate all variables by a learned joint probability distribution $p_\theta(X_1, \cdots, X_n)$.

Now let X be input data, C be discrete outputs (labels), Y be continuous outputs. We can use it for machine learning tasks:

- Regression:
$$p(Y|X) = \frac{p(X,Y)}{p(X)} = \frac{p(X,Y)}{\int p(X,Y)\mathrm{d}Y}.$$
- Classification/Clustering:
$$p(C|X) = \frac{p(X,C)}{\sum_C p(X,C)}.$$

The distinction between classification and clustering, or in general supervised and unsupervised learning, in probabilistic perspective is given by whether a random variable is observed or unobserved.

- Supervised dataset: $\{x_i, c_i\}_{i=1}^N \sim p(X,C)$.
- Unsupervised dataset: $\{x_i\}_{i=1}^N \sim p(X,C)$.
- Semi-Supervised learning: Variables are observed for some, but not all.

Like clusters, introducing assumptions about unobserved variables is a powerful modeling tool. We say a latent variable is never observed in the dataset. By introducing and modeling latent variables, we will be able to naturally describe and capture abstract features of are input data.

1.1.1 Operations on Probabilistic Model

The fundamental operations we will perform are:

- Generate data: We need know how to sample from the model.
- Estimate likelihood: When all variables are either observed or marginalized the result is a single real number which is the probability of the all variables taking on those specific values.
- Inference.
- Learning.

1.1.2 Desiderata of Probabilistic Model

We have two main goals for our joint distributions:

- Operations are efficient.
- p_θ is compact to represent.

1.2 Likelihood Function

We define log likelihood function as

$$l(\theta; x) = \ln(p(x|\theta)),$$

where x is fixed.

The process of learning is choosing θ to minimize some cost or loss function, $L(\theta)$ which includes $l(\theta)$. This can be done in may ways, including:

- Maximum likelihood estimation (MLE): $L(\theta) = l(\theta; \mathcal{D})$.
- Maximum a posteriori (MAP): $L(\theta) = l(\theta; \mathcal{D}) + r(\theta)$.

1.2.1 Maximum Likelihood Estimation

MLE is to pick values for our parameters:

$$\widehat{\theta}_{\text{MLE}} = \arg\max_{\theta} l(\theta; \mathcal{D}).$$

For i.i.d. data,

$$p(\mathcal{D}|\theta) = \prod_m p(x^{(m)}|\theta),$$

and

$$l(\theta; \mathcal{D}) = \sum_m \ln(p(x^{(m)}|\theta)).$$

Example 1.1 (Univariate Normal)**.** The model is

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

and the log likelihood function is

$$l(\theta; \mathcal{D}) = \ln(p(\mathcal{D}|\theta)) = -\frac{N}{2}\ln(2\pi\sigma^2) - \frac{1}{2}\sum_i \frac{(x-\mu)^2}{\sigma^2},$$

then

$$\frac{\partial l}{\partial \mu} = \frac{1}{\sigma^2}\sum_i (x_i - \mu).$$

We set $\frac{\partial l}{\partial \mu} = 0$, and have

$$\mu^*_{\text{MLE}} = \frac{\sum_i x_i}{N}.$$

1.3 Sufficient Statistics

A sufficient statistic is a statistic that conveys exactly the same information about the data generating process that created that data as the entire data itself. In other words, once we know the sufficient statistic $T(x)$, then our inferences are the same as would be obtained from our entire data. Mathematically, we say $T(X)$ is a sufficient statistic for X if

$$T(x^{(1)}) = T(x^{(2)}) \Rightarrow L(\theta; x^{(1)}) = L(\theta; x^{(2)}), \forall \theta.$$

Put another way,

$$P(\theta|T(X)) = P(\theta|X).$$

By the Neyman factorization theorem, we have

$$P(\theta|T(X)) = h(x, T(x))g(T(x), \theta).$$

Example 1.2 (Exponential Family)**.**

$$p(x|\eta) = h(x)e^{\eta^T T(x) - A(\eta)}$$

or equivalently

$$p(x|\eta) = h(x)A(\eta)e^{\eta^T T(x)}.$$

For univariate normal distribution,

$$\begin{aligned} p(x) &= \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2}(x-\mu)^2\right) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2}(x^2 - 2x\mu + \mu^2)\right) \\ &= \frac{1}{\sqrt{2\pi}\sigma} \exp\left(\frac{-\mu^2}{2\sigma^2}\right) \exp\left(\begin{pmatrix} \frac{\mu}{\sigma^2} & \frac{-1}{2\sigma^2} \end{pmatrix} \begin{pmatrix} x \\ x^2 \end{pmatrix}\right). \end{aligned}$$

Thus, $\eta^T = \begin{pmatrix} \frac{\mu}{\sigma^2} & \frac{-1}{2\sigma^2} \end{pmatrix}, T(x) = \begin{pmatrix} x \\ x^2 \end{pmatrix}.$

We can rewrite $p(x)$ in terms of η :

$$p(x|\eta) = (\sqrt{2\pi})^{-\frac{1}{2}} \cdot (-2\eta_2)^{\frac{1}{2}} \cdot \exp\left(\frac{\eta_1^2}{4\eta_2}\right) \cdot \exp(\eta^T T(x)),$$

where $h(x) = (\sqrt{2\pi})^{\frac{1}{2}}, A(\eta) = (-2\eta_2)^{\frac{1}{2}} \cdot \exp\left(\frac{\eta_1^2}{4\eta_2}\right).$

Example 1.3 (Bernoulli Trial)**.** Suppose $X \sim \text{Bernoulli}(\theta)$. The likelihood is

$$L = \prod_{i=1}^{N} \theta^{x_i}(1-\theta)^{1-x_i} = \theta^{\sum_{i=1}^N x_i}(1-\theta)^{N - \sum_{i=1}^N x_i}.$$

Note that the likelihood depends on $\sum_{i=1}^N x_i$, i.e., if we know this summary statistic, which we will call $T(x) = \sum_{i=1}^N x_i$, then essentially we know everything that is useful from our sample to do inference.

We have

$$l(\theta; X) = \ln(p(X|\theta)) = T(X)\ln(\theta) + (N - T(X))\ln(1-\theta),$$

then

$$\frac{\partial l}{\partial \theta} = \frac{T(X)}{\theta} - \frac{N - T(X)}{1-\theta}.$$

We set $\frac{\partial l}{\partial \theta} = 0$, and have

$$\theta^*_{\text{MLE}} = \frac{T(X)}{N}.$$

2 Directed Graphical Model

2.1 Decision Theory

When faced with a choice of actions, we should:

- Determine the value of all possible outcomes.
- Find the probability of each outcome under each action.
- Multiply the two to get expected value, summing over all outcomes.
- Choose the action with highest expected value.

2.2 Joint Distribution

The joint distribution of N random variables can be decomposed by the chain rule of probability:

$$p(x_1, \cdots, x_N) = p(x_1)p(x_2|x_1)p(x_3|x_2, x_1)\cdots p(x_N|x_{N-1:1}) = \prod_{i=1}^{N} p(x_i|x_1, \cdots, x_{N-1}).$$

2.2.1 Conditional Independence

Definition 2.1. Two random variables A, B are ***conditionally independent*** given a third variable C, denoted $X_A \perp X_B | X_C$ iff

$$p(X_A, X_B|X_C) = p(X_A|X_C)p(X_B|X_C) \Leftrightarrow p(X_A|X_B, X_C) = p(X_A|X_C) \Leftrightarrow p(X_B|X_A, X_C) = p(X_B|X_C),$$

for all X_C.

2.3 Directed Acyclic Graphical Model

Definition 2.2. A ***directed graphical model*** implies a restricted factorization of the joint distribution. In a DAG, variables are represented by nodes, and edges represent dependence.

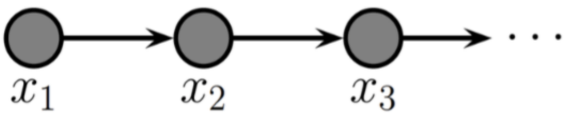
The meaning of any particular directed acyclic graphical model D is that

$$p(x_1, \cdots, x_N) = \prod_{i=1}^{N} p(x_i|\text{Parents}_M(x_i),$$

where $\text{Parents}_M(x_i)$ is the set of nodes with edges pointing to x_i. In other words, the joint distribution of a DAGM factors into a product of local conditional distributions, where each node (a random variable) is conditionally dependent on its parent node(s), which could be empty.

2.3.1 Markov Chain

Definition 2.3. ***Markov chains*** are a stochastic model describing a sequence if possible events in which the probability of each event depends only on the state attained in the previous event.

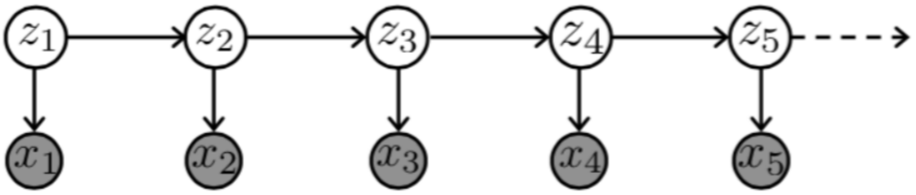


The joint probability is

$$p(x) = p(x_1)p(x_2|x_1)p(x_3|x_2)\cdots.$$

We say the model satisfies the Markov property, i.e., conditional on the present state of the system, its future and past states are independent.

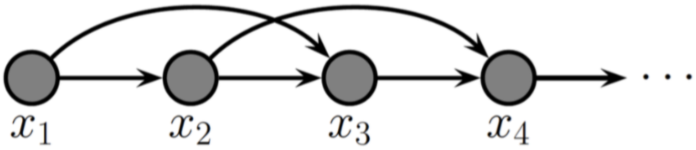
Definition 2.4. ***Hidden Markov chain*** is a statistical Markov model in which the system being modeled is assumed to be a Markov process with unobserved (hidden) state.



z_t are hidden state taking on one of discrete values and x_t are observed variables taking on values in any space. The joint probability is

$$p(x_{1:T}, z_{1:T}) = p(z_{1:T})p(x_{1:T}|z_{1:T}) = p(z_1)\prod_{t=2}^{T} p(z_t|z_{t-1})\prod_{t=1}^{T} p(x_t|z_t).$$

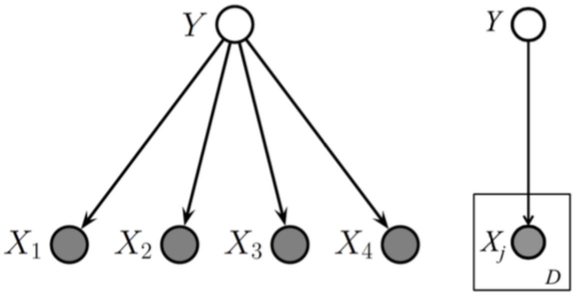
Definition 2.5. ***Second-Order Markov chain*** is a statistical Markov model in which the next state depends on two preceding ones.



The joint probability is

$$p(x_{1:T}) = p(x_1, x_2)p(x_3|x_1, x_2)\cdots = p(x_1, x_2)\prod_{t=3}^{T} p(x_t|x_{t-1}, x_{t-2}).$$

2.3.2 Plates



We can use plates where repeated quantities that are i.i.d. are put in a box. The rules of plates are simple: repeat every structure in a box a number of times given by the integer in the corner of the box, updating the plate index variable as you go, and then duplicate every arrow going into the plate and every arrow leaving the plate by connecting the arrows to each copy of the structure. We can write

$$p(x_1, \cdots, x_N|A) = \prod_{i=1}^{N} p(x_i|A).$$

Plates can be nested, in which case their arrows get duplicates also, according to the rule: draw an arrow from every copy of the source node to every copy of the destination node.

Plates can also cross (intersect), in which case the nodes at the intersection have multiple indices and get duplicated a number of times equal to the product of the duplication numbers on all the plates containing them.

2.4 D-Separation

Definition 2.6. ***D-separation*** or ***directed-separation*** is a notion of connectedness in DAGs in which two (sets of) variables may or may not be connected conditioned on a third (set of) variable(s).

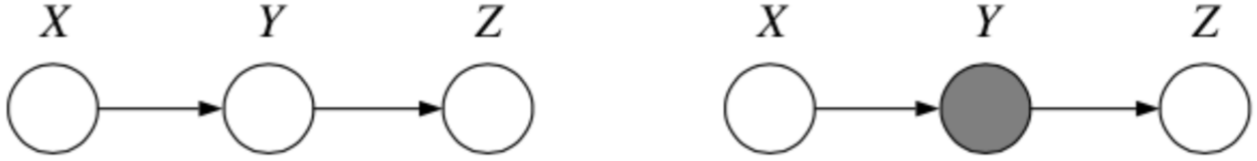
D-connection implies conditional dependence and d-separation implies conditional independence.

In particular, we say $x_A \perp x_B | x_C$ if every variable in A is d-separated from every variable in B conditioned on all the variables in C.

2.4.1 Depth-First Search Algorithm

To check if an independence is true, we can cycle through each node in A, do a depth-first search to reach every node in B, and examine the path between them. If all of the paths are d-separated, then $x_A \perp x_B | x_C$.

- Chain:

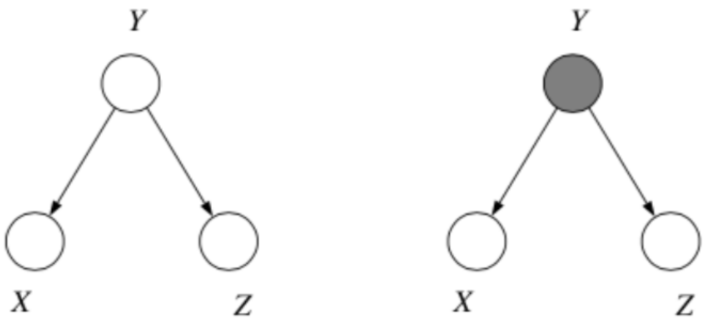


∗ We have $p(X, Y, Z) = p(X)p(Y|X)p(Z|Y)$, which implies

$$p(X, Z|Y) = \frac{p(X, Y, Z)}{p(Y)} = \frac{p(X)p(Y|X)p(Z|Y)}{p(Y)} = p(X|Y)p(Z|Y),$$

and thus $X \perp Z|Y$.

- Common cause:



∗ We have

$$p(X|Z) = \frac{p(X, Z)}{p(Z)} = \frac{\sum_Y p(X, Y, Z)}{\sum_X \sum_Y p(X, Y, Z)} \neq p(X),$$

and thus $X \not\perp Z$.

$*$ We have $p(X,Y,Z) = p(X|Y)p(Y)p(Z|Y)$, which implies

$$p(X,Z|Y) = \frac{p(X,Y,Z)}{p(Y)} = \frac{p(X|Y)p(Y)p(Z|Y)}{p(Y)} = p(X|Y)p(Z|Y),$$

and thus $X \perp Z|Y$.

- Explaining away:

$*$ We have

$$p(X|Z) = \frac{p(X,Z)}{p(Z)} = \frac{\sum_Y p(X,Y,Z)}{p(Z)} = \frac{\sum_Y p(X)p(Y|X,Z)p(Z)}{p(Z)} = p(X),$$

and thus $X \perp Z$.

$*$ We have $p(X,Y,Z) = p(X)p(Y|X,Z)p(Z)$, which implies

$$p(X,Z|Y) = \frac{p(X,Y,Z)}{p(Y)} = \frac{p(X)p(Y|X,Z)p(Z)}{p(Y)} \neq p(X|Y)p(Z|Y),$$

and thus $X \not\perp Z|Y$.

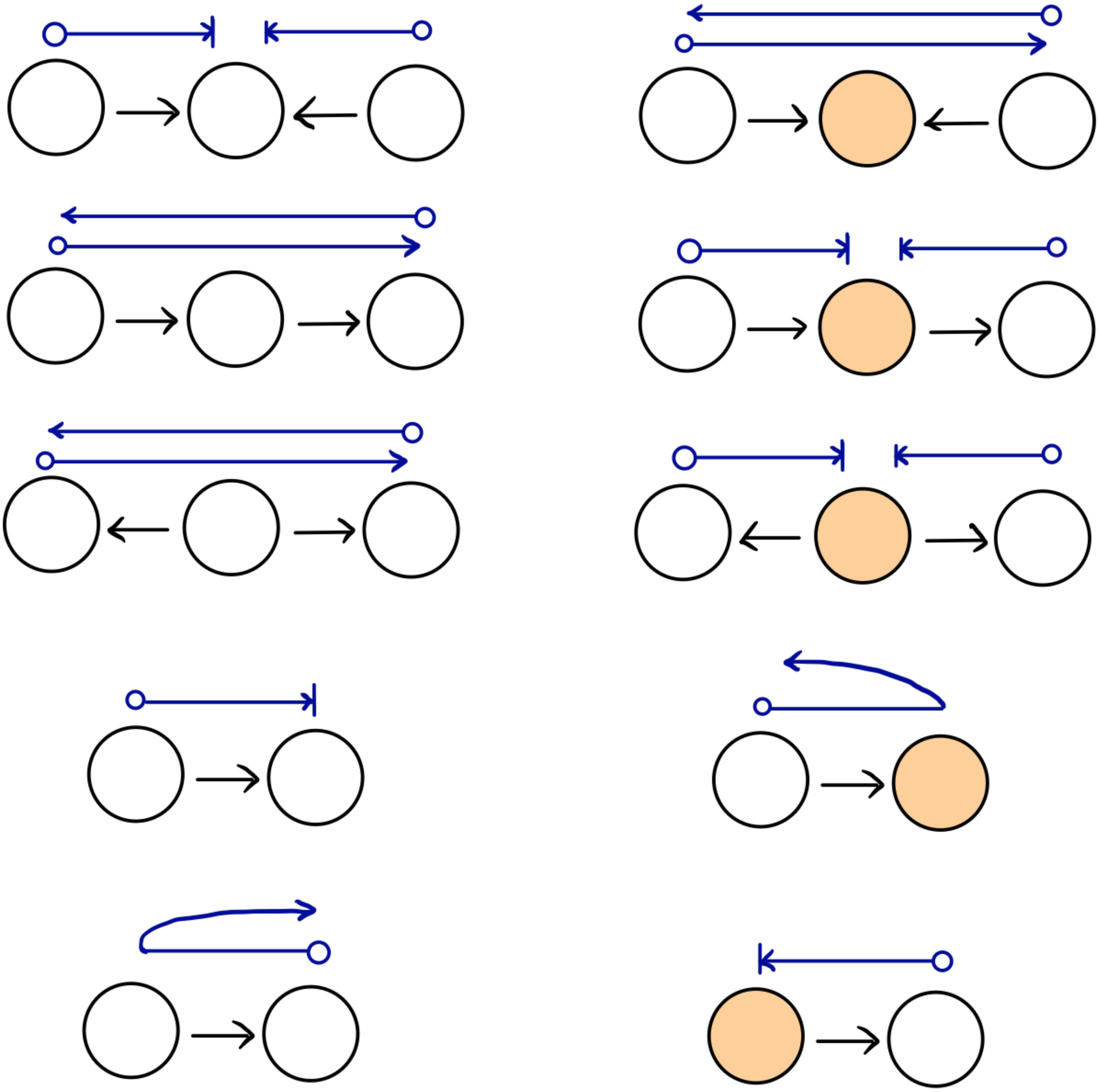
2.4.2 Bayes-Balls Algorithm

In general, the algorithm works as follows:

- Shade all nodes x_C (condition).
- Place "balls" at each node in x_A or x_B.
- Let the "balls" bounce around according to some rules.

$*$ If any of the balls reach any of the nodes in x_B from x_A or x_A form x_B, then $x_A \not\perp x_B|x_C$. Otherwise, $x_A \perp x_B|x_C$.

There are 10 basic rules:



3 Undirected Graphical Model

Definition 3.1. ***Undirected graphical model*** (UGM), also called ***Markov random field*** (MRF) or Markov network, is a set of random variables described by an undirected graph. As in DAGM, the nodes in the graph represent random variables and the edges represent probabilistic interactions between neighboring variables.

Definition 3.2. A ***clique*** is an undirected graph is a subset of its vertices s.t. every two vertices in the subset are connected by an edge. A ***maximal clique*** is a clique that cannot be extended by including one more adjacent vertex. A ***maximum clique*** is a clique of the largest possible size in a given graph.

3.1 Parameterization of an UGM

Let $x = (x_1, \cdots, x_m)$ be the set of all random variables. Unlike in DGM, there is no topological ordering associated with an undirected graph, and so we cannot use the chain rule to represent the joint distribution $p(x)$.

We associate potential functions or factors with each maximal clique in the graph. For a given clique c, we define the potential function or factor $\psi_c(x_c|\theta_c)$ to be any non-negative function, where x_c is some subset of variables in x involved in a unique and maximal clique. The joint distribution is proportional to the product of clique potentials

$$p(x) \propto \prod_{c \in \mathcal{C}} \psi_c(x_c|\theta_c).$$

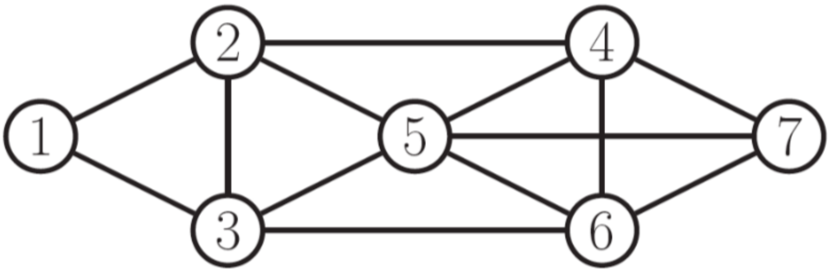
More formally, a positive distribution $p(x) > 0$ satisfies the conditional independence properties of an undirected graph G iff p can be represented as a product of factors, one per maximal clique, i.e.,

$$p(x|\theta) = \frac{1}{Z(\theta)} \prod_{c \in \mathcal{C}} \psi_c(x_c|\theta_c),$$

where $\mathcal{C}$ is the set of all maximal cliques of G and $Z(\theta)$ is the partition function defined as

$$Z(\theta) = \sum_x \prod_{c \in \mathcal{C}} \psi_c(x_c|\theta_c).$$

Example 3.1. $p(x) \propto \psi_{1,2,3}(x_1, x_2, x_3)\psi_{2,3,5}(x_2, x_3, x_5)\psi_{2,4,5}(x_2, x_4, x_5)\psi_{3,5,6}(x_3, x_5, x_6)\psi_{4,5,6,7}(x_4, x_5, x_6, x_7)$.



4 Exact Inference

4.1 Inference as Conditional Distribution

Let X_E be the observed evidence, X_F be the unobserved variable we want to infer and $X_R = X \backslash \{X_F, X_E\}$ be the remaining variables, extraneous to query. For inference, we will marginalize out these extraneous variables, focusing on the joint distribution over evidence and subject of inference:

$$p(X_F, X_E) = \sum_{X_R} p(X_F, X_E, X_R).$$

In particular, inference will focus computing the conditional probability distribution

$$p(X_F|X_E) = \frac{p(X_F, X_E)}{p(X_E)} = \frac{p(X_F, X_E)}{\sum_{X_F} p(X_F, X_E)} = \frac{p(X_F, X_E)}{\sum_{X_F, X_R} p(X_F, X_E, X_R)}.$$

4.2 Variable Elimination (VE)

Variable elimination

- Is a simple and general exact inference algorithm in any probabilistic graphical model.
- Has computational complexity that depends on the graph structure of the model.
- Can use dynamic programming to avoid enumerating all variable assignments.

Example 4.1 (Chain)**.** Suppose a simple chain

$$A \to B \to C \to D,$$

where we want to compute $P(D)$ with no observations for other variables. We have

$$X_F = \{D\}, X_E = \{\}, X_R = \{A, B, C\}.$$

The graphical model describes the factorization of the joint distribution as

$$p(A, B, C, D) = p(A)p(B|A)p(C|B)p(D|C).$$

If the goal is to compute the marginal distribution $p(D)$ with no observed variables then we marginalize over all variables but D :

$$p(D) = \sum_{A,B,C} p(A, B, C, D).$$

However, if we sum naively, it will be exponential $\mathcal{O}(k^4)$, where k is the number states:

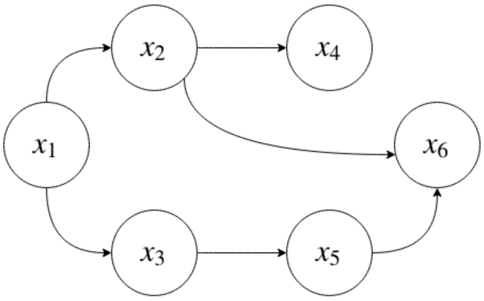
$$p(D) = \sum_C \sum_B \sum_A p(A)p(B|A)p(C|B)p(D|C).$$

If we choose elimination ordering:

$$\begin{aligned} p(D) &= \sum_C p(D|C) \sum_B p(C|B) \sum_A p(A)p(B|A) \\ &= \sum_C p(D|C) \sum_B p(C|B)p(B) \\ &= \sum_C p(D|C)p(C), \end{aligned}$$

we reduce the complexity by first computing terms that appear across the other marginalization sums. So by dynamic programming to do the computation inside out instead of outside in, we have done inference over the joint distribution represented by the chain without generating it explicitly. The cost is $\mathcal{O}(4k^2)$.

Example 4.2 (DGM)**.** Observing the state of a random variable x_6, find $p(x_1|x_6)$.



First recall that

$$p(x_1, \cdots, x_6) = p(x_1)p(x_2|x_1)p(x_3|x_1)p(x_4|x_2)p(x_5|x_3)p(x_6|x_2, x_5),$$

where

$$X_F = \{x_1\}, X_E = \{x_6\}, X_R = \{x_2, x_3, x_4, x_5\},$$

and

$$p(x_1|x_6) = \frac{p(x_1, x_6)}{\sum_{x_1} p(x_1, x_6)}.$$

We use variable elimination to compute $p(x_1, x_6)$:

$$\begin{aligned}
p(x_1, x_6) &= p(x_1) \sum_{x_2} \sum_{x_3} \sum_{x_4} \sum_{x_5} p(x_2|x_1)p(x_3|x_1)p(x_4|x_2)p(x_5|x_3)p(x_6|x_2, x_5) \\
&= p(x_1) \sum_{x_2} p(x_2|x_1) \sum_{x_3} p(x_3|x_1) \sum_{x_4} p(x_4|x_2) \sum_{x_5} p(x_5|x_3)p(x_6|x_2, x_5) \\
&= p(x_1) \sum_{x_2} p(x_2|x_1) \sum_{x_3} p(x_3|x_1) \sum_{x_4} p(x_4|x_2)p(x_6|x_2, x_3) \\
&= p(x_1) \sum_{x_2} p(x_2|x_1) \sum_{x_3} p(x_3|x_1)p(x_6|x_2, x_3) \sum_{x_4} p(x_4|x_2) \\
&= p(x_1) \sum_{x_2} p(x_2|x_1) \sum_{x_3} p(x_3|x_1)p(x_6|x_2, x_3) \\
&= p(x_1) \sum_{x_2} p(x_2|x_1)p(x_6|x_1, x_2) \\
&= p(x_1)p(x_6|x_1).
\end{aligned}$$

The complexity of variable elimination is related to the elimination ordering. Unfortunately, finding the best elimination ordering is NP-hard.

4.2.1 Intermediate Factor

In the above examples, each time we eliminated a variable it resulted in a new conditional or marginal distribution. However, in general eliminating does not produce a valid conditional or marginal distribution of the graphical model. For example, we have

$$p(X, A, B, C) = p(X)p(A|X)p(B|A)p(C|B, X)$$

and we want to marginalize over X :

$$p(A, B, C) = p(B|A) \sum_{X} p(X)p(A|X)p(C|B, X),$$

However, $\sum_X p(X)p(A|X)p(C|B,X)$ does not correspond to a valid conditional or marginal distribution since it is unnormalized.

Hence, we introduce factors ϕ, which are not necessarily normalized distributions, but which describe the local relationship between random variables.

In the above example:

$$\begin{aligned}
p(A,B,C) &= \sum_X p(X)p(A|X)p(B|A)p(C|B,X) \\
&= \sum_X \phi(X)\phi(A,X)\phi(A,B)\phi(X,B,C) \\
&= \phi(A,B)\sum_X \phi(X)\phi(A,X)\phi(X,B,C) \\
&= \phi(A,B)\tau(A,B,C).
\end{aligned}$$

The original conditional distributions are represented by factors over all variables involved, which obfuscates the dependence relationship between the variables encoded by the conditional distribution. Following marginalizing over X we introduce a new factor τ over the remaining variables.

Note that for directed acyclic graphical models, who are defined by factorizing the joint into conditional distributions, we introduce intermediate factors to only be careful about notation.

However, there are other kinds of graphical models (e.g. undirected graphical models, and factor graphs) that are not represented by factorizing the joint into a product of conditional distributions. Instead, they factorize into a product of local factors, which will need to be normalized.

4.2.2 Sum-Product Inference Algorithm

Computing $p(Y)$ for directed and undirected models is given by sum-product inference algorithm

$$\tau(Y) = \sum_z \prod_{\phi\in\Phi} \phi(z_{\mathrm{Scope}[\phi]\cap Z}, y_{\mathrm{Scope}[\phi]\cap Y}), \forall Y,$$

where Φ is a set of potentials or factors.

For directed models, Φ is given by the conditional probability distributions for all variables

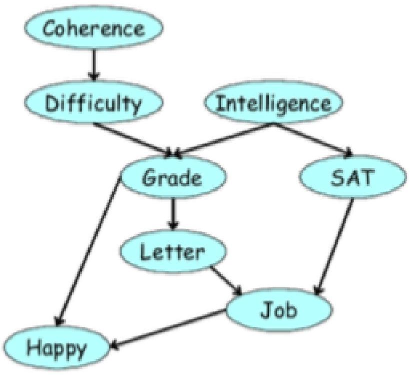
$$\Phi = \{\phi_{x_i}\}_{i=1}^N = \{p(x_i|\mathrm{Parents}(x_i))\}_{i=1}^N,$$

where the sum is over the set $Z = X - X_F$. The resulting term $\tau(Y)$ will automatically be normalized.

For undirected models, Φ is given by the set of unnormalized potentials. Therefore, we must normalize the resulting $\tau(Y)$ by $\sum_Y \tau(y)$.

Example 4.3 (Directed Graph)**.** The graph describes a factorization of the joint distribution

$$p(C,D,I,G,S,L,H,J) = p(C)p(D|C)p(I)p(G|D,I)p(L|G)p(S|I)p(J|S,L)p(H|J,G).$$



For notation convenience, we can write the conditional distributions as factors:

$$\Phi = \{\phi(C).\phi(D, C), \phi(I), \phi(G, D, I), \phi(L, G), \phi(S, I), \phi(J, S, L), \phi(H, J, G)\}.$$

Suppose now we are interested in inferring $p(J)$.

Eliminating ordering $\prec \{C, D, I, H, G, S, L\}$

$$\begin{aligned}
p(J) &= \cdots \underbrace{\sum_C \phi(C)\phi(D, C)}_{\tau(D)} \\
&= \cdots \underbrace{\sum_D \phi(G, D, I)\tau(D)}_{\tau(G,I)} \\
&= \cdots \underbrace{\sum_I \phi(I)\phi(S, I)\tau(G, I)}_{\tau(S,G)} \\
&= \cdots \underbrace{\left[\sum_H \phi(H, J, G)\right]}_{\tau(J,G)} \tau(S, G) \\
&= \cdots \underbrace{\sum_G \phi(L, G)\tau(J, G)\tau(S, G)}_{\tau(L,J,S)} \\
&= \cdots \underbrace{\sum_S \phi(J, S, L)\tau(L, J, S)}_{\tau(J,L)} \\
&= \underbrace{\sum_L \tau(J, L)}_{\tau(J)} = \tau(J).
\end{aligned}$$

Note that since our original factors correspond to conditional and marginal distributions we do not need to renormalize the final factor $\tau(J)$. However, if we started with potential factors not from a conditional distribution, we would have to normalize $\frac{\tau(J)}{\sum_J \tau(j)}$.

4.3 Complexity of VE

The complexity of the VE algorithm is

$$\mathcal{O}(mk^{N_{\max}}),$$

where m is the number pf potential functions, $|\Phi|, k$ is the number of states each random variable takes (assumed to be equal here), N_i is the number of random variables inside each sum $\sum_i$, and $N_{\max} = \arg\max_i N_i$.

Example 4.4 (Complexity of Eliminating Ordering $\prec \{C, D, I, H, G, S, L\}$)**.** We have $|\Phi| = 9, N_{\max} = N_G = 4$ and thus the complexity of the variable elimination under this ordering is $\mathcal{O}(8k^4)$.