Probabilistic Graphical Models: Principles and Techniques

Notes by Jin Park

November - December 2017

Abstract

This report contains notes to *Probabilistic Graphical Models: Principles and Techniques* by Daphne Koller and Nir Friedman. It only covers one portion of the book, addressing the problem of **representation**. Some topics covered are directed and undirected networks, temporal networks, Gaussian networks, and exponential families. These notes are not complete in topics or depth, so interested readers should further purchase the book for a more rigorous representation of the material. These notes are NOT endorsed by the authors.

Contents

1 Foundations

1.1 Probability Theory

When we refer to the **confidence** of an event occuring, then we can use probability to quantify how sure we are that the event will occur. Formally, we define an *outcome space* Ω as the space of all possible outcomes and *events* as the subset of Ω. A *probability distribution* is a mapping from events to real values, and has the following three properties:

- $P(a) \geq 0$ for all $a \in S$
- $P(\Omega) = 1$
- If $a, b \in S$ and $a \cap b = \emptyset$, then $P(a \cup b) = P(a) + P(b)$

The interpretation of probability above is known as the *subjective* view of probability. It views probability as a subjective statement about an individual's belief that an event will come about. The second interpretation of probability is known as the *frequentist* view. Probability is simply the frequency of events.

1.1.1 Basic Concepts

Here's a potpourri of basic concepts in probability.

Conditional Probability
Information may change the confidence that we have of some event occurring. How do we account for this change in probability?

$$P(b|a) = \frac{P(b \cap a)}{a} \tag{1}$$

Chain Rule and Bayes Rule
From the definition of conditional probability, we immediately get the chain rule.

$$P(a_1 \cap a_2 \cap ... \cap a_k) = P(a_1)P(a_2|a_1)...P(a_k|a_{k-1}...a_1) \tag{2}$$

We also get Bayes Rule.

$$P(a|b) = \frac{P(b|a)P(a)}{P(b)} \tag{3}$$

Random Variables
We may want to analyze an attribute (age group or symptoms). *Random variables* are a formal machinery for discussing attributes and the values of different outcomes. It is usually denoted by $X_{age=17}$ but it is actually a function $X(age = 17) \to \mathbb{R}$.
We may be given a *joint distribution* $P(X_1...X_k)$. To find the *marginal distribution*, we must sum up all the possible assignments of the other variables.

$$P(X_1) = \sum_i P(X_1, x_2^{(i)}, ...x_k^{(i)})$$

Note that random variables are simply sets of events which conforms to an attribute. This means that all rules that apply to events (conditioning, chain rule, bayes rule) apply to random variables as well.

Independence
Some information is not useful. We use *independence* as a way to describe the case when $P(a|b) = b$.

$$P \models (a \perp b) \qquad \text{if } P(a|b) = P(a) \text{ or } P(B) = 0 \qquad (4)$$
$$P \models (a \perp b) \qquad \text{iff } P(a \cap b) = P(a)P(b) \qquad (5)$$

Conditional Independence
Two events may be independent given certain information. Conditional independence is defined:

$$P \models (a \perp b|\gamma) \qquad \text{if } P(a \cap b|\gamma) = P(a|\gamma)P(b|\gamma) \qquad (6)$$

Some independence properties include

$$(X \perp Y|Z) \Rightarrow (Y \perp X|Z) \qquad \text{Symmetry} \qquad (7)$$
$$(X \perp Y, W|Z) \Rightarrow (X \perp Y|Z) \qquad \text{Decomposition} \qquad (8)$$
$$(X \perp Y, W|Z) \Rightarrow (X \perp Y|Z, W) \qquad \text{Weak union} \qquad (9)$$
$$(X \perp W|Z, Y) \wedge (X \perp Y|Z) \Rightarrow (X \perp Y, W|Z) \qquad \text{Contraction} \qquad (10)$$

For positive distributions:

$$(X \perp Y|Z, W) \wedge (X \perp W|Z, Y) \Rightarrow (X \perp Y, W|Z) \quad \text{Intersection} \qquad (11)$$

1.1.2 Querying a Distribution

Throughout the book, we will be discussing the distributions of a subset of random variables given some information. A *probability query* consists of two parts:

- Evidence: subset E of random variable assignments in the model
- Query variables: subset Y of random variables

We will attempt to compute the *posterior probability distribution*, $P(Y|E = e)$.

We may also want to find the most probable assignment $y*$ given the information e. This is known as the *maximum a posteriori query* (MAP query) or *most probable explanation* (MLE)

$$MAP(W|e) = argmax_w P(w, e)$$

where $W = \mathbb{X} - E$, all other random variables besides E.

We may not want to find the most probable assignment for all other random variables W. Using only a subset of the variables $Y \subseteq W$ and $Z = W - Y$, we define *marginal MAP query* as

$$MAP(Y|e) = argmax_Y \sum_Z P(Y, Z|e)$$

1.1.3 Continuous Spaces

Some random variables, like blood pressure, is continuous. We use a *probability density function* to describe the distribution.

$$P(a \leq X \leq b) = \int_a^b p(x)dx$$

$$\int_{Val(X)} p(x)dx = 1$$

Two important continuous distributions are the uniform and Gaussian distributions.

$$p(x) = \begin{cases} \frac{1}{b-a} & b \leq x \leq a \\ 0 & else \end{cases} \qquad \text{Uniform} \qquad (12)$$

$$p(x) = \frac{1}{\sqrt{2\pi\sigma}} exp(-\frac{(x-\mu)^2}{2\sigma^2} \qquad \text{Gaussian} \qquad (13)$$

1.2 Information Theory

Information theory is the theory of effectively coding and transmitting information. We must consider how to efficiently encode data to maximize the amount of data per channel and how to deal with noisy channels.

1.2.1 Compression and Entropy

Say that we want to send a large corpus of English text through a channel. One possible way to send it is to send it as an ASCII text. Another more efficient way is to create a dictionary of the words in the corpus and change each word in the corpus into a word index specified by the dictionary. The final way is *Huffman encoding*. The main idea of Huffman encoding is to assign variable-length codes to input characters of which lengths correspond to the frequency of the corresponding word. The most frequent word gets the smallest character and the least frequent word gets the longest character. To create a Huffman tree, you must

1. Create a leaf node for each unique (word, frequency pair). Add these nodes into a priority queue.
2. Extract two leaf nodes with the lowest frequency.
3. Create an internal node with frequency equal to the sum of the two node frequencies. Add the first extracted node as the left child and the second extracted node as the right child.
4. Repeat step 2 and 3 until the priority queue has one node.

Is Huffman encoding the best we can do? Surprisingly, yes. The notion of entropy gives us the precise lower bound for the expected number of bits required to encode instances sampled from a large corpus. The *entropy* of a distribution over a random variable X is defined

$$H_P(X) = E_P[log\frac{1}{P(x)}] = \sum_x P(x)log\frac{1}{P(x)}$$

where we treat $0log(\frac{1}{0}) = 0$.

Another way to view entropy is as a measure of uncertainty about the value of X. Consider a game of asking yes/no questions until we pinpoint the value X. The entropy of X is the average number of questions we need to ask to get the answer. It might be tempting to draw analogies between entropy and variance. However, they are very different. Consider a bimodal distribution. Variance increases as the distance between the peaks increase, but entropy does not.

1.2.2 Conditional Entropy

Say that we want to encode values X and Y. What is the cost of encoding X if we already encoded Y? *Conditional entropy* is defined as

$$H_P(X|Y) = H_P(X,Y) - H_P(Y) - E_P[log\frac{1}{P(X|Y)}]$$

We can also find the joint distribution with the chain rule.

$$\begin{aligned} H_P(X_1...X_k) &= E[\frac{1}{P(X_1...X_k)}] \\ &= H_P(X_1) + H_P(X_2|X_1) + ... + H_P(X_k|X_1...X_k) \end{aligned}$$

We know $H_P(X|Y) \leq H_P(X)$ but by how much? In other words, how much information did Y give about X? The *mutual information* between X and Y is

$$I_P(X;Y) = H_P(X) - H_P(X|Y) = E_P[log\frac{P(X|Y)}{P(X)}]$$

Mutual information satisfies several properties:

- $0 \leq I_P(X;Y) \leq H_P(X)$
- $I_P(X;Y) = I_P(Y;X)$
- $I_P(X;Y) = 0$ iff $X \perp Y$

1.2.3 Relative Entropy and Distance between Distributions

We may want to compare two distributions. For example, we might want to approximate a distribution with a simpler one and evaluate the quality of the approximate distribution. A *distance metric* satisfies the following properties.

- Positivity: $d(P;Q) \geq 0$ and $d(P;Q) = 0$ iff $P = Q$
- Symmetry: $d(P;Q) = d(Q;P)$
- Triangle Inequality: $d(P;R) \leq d(P;Q) + d(Q;R)$

Although it is not a distance metric, relative entropy is often used to compare two distributions. Also known as the KL-divergence, we define *relative entropy* as

$$D(P||Q) = E_P[log \frac{P(X_1 ... X_n)}{Q(X_1 ... X_n)}]$$

Relative entropy does not satisfy symmetry and the triangle inequality, so it is not a distance metric. Relative entropy does, however, include conditioning

$$D(P(X|Y)||Q(X|Y)) = E_P[log \frac{P(X|Y)}{Q(X|Y)}$$

and so satisfies the chain rule

$$\begin{aligned} D(P||Q) = & D(P(X_1)||Q(X_1)) + \\ & D(P(X_2|X_1)||Q(X_2|X_1)) + ... \\ & D(P(X_n|X_1 ... X_{n-1})||Q(X_n|X_1 ... X_{n-1})) \end{aligned}$$

The relative entropy of marginal distributions is upper-bounded by the relative entropy of joint distributions.

$$D(P(X)||Q(X)) \leq D(P(X,Y)||Q(X,Y))$$

If $(X \perp Y)$ then

$$D(P(X,Y)||Q(X,Y)) = D(P(X)|Q(X)) + D(P(Y)|Q(Y))$$

Other distance metrics include

- L_1 distance: $||P - Q||_1 = \sum_x |P(x) - Q(x)|$
- L_2 distance: $||P - Q||_2 = (\sum_2 (P(x) - Q(x))^2$
- L_∞ distance: $||P - Q||_\infty = max_x |P(x) - Q(X)|$
- Variational Distance: $D_{var}(P;Q) = max_{a \in S} |P(a) - Q(a)|$

Variational distance is the maximal distance in probability for any event. It turns out that variational distance is half of L_1 norm!

Although these distance metrics are useful, relative entropy is usually more applicable to probability distributions because it follows the chain rule. Luckily, relative entropy is an upper bound for L_1 norm and consequently variational distance.

$$||P - Q||_1 \leq ((2ln2)D(P||Q))^{\frac{1}{2}}$$

2 Bayesian Networks

2.1 Exploiting Independence Properties

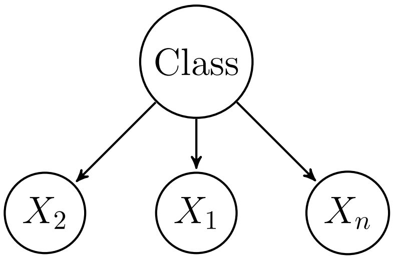
The main question of the representation problem is how to represent high dimensional distributions compactly. Using independent properties, we can reduce the amount of parameters needed to represent the distribution.

Consider a series of independent coin tosses. Assigning the result of each coin toss to random variable X_i, we would need 2^n parameters to specify the distribution; each assignment for $(X_1...X_n)$ requires a probability. However, we recognize that the probability of each coin toss is independent of each other. Then, we can specify θ_i for the probability that the coin toss will be heads. Then, $P(x_i...x_n) = \prod_i \theta_i$.

Formally, the space of all joint distributions $p_1...p_n$ is a 2^n subspace of $\mathbb{R}$, the set $\{(p_1...p_n) \in \mathbb{R}^{2^n} : p_1 + ... + p_n = 1\}$. On the other hand, the factorization of the distribution is an n-dimensional manifold in $\mathbb{R}^{2^n}$. This factorization, while being compact, does not have the same expressive power as the joint distribution.

Bayesian networks are based on the *conditional independence properties*. It will attempt to change the joint distribution $P(I, S)$ into the conditional distribution $P(I)P(S|I)$ via the chain rule.

A simple Bayesian network model is the *Naive Bayes model*. This model attempts to predict a class C based on individual features X_i.



The Naive Bayes assumption is that each feature are conditionally independent given the class: $(X_i \perp X_{-i}|C)$ for all i. With this assumption, we can factorize the distribution.

$$P(C, X_1, ..., X_n) = P(C) \prod_{i=1}^{n} P(X_i|C_i)$$

If we know each feature, we can predict the the class by maximizing the *class bias*.

$$\frac{P(C = c^1|x_1...x_n)}{P(C = c^2|x_1...x_n)} = \frac{P(C = c^1)}{P(C = c^2)} \prod \frac{P(x_i|c^1)}{P(x_i|c^2)}$$

In the case of two classes, this model uses 2n+1 parameters! Unfortunately the Naive Bayes assumption is the source of this model's flaws. In many cases, features tend to be correlated to with each other, such as symptoms for medical diagnosis or pixels in computer vision. In these cases, the Naive Bayes "overcounts" correlated features.

2.2 Bayesian Networks

At the core of the Bayesian network is a directed acyclic graph G. The graph structure can be viewed in two ways. One, it is a data structure that provides a skeleton for representing a joint distribution compactly through *conditional probability distributions* (CPDs). Second, it is a compact representation for a set of conditional independence assumptions about a distribution. We will examine both viewpoints in depth throughout the chapter.

Bayesian networks are very useful for several types of reasoning. *Causal reasoning* is the prediction of effects from causal factors. *Evidential reasoning* is the explanation of causes from effects. In evidential reasoning, there may be many causes for an event. *Intercausal reasoning* is the "explaining away" of things, where different causes of the same effect interact.

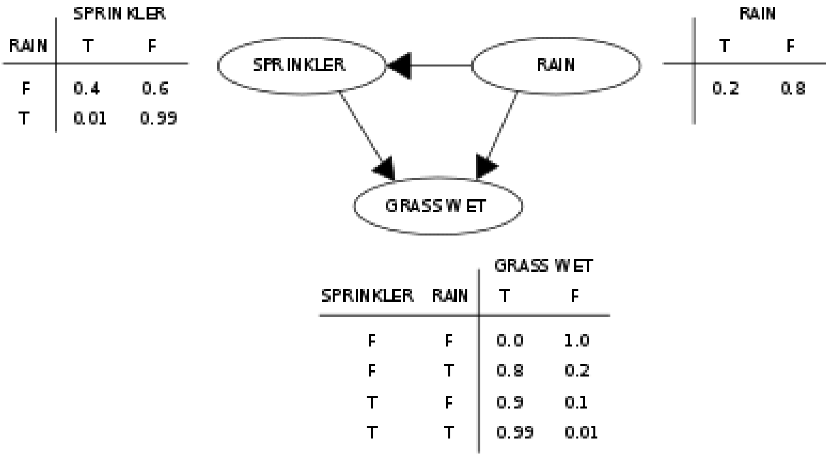
Formally, a *Bayesian network structure* is a directed acyclic graph whose nodes represent random variables $X_1 ... X_n$. Let Pa_{X_i} denote the parents of X_i in G and $Nondescendants_{X_i}$ denote variables that are not descendants of X_i. For each X_i

$$(X_i \perp Nondescendants_{X_i} | Pa_{X_i})$$

also called *local independencies*, $I_l(G)$. We will see in the following section that these independencies are equivalent to the conditional factorization discussed in section 2.1.

2.2.1 From Graphs to Distributions

We have defined our Bayesian network structure based only on the local independencies. In this section, we will see how this representation can be used to specify the standard way of representing a Bayesian, as a graph annotated by conditional probability distributions.



	SPRINKLER	
RAIN	T	F
F	0.4	0.6
T	0.01	0.99

RAIN	
T	F
0.2	0.8

		GRASS WET	
SPRINKLER	RAIN	T	F
F	F	0.0	1.0
F	T	0.8	0.2
T	F	0.9	0.1
T	T	0.99	0.01

The Bayesian network G is an *I-map* of probability distribution P if G associates with the set of independencies of P, $I(P)$. If G is an I-map of P, we can compress the joint representation P with the following factorization

$$P(X_1 ... X_n) = \prod_{i=1}^{n} P(X_i | Pa_{X_I}) \tag{14}$$

also called the *chain rule for Bayesian networks.* In a distribution of n binary random variables, we reduce the 2^n independent parameters into $n * 2^k$ parameters, where k is the maximum number of parents per node.

The above illustrates the statement that a set of independencies can construct the factorization to local conditional probability models. The converse is also true; if P is a joint distribution that factorizes according the G, aka satisfies eq. 14, G is an I-map for P. All distributions that can be factored to the chain rule of Bayesian networks can be represented by the independencies associated with the Bayesian network.

2.3 More Independencies

Independencies gives a useful framework for the task of inference, allowing us to substantially reduce the computation of a probability query. In the previous sections, we have only analyzed local independencies of Bayesian networks $I_l(G)$. What other independencies are implied by G? In other words, **given $I_l(G)$, for what subsets of random variables $X, Y, Z \in G$ can we guarantee $(X \perp Y|Z)$?**

2.3.1 D-separation

As a building block to global independencies, we'll consider four *local dependency trails* that will cause r.v. X, Y to be dependent given Z.

- Causal trail $(X \rightarrow Z \rightarrow Y)$: active iff Z is not observed
- Evidential trail $(X \leftarrow Z \leftarrow Y)$: active iff Z is not observed
- Common cause $(X \leftarrow Z \rightarrow Y)$: active iff Z is not observed
- Common effect $(X \rightarrow Z \leftarrow Y)$: active iff Z or descendant is observed. Also known as *v-structure*

A trail $X_1 \rightleftharpoons ... \rightleftharpoons X_n$ is active if it follows all of these local independencies.

Now, we can use active trails to define d-separation. Let X, Y, Z be three sets of nodes in G. X and Y are d-separated give Z if there is no active trail between any node in X, Y given Z.

$$I(G) = (X \perp Y|Z) : d - sep_G(X; Y|Z)$$

are called *global Markov independencies.* D-separation as a method of finding independencies is sound and complete.

D-separation as a way to infer independencies would be useless if there were not an efficient way for determining d-separation between two sets of random variables X, Y given Z. One way would be to check for active trails between every pair of random variable between X, Y. However, this may take an exponential amount of time depending on the graph. Fortunately there is a much faster algorithm only requiring linear time. It follows two phases

1. From leaves to roots, mark all parents of nodes in Z
2. Use breadth first search to follow local dependency trails.

2.3.2 I-Equivalence

It might be useful to analyze the similarity of one Bayesian network with another. Two graph structures K_1 and K_2 are *I-equivalent* if $I(K_1) = I(K_2)$. The set of all graphs over P is partitioned into a set of mutually exclusive and exhaustive I-equivalent classes, which are the set of equivalence classes induced by the I-equivalence relation. This means that for a given probability distribution P, there may be multiple Bayesian networks associated the distribution, $I(P) = I(G_1) = I(G_2)$. No intrinsic property allows us to favor one graph over the other. Luckily, there is a way to specify the entire class of Bayesian networks associated with P, a topic discussed further in the next section.

Which Bayesian networks are I-equivalent? If G_1 and G_2 have the same skeleton and the same set of v-structures, then they are I-equivalent. The skeleton of a Bayesian network is simply an undirected graph with edges between every pair of adjacent vertices X, Y in G. Unfortunately, this characterization is only sufficient, not complete; there exists I-equivalent graphs that do not have the same v-structure. Consider complete graphs. The global independencies of these graphs are empty, but any two may have different set of v-structures.

The reason for non-uniqueness in this example is the *covering edge.* In a v-structure, a covering edge is the edge between the parents. For $(X \rightarrow Z \leftarrow Y)$, $(X \rightarrow Y)$ or $(X \leftarrow Y)$ is a covering edge. If there is no covering edge, the v-structure is known as an *immorality.* Hence, we have our sufficient and complete condition for I-equivalence: G_1 and G_2 are I-equivalent iff they have the same skeleton and immoralities.

2.4 From Distribution to Graph

So far, we have seen the usefulness of Bayesian network as a compact representation of P and its independencies. However, for distributions in real life, P will not follow the nice factorization of Bayesian Networks. To what extent can we construct a graph G whose independencies are reasonable surrogates for the independencies in P?

One way to represent distribution P may be to take any Bayesian network that is an I-map for P. The problem with this solution is obvious; a complete graph is an I-map for every distribution, since its conditional factorization is the joint distribution. A nontrivial representation is a *minimal I-map*, an I-map for which the removal of a single edge renders it not an I-map of P. The minimal I-map can be found with the Build-Minimal-I-Map algorithm.

The main idea Build-Minimal-I-Map is simple. Let Construct a Bayesian network by adding one node at a time. Find parents of node X_i by finding the minimal set U that satisfies $(X_i \perp \{X_1 ... X_i - 1\} - U | U)$. Set these as parents because they are d-separate the current node from all other existing nodes.

Unfortunately, there are many different minimal I-maps for a particular distribution, dependent on the ordering of random variables inputted into Build-Minimal-I-Map 1. In fact, minimal I-maps may capture little or no independency at all, since it only needs to satisfy the condition that removing one node will render it not an I-Map of the distribution.

Algorithm 1 Procedure to build a minimal I-map given an ordering

```
procedure Build-Minimal-I-Map($X_1 ... X_n$ an ordering of random variables, $I$ a set of independencies.)
    Set $G$ to an empty graph over $X$
    for $i = 1, ..., n$ do
        $a \leftarrow \{X_1, ..., X_{i-1}\}$ // U is the current candidate for parents of $X_i$
        for $U' \subseteq \{X_1, ..., X_{i-1}\}$ do
            if $U' \subset U$ and $(X_i \perp \{X_1, ..., X_{i-1}\} - U' | U') \in I$ then
                $U \leftarrow U'$
            end if
        end for
        // $U$ is a minimal set satisfying $(X_i \perp \{X_1, ..., X_{i-1}\} - U | U)$
        // Now set U to be the parents of $X_i$
        for $X_j \in U$ do
            Add $X_j \rightarrow X_i$ to $G$
        end for
    end for
end procedure
```

A graph K is a *perfect I-map* (P-map) for a set of independencies of distribution P if $I(K) = I(P)$. Like a minimal I-map, there are many different P-maps for a set of independencies. To resolve this issue, we will create a partially directed acyclic graph (PDAG) that represents the all P-maps of P. Although P-maps are not unique, they are unique up to I-equivalence between networks. In the discussion in section 2.3.2, the class of I-equivalent Bayesian networks are defined by its skeleton and immoralities. We use these two components to define the PDAG $G*$ used to represent the P-map class of the distribution.

The first task is to identify the undirected skeleton of $G*$. If X and Y are two variables not adjacent in $G*$, then there exists a *witness set* U such that $(X \perp Y | U)$. The witness set U is a witness to X and Y's independence. Also note that the size of the witness set is bounded by the maximum indegree of the graph, because each node is independent of all other nodes given its parents. With this fact, we can obtain the skeleton of the distribution.

Algorithm 2 Recovering the undirected skeleton for a distribution

```
procedure Build-PMap-Skeleton($X_1, ..., X_n$ = Set of random variables, $P$ = distribution, $d$ = Bound on witness set)
    Let $H$ be the complete undirecteed graph over X
    for $X_i, X_j$ do
        $U_{X_i,X_j} \leftarrow \emptyset$
        for $U \in Witnesses(X_i, X_j, H, d)$ do
            // Consider U as a witness set for $X_i, X_j$
            if $P \models (X_i \perp X_j | U)$ then
                $U_{X_i,X_j} \leftarrow U$
                Remove $X_i - X_j$ from $H$
                break
            end if
        end for
    end for
    return $(H, \{U_{X_i,X_j}\})$
end procedure
```

This algorithm is in $O(n^2 * \binom{n-2}{d}) = O(n^{d+2})$.

The second task is to mark immoralities within the skeleton created by Build-PMap-Skeleton. A *potential immorality* in skeleton H is three nodes $X - Z - Y$ that do not contain an edge between X and Y. There are four cases to consider, $X \rightarrow Z \rightarrow Y$, $X \leftarrow Z \leftarrow Y$, $X \leftarrow Z \rightarrow Y$, and $X \rightarrow Z \leftarrow Y$. If $G*$ indeed contains the immorality $X \rightarrow Z \leftarrow Y$, we know Z will not be contained in any witness sets $U_{X,Y}$. Also, if $G*$ contains the first three cases, we know Z will be contained in all witness sets $U_{X,Y}$. Combining these two results, $X - Z - Y$ **is an immorality iff Z is not in the witness set for X and Y.** This motivates the following algorithm.

Algorithm 3 Marking immoralities in the construction of a perfect map

```
procedure Mark-Immoralities($X_1 ... X_n$ = the set of random variables, $S$ = skeleton, $U_{X_i,X_j}$ = witnesses found by Build-PMap-Skeleton)
    $K \leftarrow S$
    for $X_i, X_j, X_k$ such that $X_i - X_j - X_k \in S$ and $X_i - X_k \notin S$ do
        if $X_j \notin U_{X_i,X_j}$ then
            Add the orientations $X_i \rightarrow X_j$ and $X_j \leftarrow X_k$ to $K$
        end if
    end for
end procedure
```

The final task is to resolve any inconsistencies with the graph. The first rule (R1) is the case when $(X \rightarrow Y - Z)$. The undirected edge $Y - Z$ must be oriented right, or it would be an immorality. The second rule (R2) is derived from the acyclicity constraint. If $(X \rightarrow Y \rightarrow Z)$ and $X - Z$, then $X - Z$ must be oriented toward the right to not create a cycle. The final rule (R3) is more complex, but uses acyclic and immorality constraints.

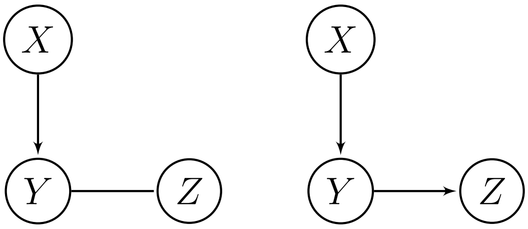


Figure 1: R1

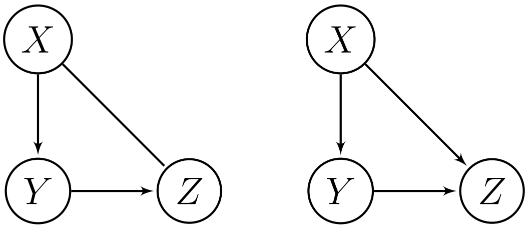


Figure 2: R2

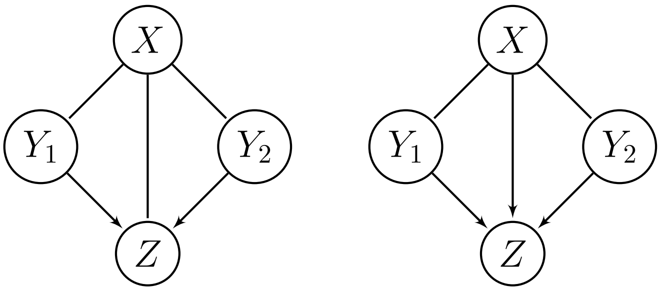


Figure 3: R3

Using these three rules, we can propagate constraints through our skeleton oriented by Mark-Immoralities. The algorithm is implemented as follows.

Algorithm 4 Finding the class PDAG characterizing the P-map of distribution

```
procedure Build-PDAG($X_1...X_n$ = the set of random variables, $P$ = distribution)
    $S, \{U_{X_i,X_j}\} \leftarrow$ Build-PMap-Skeleton($X_1...X_n, P$)
    $K \leftarrow$ Find-Immoralites($X_1...X_n, S, \{U_{X_i,X_j}\}$)
    while not converged do
        Find a subgraph in $K$ matching the left-hand side of rules R1-R3
        Replace the subgraph with the right-hand side of the rule
    end while
    return $K$
end procedure
```

This algorithm is sound and complete for all distributions P that have a perfect map.

3 Undirected Graphical Models

There are some probabilistic distributions that cannot be captured with Bayesian Networks. Consider the following. There are four students Alice, Betty, Carl, and Debbie, who are collaborating in homework with each other. However, Alice refuses to work with Carl, and Betty will not work with Debbie. To find the probability that all four students receive full credit on their homework, the distribution $P(A, B, C, D)$ satisfies only the independencies $(A \perp C|B, D)$ and $(B \perp D|A, C)$. Any Bayesian network I-map would have extraneous edges, and so would not capture one of the independencies.

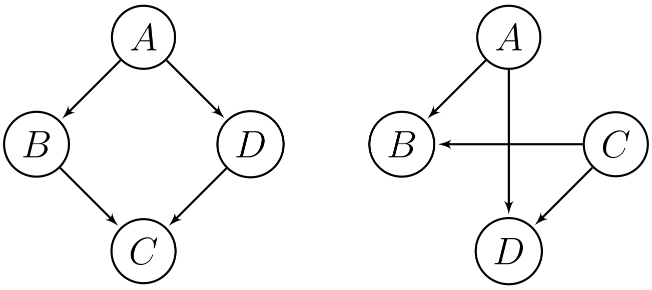


Figure 4: Attempt at a Bayesian Network

The first graph includes the extra independency $(B \perp D|A)$. The second graph also includes an extra independency $(A \perp C)$.

A more natural representation is to have undirected edges. As you can see below, an undirected graph perfectly captures the set of independencies. This graphical model is called a *Markov network*, a network in which nodes are variables, and edges are "probabilistic relationships".

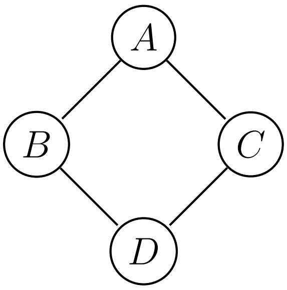


Figure 5: Markov Network

In a Bayesian network, we used CPDs to weight edges. However, in a Markov network, there is no direction. We instead use *factors*; a factor ϕ is a function from a set of random variables $D = \{X_1 ... X_k\}$ to $\mathbb{R}$. D is known as the *scope* and is also denoted $Scope[\phi]$.

3.1 Parameterization

It might be tempting to associate factors directly to edges in a graph. However, this limits the expressive power of the model. Consider a fully connected Markov network over P. If all variables are binary, each factor over each edge has 4 parameters, so the total number of parameters would be $4\binom{n}{2}$. However, P has no independencies and therefore specifies a fully joint distribution, requiring $2^n - 1$ parameters. Associating factors directly to edges only encodes pairwise relationships. A more general representation is to allow factors over arbitrary number of variables.

We may want to specify the fully joint distribution of a Markov network. A *Gibbs distribution* parameterized by a set of factors $\Phi = \{\phi_1(D_1), ..., \phi_k(D_k)\}$ if

$$P_\Phi(X_1...X_n) = \frac{1}{Z}\widetilde{P}_\Phi(X_1...X_n)$$

where

$$\begin{aligned} \widetilde{P}_\Phi &= \phi_1(D_1) * ... * \phi_k(D_k) && \textit{(unnormalized measure)} \\ Z &= \sum_{X_1...X_n} \widetilde{P}_\Phi(X_1...X_n) && \textit{(partition function)} \end{aligned}$$

Each D_k is called a clique potential.

We may want to condition our Markov network to a certain context U. Say that we have some new information $U \subset Y$ and factor $\phi(Y)$. A *factor reduction* of ϕ to the context $U = u$ is a factor over scope $Y' = Y - U$ such that

$$\phi[u](y') = \phi(y', u)$$

This simply removes all entries within the original factor $\phi(y)$ that is inconsistent with u. A *reduced Gibbs distribution* $P_\Phi[u]$ to the context u is a Gibbs distribution defined by the set of factors $\Phi[u] = \{\phi_1[u], ..., \phi_k[u]\}$. Note that $P_\Phi[u] = P_\Phi(W|u)$ where $W = X - U$.

Let H be a Markov network over X and context $U = u$. A *reduced Markov network* $H[u]$ is a Markov network over the nodes $W = X - U$, where we have an edge $X - Y$ if there is an edge $X - Y$ in H. Note that this is the same operation as reducing the Gibbs distribution but in graphical form.

3.2 Markov Network Independencies

Like the Bayesian network, we must ensure that Markov networks can encode a set of independencies. Intuitively, separation between X, Y is when influence cannot "flow" between one node and the other. An *active path* is a path $X_1 - ... - X_n$ where there is no observed variable Z that blocks it. *Separation* occurs when there is no active path between two subsets of random variables X, Y. We define *globalindependencies* encoded by Markov network H as

$$I(H) = \{(X \perp Y|Z) : sep_H(X; Y|Z)\}$$

Separation is monotonic as Z increases. That is, observing more variables cannot induce an active path.

With these definitions, we can prove that a Gibbs distribution P over graph H follows the independencies in I-map H, $I(H)$. Proving the sufficiency of this equivalence is left for exercise. The completeness only holds if P is a positive distribution and is known as the Hammersly-Clifford theorem.

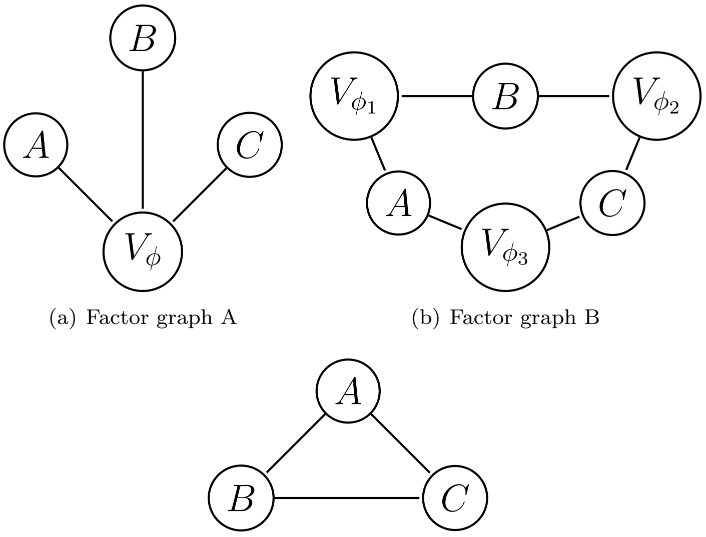
There are three types of Markov independencies.

- Global: $I(H) = \{(X \perp Y|Z) : sep_H(X;Y|Z)\}$
- Pairwise: $I_p(H) = \{(X \perp Y|\mathbb{X} - X, Y) : X - Y \notin H\}$
- Local (Markov blanket): $I_l(H) = \{(X \perp \mathbb{X} - X - MB_H(X))|MB_H(X)) : X \in \mathbb{X}\}$

The set of global independencies will always be larger than the set of local independencies, and the set of local independencies will be larger than pairwise independencies, $I(H) \supseteq I_l(H) \supseteq I_p(H)$. For positive distributions, all of them are equal, $I(H) = I_l(H) = I_p(H)$.

3.3 Parameterization Revisited

A Markov network does not generally reveal all the stucture in a Gibbs parameterization. In a complete subgraph, we cannot tell whether the factors are pairwise or over different subsets of the clique.



(a) Factor graph A

(b) Factor graph B

(c) Markov network

Figure 6: Different Factor graphs for the same Markov Network

A *factor graph* F is an undirected graph with variable nodes and factor nodes V_ϕ. A distribution factorizes over F if it can be represented as the set of factors represented by V_ϕ.

3.3.1 Log-linear models

Sometimes patterns that arise between different variables can be easier seen in log-space. An *energy function* is a factor in log-space

$$\epsilon(D) = -ln(\phi(D))$$

The energy function is a *feature*, a function from $Val(D)$ to $\mathbb{R}$, where D is the scope. A feature is analogous to a factor without nonnegativitivity constraints. We can rewrite the Gibbs distribution in feature form, giving us the log-linear model.

$$P(X_1...X_n) = \frac{1}{Z} exp[-\sum_{i=1}^{k} w_i \epsilon_i(D_i)]$$

There are three types of representation of Markov networks.

1. product over clique potentials (Markov factorization)

2. product over factors (factor graph)

3. product over feature weights (log-linear model)

Each factorization is more fine-grained than the previous; that is, a factor graph can model all Markov factorizations, and a log-linear model can model all factor graphs. All representations are useful. Markov models are useful for analyzing independence assertions. Factor graphs are useful for inference, and features are useful for parameterization in learning as they provide the biggest capacity.

An application of log-linear models are the Boltzman distribution. The Boltzman distribution is driven by the idea that a neuron is activated by the strength of the signal coming from neighboring neurons. The probability of X_i is

$$P(X_i) = sigmoid(-\sum_j (w_{i,j} x_j) - w_i)$$

where w_i is the weight of the connection between X_i and its neighbor X_j, x_j is the activation of neighbor X_j, and w_i is some bias on neuron X_i. This the most popular mathematical approximation of the function employed by a neuron in the brain. Thus, if we imagine a process by which the network continuously adapts its assignment by resampling the value of each variable as a stochastic function of its neighbors, then the "activation" probability of each variable resembles a neuron's activity. This model is a very simple variant of a stochastic, recurrent neural network.

3.3.2 Eliminating Ambiguity

Unfortunately there are infinitely ways of parameterizing a log-linear model. Consider a distribution $P(A, B, C)$ such that there are two energy functions $\epsilon_1(A, B)$ and $\epsilon_2(B, C)$. For any constant λ, we can redefine

$$\epsilon'_1(a, b^i) := \epsilon_1(a, b^i) + \lambda$$
$$\epsilon'_2(b^i, c) := \epsilon_2(b^2, c) + \lambda$$

and these new energy functions would be valid parameterizations of P.

The *canonical parameterization* of a Gibbs distribution resolves this ambiguity. Let x_Z be the assignment in x to Z, and ξ_{-Z} be the assignment of all other variables outside Z. Then, the *canonical energy function* is

$$\epsilon^*_D(d) = \sum_{Z \subseteq D} (-1)^{|D-Z|} l(d_Z, \xi^*_{-Z})$$

and the *canonical parameterization* of the Gibbs distribution is

$$P(\xi) = exp[\sum_i \epsilon^*_{D_i}(\xi(D_i))]$$

3.4 From Bayesian networks to Markov networks

The factorization of Bayesian networks can be reduced to a Gibbs distribution.

$$\begin{aligned} P(X_1...X_n) &= \prod_i P(X_i|Pa_{X_i}) \\ &= \prod_i \phi_{X_i}(X_i, Pa_{X_i}) \end{aligned}$$

A Bayesian network conditioned on evidence $E = e$ induces a Gibbs distribution reduced to the context $E = e$.

To create a Markov graph structure from a Bayesian network G, we can moralize the skeleton of G. A *moralized graph* $M(G)$ of Bayesian network G is an undirected graph that contains an edge $X - Y$ if: 1. X and Y share an edge in G or 2. X and Y are parents of the same node. The term "moralize" comes from the fact that two parents within an "immorality" will be "married". The moralized graph is guaranteed to be a minimal I-map for G. If G contains no immoralities then $M(G)$ is a perfect map for G.

3.5 Partially Directed Models

In this section, we will unify directed and undirected graphical models. This way, we are able to express both directed and undirected independencies.

A *conditional random field* is an undirected graph H whose nodes correspond to $X \cup Y$. The network encodes a conditional distribution

$$\begin{aligned} P(Y|X) &= \frac{1}{Z(X)}\widetilde{P}(Y,X) \\ \widetilde{P}(Y,X) &= \prod_{i=1}^{m} \phi_i(D_i) \\ Z(X) &= \sum_Y \widetilde{P}(Y,X) \end{aligned}$$

such that each $D_i \not\supseteq X$.

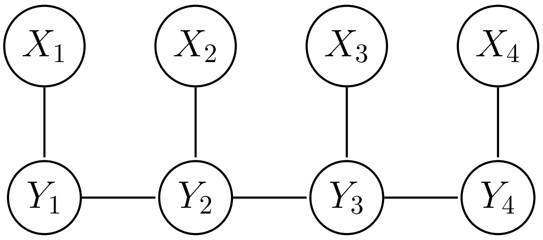


Figure 7: Conditional Random Field

We avoid encoding the distribution over variables in X. This allows the model a rich set of observed variables X. Defining a conditional distribution lets the observed variables be more expressive!

A *chain graph* is a partially acyclic directed graph such that each chain component is a conditional random field on its parent's chain component. The moralized graph corresponds to fully connecting parent components together. Let $P(K_i|Pa_{K_i})$ be a CRF over the parents and K_i, then we can factorize the distribution P over the chain graph

$$P(X_1...X_n) = \prod_{i=1}^{l} P(K_i|Pa_{K_i})$$

We can define all the different independencies in PDAG K.

- Pairwise: $I_p(K) = \{(X \perp Y|(Nondescendants_X - X - Y) : X, Y$ is nonadjacent, $Y \in Nondescendants_X\}$
- Local: $I_l(K) = \{(X \perp Nondesc_X - Boundary_X|Boundary_X)\}$
- Global: $(X \perp Y|Z)$ if X is separated from Y in $M[K[X \cup Y \cup Z]]$ given Z

All positive distributions P factorizes over PDAG K iff $P \models I(K)$.