Representing Complex Distributions, contd.

Daphne Koller
Stanford University
January 11, 2000

1 Markov Networks

Bayesian networks (directed graphical models) are the most common way now used to represent distributions. However, there is also a notion of undirected graphical model, which is quite useful, both on its own, and as an alternative way of understanding inference in directed graphical models.

1.1 Independencies in Markov networks

Definition 1.1: A Markov network structure $H$ is an undirected graph whose nodes represent random variables $X_1, \ldots, X_n$. Let $X$, $Y$ and $Z$ be disjoint subsets of nodes in the graph. We say that $Z$ separates $X$ and $Y$ in $H$, denoted $sep_H(X; Y \mid Z)$, if every path between a node in $X$ and a node in $Y$ must pass through a node in $Z$.

Intuitively, the Markov network $H$ encodes the following set of independence assumptions: For each $X$, $Y$, and $Z$ such that $sep_H(X; Y \mid Z)$ holds, we have the assumption that $I(X; Y \mid Z)$. More formally,

Definition 1.2: We say that a Markov network $H$ is an $I$-map (independency mapping) for a distribution $P$ if whenever $sep_H(X; Y \mid Z)$ holds, we have that $P \models I(X; Y \mid Z)$.

As usual, the complete graph implies no independence assumptions, and is therefore an $I$-map for any distribution. We can therefore define, as for Bayesian networks, the notion of a minimal $I$-map:

Definition 1.3: We say that a Markov network $H$ is a minimal $I$-map for a distribution $P$ if it is an $I$-map, and if the removal of any edge from $H$ renders is not an $I$-map.

How can we construct a minimal $I$-map for a distribution $P$? Some edges clearly have to be in $H$. If we do not connect a pair of nodes $A$ and $B$ via a direct edge, then we will be able to separate them somehow, e.g., by instantiating every other node in the graph. Hence, if $P \not\models I(A; B \mid X - \{A, B\})$ (where $X$ is the set of nodes in $H$), then we must connect $A$ and $B$ using a direct edge. This process adds edges it would clearly be bad to omit; any $I$-map must contain at least these edges. Surprisingly, in most cases, this is enough. If we put only these edges in the graph, we get an $I$-map.
Theorem 1.4: Let $P$ be a positive distribution over $X$, i.e., one where for all $x \in \text{Val}(X)$, $P(x) > 0$. Consider a Markov network $H$ whose vertices are $X$ and whose edges $E$ are as follows:

$$(A, B) \in E \iff P \nmid I(A; B \mid X - \{A, B\})$$

Then $H$ is the unique minimal I-map for $P$.

If we show that this graph is, in fact, an I-map, the minimality is ensured: we have already shown that any I-map must contain these edges. Similarly, we are also guaranteed that this minimal I-map is unique.

The positivity assumption is critical for this result. Consider, for example, a distribution over the four binary variables $A, B, C, D$, which has $P(a^1, b^1, c^1, d^1) = P(a^0, b^0, c^0, d^0) = 1/2$, and $P$ is 0 everywhere else. In this case, this algorithm returns the empty network: for example, when selecting edges for $A$, it is the case that $P \models I(A; B \mid C, D)$ and similarly for the other two nodes. Clearly, the resulting network is not an I-map for $P$.

Definition 1.5: Let $P$ be a positive distribution. The set of nodes that are the immediate neighbors of a node $X_i$ in the minimal I-map $H$ for $P$ is called the Markov blanket of $X_i$.

It is clear that a node is independent of the rest of the network given its Markov blanket.

What independencies does a Markov network capture exactly. If $H$ is a minimal I-map for $P$, does it necessarily reflect its independencies exactly? More formally:

Definition 1.6: We say that $H$ is a perfect map for $P$ if for any $X, Y, Z$, we have that

$$\text{sep}_H(X; Y \mid Z) \iff P \models I(X; Y \mid Z)$$

Does every distribution have a perfect map? Clearly, the answer is no. Consider a distribution arising from a three-node Bayesian network with a v-structure, e.g., the distribution induced in the Alarm example over the nodes Burglary, Earthquake, and Alarm. In the Markov network for this distribution, we must clearly have an edge between $B$ and $A$ and between $E$ and $A$. Can we omit the edge between $B$ and $E$? No, because we do not have that $I(B; E \mid A)$ holds for the distribution; rather, we have the opposite: $B$ and $E$ are dependent given $A$. Therefore, the only minimal I-map for this $P$ is the fully connected graph, which does not capture the marginal independence $I(B; E)$ that holds in $P$.

This example illustrates what happens when we construct the minimal Markov network I-map for a Bayesian network distribution. Consider any pair of nodes $X$ and $Y$ that are parents of the same node $Z$. When we condition on $Z$, $X$ and $Y$ become correlated; hence, it is not the case that $I(X; Y \mid X - \{X, Y\})$. Thus, in the minimal Markov network I-map, there must be an edge between $X$ and $Y$.

Definition 1.7: The moral graph $H[G]$ of a Bayesian network structure $G$ is the undirected graph over the same set of nodes that contains an undirected edge between $X$ and $Y$ if: (a) there is a directed edge between them (in either direction), or (b) $X$ and $Y$ are both parents of the same node.\footnote{The name moral graph originated because of the supposed “morality” of marrying the parents of a node.}
Proposition 1.8: The moral graph $\overline{H}[G]$ of a Bayesian network structure $G$ is the minimal Markov network that satisfies all of the independence assumptions that are induced by $G$.

It is also easy to see that the Markov blanket of a node in the Bayesian network (as defined before) is the same as its Markov blanket in the induced moral graph.

1.2 Factorization for Markov networks

Like in a Bayesian network, the independence assumptions induced by a Markov network allow us to factorize the distribution into smaller pieces. In this case, the factorization is somewhat different.

We define a complete subgraph of $H$ to be a subset of the vertices of $H$ where each vertex is connected by an edge to all other vertices in the subgraph. Naturally, a complete subgraph has no independences within the subgraph. We define a clique to be a maximal complete subgraph, i.e., one to which no additional vertices can be added. Any graph can be decomposed into cliques (which are not necessarily disjoint). Figure 1 contains examples of decompositions of graphs into their cliques.

The decomposition of a Markov network into cliques induces a similar decomposition on the distribution.

Definition 1.9: Let $\mathbf{Y}$ be a set of random variables. We define a factor to be a function from $\text{Val}(\mathbf{Y})$ to $\mathbb{R}$. ■

Definition 1.10: Let $H$ be a Markov network, and let $\mathbf{C}_1, \ldots, \mathbf{C}_k$ be the cliques in $H$. A distribution $P$ factorizes over $H$ if there exists factors $f_1, \ldots, f_k$ over $\mathbf{C}_1, \ldots, \mathbf{C}_k$ such that

$$P(X_1, \ldots, X_n) = \alpha f_1(\mathbf{C}_1) \times f_2(\mathbf{C}_2) \times \cdots \times f_k(\mathbf{C}_k)$$

where $\alpha$ is a normalizing constant. The factors $f_i$ are called clique potentials. ■

---

2Note that this is nonstandard usage of the word clique. Usually, a clique is any complete subgraph, not necessarily a maximal one. However, it is the standard interpretation of the word within the graphical models community.
We can now state an analogous theorem to the one we stated for Bayesian networks:

**Theorem 1.11:** Let $P$ be a positive distribution over $X_1, \ldots, X_n$ and $H$ a Markov network over the same variables. Then $P$ factorizes over $H$ if and only if $H$ is an I-map for $P$.

Markov networks are very useful for vision applications. There, for example, we have nodes associated with pixels in the image, and edges that connect neighboring pixels. The clique potentials are used to represent the correlations between the intensity values of neighboring pixels. In general, however, the parameterization of Markov networks is not very intuitive, as they do not correspond either to probabilities or to conditional probabilities. As a consequence, the numbers are hard for people to understand, making them hard to elicit; furthermore, it turns out that, for the same reason, these parameters are harder to estimate from data. As a consequence, directed graphical models are more commonly used in most applications. Nevertheless, undirected graphical models will turn out to be a useful internal data structure in certain inference algorithms.

## 2 Knowledge engineering

Above, we interpreted the process of constructing a BN in as the process of going from a given distribution to a BN. Real life is not like that. We have a vague model of the world, and we need to crystallize it into a BN.

### 2.1 Picking variables

We must define our variables very precisely. The *clarity test* is a good way of figuring out whether they are well-defined enough: Assume that we are a million years after the events described in the BN; can an omniscient being, one who saw everything, determine the value of the variable? For example, consider a *Weather* variable, with a value *sunny*. To be absolutely precise, we must define: where do we check the weather, at what time, what fraction of the sky must be clear in order for it to be sunny. For a variable such as *Heart-attack*, we must specify how large does the heart attack have to be, in what period of time does it have to happen, ... By contrast, a variable such as *Risk-of-heart-attack* is meaningless.

If we are not careful, we’ll have a hard time making sure that evidence observed and conclusions made are coherent.

It is also important to pick reasonable values for our variables. In particular, if our partition is not fine enough, conditional independence assumptions may be false. For example, we may want to construct a model where we have a person’s cholesterol level, and two tests that are conditionally independent given that. We define “normal” to be up to 200, and “high” to be above that. But it may be the case that both tests are more likely to fail if the person’s cholesterol is marginal (200–240). In this case, the assumption of conditional independence given the value (high/normal) of the cholesterol test is false. It is only true if we add a “marginal” value.

Generally speaking, we want our model to contain variables that we can potentially observe or that we may want to query. However, sometimes we want to put in *hidden variables* that are neither. Why would we want to do that? Let’s go back to the cholesterol test. In order for the answers to be accurate, the person has to not eat since 10pm the following evening. If the person eats (having no willpower), the results are consistently off. We don’t really care about this variable, nor can we observe it. However, without it, all of the different cholesterol tests become correlated. In order to avoid graphs where all the tests are correlated, it’s better to put in this additional
hidden variable, rendering them conditionally independent given the true cholesterol level and the person's willpower.

On the other hand, it's not necessary to add every variable that might be relevant. For example, in our Alarm example, perhaps our neighbor is more likely to call when an ambulance with a siren passes by. Is this fact important to represent? The probabilities already account for the fact that he might be unreliable. It is not worthwhile including this variable if I cannot observe it or something related to it.

2.2 Picking structure

Causal structures tend to work well. Either because of some real locality of influence in the world, or because of the way people perceive the world, causal connection graphs tend to be sparser. Note: causal is in the world, not in our inference process. For example, in an automobile insurance network, it is tempting to put Previous-accident as a parent of Good-driver, because that is how the insurance company thinks about the problem. This is not the causal order in the world, because being a bad driver causes previous (and future) accidents. In principle, there is nothing to prevent us from directing the edges in this way. However, as we saw in the minimal I-maps for the Alarm example, a non-causal ordering often requires that we introduce a lot of additional edges to account for induced dependencies.

Another issue is that, in real life, we often have to make compromises. Models with all of the variables and edges may be too complicated to reason with. For example, is MSNBC's web-based system for pregnancy and child care, they originally began with a model that included a node for every possible disease. The network turned out to be too complex to run in real-time on a shared web-server. Therefore, they compromised by aggregating all of the disease nodes into a single Primary-disease node. The first one is much more realistic, but the second is not too bad an approximation in practice, and is much faster computationally.

In general, our model is always an approximation. Everything is dependent on everything else (butterfly in China). For example, in our alarm example, we assumed that the neighbor calling was unrelated to the earthquake. This assumption might not be true: if there is an earthquake, the neighbor might be less likely to call. In general, we can put a whole bunch of weak but existing influences into the model, but we end up with a useless one. So we make reasonable approximations.

2.3 Picking numbers

Numbers are very hard to elicit from people. They really dislike committing to an exact estimate. There have been several approaches developed for doing this process, e.g., something called a probability wheel, asking people how they would compare the probability in question to certain predefined lotteries or to other things, etc. Nevertheless, it is a very long, drawn out process.

The good news: empirical evidence shows that the answers given by a BN are not all that sensitive to many aspects of the probabilities. In general, 0.53 versus 0.54 doesn't make much difference. What does matter?

- zeros: you simply can't condition away a zero probability event no matter how hard you try.

In the first lecture, I mentioned the Pathfinder system, the first real BN-based expert system. In one of the first versions of the system (one based on naive Bayes), 10% of the cases were misdiagnosed because the expert had given a probability of zero to an unlikely but possible event. Realistically, very few things (except definitions) have probability zero, and we must be careful in assigning zeros.
• relative values: the qualitative behavior of the system will be affected by the relative sizes of
$P(x \mid y)$ for different values $y$ of $P(x)$. For example, it’s important that my neighbor is more
likely to call given an alarm than not given an alarm. It’s also important that the alarm is
more likely to be set off by a burglary than by an earthquake.

• orders of magnitude: an extreme case of relative sizes is when one thing is an order of
magnitude more likely than another. Differences like this can radically change the behavior
of the system.

Fortunately, BNs give you a good tool for deciding how much a given number affects the outcome. This process, called sensitivity analysis, can be applied as a byproduct of standard inference
techniques.

Finally, we note that causal models make it easier to elicit CPDs, since they tend represent very
natural quantities. For example, the probability that the alarm will go off in case of a burglary. The
probabilities of burglary, earthquake can be obtained from the neighborhood police station and the
USGS. In general, the causal direction allows us to isolate the sensor, using its specs to determine
the CPTs. In fact, one of the standard ways of doing knowledge engineering is to build the network
structure in a way that makes it easy to elicit probabilities. This process is called “extending the
classroom”. You start out with a variable such as “lung tumor”. You ask the expert if he can
give beliefs over that. He says “it depends”, you ask on what. The appropriate parent variables,
e.g., lung cancer are added. This continues until root nodes are reached, i.e., until expert can give
you probabilities. E.g., add smoking and exposure to toxis, then add age and gender.