Lecture Notes for CS 726 - Spring 2022

Eeshaan Jain

January 26, 2022

These are my lecture notes taken during the Advanced Machine Learning (CS 726) course at IIT Bombay during the Spring 2021 session.

Contents

Probabilistic Modeling

Probability Theory

We will briefly review probability in a rigorous sense.

We define events considering we have a space of possible outcomes denoted by Ω. $\mathcal{S}$ is a set of measurable events, to which we assign probabilities, and each event $\alpha \in \mathcal{S}$ is a subset of Ω.

The event space necessarily satisfies three properties -

1. It contains the empty event $\emptyset$ and the trivial event Ω.
2. It is closed under union, i.e if $\alpha, \beta \in \mathcal{S}$, so is $\alpha \cup \beta$.
3. It is closed under complementation, i.e if $\alpha \in \mathcal{S}$, so is $\Omega - \alpha$.

In a single coin toss, we have

$$\Omega = \{H, T\}$$

Definition 1 (Probability distribution)**.** A probability distribution P over $(\Omega, \mathcal{S})$ is a mapping of events in $\mathcal{S}$ to real values satisfying

- $P(\alpha) \geq 0$ for all $\alpha \in \mathcal{S}$
- $P(\Omega) = 1$
- If $\alpha, \beta \in \mathcal{S}$ and $\alpha \cap \beta = \emptyset$, then $P(\alpha \cup \beta) = P(\alpha) + P(\beta)$

A direct consequence of the properties is:

$$P(\emptyset) = 0$$
$$P(\alpha \cup \beta) = P(\alpha) + P(\beta) - P(\alpha \cap \beta)$$

Conditional probability answers the question - after learning that event α is true, how does our belief about β change? Formally, we define

$$P(\beta|\alpha) = \frac{P(\alpha \cap \beta)}{P(\alpha)} \tag{1}$$

It can be checked that this satisfies Definition 1 and is a probability distribution. Noting that $P(\alpha \cap \beta) = P(\alpha)P(\beta|\alpha)$, we define the chain rule of conditional probabilities

$$P(\alpha_1 \cap \cdots \cap \alpha_k) = P(\alpha_1)P(\alpha_2|\alpha_1) \cdots P(\alpha_k|\alpha_1 \cap \cdots \cap \alpha_{k-1}) \tag{2}$$

We further define the Bayes' rule

$$P(\alpha|\beta) = \frac{P(\beta|\alpha)P(\alpha)}{P(\beta)} \tag{3}$$

Here, $P(\alpha|\beta)$ is called the *posterior*, $P(\beta|\alpha)$ is the *likelihood*, $P(\alpha)$ is the *prior* and $P(\beta)$ is the *marginal probability* of the structure in context. We can generalize Equation 3 as

$$P(\alpha|\beta \cap \gamma) = \frac{P(\beta|\alpha \cap \gamma)P(\alpha|\gamma)}{P(\beta|\gamma)} \tag{4}$$

Now, we formally define the notion of random variables, which intuitively can be considered to be attribute reporters.

Definition 2 (Random Variable)**.** A random variable X is a *measurable* function $X : \Omega \to \mathcal{S}$. The probability that X takes values in a set $s \in \mathcal{S}$ is written as

$$\Pr(X \in s) = \Pr(\{\omega \in \Omega | X(\omega) \in s\}) \tag{5}$$

The marginal distribution over a random variable X is the distribution over events that can be described using X, and is denoted by $P(X)$. More generally, if we want to describe a distribution over a set of random variables $\mathcal{X} = \{x_1, \cdots, x_n\}$ called the *joint distribution* denoted as $P(x_1, \cdots, x_n)$. The full assignment to the variables is denoted as $\xi \in \text{Val}(\mathcal{X})$. The space corresponding to the joint assignment in $\mathcal{X}$ is called the *canonical outcome space*.

Now, we glance at independencies, a core component of Probabilistic Graphical Models.

Definition 3 (Independence)**.** An event α is independent of an event β denoted by $P \models (\alpha \perp\!\!\!\perp \beta)$, if $P(\alpha|\beta) = P(\alpha)$ or $P(\beta) = 0$.

Proposition 4. *A distribution satisfies $(\alpha \perp\!\!\!\perp \beta)$ if and only if*

$$P(\alpha \cap \beta) = P(\alpha)P(\beta) \tag{6}$$

Proof. Skipped (hint: Use the definition of conditional probability). □

Definition 5 (Conditional Indpendence)**.** An event α is conditionally independent of event β given γ in P, denoted by $P \models (\alpha \perp\!\!\!\perp \beta|\gamma)$ if $P(\alpha|\beta \cap \gamma) = P(\alpha|\gamma)$ or if $P(\beta \cap \gamma) = 0$.

Proposition 6. *P satisfies $(\alpha \perp\!\!\!\perp \beta|\gamma)$ if and only if*

$$P(\alpha \cap \beta|\gamma) = P(\alpha|\gamma)P(\beta|\gamma) \tag{7}$$

Definition 7. Let $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$ be sets of random variables. $\mathbf{X}$ is conditionally independent of $\mathbf{Y}$ given $\mathbf{Z}$ in a distribution P if P satisfies $(\mathbf{X} = \mathbf{x} \perp\!\!\!\perp \mathbf{Y} = \mathbf{y}|\mathbf{Z} = \mathbf{z})$ for all values of $\mathbf{x} \in \text{Val}(\mathbf{X}), \mathbf{y} \in \text{Val}(\mathbf{Y})$ and $\mathbf{z} \in \text{Val}(\mathbf{Z})$. We say that the variables in $\mathbf{Z}$ are *observed*. If $\mathbf{Z}$ is empty, then we say that $\mathbf{X}$ and $\mathbf{Y}$ are marginally independent.

Proposition 8. *The distribution P satisfies $(\mathbf{X} \perp\!\!\!\perp \mathbf{Y}|\mathbf{Z})$ if and only if*

$$P(\mathbf{X}, \mathbf{Y}|\mathbf{Z}) = P(\mathbf{X}|\mathbf{Z})P(\mathbf{Y}|\mathbf{Z}) \tag{8}$$

The following properties hold for conditional independencies:

1. *Symmetry:* $(\mathbf{X} \perp\!\!\!\perp \mathbf{Y}|\mathbf{Z}) \implies (\mathbf{Y} \perp\!\!\!\perp \mathbf{X}|\mathbf{Z})$
2. *Decomposition:* $(\mathbf{X} \perp\!\!\!\perp \mathbf{Y}, \mathbf{W}|\mathbf{Z}) \implies (\mathbf{X} \perp\!\!\!\perp \mathbf{Y}|\mathbf{Z})$
3. *Weak Union:* $(\mathbf{X} \perp\!\!\!\perp \mathbf{Y}, \mathbf{W}|\mathbf{Z}) \implies (\mathbf{X} \perp\!\!\!\perp \mathbf{Y}|\mathbf{Z}, \mathbf{W})$

Decomposition can also be stated as

$$\mathbf{X} \perp\!\!\!\perp \{\mathbf{Y}, \mathbf{Z}\} \implies \mathbf{X} \perp\!\!\!\perp \mathbf{Y}, \mathbf{X} \perp\!\!\!\perp \mathbf{Z}$$

Weak Union can also be stated as

$$\mathbf{X} \perp\!\!\!\perp \{\mathbf{Y}, \mathbf{Z}\} \implies \mathbf{X} \perp\!\!\!\perp \mathbf{Y}|\mathbf{Z}$$

But note that, if $\mathbf{X} \perp\!\!\!\perp \mathbf{Y}$ and $\mathbf{Z} \not\perp\!\!\!\perp \{\mathbf{X}, \mathbf{Y}\}$ then it is not necessary to have $\mathbf{X} \perp\!\!\!\perp \mathbf{Y}|\mathbf{Z}$

4. *Contraction:* $(\mathbf{X} \perp\!\!\!\perp \mathbf{W}|\mathbf{Z}, \mathbf{Y})$ & $(\mathbf{X} \perp\!\!\!\perp \mathbf{Y}|\mathbf{Z}) \implies (\mathbf{X} \perp\!\!\!\perp \mathbf{Y}, \mathbf{W}|\mathbf{Z})$

If our distribution is positive (i.e, for all non-empty $\alpha \in \mathcal{S}, P(\alpha) > 0$), we have another property

Contraction can also be stated as

$$\mathbf{X} \perp\!\!\!\perp \mathbf{Y}|\mathbf{Z}, \mathbf{X} \perp\!\!\!\perp \mathbf{Z} \implies \mathbf{X} \perp\!\!\!\perp \{\mathbf{Y}, \mathbf{Z}\}$$

- ◇ *Intersection:* $(\mathbf{X} \perp\!\!\!\perp \mathbf{Y}|\mathbf{Z}, \mathbf{W})$ & $(\mathbf{X} \perp\!\!\!\perp \mathbf{W}|\mathbf{Z}, \mathbf{Y}) \implies (\mathbf{X} \perp\!\!\!\perp \mathbf{Y}, \mathbf{W}|\mathbf{Z})$

Theorem 9. *Consider $\mathcal{X} = \{\mathbf{X}, \mathbf{Y}, \mathbf{Z}\}$. Then $P \models (\mathbf{X} \perp\!\!\!\perp \mathbf{Y}|\mathbf{Z})$ if and only if we can write*

$$P(\mathcal{X}) = \phi_1(\mathbf{X}, \mathbf{Z})\phi_2(\mathbf{Y}, \mathbf{Z}) \tag{9}$$

Proof. Skipped. □

Probabilistic Graphical Models

Given a set of n random variables $\mathcal{X} = \{x_1, x_2, \cdots x_n\}$ where n is large, we want to build a joint probability distribution P over this set. Explicitly representing the joint distribution is computationally expensive, since just having binary values variables requires the joint distribution to specify $2^n - 1$ numbers, and for more practical variables, the count is too large.

We want to efficiently represent, estimate and answer inference queries on the distribution.

An example of a query can be - `Estimate the fraction of people with a bachelor's degree.`

Alternatives to explicit joint distributions

▷ Can we assume all columns are independent? **NO** - this is obviously a very bad assumption.
▷ Can we use data to detect highly correlated column pairs, and estimate their pairwise frequencies? **MAYBE** - but there might be too many correlated pairs, and the method is ad hoc.
To solve the above two not so good ways, we explore conditional independencies. It may be possible that `income` $\not\perp\!\!\!\perp$ `age` but `income` $\perp\!\!\!\perp$ `age|experience`.

Note that we write that a set X is conditionally independent of Y given Z, i.e $X \perp\!\!\!\perp Y|Z$ if

$$\Pr(X|Y, Z) = \Pr(X|Z)$$

Probabilistic graphical models use a graph-based representation as the basis for compactly encoding a complex distribution over a high-dimensional space.

It is convenient to represent the independence assumption using a graph. The so called graphical model has nodes as the variables (continuous or discrete), and the edges represent direct interaction. If we consider directed edges, we talk about Bayesian Networks, and if we consider undirected edges, we talk about Markov Random Fields.

Essentially the graphical model is a combination of the graph and potentials.

Definition 10 (Potentials)**.** Potentials $\psi_c(\mathbf{x}_c)$ are scores for assignment of values to subsets c of directly interacting variables. We factorize

the probability as a product of these potentials, i.e

$$\Pr(\mathbf{x} = x_1, \cdots, x_n) \propto \prod \psi_s(\mathbf{x}_s) \tag{10}$$

Bayesian Networks

Bayesian Networks, also referred to as *directed graphical models* are a family of probability distributions that has a compact parameterization representable using a directed graph.

It is known that

$$\Pr(x_1, x_2, \cdots, x_n) = \Pr(x_1)\Pr(x_2|x_1)\Pr(x_3|x_2, x_1)\cdots\Pr(x_n|x_{n-1}, \cdots, x_1) \tag{11}$$

A compact Bayesian Network is a distribution in which each factor in the above equation depends on the *parent* variables represented by $\text{Pa}(x_i)$ for variable x_i. Thus, we have

$$\Pr(x_i|x_{i-1}, x_{i-2}, \cdots, x_1) = \Pr(x_i|\text{Pa}(x_i)) \tag{12}$$

and the corresponding potentials at each node in terms of its parents are

$$\psi_i(x_i, \text{Pa}(x_i)) = \Pr(x_i, \text{Pa}(x_i)) \tag{13}$$

Thus,

$$\Pr(x_1, x_2, \cdots, x_n) = \prod_{i=1}^{n} \Pr(x_i|\text{Pa}(x_i)) \tag{14}$$

Consider the situation when each variable can take d values. The naive approach gives us $\mathcal{O}(d^n)$ parameters. If we think of the potentials as probability tables (with the rows corresponding to $\text{Pa}(x_i)$) and columns corresponding to the values of x_i, with entries as $\psi_i(x_i, \text{Pa}(x_i))$, we can notice that if $|\text{Pa}(x_i)| \leq k$, then the number of parameters are $\mathcal{O}(d^{k+1})$, and for n variables, we have $\mathcal{O}(nd^{k+1})$, which provides us the compact representation.

Definition

Now we formally define these -

Definition 12 (Bayesian Network)**.** A Bayesian Network is a directed graph $G = (V, E)$ together with

- ◇ a random variable x_i for each node $i \in V$
- ◇ a potential $\psi_i(x_i, \text{Pa}(x_i))$ for each node $i \in V$

For a variable x_i in our Bayesian Network $\mathcal{G}$, denote $\text{ND}(x_i)$ as the non-descendents of x_i. The following local conditional independencies hold in $\mathcal{G}$ -

$$x_i \perp\!\!\!\perp \text{ND}(x_i)|\text{Pa}(x_i) \tag{15}$$

Example 11 shows the independencies in a simple Bayesian Network.

Example 11.

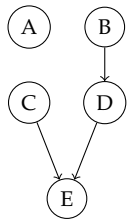

Figure 1: Sample BN

Consider the BN above. We will consider each variable at a time.

- ◇ A has no parent, and has no descendent. Thus,

$$A \perp\!\!\!\perp B, C, D, E$$

- ◇ B has no parent, but has D as a descendent. Thus,

$$B \perp\!\!\!\perp A, C$$

- ◇ C has no parent, but has E as a descendent. Thus,

$$C \perp\!\!\!\perp A, B, D$$

- ◇ D has B as a parent, and has E as the descendent. Thus,

$$D \perp\!\!\!\perp A, C|B$$

- ◇ E has C and D as parents, but has no descenent. Thus,

$$I \perp\!\!\!\perp A, B|C, D$$

Definition 13 (Factorization)**.** Let $\mathcal{G}$ be a Bayesian Network graph over the variables $\{X_i\}_{i=1}^n$. We say that a distribution P over the same space factorizes according to $\mathcal{G}$ if P can be expressed as a product described in Equation 14. Such factorization is also known as the chain rule for Bayesian Networks, and is denoted as Factorize$(P, \mathcal{G})$.

Definition 14. Let P be a distribution over $\mathcal{X}$. We define $\mathcal{I}(P)$ to be the set of independent assertions of the form $\mathbf{X} \perp\!\!\!\perp \mathbf{Y} | \mathbf{Z}$ that hold in P.

We can now write "P satisfies the local independencies associated with $\mathcal{G}$" as $\mathcal{I}_\ell(\mathcal{G}) \subseteq \mathcal{I}(P)$.

Definition 15 (Independency-Map)**.** Let $\mathcal{K}$ be any graph object associated with a set of independencies $\mathcal{I}(\mathcal{K})$. We call $\mathcal{K}$ an I-map for a set of independencies $\mathcal{I}$ if $\mathcal{I}(\mathcal{K}) \subseteq \mathcal{I}$.

Thus for $\mathcal{G}$ to be an I-map for P, any independence that asserts in $\mathcal{G}$ must also assert in P, but P can have additional independencies not reflected in $\mathcal{G}$.

Remark 16 (Notation Alert)*.* Note that we will use the following interchangeably - P satisfies the local conditional independencies satisfied by $\mathcal{G}$ and $\mathcal{G}$ is an I-map for P, i.e

$$\text{Local-CI}(P, \mathcal{G}) \equiv \mathcal{I}_\ell(\mathcal{G}) \subseteq \mathcal{I}(P) \tag{16}$$

Definition 17 (I-equivalence)**.** Two Bayesian Networks $\mathcal{G}_1$ and $\mathcal{G}_2$ are $\mathcal{I}$-equivalent, if the encode the same dependencies, i.e

$$\mathcal{I}(\mathcal{G}_1) = \mathcal{I}(\mathcal{G}_2) \tag{17}$$

Theorem 18. *If $\mathcal{G}_1$ and $\mathcal{G}_2$ have the same skeleton, and the same v-structures (see D-separation), then they are $\mathcal{I}$-equivalent.*

Proof. Skipped. □

Theorem 19. *Given a distribution $P(x_1, x_2, \cdots, x_n)$ and a directed acyclic graph (DAG) $\mathcal{G}$,*

$$\textit{Local-CI}(P, \mathcal{G}) \iff \textit{Factorize}(P, \mathcal{G}) \tag{18}$$

Proof. ($\implies$) We essentially need to show that if $\mathcal{G}$ is an I-map for P, then P factorizes according to $\mathcal{G}$. Consider a topologically sorted order $x_1, x_2, \cdots, x_n$ in $\mathcal{G}$. Local-CI$(P, \mathcal{G})$ tells us that

$$\Pr(x_i | x_1, \cdots, x_{i-1}) = \Pr(x_i | \text{Pa}(x_i))$$

We can write

$$P(x_1, x_2, \cdots, x_n) = \prod_{i=1}^{n} P(x_i | x_1, \cdots, x_{i-1})$$

Each term in the product can be simplified due to the notion of Local-CI stated above, and we reach Equation 14, proving factorization.

($\impliedby$) Proof has been skipped. □

Minimal Construction

Our goal is to construct a minimal and correct BN $\mathcal{G}$ to represent P. A DAG $\mathcal{G}$ is correct if all Local-CIs that are implied in $\mathcal{G}$ hold in P, and a DAG $\mathcal{G}$ is minimal if we cannot remove any edge(s) from $\mathcal{G}$ and still get a correct BN for P.

In the setting, we define our oracle $\mathcal{O}$ to whom we can ask any query of the type "`Is X` $\perp\!\!\!\perp$ `Y|Z`?" pertaining to P and get a boolean answer. We will query the oracle several times to build up our BN. The following algorithm constructs such a BN -

Variables: $x_1, x_2, \cdots, x_n \longleftarrow$ ordered variables in $\mathcal{X}$
Independencies: $\mathcal{I} \longleftarrow$ set of independencies
$\mathcal{G} \longleftarrow$ Empty graph over $\mathcal{X}$
for $i = 1$ *to* n **do**
  $\mathbf{U} \longleftarrow \{x_1, \cdots, x_{i-1}\}$ // Set of candidate parents of x_i
  for $U' \subseteq \{x_1, \cdots, x_{i-1}\}$ **do**
    if $U' \subset U$ *and* $(x_i \perp\!\!\!\perp \{x_1, \cdots, x_{i-1}\} - U' | U') \in \mathcal{I}$ **then**
      $\mathbf{U} \longleftarrow U'$
    end
  end
  // Now we have the minimal set **U** satisfying $(x_i \perp\!\!\!\perp \{x_1, \cdots, x_{i-1}\} - \mathbf{U} | \mathbf{U})$
  // Now we set **U** to be the parents of x_i
  for $x_j \in \mathbf{U}$ **do**
    Add $x_j \to x_i$ in $\mathcal{G}$
  end
end
return $\mathcal{G}$

Algorithm 1: Minimal Bayesian Network Construction (I-Map)

We know sketch rough proofs for the claims of the algorithm.

Theorem 20. *The BN $\mathcal{G}$ constructed by algorithm 1 is minimal, i.e we cannot remove any edge from the BN while maintaining the correctness of the BN for P.*

Proof. By construction. A subset of $\text{ND}(x_i)$ were available when we chose parents of $\mathbf{U}$ minimally. □

Theorem 21. *$\mathcal{G}$ constructed by the above algorithm is correct, i.e, the local-CIs induced by $\mathcal{G}$ hold in P.*

Proof. The construction is such that $\text{Factorize}(P, \mathcal{G})$ holds everytime. Since $\text{Factorize}(P, \mathcal{G}) \implies \text{Local-CI}(P, \mathcal{G})$, the constructed BN satisfies the local-CIs of P. □

Question 22 (Construction of BN)**.** *Draw a Bayesian network over five variables $x_1, \cdots, x_5$ assuming the variable order x_1, x_2, x_3, x_4, x_5. For this*

ordering, assume that the following set of local CIs hold in the distribution:

$$x_1 \perp\!\!\!\perp x_2 \quad x_3 \perp\!\!\!\perp x_2 | x_1 \quad x_4 \perp\!\!\!\perp x_1, x_3 | x_2 \quad x_5 \perp\!\!\!\perp x_1, x_2 | x_3, x_4$$

Answer 23. Due to the ordering, we begin by inserting x_1 into the BN. Then we follow Algorithm 1 as follows:

1. x_2: Predecessor - x_1
   - *Query 1* - Is $x_2 \perp\!\!\!\perp x_1 | \emptyset$? : *Result 1* - True

   Thus, x_2 has no parents.
2. x_3: Predecessors - x_1, x_2
   - *Query 1* - Is $x_3 \perp\!\!\!\perp \{x_1, x_2\} | \emptyset$? : *Result 1* - False
   - *Query 2* - Is $x_3 \perp\!\!\!\perp x_2 | x_1$? : *Result 2* - True

   Thus, x_3 has x_1 as a parent, and x_2 as a non-descendent.
3. x_4: Predecessors - x_1, x_2, x_3
   - *Query 1* - Is $x_4 \perp\!\!\!\perp \{x_1, x_2, x_3\} | \emptyset$? : *Result 1* - False
   - *Query 2* - Is $x_4 \perp\!\!\!\perp \{x_2, x_3\} | x_1$? : *Result 2* - False
   - *Query 3* - Is $x_4 \perp\!\!\!\perp \{x_1, x_3\} | x_2$? : *Result 3* - True

   Thus, x_4 has x_2 as a parent, and x_1, x_3 as non-descendents.
4. x_5: Predecessors - x_1, x_2, x_3, x_4
   Check that it has x_3, x_4 as parents and x_1, x_2 as non-descendents.

Thus, finally we get the BN as in Figure 2

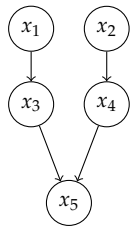

Figure 2: BN Example

Remark 24 (Importance of ordering). It is possible that a different ordering in $\mathcal{X}$ gives rise to a different BN, which although may be minimal, but may not be *optimal*. A minimal BN is defined for a given ordering, while an optimal BN is defined over all orderings. Example 25 shows such a case.

Example 25. Consider the BN shown in Figure 3, with the ordering x_1, x_2, x_3. We have $x_1 \perp\!\!\!\perp x_2$ as the only CI holding. Now consider our order changes to x_3, x_2, x_1. We follow the procedure as before, first inserting x_3 into the BN. Now

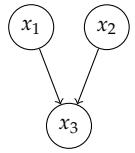

Figure 3: BN Ordering (optimal)

1. x_2: Predecessor - x_3
   - *Query 1* - Is $x_2 \perp\!\!\!\perp x_3 | \emptyset$? : *Result 1* - False

   Thus, we have x_3 as a parent of x_2.
2. x_1: Predecessor - x_3, x_2

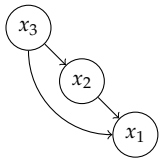

Figure 4: BN Ordering (non-optimal)

- *Query 1* - Is $x_1 \perp\!\!\!\perp \{x_2, x_3\} | \emptyset$? : *Result 1* - False
- *Query 2* - Is $x_1 \perp\!\!\!\perp x_2 | x_3$? : *Result 2* - False
- *Query 3* - Is $x_1 \perp\!\!\!\perp x_3 | x_2$? : *Result 3* - False

Thus, we have both x_2 and x_3 as parents of x_1.

Thus we get the BN as in Figure 4, and see that the BN is minimal, but not optimal.

D-Separation

Our goal is to know when we can guarantee $\mathbf{X} \perp\!\!\!\perp \mathbf{Y} | \mathbf{Z}$ holds given a BN $\mathcal{G}$. The further discussion provides some cases where we can guarantee $\mathbf{X} \not\perp\!\!\!\perp \mathbf{Y} | \mathbf{Z}$.

1. **Direct Connection:** If there is an edge $X \to Y$, then regardless of any $\mathbf{Z}$, we can find examples where they influence each other.
2. **Indirect Connection:** This means that there is a trail between the nodes in the graph. We consider the simple case when we a 3-node graph and Z is between X and Y. Consider the 4 diagrams to the left for reference.

   (a) *Indirect causal effect:* X cannot influence Y via Z if Z is observed.

   (b) *Indirect evidential effect:* This is similar to the previous case as dependence is a symmetric notion. Thus, X can influence Y via Z, only if Z is not observed.

   (c) *Common cause:* The conclusion is similar to (a) and (b).

   (d) *Common effect:* (v-structure) This case is a bit tricky to understand, but the crux is that X can influence Y when either Z or one of Z's descendents is observed.

   If we have flow of influence from X to Y via Z, we say that the trail $X \rightleftharpoons Y \rightleftharpoons Z$ is active.

$$\left.\begin{array}{r}\text{Causal trail: } X \to Z \to Y \\ \text{Evidential trail: } Y \to Z \to X \\ \text{Common cause: } X \leftarrow Z \to Y\end{array}\right\} \text{Active if and only if } Z \text{ is observed}$$
$$\left.\star \text{ Common effect: } X \to Z \leftarrow Y\right\}$$
$$\hookrightarrow \text{Active if and only if } Z \text{ or one of } Z\text{'s descendent is observed} \tag{19}$$

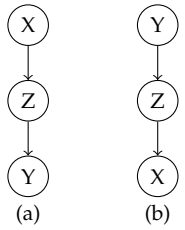

Figure 5: Causal and evidential effect

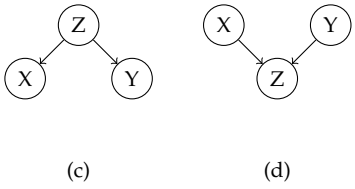

Figure 6: Common cause and common effect

Now, we can create a general notion of trails -

Definition 26. Let $\mathcal{G}$ be a BN, and $x_1 \rightleftharpoons \cdots \rightleftharpoons x_n$ be a trail in $\mathcal{G}$. Let $\mathbf{Z} \subset \{\text{observed variables}\}$. The trail is active given $\mathbf{Z}$ if

- ◇ Whenever we have a v-structure $x_{i-1} \to x_i \leftarrow x_{i+1}$, then x_i or one of its descendents are in $\mathbf{Z}$
- ◇ No other node along the trail is in $\mathbf{Z}$.

We can see that if $x_1 \in \mathbf{Z}$ or $x_n \in \mathbf{Z}$, then the trail is inactive.

Essentially in a DAG, $\mathbf{Z}$ d-separates $\mathbf{X}$ from $\mathbf{Y}$ if all paths $\mathcal{P}$ from any $\mathbf{X}$ to $\mathbf{Y}$ is blocked by $\mathbf{Z}$.

A path $\mathfrak{P}$ is *blocked* if it is inactive, i.e there is no flow of influence.

Definition 27 (d-separation)**.** Let $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$ be three sets of nodes in $\mathcal{G}$. We say that $\mathbf{X}$ and $\mathbf{Y}$ are d-separated given $\mathbf{Z}$, i.e d-sep$_{\mathcal{G}}(\mathbf{X}; \mathbf{Y}|\mathbf{Z})$ if there is no active trail between any node $x \in \mathbf{X}$ and $y \in \mathbf{Y}$ given $\mathbf{Z}$.

We use the same notation as $\mathcal{I}(P)$ as we can show that the independencies in $\mathcal{I}(\mathcal{G})$ are those guaranteed to hold for every distribution over $\mathcal{G}$ (Theorem 29).

Definition 28 (Global Markov independencies)**.** The set

$$\mathcal{I}(\mathcal{G}) \stackrel{\text{def}}{=} \{(\mathbf{X} \perp\!\!\!\perp \mathbf{Y}|\mathbf{Z}) : \text{d-sep}_{\mathcal{G}}(\mathbf{X}; \mathbf{Y}|\mathbf{Z})\} \tag{20}$$

denoting the set of independencies corresponding to d-separation is the set of global Markov independencies.

Theorem 29. *The d-separation test identifies the complete set of conditional independencies that hold in all distributions that conform to a given Bayesian Network.*

Proof. Skipped. □

Now, we look at another way to check d-separation over BNs, but first we define some terms.

Definition 30 (Ancestral Graph)**.** Given a graph $G = (V, E)$ and a set of nodes to focus on, say $V^* \subseteq V$, the ancestral graph G^A is a subgraph induced by $V^A = V^* \cup \mathcal{A}(V^*)$ where $\mathcal{A}(V^*)$ denotes the ancestors of V^*. Thus,

$$G^A = G\langle V^A \rangle = (V^A, \{(u, v)|(u, v) \in E \text{ and } u, v \in V^A\}) \tag{21}$$

Definition 31 (Markov Blanket)**.** Given a random variable Y in a random variable set $\mathcal{X} = X_1, X_2, \cdots, X_n$, it's Markov Blanket is any subset $\mathcal{S}$ of $\mathcal{X}$, conditioned on which other variables are independent with Y, i.e

$$Y \perp\!\!\!\perp \mathcal{X} \setminus \mathcal{S}|\mathcal{S} \tag{22}$$

Thus, we can infer Y from $\mathcal{S}$ itself, and the rest of the elements are redundant in observation.

Definition 33 (Moral graph)**.** A moral graph of a directed acyclic graph G is an undirected graph in which each node of the original G is now connected to its *Markov Blanket*.

Remark 32. Essentially we are finding an equivalent undirected graph for a DAG. We find all pairs of non-adjacent nodes having a common child, and add an undirected edge between them. Then we transform all directed edges in the resulting graph to undirected edges.

Given: Bayesian Network $\mathcal{G}$, Condition to check $\mathcal{C}$: $\mathbf{X} \perp\!\!\!\perp \mathbf{Y} | \mathbf{Z}$
$\mathcal{C} \leftarrow$ False
$G = (V, E) \leftarrow$ Underlying DAG in $\mathcal{G}$.
$G^A = (V^A, E^A) \leftarrow$ Ancestral graph of G
$G_M^A = (V_M^A, E_M^A) \leftarrow$ Moral graph of G^A using Note 32
// Delete the nodes in $\mathbf{Z}$ and all its connections
for $z \in \mathbf{Z}$ **do**
    $\Xi \leftarrow \{\}$
    for $u \in V$ *such that* $\xi = (u, z) \in E_M^A$ **do**
        $\Xi \leftarrow \Xi \cup \xi$
    end
    $E_M^A \leftarrow E_M^A \setminus \Xi$
    $V_M^A \leftarrow V_M^A \setminus \{z\}$
end
if $\mathbf{X}$ *and* $\mathbf{Y}$ *are disconnected in the resulting graph* **then**
    $\mathcal{C} \leftarrow$ True
end

Algorithm 2: Checking for independence in a BN

Definition 34 (Perfect map)**.** A graph $\mathcal{G}$ is a perfect map (P-map) for a set of independencies $\mathcal{I}$ if $\mathcal{I}(\mathcal{G}) = \mathcal{I}$. Also, $\mathcal{G}$ is P-map for a distribution P if $\mathcal{I}(\mathcal{G}) = \mathcal{I}(\mathcal{P})$.

Limitations

Consider the following set of CIs

$$x \perp\!\!\!\perp y \quad y \perp\!\!\!\perp z \quad z \perp\!\!\!\perp x \quad x \not\perp\!\!\!\perp \{y, z\} \tag{23}$$

If you try to draw the BN for any ordering of the variables $\{x, y, z\}$, you can check that you get extraneous edges, and we fail to capture at least one of the conditions above. Thus, a symmetric dependency of this sort is not possible to be represented by Bayesian Networks. This, gives rise to a different field of graphical modeling using *Markov Networks*, which can represent some of these situations.

Markov Random Fields

Intuition

We saw previously that we cannot draw a *perfect* I-map such that $\mathcal{I}(\mathcal{G}) = \mathcal{I}(P)$ for any distribution P using directed graphical models. Such too is the case with undirected graphical models, but they help us to represent some of these independencies which directed graphs couldn't.

To be added.

Cliques

Definition 35 (Complete Graph)**.** A complete graph is a simple undirected graph in which every pair of distinct vertices is connected by a unique edge.

Definition 36 (Clique)**.** A clique C in an undirected graph $G = (V, E)$ is a subset of vertices, $C \subseteq V$ such that every two distinct vertices are adjacent. Thus, the subgraph induced by C, i.e $G\langle C \rangle$, is a complete graph.

Definition 37 (Maximal Clique)**.** A clique that cannot be extended by including one more adjacent vertex (i.e it does not exist exclusively within the vertex set of a larger clique) is a maximal clique.

Definition 38 (Maximum Clique)**.** A maximum clique of a graph G, is a clique such that there is no other clique with more vertices.

With each clique C, we associate a potential function ψ, which is a provisional function of its arguments that assigns a pre-probabilistic score of their joint distribution. It is to note that ψ must be non-negative, but it shouldn't be interpreted as probability.

Gibbs Fields

A Gibbs Field is a representation of a set of random variables and their relationships. An example is in Figure 7. In this, the edges are undirected and imply some correlation between the connected nodes.

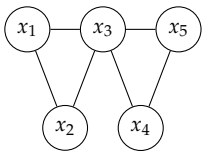

Figure 7: A Gibbs Field

Consider clique potentials as $\psi_i(c_i)$. Then the joint probability for any set of random variables $\mathcal{X} = \{x_1, \cdots, x_n\}$ represented by a Gibbs Field can be written as a product of clique potentials

$$P(\mathcal{X}) = \frac{1}{Z} \prod_{c_i \in C} \psi_i(c_i) \tag{24}$$

Z is a normalizing constant required to create a valid probability distribution, i.e

$$Z = \sum_{x} \prod_{c_i \in C} \psi_i(C_i) \tag{25}$$

Remark 39. For any Gibbs Field, there is a subset $\hat{C}$ of C consisting of only maximal cliques, which are not proper subsets of any other cliques. We write the potentials for these maximal cliques as products of all potentials of their sub-cliques, and thus state the joint probability as

$$P(\mathcal{X}) = \frac{1}{Z} \prod_{c_i \in \hat{C}} \hat{\psi}_i(c_i) \tag{26}$$

Formal Definition

Definition 40 (Markov Random Field)**.** A Markov Random Field (MRF) is a probability distribution P over variables $x_1, \cdots, x_n$ defined by an undirected graph G in which nodes correspond to variables x_i and has the form

$$P(x_1, x_2, \cdots, x_n) = \frac{1}{Z} \prod_{c \in C} \psi_c(x_c) \tag{27}$$

where

$$Z = \sum_{x_1, \cdots, x_n} \prod_{c \in C} \psi_c(x_c) \tag{28}$$

is the ***partition function*** which is the normalizing constant ensuring the distribution sums to 1.

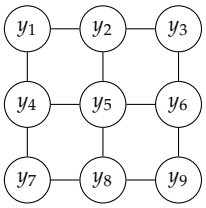

Figure 8: Relations in image pixels

As we saw earlier, if we have symmetric interactions, then UGMs become useful (such as labeling pixels in an image - see Figure 8). Define $y_i = 1$ if the pixel is a part of the foreground, and 0 else. Taking cliques of size 1, we have the potential functions $\psi_1(0)$ to $\psi_9(0)$ and $\psi_1(1)$ to $\psi_9(1)$. Now considering cliques of size 2, we have $\psi(0,0), \psi(0,1), \psi(1,0)$ and $\psi(1,1)$. Thus we write

$$\Pr(y_1, \cdots, y_9) \propto \prod_{k=1}^{9} \psi_k(y_k) \prod_{(i,j) \in E(G)} \psi(y_i, y_j) \tag{29}$$

Conditional Independencies

From now on, we will work on the UGM in Figure 8. Let

$$V = \{y_1, \cdots, y_9\}$$

. We define three types of CIs in UGMs as follows

In a graph G with vertices $V = \{x_1, \cdots, x_n\}$, $\mathcal{N}(x_i)$ denotes the neighbors of x_i in the graph

1. **Local CI:** $y_i \perp\!\!\!\perp V - \mathcal{N}(y_i) - \{y_i\} | \mathcal{N}(y_i)$

   $$y_1 \perp\!\!\!\perp y_3, y_5, y_6, y_7, y_8, y_9 | y_2, y_4$$

2. **Pairwise CI:** $y_i \perp\!\!\!\perp y_j | V - \{y_i, y_j\}$ if $(y_i, y_j) \notin E(G)$

   $$y_1 \perp\!\!\!\perp y_3 | y_2, y_4, y_5, y_6, y_7, y_8, y_9$$

3. **Global CI: $\mathbf{X} \perp\!\!\!\perp \mathbf{Y} | \mathbf{Z}$** if $\mathbf{Z}$ separates $\mathbf{X}$ and $\mathbf{Y}$ in the graph

$$y_1, y_2, y_3 \perp\!\!\!\perp y_7, y_8, y_9 | y_4, y_5, y_6$$

Checking for CI in MRFs is much more easier than BNs. The way to check is through graph separability. Consider the example given in **Global CI**. If we remove y_4, y_5 and y_6 from the graph along with their edges, we see that the components y_1, y_2, y_3 is disconnected from y_7, y_8, y_9, and hence the CI holds.

Theorem 41. *Let G be an undirected graph of $V = \{x_1, \cdots, x_n\}$ nodes, and let $P(x_1, \cdots, x_9)$ be a distribution. If P is represented by G, that is, if it can be factorized as per the cliques of G, then P will also satisfy the global-CIs of G. Thus*

$$\textit{Factorize}(P, G) \implies \textit{Global-CI}(P, G) \tag{30}$$

Note that for any arbitrary distribution, the converse doesn't hold, i.e in general for a distribution P

$$\text{Factorize}(P, G) \not\Longrightarrow \text{Global-CI}(P, G) \tag{31}$$

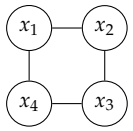

Figure 9: Sample UGM

We see this through a counter example. Consider the UGM in Figure 9, for the probability distribution $P(x_1, x_2, x_3, x_4)$ such that $P(x_1, x_2, x_3, x_4) = \frac{1}{8}$ when x_1, x_2, x_3, x_4 can take values from $\{0000, 1000, 1100, 1110, 1111, 0111, 0011, 0001\}$ else 0. It can be manually checked that all 4 Global-CIs hold in the graph, for example $x_1 \perp\!\!\!\perp x_3 | x_2, x_4$. Now consider the factors in the edges as $\psi(x_i, x_j)$. These will be positive, but that cannot represent the probability for $x_1, x_2, x_3, x_4 = 0101$.

Also, it is trivial to see that

$$\text{Global-CI} \implies \text{Local-CI} \tag{32}$$

But again through a counter example, we will show that the converse doesn't hold, i.e

$$\text{Local-CI} \not\Longrightarrow \text{Global-CI} \tag{33}$$

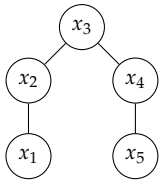

Figure 10: Sample UGM

Consider a distribution over 5 binary variables $P(x_1, \cdots, x_5)$ where $x_1 = x_2, x_4 = x_5$ and $x_3 = x_2 \wedge x_4$. Consider G as in Figure 10. Notice that all 5 Local-CIs hold in the graph, for example $x_1 \perp\!\!\!\perp \{x_3, x_4, x_5\} | x_2$. But notice that the graph also tells us that $x_2 \perp\!\!\!\perp x_4 | x_3$, but this is not present in the distribution P.

We also notice that

$$\text{Local-CI} \implies \text{Pairwise-CI} \tag{34}$$

But again through a counter example, we show that the converse doesn't hold, i.e

$$\text{Pairwise-CI} \not\Longrightarrow \text{Local-CI} \tag{35}$$

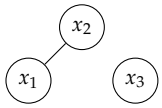

Figure 11: Sample UGM

Consider $P(x_1, x_2, x_3)$ defined over 3 binary variables such that $P(x_1, x_2, x_3) = \frac{1}{2}$ if $x_1 = x_2 = x_3$ and 0 else. Let G be as in Figure 11. See that both the Pairwise-CIs, i.e $x_1 \perp\!\!\!\perp x_3 | x_2$ and $x_2 \perp\!\!\!\perp x_3 | x_1$ hold in the graph, but the local CI $x_1 \perp\!\!\!\perp x_3$ doesn't hold.

We have made a lot of statements about converses not holding in arbitrary distributions, but the natural question to arise is, can we find distributions where all the relations hold? The answer is yes, and is shown by the following theorem, also called the *fundamental theorem of random fields* -

Theorem 42 (Hammerseley Clifford Theorem)**.** *If a positive distribution $P(x_1, \cdots x_n)$ confirms to the Pairwise-CIs of a UDGM G, then it can be factorized as per the cliques C of G as*

A distribution $P(\mathbf{x})$ is positive, if $P(\mathbf{x}) > 0 \; \forall \; \mathbf{x}$.

$$P(x_1, \cdots, x_n) \propto \prod_{C \in G} \psi_C(\mathbf{y}_C) \tag{36}$$

Proof. Skipped. □

Thus, in summary, for any arbitrary distribution P and UGM H,

$$\begin{aligned} &\text{Factorize}(P, H) \implies \text{Global-CI}(P, H) \implies \\ &\text{Local-CI}(P, H) \implies \text{Pairwise-CI}(P, H) \end{aligned} \tag{37}$$

and if P is positive, then

$$\text{Pairwise-CI}(P, H) \implies \text{Factorize}(P, H) \tag{38}$$

Hence, for a positive distribution, all three types of CIs are *equivalent*.

Minimal Construction

The question to answer is, given a positive distribution $P(x_1, \cdots, x_n)$ as an oracle $\mathcal{O}$ to which we can ask the query - is $X \perp\!\!\!\perp Y | Z$ and get a boolean answer, we need to draw a minimal and correct UGM G to represent P.

Denote $V = \{x_1, x_2, \cdots, x_n\}$ as the set of all variables. We see that there are two methods to draw the UGM -

1. *Using Pairwise-CIs:* For each pair of vertices (x_i, x_j), if $x_i \perp\!\!\!\perp x_j | V - \{x_i, x_j\}$ in P, add an edge between x_i and x_j in G.
2. *Using Local-CIs:* For each vector x_i, find the smallest subset U such that $x_i \perp\!\!\!\perp V - U - \{x_i\} | U$ in P. Then, add U to $\mathcal{N}(x_i)$ in P.

Example 43. To be added.

We had seen Markov Blankets before, but we re-define them in terms of a UGM.

Definition 44 (Markov Blanket)**.** The Markov Blanket (MB) of a variable x_i is the smallest subset of variables V that makes x_i conditionally independent of others given the MB, i.e

$$x_i \perp\!\!\!\perp V - MB(x_i) - \{x_i\} | MB(x_i) \tag{39}$$

Theorem 45. *The MB of a variable is always unique for a positive distribution.*

Proof. We will prove the following by contradiction. Let $x_i \in V$ and M_1, M_2 be two MBs. Let $\alpha = M_1 - M_2$ and $\beta = M_2 - M_1$, $M = M_1 \cap M_2, W = V - (M_1 \cup M_2)$. Note that, by definition, $x_i \perp\!\!\!\perp V - M_2 | M_2$ and $x_i \perp\!\!\!\perp V - M_1 | M_1$. Using this, we can write

$$x_i \perp\!\!\!\perp W, \alpha | M, \beta \quad x_i \perp\!\!\!\perp W, \beta | M, \alpha$$

For positive distributions, using intersection property, we can write

$$x_i \perp\!\!\!\perp W, \alpha, \beta | M$$

This implies that M is also a MB, but that is a contradiction since M_1 and M_2 were supposed to be minimal. Hence, the MB is unique. □

Definition 46 (Immorality)**.** In a directed acyclic graph, the structure of the form $x \to y \leftarrow z$ is an immorality provided there is no edge between x and z.

With this, we can restate the equivalence of BNs -

*Two BNs $\mathcal{G}_1$ and $\mathcal{G}_2$ are equivalent **iff** they have the same skeleton structure and the same set of immoralities.*

Interesetingly, UGMs were initially used to model interactions of atoms in gases and solids in 1800. A few other places where they are used are in

1. Markov Random Fields - Image Segmentation
2. Conditional Random Fields - Information Extraction
3. Social Networks
4. Bio-informatics - Annotating active sites in proteins

Conversion to and from Bayesian Networks

Theorem 47. *In a Bayesian Network $\mathcal{G}$, the Markov Blanket of a variable x_i is given as*

$$MB(x_i) = Pa(x_i) \cup Ch(x_i) \cup Sp(x_i) \tag{40}$$

where $Pa(x_i)$, $Ch(x_i)$ and $Sp(x_i)$ denote the parents, children and spouses (unmarried shared parent) of the children of x_i (if exists).

Proof. Only a flavor of the proof is provided. We have seen moralization of the Bayesian Network $\mathcal{G}$, and when we get $\mathcal{G}^M$ (i.e the moralized graph), notice that removing the parents, children and spouses disconnects the node from the graph. □

For example, in Figure 12, the MB of x_2 is given as

$$MB(x_2) = \{x_4\} \cup \{x_1\} \cup \{x_3\}$$

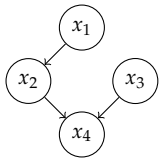

Figure 12: Sample BN

Theorem 48. *A Bayesian Network will have a perfect MRF if it has no immoralities.*

Proof. Skipped. □

We can ask the reverse question too. What condition should be posed on the MRF to have a perfect BN?

Definition 49 (Chordal Graph)**.** A cordal graph is a simple graph in which every graph cycle of length four or greater has a cycle chord.

With this, we can state that

Theorem 50. *An MRF can be perfectly converted to a BN if and only if it is chordal.*

Proof. Skipped. □

Inference Queries

We have seen two major types of compact representations of joint probability distributions in terms of graphs. Summarizing the expressions of the joint distribution, we can write

$$\text{For UGM: } \Pr(x_1, \cdots, x_n) = \frac{1}{Z} \prod_C \psi_C(x_C) \tag{41}$$

$$\text{For DGM: } \Pr(x_1, \cdots, x_n) = \prod_i \Pr(x_i | \text{Pa}(x_i)) \tag{42}$$

We get a very compact representation if $\text{Pa}(x_i)$ is small.

Given a probability distribution P, we can ask two major types of queries -

1. *Marginal probability queries over a sm'all subset of variables:* Given P, what is the marginal probability of x_1?.

$$\begin{aligned} \Pr(x_1) &= \sum_{x_2, \cdots, x_n} \Pr(x_1, \cdots, x_n) \\ &= \sum_{x_2=1}^{m} \cdots \sum_{x_n=1}^{m} \Pr(x_1, \cdots, x_n) \end{aligned} \tag{43}$$

   We can see that if each variable takes m values, then the brute-force computation of the marginal probability will take $\mathcal{O}(m^{n-1})$ time.

2. *Most likely labels of remaining variables (MAP queries):* Here, we ask questions of the form,

$$\mathbf{x}^* = \arg\max_{x_1, \cdots, x_n} \Pr(x_1, \cdots, x_n) \tag{44}$$

   An example of such a query could be - find the most likely entity labels of all words in a sentence.

Example 51 (Exact Inference)**.** Say we have a probability distribution over three binary variables as

$$P(x_1, x_2, x_3) = \frac{1}{Z} \psi_{12}(x_1, x_2) \psi_{23}(x_2, x_3)$$

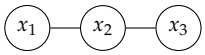

Figure 13: UGM for $P(x_1, x_2, x_3)$

The UGM for this is shown in Figure 13. Say we have the potential tables (each entry being $\psi_{ij}(a, b)$ representing the potential) as

x_1 \ x_2	0	1
0	5	2
1	1	4

x_2 \ x_3	0	1
0	2	10
1	5	3

For example, we see that $\psi_{12}(0, 0) = 5$. Let us find $P(x_1)$.

$$P(x_1) = \frac{1}{Z} \sum_{x_2 \in \{0,1\}} \sum_{x_3 \in \{0,1\}} \psi_{12}(x_1, x_2) \psi_{23}(x_2, x_3)$$

We multiply the above two tables to get an intermediate potential distribution $\psi_{123}(x_1, x_2, x_3)$ and get a three dimensional table as follows (note that the columns denote x_2 and the rows denote x_1)

$x_3 = 0$

	0	1
0	10	10
1	2	20

$x_3 = 1$

	0	1
0	50	6
1	10	12

For example, $\psi_{12}(0,0)\psi_{23}(0,0) = 2 \times 5 = 10$. The next computation is to sum over x_3.

$$P(x_1) = \frac{1}{Z} \sum_{x_2 \in \{0,1\}} \psi^*_{12}(x_1, x_2)$$

The table after sum denoting $\psi^*_{12}(x_1, x_2)$ is

		x_2 = 0	x_2 = 1
x_1	0	60	16
	1	12	32

Now we eliminate x_2 by summing over the row values, thus finally

$$\psi^*_1(x_1) = \frac{1}{Z} \begin{bmatrix} 76 \\ 44 \end{bmatrix}$$

Since this $P(x_1) = \psi^*_1(x_1)$, we immediately get to know that $Z = 76 + 44 = 120$.

Clearly, we see through the example that the calculation, even for three variables is cumbersome. Image doing this for thousands!

From the table in the above example, we can also calculate the assignment which gives the maximum probability. Note the ψ_{123} table made, and see that $x_1 = 0, x_2 = 0, x_3 = 1$ has the score of 50 giving the highest probability. But let us write this in a more algorithmic way

$$\mathbf{x}^* = \arg\max_{x_2} \arg\max_{x_2} \arg\max_{x_3} \psi_{12}(x_1, x_2)\psi_{23}(x_2, x_3)$$

Let us construct the table $\psi^{\max}_{12}(x_1, x_2)$ from the ψ_{123} table

		x_2 = 0	x_2 = 1
x_1	0	50 for $x_3 = 1$	10 for $x_3 = 0$
	1	10 for $x_3 = 0$	20 for $x_3 = 0$

Similarly, $\psi^{\max}_1(x_1)$ will be

$$\begin{bmatrix} 50 \text{ for } x_2 = 0, x_3 = 1 \\ 20 \text{ for } x_2 = 1, x_3 = 0 \end{bmatrix}$$

At last, we can do an argmax over x_1 to get the assignment $x_1 = 0, x_2 = 0, x_3 = 1$ for the score of 50.

Clearly, after the example, it is clear that we want to avoid the exponential overhead that brute-force approach applies.

Exact Inference on Chains

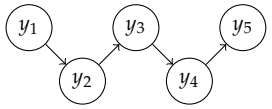

Figure 14: Chain graph

Consider the chain show in Figure 14.

We see that in the graph we would have potentials of the form $\psi_i(y_i, y_{i+1})$, and

$$\Pr(y_1, \cdots, y_n) = \prod_i \psi_i(y_i, y_{i+1}) \tag{45}$$

Note: Since we don't have immoralities, the MRF is equivalent to the undirected version of the graph. Say we want to calculate

$$\Pr(y_5 = 1) = \sum_{y_1, \cdots, y_4} \Pr(y_1, y_2, y_3, y_4, 1) \tag{46}$$

The key idea to reducing computations is to push summations past the multiplications, i.e

$$\begin{aligned}
\Pr(y_5 = 1) &= \sum_{y_1, \cdots, y_4} \Pr(y_1, y_2, y_3, y_4, 1) \\
&= \sum_{y_1} \sum_{y_2} \sum_{y_3} \sum_{y_4} \psi_1(y_1, y_2) \psi_2(y_2, y_3) \psi_3(y_3, y_4) \psi_4(y_4, 1) \\
&= \sum_{y_1} \sum_{y_2} \psi_1(y_1, y_2) \sum_{y_3} \psi_2(y_2, y_3) \sum_{y_4} \psi_3(y_3, y_4) \psi_4(y_4, 1) \\
&= \sum_{y_1} \sum_{y_2} \psi_1(y_1, y_2) \sum_{y_3} \psi_2(y_2, y_3) \mathcal{B}_3(y_3) \\
&= \sum_{y_1} \sum_{y_2} \psi_1(y_1, y_2) \mathcal{B}_2(y_2) \\
&= \sum_{y_1} \mathcal{B}_1(y_1)
\end{aligned} \tag{47}$$

We denote $\mathcal{B}_i(y_i)$ as the *belief* which flows from node $i+1$ to i. This is an efficient computation. In general, if we have a chain with n variables and each can take m values, the above algorithm (breaking into beliefs) takes time in order of $\mathcal{O}(nm^2)$.

Notice that we did the efficient computation for chains, the natural question is, for what other graphs can this be done?

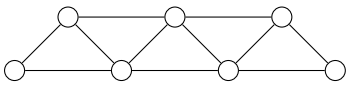
Figure 15: Triangular graph

Another one is shown in Figure 15. We define potential over each triangle (say ψ_{123}). If we follow a similar idea as the algorithm above, the time required for this computation will be $\mathcal{O}(nm^3)$.

Hardness of Inference and 3-SAT

The above discussion might lead to the thought that any graph G which can be factorized into small clique sizes might have an efficient

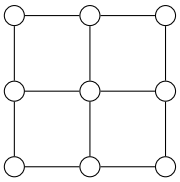

Figure 16: Grid graph

computation method (i.e polynomial time) of calculating the marginal probability.

The answer sadly is no, and a counter example is the grid graph shown in Figure 16.

We will now reduce the 3-SAT to inference in Bayesian Networks.

Definition 52 (3-SAT Problem)**.** Given n boolean variables $x_1, \cdots x_n$ such that $x_i \in \{T, F\}$. We define a literal ℓ to be the variable x_i or its negation $\neg x_i$ or $\bar{x}_i$. Given a set of K clauses $C_1, C_2, \cdots, C_K$ with each clause being

$$C_j = \ell_{j_1} \vee \ell_{j_2} \vee \ell_{j_3} \tag{48}$$

The 3-SAT problem is to decide if there exists an assignment of values to the n variables such that

$$C_1 \wedge C_2 \wedge \cdots \wedge C_K = T \tag{49}$$

Example 53. Consider $n = 4, K = 3$ and

$$C_1 = x_1 \vee \bar{x}_2 \vee \bar{x}_3$$
$$C_2 = x_2 \vee x_3 \vee \bar{x}_4$$
$$C_3 = x_4 \vee \bar{x}_1 \vee \bar{x}_2$$

In this case, having all $x_i = T$ for $i = \{1, 2, 3, 4\}$ solves the problem.

In the above example, we by chance got lucky and solved the problem, but in general for a large number of variables, it is not possible to go over all possible combinations of values, since it requires an exponential amount of time.

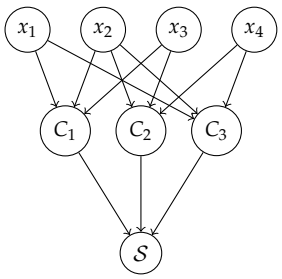

Figure 17: 3-SAT as BN

Now we represent 3-SAT as a Bayesian Network.

Let us do that in a *layer* sense. Let the first layer have all the variables as nodes and the next layer have all the clauses. Each clause will have 3 parents due to Equation 48. Finally, the third layer would have $\mathcal{S}$, which is the satisfiability (Equation 49), and it's parents would be all the clauses. Figure 17 shows the BN of Example 53.

Coming back to the general setting, for each variable x_i, we denote

$$\Pr(x_i) = \begin{cases} \frac{1}{2} & x_i = F \\ \frac{1}{2} & x_i = T \end{cases} \tag{50}$$

We also need to define $\Pr(C_j | \ell_{j_1}, \ell_{j_2}, \ell_{j_3})$. To do this, we assign a non-zero probability to only those which make $C_j = T$. This can be done uniformly (say out of the 8 assignments, 5 give a non-zero value, then 1 for each of those assignments, and 0 to rest - this is done because each C_j is a deterministic function of the literals). Finally, we write the last probability $\Pr(\mathcal{S} | C_1, \cdots, C_K)$ as 1 if $C_1, \cdots, C_K = T$, i.e all are true, and in the rest of the cases, we assign it as zero (note the

difference here - the table for each C_i had 8 rows, and the table for $\mathcal{S}$ has 2^K rows). The 2^K shows that it is not polynomial. This is again, not efficient.

One small change we can do is that instead of having a single $\mathcal{S}$ in the last layer, have $K-1$, such that each $\mathcal{S}_i$ is connected to C_{i-1} and C_i as parents, and each $\mathcal{S}_i$ is a parent of $\mathcal{S}_{i+1}$. This allows us to create the probability table as $\Pr(\mathcal{S}_j|\mathcal{S}_{j-1}, C_{j-1}, C_j)$ which represents the logic

$$\mathcal{S}_j = \mathcal{S}_{j-1} \wedge C_{j-1} \wedge C_j \tag{51}$$

This allows each $\mathcal{S}_j$ with 8 variables, bringing in the needed efficiency. More specifically, the space required now is polynomial, since each S_j requires only 2^4 space, each C_j requires 2^5 space and each x_j requires just constant (2) space. Thus overall the space required is $\mathcal{O}((K-1) \cdot 2^4 + K \cdot 2^5 + 2)$

Finally, if we can answer $\Pr(\mathcal{S}_j = 1) > 0$ positively, then we know that a 3-SAT assignment exists, else it does not.

Variable Elimination on General Graphs

We saw that using brute-force (i.e an exponential number of operations), we could calculate the normalizer Z. This is impractical, and hence we need a more efficient way to do so. Let's define the problem again -

Given an arbitrary set of potentials $\psi_C(x_C)$ in a graph G where C are the cliques in G, we need to find

$$Z = \sum_{x_1, \cdots, x_n} \prod_C \psi_C(x_C)$$

The algorithm to do so is as follows:

Input: Graph G
Variables: $x_1, x_2, \cdots, x_n$ present in a *good* ordering
$\mathcal{F} \longleftarrow \{\psi_C(x_C)$ where $C =$ cliques in $G\}$
for $i = 1$ *to* n **do**
  $\mathcal{F}_i \longleftarrow$ factors in $\mathcal{F}$ containing x_i
  $\mathcal{M}_i \longleftarrow$ product of factors in $\mathcal{F}_i$
  $m_i \longleftarrow \sum_{x_i} \mathcal{M}_i$
  $\mathcal{F} \longleftarrow (\mathcal{F} - \mathcal{F}_i) \cup \{m_i\}$
end

Algorithm 3: Variable Elimination

At the end, $\mathcal{F}$ consists of only a constant. Note that the product of factors isn't trivial, i.e we would need to multiply probability tables. To understand Algorithm 3, let's see an example.

Example 54. Say we have been given 5 variables, and the cliques are

$$\psi_{12}(x_1, x_2), \psi_{24}(x_2, x_4), \psi_{23}(x_2, x_3), \psi_{45}(x_4, x_5), \psi_{35}(x_3, x_5)$$

The corresponding graph is in Figure 18. We can see that

$$Z = \sum_{x_1 \cdots x_5} \psi_{12}(x_1x_2)\psi_{24}(x_2x_4)\psi_{23}(x_2x_3)\psi_{45}(x_4x_5)\psi_{35}(x_3x_5)$$

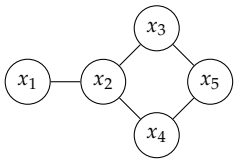

Figure 18: UGM for Example

Say our good ordering is x_1, x_2, x_3, x_4, x_5. So we start

◇ First variable x_1 -

$$\begin{aligned}
\mathcal{F}_1 &= \{\psi_{12}(x_1, x_2)\} \\
\mathcal{M}_1(x_1, x_2) &= \psi_{12}(x_1, x_2) \\
m_1(x_2) &= \sum_{x_1} \mathcal{M}_1 \\
\mathcal{F} &= \{\psi_{24}(x_2, x_4), \psi_{23}(x_2, x_3), \psi_{45}(x_4, x_5), \psi_{35}(x_3, x_5), m_1(x_2)\}
\end{aligned}$$

◇ Second variable x_2 -

$$\begin{aligned}
\mathcal{F}_2 &= \{\psi_{24}(x_2, x_4), \psi_{23}(x_2, x_3), m_1(x_2)\} \\
\mathcal{M}_2(x_2, x_3, x_4) &= \psi_{12}(x_1, x_2)\psi_{23}(x_2, x_3)m_1(x_2) \\
m_2(x_3, x_4) &= \sum_{x_2} \mathcal{M}_2 \\
\mathcal{F} &= \{\psi_{45}(x_4, x_5), \psi_{35}(x_3, x_5), m_2(x_3, x_4)\}
\end{aligned}$$

◇ Third variable x_3 -

$$\begin{aligned}
\mathcal{F}_3 &= \{\psi_{35}(x_3, x_5), m_2(x_3, x_4)\} \\
\mathcal{M}_3(x_3, x_4, x_5) &= \psi_{35}(x_3, x_5)m_2(x_3, x_4) \\
m_3(x_4, x_5) &= \sum_{x_3} \mathcal{M}_3 \\
\mathcal{F} &= \{\psi_{45}(x_4, x_5), m_3(x_4, x_5)\}
\end{aligned}$$

◇ Fourth variable x_4 -

$$\begin{aligned}
\mathcal{F}_4 &= \{\psi_{45}(x_4, x_5), m_3(x_4, x_5)\} \\
\mathcal{M}_4(x_4, x_5) &= \psi_{45}(x_4, x_5)m_3(x_4, x_5) \\
m_4(x_5) &= \sum_{x_4} \mathcal{M}_4 \\
\mathcal{F} &= \{m_4(x_5)\}
\end{aligned}$$

The above example showed how $\mathcal{F}$ is a singleton set at the end. We can also modify Algorithm 3 to get $\Pr(x_i)$ as follows -

◇ In line 1 of Algorithm 3, we choose a good ordering such that x_i is last

◇ The for loop in line 3 runs only for $n - 1$ iterations

◇ After this, at the end, $\mathcal{F}$ will consist of unnormalized values, sum of which will give Z, and each term divided by Z will give the required probability.

What if we want to compute the MAP query? For that, we do the follwing -

◇ In line 6, we have $\hat{m}_i = \max_{x_i} \mathcal{M}_i$ and we have to keep around the maximizing assignment

◇ In the end $\mathcal{F}$ consists of the required argmax.

Theorem 55. *The complexity of the Variable Elimination algorithm is $\mathcal{O}(nm^w)$ where w is the maximum number of variables in any factor.*

Sketch of proof. The bottleneck step in the algorithm's for loop is computing the product of factors, and in general if the factor has κ variables, then the time to do the product will be $\mathcal{O}(m^\kappa)$. □

In Example 54, we see that the time complexity is $\mathcal{O}(nm^3)$. If we started with x_2, our time complexity would've been $\mathcal{O}(nm^4)$.

More interestingly, if we have a star graph (Figure 19), and if we start with the centre node first, we encounter a very severe penalty in terms of time complexity. This elimination order will give you $\mathcal{O}(m^n)$ running time, while removing the non-central nodes first gives you just $\mathcal{O}(nm^2)$ running time.

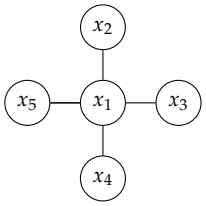

Figure 19: Star Graph

Unfortunately, choosing the optimal elimination order is NP hard in general. But for chordal (triangulated) graphs, the algorithm is polynomial time. But another problem we stumble upon is that if our graph is not triangular, optimal triangulation is NP hard (but there exist many heuristics to do this in polynomial time).

Definition 56 (Simplicial)**.** A vertex in a graph G is simplicial if its neighbors form a complete set.

Theorem 57. *Every triangulated graph is either complete or has at least two non-adjacent simplicial vertices.*

Proof. To be added. □

The goal is to find an optimal ordering for inferring $\Pr(x_1)$, which means x_1 should be last.

```
1 Input: Graph G, n = number of vertices in G
2 for i = 1 to n do
3 |   π_i ⟵ any simplicial vertiex in G except 1
4 |   Remove π_i from G
5 end
6 return ordering π_1, ··· , π_{n−1}
```

Algorithm 4: Optimal ordering for triangulated graph

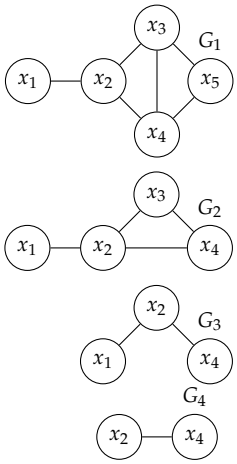

Figure 20: Sequence of graphs

Example 58. Consider the triangulated graph G_1 given on the right for which we have to find the optimal ordering. We go over the iterations as follows

1. In G_1, we have x_1 and x_5 as simplicial vertices. Say we remove x_5 first, to get G_2.
2. In G_2, we have x_1, x_3 and x_4 as the simplicial vertices. Say we remove x_3 to get G_3.
3. In G_3, we have x_1 and x_4 as simplicial vertices. Say we remove x_1 to get G_4.
4. In G_4 we have x_2 and x_4 as simplicial vertices. Say we remove x_2.

The sequence of graphs is shown in Figure 20.

Thus, finally we get the ordering x_5, x_3, x_1, x_4, x_2 as an optimal ordering.

Multiple Inference Queries

The above subsection showed how we can calculate the optimal ordering and a single inference query. But say, we have been given a chain graph with potentials as $\psi_{i,i+1}(x_i, x_{i+1})$, say we need all $\Pr(x_1), \cdots, \Pr(x_n)$, can we do that faster? A no-brain method would be to use variable elimination n times to get $\mathcal{O}(n^2m^2)$.

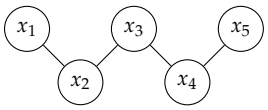

Figure 21: Chain graph

Say I have the chain graph in Figure 21. If we need to calculate $\Pr(x_1)$, we first remove x_5. This is followed by removing x_4, x_3 and x_2.

Now if we want to calculate $\Pr(x_2)$. We can reuse the computation done in removing x_5, x_4 and x_3.

We will see that if we skillfully reuse such computation, if each variable elimination run takes time $\mathfrak{T}$, the time for n inference queries will take just $2\mathfrak{T}$.