Conditional Decompositions

Michael Betancourt

August 2026

All probability theory chapters can be found here.

A repository containing all of the files used to generate this chapter is available on GitHub.

Table of contents

On product spaces, conditional probability theory becomes particularly useful when we apply it more than once. After disintegrating a joint probability distribution into a marginal distribution and conditional probability kernel, we can often disintegrate that marginal distribution into even simpler pieces.

In this chapter, we'll explore these recursive decompositions and their organization into *conditional decompositions*.

1 Recursive Decomposition

Recall that any subsequence of component indices,

$$\mathsf{i} \vec{\subset} 1 : I,$$

defines a projection function ϖ_{i} that disintegrates a joint probability distribution

$$\pi(\mathsf{x}_{1:I})$$

into a marginal distribution over the thinned product space X^{i},

$$\pi(\mathsf{x}_{\mathsf{i}}),$$

and a conditional probability kernel over the cross sections $X^{\mathsf{i}^c} \times \{x_{\mathsf{i}}\}$,

$$\pi(\mathsf{x}_{\mathsf{i}^c} \mid x_{\mathsf{i}}).$$

The marginal and conditional probability distributions will always be simpler than the joint distribution, but they might not be simpler enough for practical applications. Fortunately, the marginal distribution is amenable to decomposition of its own.

A smaller subsequence of component indices,

$$\mathsf{i}' \vec{\subset} \mathsf{i} \vec{\subset} 1 : I,$$

defines a new projection function,

$$\varpi_{\mathsf{i}'} : X^{\mathsf{i}} \to X^{\mathsf{i}'},$$

that disintegrates $\pi(\mathsf{x}_{\mathsf{i}})$ into a new marginal distribution

$$\pi(\mathsf{x}_{\mathsf{i}'})$$

over the thinner product space

$$X^{\mathsf{i}'}$$

and a new conditional probability kernel

$$\pi(\mathsf{x}_{\mathsf{i}\,\mathsf{i}'} \mid x'_{\mathsf{i}})$$

over the cross sections

$$X^{\mathsf{i}\,\mathsf{i}'} \times \{x'_{\mathsf{i}}).$$

Given this two-step decomposition, we can compute joint expectation values by nesting the two conditional expectations and one marginal expectation,

$$\begin{aligned}
\mathbb{E}_{\pi}[g] &= \int \pi(\mathrm{d}x_{1:I})\, g(x_{\mathsf{i}^{\mathsf{c}}}, x_{\mathsf{i}\,\mathsf{i}'}, x_{\mathsf{i}'}) \\
&= \int \pi(\mathrm{d}x_{\mathsf{i}'}) \int \pi(\mathrm{d}x_{\mathsf{i}\,\mathsf{i}'} \mid x_{\mathsf{i}'}) \int \pi(\mathrm{d}x_{\mathsf{i}^{\mathsf{c}}} \mid x_{\mathsf{i}\,\mathsf{i}'})\, g(x_{\mathsf{i}^{\mathsf{c}}}, x_{\mathsf{i}\,\mathsf{i}'}, x_{\mathsf{i}'}).
\end{aligned}$$

Symbolically, we often write this as

$$\pi(\mathrm{d}x_{1:I}) = \pi(\mathrm{d}x_{\mathsf{i}'})\, \pi(\mathrm{d}x_{\mathsf{i}\,\mathsf{i}'} \mid x_{\mathsf{i}'})\, \pi(\mathrm{d}x_{\mathsf{i}^{\mathsf{c}}} \mid x_{\mathsf{i}\,\mathsf{i}'}).$$

When working with probability density functions, this becomes even cleaner. The joint expectation

$$\mathbb{E}_{\pi}[g] = \int \pi(\mathrm{d}x_{1:I})\, g(x_{1:I})$$

can be written as the nested integrals

$$\begin{aligned}
\mathbb{E}_{\pi}[g] &= \int \pi(\mathrm{d}x_{\mathsf{i}'}) \int \pi(\mathrm{d}x_{\mathsf{i}\,\mathsf{i}'} \mid x_{\mathsf{i}'}) \int \pi(\mathrm{d}x_{\mathsf{i}^{\mathsf{c}}} \mid x_{\mathsf{i}\,\mathsf{i}'})\, g(x_{\mathsf{i}^{\mathsf{c}}}, x_{\mathsf{i}\,\mathsf{i}'}, x_{\mathsf{i}'}) \\
&= \int \nu^{\mathsf{i}'}(\mathrm{d}x_{\mathsf{i}'})\, p(x_{\mathsf{i}'}) \\
&\qquad \int \nu^{\mathsf{i}\,\mathsf{i}'}(\mathrm{d}x_{\mathsf{i}\,\mathsf{i}'})\, p(x_{\mathsf{i}\,\mathsf{i}'} \mid x_{\mathsf{i}'}) \\
&\qquad\qquad \int \nu^{\mathsf{i}^{\mathsf{c}}}(\mathrm{d}x_{\mathsf{i}^{\mathsf{c}}})\, p(x_{\mathsf{i}^{\mathsf{c}}} \mid x_{\mathsf{i}\,\mathsf{i}'})\, g(x_{\mathsf{i}^{\mathsf{c}}}, x_{\mathsf{i}\,\mathsf{i}'}, x_{\mathsf{i}'}) \\
&= \int \nu^{\mathsf{i}'}(\mathrm{d}x_{\mathsf{i}'}) \int \nu^{\mathsf{i}\,\mathsf{i}'}(\mathrm{d}x_{\mathsf{i}\,\mathsf{i}'}) \int \nu^{\mathsf{i}^{\mathsf{c}}}(\mathrm{d}x_{\mathsf{i}^{\mathsf{c}}}) \\
&\qquad \Big[p(x_{\mathsf{i}'})\, p(x_{\mathsf{i}\,\mathsf{i}'} \mid x_{\mathsf{i}'})\, p(x_{\mathsf{i}^{\mathsf{c}}} \mid x_{\mathsf{i}\,\mathsf{i}'}) \Big]\, g(x_{\mathsf{i}^{\mathsf{c}}}, x_{\mathsf{i}\,\mathsf{i}'}, x_{\mathsf{i}'}).
\end{aligned}$$

This implies that the joint probability density function decomposes as

$$\begin{aligned}
p(x_{1:I}) &= p(x_{\mathsf{i}^{\mathsf{c}}}, x_{\mathsf{i}\,\mathsf{i}'}, x_{\mathsf{i}'}) \\
&\overset{\nu^{1:I}}{=} p(x_{\mathsf{i}'})\, p(x_{\mathsf{i}\,\mathsf{i}'} \mid x_{\mathsf{i}'})\, p(x_{\mathsf{i}^{\mathsf{c}}} \mid x_{\mathsf{i}\,\mathsf{i}'}).
\end{aligned}$$

If i' contains more than one component index, then we can iterate this process, at each step disintegrating the new marginal distribution until we are satisfied or left with a single component space.

2 Conditional Decompositions

One of more difficult aspects of trying to implement repeated, recursive disintegrations in practice is keeping track of all of the intermediate spaces. Fortunately, we can we systematize this process by organizing each step into a subset of active component indices, similar to how we organized recursive decompositions of product spaces in Chapter 3, Section 4.3.2.

2.1 General Conditional Decompositions

To set up the decomposition we begin with a partition of the component indices $1 : I$ into disjoint subsequences

$$(\mathsf{i}_1, \ldots, \mathsf{i}_K)$$

with

$$\mathsf{i}_k \vec{\subset} 1 : I.$$

Next, we iteratively remove the component indices in each i_k to define a sequence of increasingly thin product spaces, X^{j_k} with

$$\mathsf{j}_k = \vec{\cup}_{k'=k}^{K} \mathsf{i}_{k'}.$$

The first element of the partition defines a projection function

$$\varpi_{\mathsf{i}_1} : X^{\mathsf{j}_1} \to X^{\mathsf{j}_2},$$

which decomposes the full product space

$$X^{\mathsf{j}_1} = X^{1:I}$$

into the thinned product space X^{j_2} and the cross sections

$$X^{\mathsf{i}_1} \times \{x_{\mathsf{j}_2}).$$

At the same time, the projection function disintegrations any joint distribution over the full product space into a marginal distribution over the thinned product space,

$$\pi(\mathsf{x}_{\mathsf{j}_2}),$$

and a conditional probability kernel over the cross sections,

$$\pi(\mathsf{x}_{\mathsf{i}_1} \mid \mathsf{j}_2).$$

From here, we iterate by following the partition of the component indices. At the kth step, the projection function

$$\varpi_{\mathsf{i}_k} : X^{\mathsf{j}_k} \to X^{\mathsf{j}_{k+1}},$$

decomposes the initial product space X^{j_k} into the product space $X^{\mathsf{j}_{k+1}}$ and the cross sections

$$X^{\mathsf{i}_k} \times \{x_{\mathsf{j}_{k+1}}),$$

while disintegrating

$$\pi(\mathsf{x}_{\mathsf{j}_k}),$$

into a new marginal distribution,

$$\pi(\mathsf{x}_{\mathsf{j}_{k+1}}),$$

and a new conditional probability kernel,

$$\pi(\mathsf{x}_{\mathsf{i}_k} \mid x_{\mathsf{j}_{k+1}}).$$

Altogether, this procedure disintegrates a joint probability distribution into a sequence of $K-1$ conditional probability kernels

$$\pi(\mathsf{x}_{\mathsf{i}_k} \mid x_{\mathsf{j}_{k+1}})$$

and one terminal marginal probability distribution,

$$\pi(\mathsf{x}_{\mathsf{j}_K}).$$

I will refer to this collection of probabilistic objects, or sometimes just the partition of component indices that implicitly defines it, as a **conditional decomposition**.

Every ordered partition of the component indices defines a unique conditional decomposition. For example, when working with a three-component product space

$$X^{1:3} = X_1 \times X_2 \times X_3,$$

the index partitions

$$\begin{aligned}
&((1,2,3)); \quad ((1,2),(3)); \quad ((3),(1,2));\\
&((1),(2,3)); \quad ((2,3),(1)); \quad ((1,3),(2));\\
&((2),(1,3)); \quad ((1),(2),(3)); \quad ((1),(3),(2));\\
&((2),(1),(3)); \quad ((2),(3),(1)); \quad ((3),(1),(2));\\
&((3),(2),(1))
\end{aligned}$$

all define a unique conditional decomposition. In particular, the partition $((1),(2,3))$ defines the conditional decomposition

$$\pi(\mathrm{d}x_1, \mathrm{d}x_2, \mathrm{d}x_3) = \pi(\mathrm{d}x_1 \mid \mathrm{d}x_2, \mathrm{d}x_3)\, \pi(\mathrm{d}x_2, \mathrm{d}x_3).$$

Equivalently, this partition defines a decomposition of the joint probability density function into conditional and marginal probability density functions,

$$p(x_1, x_2, x_3) = p(x_1 \mid x_2, x_3)\, p(x_2, x_3).$$

Each of these conditional decompositions can be more or less useful in different applications. Some, for instance, might be more interpretable in the context of a given application, while others might facilitate the evaluation of certain probabilistic operations. We will return to the problem of identifying productive decompositions in probabilistic modeling applications in future chapters.

That said, the number of possible conditional decompositions is so large that trying to work through them all can be overwhelming. Given I component spaces, we can construct $H_d(I)$ distinct conditional decompositions, where $H_d(n)$ is the nth ordered Bell number (Knopfmacher and Mayes 2005). The ordered Bell numbers grow wildly fast as n increases (Figure 1).

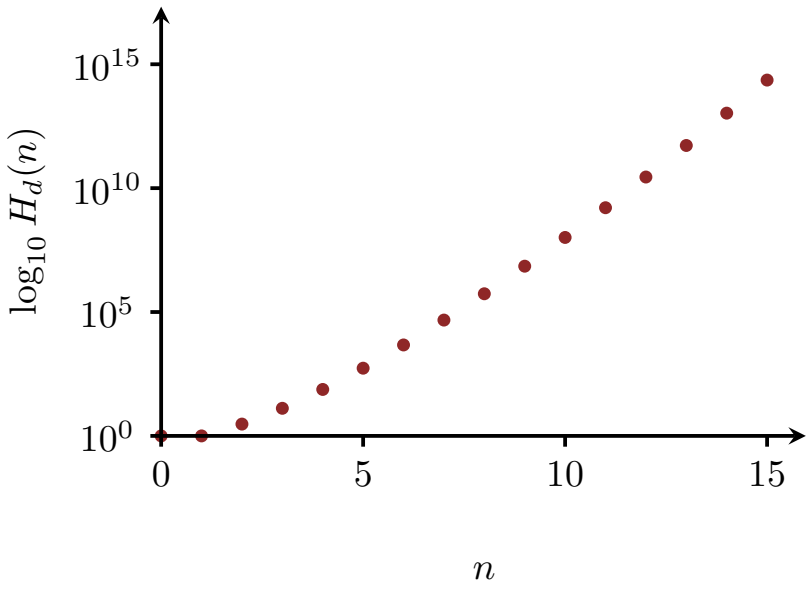


Figure 1: The nth ordered Bell number counts the number of ordered partitions of a finite set with n elements. As the size of a set increases, the number of ordered partitions increases extraordinarily quickly. Even on a log scale the growth appears to be exponential!

2.2 Atomic Conditional Decompositions

That said, most practical applications consider only **atomic conditional decompositions** that decompose joint probability distributions as completely as possible.

An atomic conditional decomposition is defined by an atomic partition, where each cell consists of only a single component index,

$$\mathsf{i}_k = \{i_k\}.$$

This results in the sequence of conditional probability kernels

$$\pi(\mathsf{x}_{\{\mathsf{i}_k\}} \mid x_{\mathsf{j}_{k+1}})$$

and the marginal probability distribution,

$$\pi(\mathsf{x}_{\mathsf{j}_I}).$$

These conditional decomposition "peel" off one component space at a time in the prescribed order.

Given I component spaces, there are only $I!$ of these atomic conditional decompositions, each corresponding to a particular ordering of the components. The number of atomic conditional decompositions grows quickly with as the number of component spaces, but far less so than the number of general conditional decompositions does.

For instance, when working with the product space

$$X^{1:3} = X_1 \times X_2 \times X_3,$$

we have only six unique atomic conditional decompositions corresponding to the partitions

$$\begin{array}{lll} (\ (1), (2), (3)\ ); & (\ (1), (3), (2)\ ); & (\ (2), (1), (3)\ ); \\ (\ (2), (3), (1)\ ); & (\ (3), (1), (2)\ ); & (\ (3), (2), (1)\ ). \end{array}$$

To exhaustively demonstrate, the partition $(\ (2), (3), (1)\ )$ defines the atomic conditional decomposition

$$\pi(\mathrm{d}x_1, \mathrm{d}x_2, \mathrm{d}x_3) = \pi(\mathrm{d}x_2 \mid \mathrm{d}x_1, \mathrm{d}x_3)\, \pi(\mathrm{d}x_3 \mid \mathrm{d}x_1)\, \pi(\mathrm{d}x_1).$$

This is equivalent to the decomposition of the joint probability density function into conditional and marginal probability density functions,

$$p(x_1, x_2, x_3) = p(x_2 \mid x_1, x_3)\, p(x_3 \mid x_1)\, p(x_1).$$

3 Sequential Construction of Joint Probability Distributions

While the names suggests destruction, conditional decompositions can also be used to *construct* joint probability distributions from simpler probabilistic pieces.

Given a partition of the component indices, we can construct a joint distribution by working through the subsequences of component indices *in reverse*. After engineering an appropriate marginal distribution

$$\pi(\mathsf{x}_{\mathsf{j}_K})$$

over the thinned product space X^{j_K}, we engineer conditional probability kernels,

$$\pi(\mathsf{x}_{\mathsf{i}_k} \mid x_{\mathsf{j}_{k+1}})$$

over the cross sections

$$X^{\mathsf{i}_k} \times \{x_{\mathsf{j}_{k+1}}\}.$$

The ultimate conditional probability kernel lifts $\pi(\mathsf{x}_{\mathsf{j}_K})$ into a joint distribution over $X^{\mathsf{j}_{K-1}}$, before the penultimate conditional probability kernel lifts that joint distribution into a joint distribution over $X^{\mathsf{j}_{K-2}}$, and so on. Altogether, the conditional probability kernels lift the marginal distribution into a joint distribution over the full product space.

When the i_k are small, the intermediate cross sections will be simple compared to the full product space. In this case engineering probabilistic objects over the cross sections will typically be more manageable than trying to work with the entire product space at once. For example, when building off of an atomic decomposition, we will only ever have to work with a single active component space at a time, starting with

$$\pi(\mathsf{x}_{i_I}),$$

and then proceeding to each

$$\pi(\mathsf{x}_{i_k} \mid x_{i_{k+1}}, \ldots, x_{i_I}.)$$

In other words, conditional decompositions define a kind of armature around which we can design joint distributions. Once the structure of armature has been established with the choice of a conditional decomposition, we can focus on the details, namely the choice of the marginal distribution and each conditional probability kernel.

4 Structured Conditional Decompositions

In many applications, the conditional probability kernels in a conditional decomposition will behave simpler than what is strictly possible. These simplifications, in turn, encode meaningful structure of the corresponding joint distributions. To take advantage of this structure in practice, however, we will need to establish an appropriate vocabulary.

5 Conditional Dependencies

At any step of a conditional decomposition, the behavior of the conditional probability distributions will, in general, depend on the behavior of *all* of the remaining component spaces,

$$\pi(\mathsf{x}_{\mathsf{i}_k} \mid x_{\mathsf{j}_{k+1}})$$

where

$$\mathsf{j}_{k+1} = \vec{\cup}_{k'=k+1}^{K} \mathsf{i}_{k'}.$$

In this case, we say that the conditional probability kernel is **conditionally dependent** on the behavior in each thinned product space

$$X^{\mathsf{i}_{k+1}}, X^{\mathsf{i}_{k+2}}, \ldots, X^{\mathsf{i}_{K-1}}, X^{\mathsf{i}_K}.$$

Sometimes, however, the behavior of the conditional probability distributions at one step will be invariant to the behavior in some of these thinned product spaces. For example, if

$$\pi(\mathsf{x}_{\mathsf{i}_k} \mid x_{\mathsf{i}_{k+3} \vec{\cup} \mathsf{i}_{K-1}}),$$

then the conditional probability kernel will be conditionally dependent on only $X^{\mathsf{i}_{k+3}}$ and $X^{\mathsf{i}_{K-1}}$.

Note that we are considering dependencies on not individual component spaces but rather the particular thinned product spaces defined by the component index partition. If we want to allow for more precise conditional dependencies, then we need to use a finer partition of the component indices. In this sense, the conditional dependencies from atomic conditional decompositions are the most informative.

Conditional dependencies also neatly translate into the structure of conditional probability density functions. For example, if

$$\pi(\mathsf{x}_{\mathsf{i}_k} \mid x_{\mathsf{i}_{k+3} \vec{\cup} \mathsf{i}_{K-1}})$$

then we would also have

$$p(x_{\mathsf{i}_k} \mid x_{\mathsf{i}_{k+3} \vec{\cup} \mathsf{i}_{K-1}}).$$

5.1 Conditional Independencies

An important, but subtle, aspect of conditional dependencies is that define only *direct* or *local* dependencies. For instance, the conditional behavior of x_{i_k} may not conditionally depend on the behavior of $x_{\mathsf{i}_{k'}}$. If, however, x_{i_k} conditionally depends on $x_{\mathsf{i}_{k''}}$ and $x_{\mathsf{i}_{k''}}$ itself conditionally depends on $x_{\mathsf{i}_{k'}}$, then $x_{\mathsf{i}_{k'}}$ will *indirectly* moderate x_{i_k} through $x_{\mathsf{i}_{k''}}$.

To check for these indirect dependencies we have to push the joint probability distribution forward along the projection function

$$\varpi_{\mathsf{i}_k \vec{\subset} \mathsf{i}_{k'}} : X^{1:I} \to X^{\mathsf{i}_k \vec{\subset} \mathsf{i}_{k'}},$$

and then disintegrate that pushforward distribution with respect to the projection function

$$\varpi_{\mathsf{i}_{k'}}^{\mathsf{i}_k \vec{\subset} \mathsf{i}_{k'}} : X^{\mathsf{i}_k \vec{\subset} \mathsf{i}_{k'}} \to X^{\mathsf{i}_{k'}}.$$

If the resulting conditional probability kernel is invariant to the behavior in $X^{\mathsf{i}_{k'}}$,

$$\pi(\mathsf{x}_{\mathsf{i}_k} \mid \mathsf{i}_{k'}) = \pi(\mathsf{x}_{\mathsf{i}_k}),$$

then the conditional behavior of x_{i_k} will be completely independent of $x_{\mathsf{i}_{k'}}$, regardless of the behavior of any other component spaces. In this case, we say that x_{i_k} is **conditionally independent** of $x_{\mathsf{i}_{k'}}$.

More generally, if we apply this procedure not to joint distribution

$$\pi(\mathsf{x}_{1:I})$$

but rather to the conditional probability kernel

$$\pi(\mathsf{x}_{\mathsf{i}_{k''}^c} \mid x_{\mathsf{i}_{k''}})$$

and

$$\pi(\mathsf{x}_{\mathsf{i}_k} \mid x_{\mathsf{i}_{k'} \vec{\subset} \mathsf{i}_{k''}}) = \pi(\mathsf{x}_{\mathsf{i}_k} \mid x_{\mathsf{i}_{k''}}),$$

then we say that x_{i_k} is conditionally independent of $x_{\mathsf{i}_{k'}}$ given $x_{\mathsf{i}_{k''}}$.

While conditional dependencies are straightforward, conditional independencies are often counter-intuitive and require careful consideration. For much more on conditional independences for atomic conditional decompostions, see Bishop (2006).

5.2 Dependency Subsequences

Conditional dependencies are not unique to any one choice of conditional probability kernel. Indeed, different conditional probability kernels can share the same conditional dependencies. Fortunately, abstracting the conditional dependencies from any particular conditional probability kernel is straightforward.

At the kth step of a conditional decomposition, any valid conditional probability kernel can depend on the thinned product spaces given by the subsequences of component indices

$$\mathsf{i}_{k+1}, \mathsf{i}_{k+2}, \ldots, \mathsf{i}_{K-1}, \mathsf{i}_K.$$

We can neatly communicate the dependencies between these subsequences by collecting them into subsets,

$$\mathsf{d}_k = \{\mathsf{i}_{k'} \mid k' \in \mathsf{k}_k \subset \{k+1, \ldots, K\}\}.$$

This allows us to write the conditional probability kernel more precisely as

$$\pi(\mathsf{x}_{\mathsf{i}_k} \mid \mathsf{d}_k).$$

I will refer to these subsequences as **dependency subsequences**.

For example, the last dependency subsequence is always empty because there are no component spaces left on which behavior can depend,

$$\mathsf{d}_K = ().$$

The second-to-last dependency subsequence, however, could be one of two possibilities,

$$\mathsf{d}_{K-1} = ()$$

or

$$\mathsf{d}_{K-1} = \mathsf{i}_K,$$

while the third-to-last could be one of four possibilities,

$$\begin{aligned}
\mathsf{d}_{K-2} &= () \\
\mathsf{d}_{K-2} &= \mathsf{i}_{K-1} \\
\mathsf{d}_{K-2} &= \{\mathsf{i}_{K-1}, \mathsf{i}_K\} \\
\mathsf{d}_{K-2} &= \mathsf{i}_K.
\end{aligned}$$

Dependency subsequences allow us to incorporate even more information into a conditional decomposition without having to commit to particular conditional probability kernels and marginal distributions. Specifically, we can generalize a conditional decomposition to be a sequence of pairs,

$$((\mathsf{i}_1, \mathsf{d}_1), \ldots, (\mathsf{i}_K, \mathsf{d}_K),$$

where the i_k partition the component indices and the d_k communicate the precise conditional dependencies.

If, for instance, we have an atomic conditional decomposition and

$$\mathsf{d}_k = \emptyset$$

for all k, then every conditional probability distribution will be independent of the behavior in every subsequent component space,

$$\pi(\mathsf{x}_{i_k} \mid \mathsf{d}_k) = \pi(\mathsf{x}_{i_k}).$$

In this case, the joint distribution will be an independent product distribution,

$$\pi(\mathrm{d}x_1, \ldots, \mathrm{d}x_I) = \pi(\mathrm{d}x_1) \ \ldots \ \pi(\mathrm{d}x_i) \ \ldots \ \pi(\mathrm{d}x_I).$$

Conditional dependencies give us yet another way to characterize independent product distributions!

For a more specific example, consider the product space

$$X = X_1 \times X_2 \times X_3 \times X_4$$

with the atomic conditional decomposition

$$(\ (4), \{(1), (3)\}\ ), \quad (\ (3), \{(1), (2)\}\ ), \quad (\ (1), \{(2)\}\ ), \quad (\ (2), \emptyset\ ).$$

Any joint probability density function compatible with this conditional decomposition will be of the form

$$\begin{aligned} p(x_1, x_2, x_3, x_4) = \ & p(x_4 \mid x_1, x_3) \\ & \cdot p(x_3 \mid x_1, x_2) \\ & \cdot p(x_1 \mid x_2) \\ & \cdot p(x_2). \end{aligned}$$

With dependency subsequences, our probabilistic modeling armature becomes even more informative. We can now construct joint distributions in three stages: partitioning the component variables into subsequences, specifying the precise conditional dependencies between these subsequences, and then finally specifying particular conditional probability kernels and a particular marginal distribution.

6 Directed Graphical Models

While richly informative, component index partitions and dependency subsequences can be awkward to work with in practice. Reading off conditional dependencies from the structure of conditional probability densities is less explicit but often more intuitive to many practitioners. In this section, we will consider a *visual* representation of conditional decompositions that neatly complements these approaches.

6.1 Directed Acyclic Graphs

A **graph** is a collection of **nodes** with **edges** connecting certain pairs of nodes. Typically, nodes represent mathematical objects and edges the relationships, or lack thereof, between them.

Directed graphs distinguish between edges that originate from one node and terminate in another, and vice versa. The structure of a directed graph is particularly straightforward to visualize, with two-dimensional shapes representing each node and one-dimensional arrows the directed edges connecting them (Figure 2).

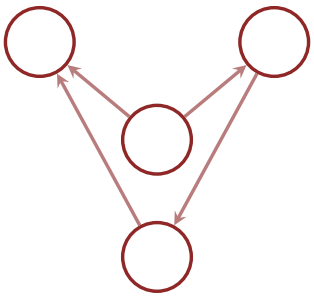


Figure 2: Directed graphs are typically visualized with shapes to represent each node and arrows to represent the directed edges between them. When the nodes of a directed graph are named, we can also label the shapes accordingly.

The nodes at which a directed edge originates is referred to as the **parent** of the node at which the edge terminates. Similarly, the terminating node is referred to as a **child** of originating node (Figure 3). This familial language motivates a variety of related vocabulary, such as referring to two nodes that share a common parent as siblings.

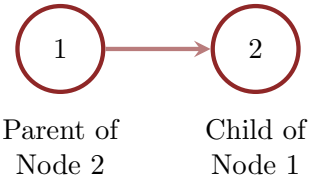
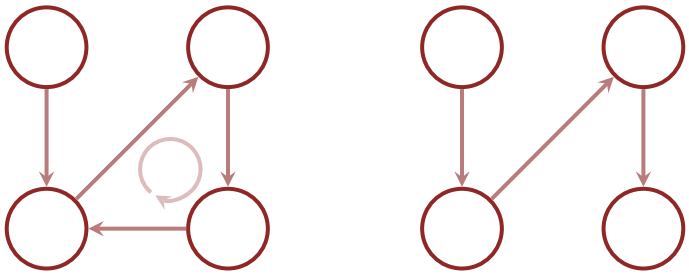


Figure 3: Two nodes connected by a directed edge are referred to as the parent and child of each other. This terminology allows us to compactly describe relationships between nodes in a directed graph.

A sequence of directed edges connecting parent nodes to child nodes defines a **path** through the graph. In general, these paths can circle back on themselves, forming **cycles** (Figure 4a). Directed graphs that do not exhibit any cycles (Figure 4b) are known as **directed acyclic graphs**, or just **DAGs** for short.

Because of the lack of cycles, directed acyclic graphs exhibit a well-defined notion of directionality (Figure 5). On one side of the graph we have **root nodes** with no parents, and on the



(a) (b)

Figure 4: (a) Cycles in a directed graph are formed by paths of directed edges that connect back on themselves. (b) In a directed acyclic graph, each path originates and terminates at a distinct node.

other we have **leaf nodes** with no children.

Although visualizations of directed graphs don't technically have to respect this directionality, they tend to be more effective when they do. That said, there is no universal convention for whether root notes should be on the top, bottom, left, or right of a visualization. We are free to pick whichever convention is most convenient, although it's best to maintain that convention consistently.

6.2 Graphical Models of Conditional Decompositions

Conveniently, every atomic conditional decomposition defines a unique directed acyclic graph, which can then be used to visualize the conditional structure. A conditional decomposition represented by a directed acyclic graph is known as a **directed graphical model**, **graphical model**, or even just **DGM**.

To see how to translate a conditional decomposition into a graphical model, let's consider the general conditional decomposition

$$((\mathsf{i}_1, \mathsf{d}_1), \quad \ldots, \quad (\mathsf{i}_K, \mathsf{d}_K)).$$

Each subsequence of active component indices,

$$\mathsf{i}_k$$

can be represented by a node labeled with the corresponding component variables. The elements of d_k then define arrows that originate at the node representing the corresponding component indices and terminating at i_k.

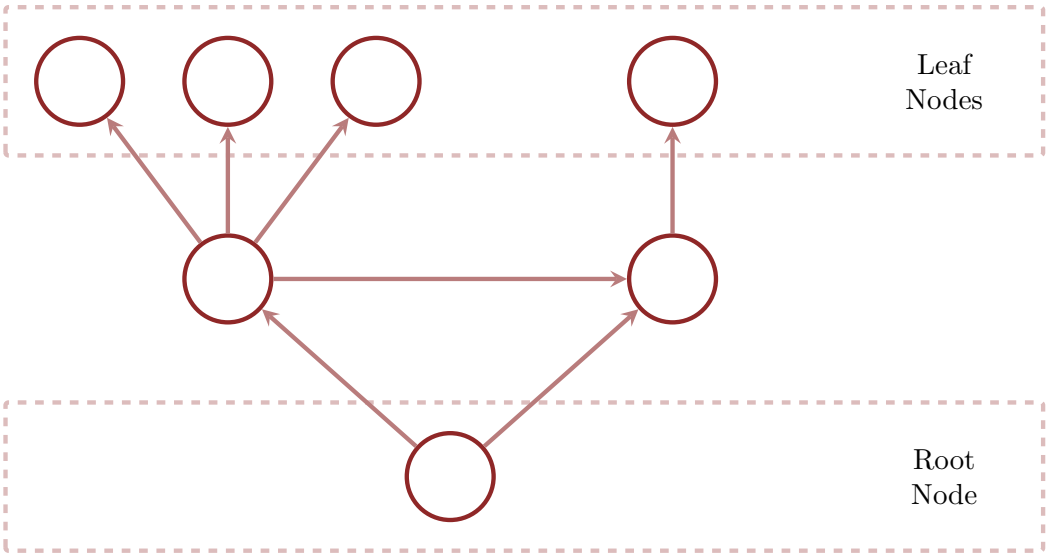


Figure 5: Directed acyclic graph can be arranged in a natural direction, starting with root nodes that have no parents and ending with leaf nodes that have no children.

To demonstrate, let's take $I = 8$ and build a graphical model for the the conditional decomposition

$$\begin{array}{lll} (& ((1,7), & \{(2),(3,4,5)\}), \\ & ((3,4,5), & \{(2) \qquad\quad \}), \\ & ((2), & \{(6,8) \qquad \}), \\ & ((6,8), & \{ \qquad\quad \}), \ ) \end{array}$$

This corresponds to the joint probability density function decomposition

$$\begin{aligned} p(x_1,&x_2,x_3,x_4,x_5,x_6,x_7,x_8) \\ &= \ p(x_1,x_7 \mid x_2,x_3,x_4,x_5) \\ &\quad \cdot p(x_3,x_4,x_5 \mid x_2) \\ &\quad \cdot p(x_2 \mid x_6,x_8) \\ &\quad \cdot p(x_6,x_8). \end{aligned}$$

The four component index subsequences

$$\begin{aligned} \mathsf{i}_1 &= (1,7) \\ \mathsf{i}_2 &= (3,4,5) \\ \mathsf{i}_3 &= (2) \\ \mathsf{i}_4 &= (6,8) \end{aligned}$$

each define a node in the corresponding graphical model (Figure 6a).

Because x_{i_1} directly depends on both x_{i_2} and x_{i_3}, we next place two direct edges between the corresponding nodes (Figure 6b). Similarly, we place a lone directed edge that terminates at x_{i_2}. (Figure 6c). Finally, we place one last directed edge from the node representing x_{i_4} to the node representing x_{i_3} (Figure 6d). Lacking any conditional dependencies, no directed edges terminate at x_{i_4}. In this case, the node representing x_{i_1} defines a lone leaf node, while the node representing x_{i_4} defines the only root node.

In order to probe more detailed conditional dependencies, we need to start with more refined partitions of the component indices. For example, breaking the subsequence $\mathsf{i}_4 = (6,8)$ up into two subsequences (6) and (8) allows us to consider whether or not x_{i_3} depends on x_6, x_8, or both.

That said, conditional decompositions cannot be refined beyond atomic conditional decompositions. The corresponding directed graphical models are a bit more straightforward, with each node representing a single component space and each directed edge connecting pairs of

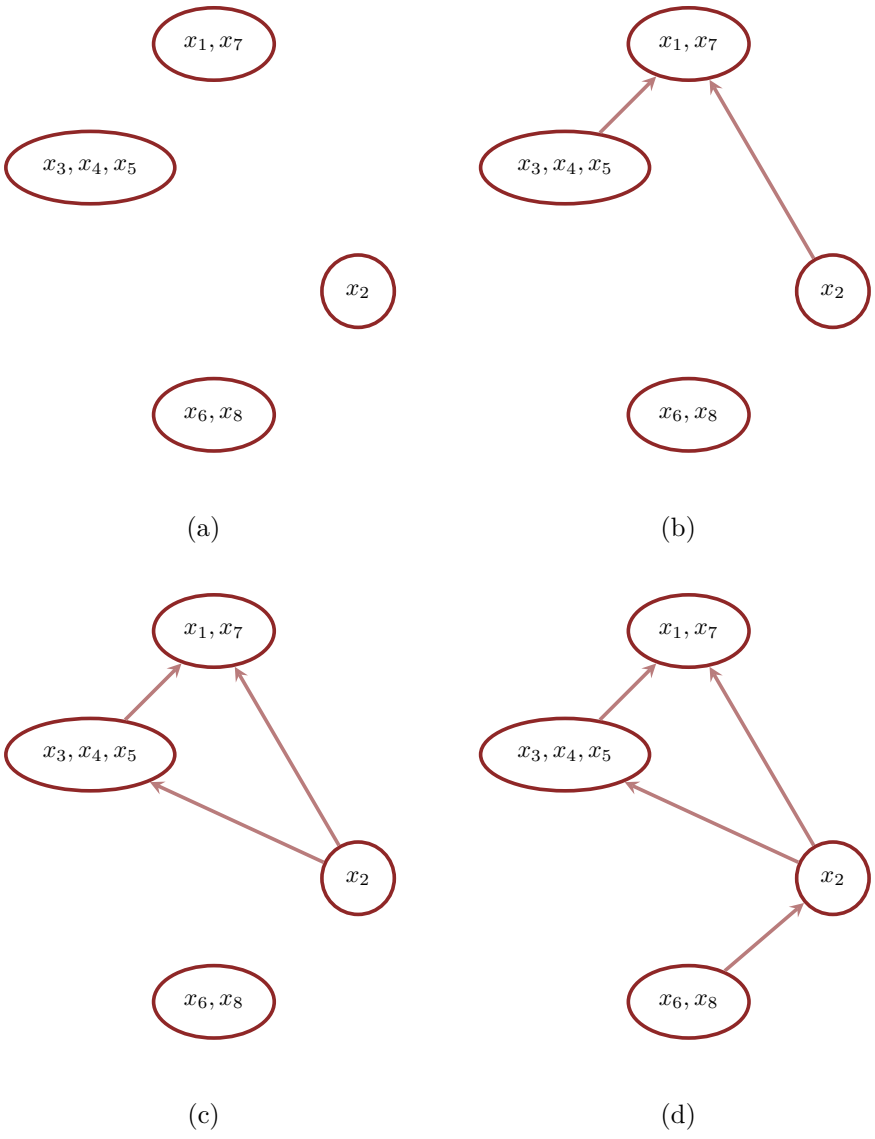


Figure 6: Graphical models are straightforward to build for atomic conditional decompositions. (a) After placing nodes for each subsequence of active component spaces, (b-d) we work down the conditional decomposition, placing directed arrows nodes that are conditionally dependent on each other.

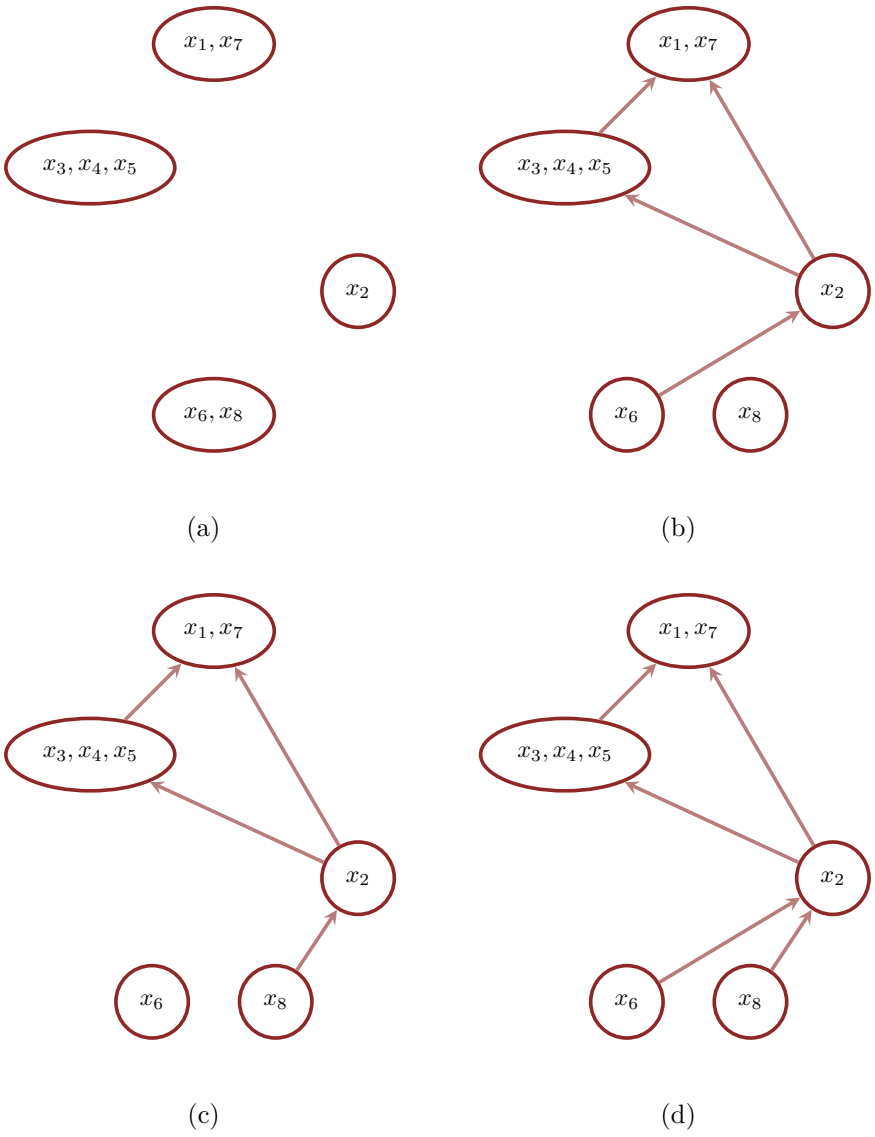
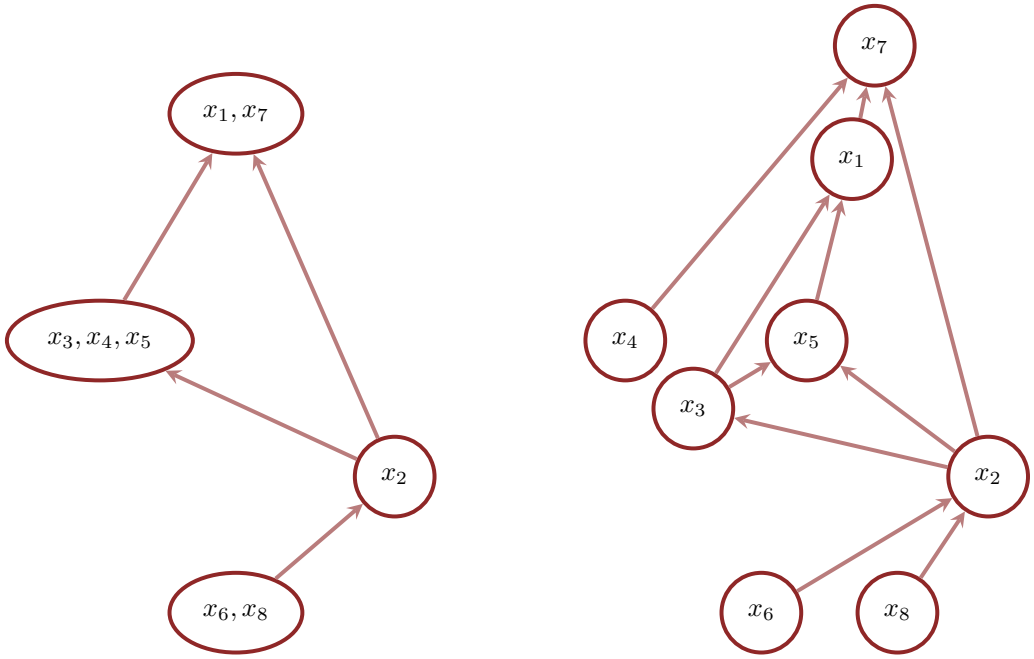


Figure 7: (a) Directed graphical models that visualize crude conditional decompositions can be elaborated into (b-d) directed graphical models that visualize more more refined conditional decompositions. In general, the many refinements will be consistent with a given non-atomic conditional decomposition.

component spaces. Consider, for instance, the atomic conditional decomposition

$$
\begin{array}{ll}
(& (\,(7), \{(1), (2), (4)\}\,), \\
 & (\,(1), \{(3), (5)\quad\}\,), \\
 & (\,(4), \{\quad\}\,), \\
 & (\,(5), \{(2), (3)\quad\}\,), \\
 & (\,(3), \{(2)\quad\}\,), \\
 & (\,(2), \{(6), (8)\quad\}\,), \\
 & (\,(6), \{\quad\}\,), \\
 & (\,(8), \{\quad\}\,)\quad) \\
\end{array}
$$

This is compatible with our initial example of a conditional decomposition, but conveys much more detailed information about the structure of the joint probability distribution (Figure 8).



(a) (b)

Figure 8: (a) The directed graphical models derived from cruder compositional decompositions always contain convey less information than (b) directed graphical models derived from atomic conditional decompositions.

The more coupled the behaviors in the component spaces are to each other, the more directed edges the corresponding graphical model will feature. Often the most informative aspect of a graphical model is the *absence* of directed edges.

6.3 Bells and Whistles

The nodes and edges in directed graphical models convey a wealth of information. We can, however, pack even more information into directed graphical models by augmenting this basic graph structure with some special features.

6.3.1 Auxiliary Nodes

Many applications consider joint distributions whose behavior depends on the behavior of auxiliary spaces that are not equipped with any probabilistic structure of their own. In other words, these joint distributions are *parameterized* by the configuration of the auxiliary spaces.

These non-probabilistic dependencies can be represented in graphical models by introducing multiple types of nodes, one to represent the probabilistic spaces and one to represent the auxiliary spaces. I like to use *rounded* shapes to represent probabilistic nodes and *rectangular* shapes to represent auxiliary nodes.

For instance, the conditional structure of

$$\begin{aligned} p(x_1, x_2, x_3, x_4; y) = \; & p(x_4 \mid x_1, x_3) \\ & \cdot p(x_1 \mid x_3; y) \\ & \cdot p(x_3 \mid x_2) \\ & \cdot p(x_2) \end{aligned}$$

can be represented by the graphical model in Figure 9.

6.3.2 Pushforward Nodes

Sometimes the behavior of a component space depends on the behavior of other component spaces through only the output of some intermediate function. For instance, instead of

$$\begin{aligned} p(x_1, x_2, x_3, x_4; y) = \; & p(x_4 \mid x_1, x_3) \\ & \cdot p(x_1 \mid x_3; y) \\ & \cdot p(x_3 \mid x_2) \\ & \cdot p(x_2) \end{aligned}$$

we might have

$$\begin{aligned} p(x_1, x_2, x_3, x_4; y) = \; & p(x_4 \mid f(x_1, x_3)) \\ & \cdot p(x_1 \mid x_3; y) \\ & \cdot p(x_3 \mid x_2) \\ & \cdot p(x_2). \end{aligned}$$

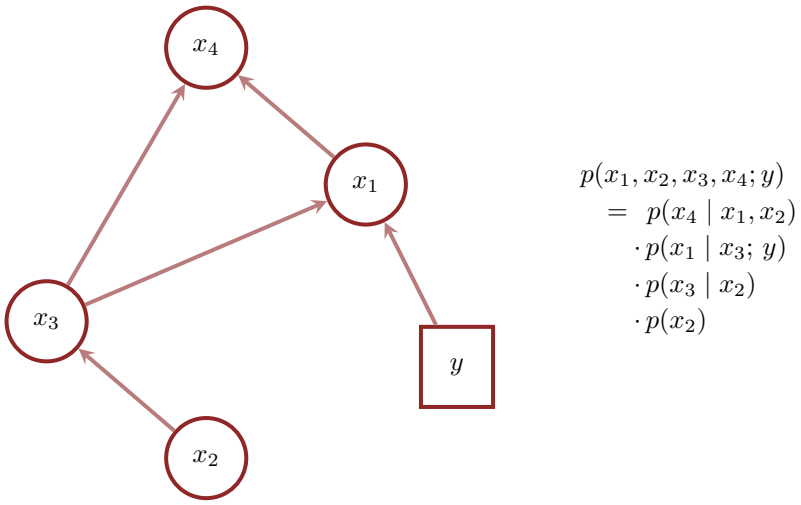


Figure 9: Rectangular nodes communicate non-probabilistic dependencies in joint probability distributions. Here the conditional density functions $p(x_1 \mid x_3; y)$ are parameterized by the configuration of a non-probabilistic space, $y \in Y$.

In this case, the behavior in X_4 doesn't directly depend on the behavior in $X_1 \times X_3$, but rather the behavior in $X_1 \times X_3$ pushed forward through f.

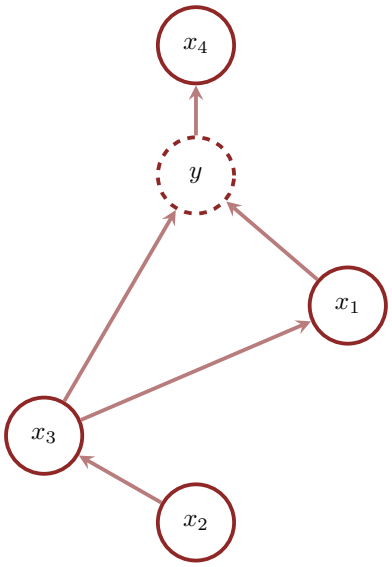
One way to communicate these functional dependencies in a graphical model is to introduce an explicit variable for the function output,

$$\begin{aligned} p(x_1, x_2, x_3, x_4; y) = \; & p(x_4 \mid y = f(x_1, x_3)) \\ & \cdot p(x_1 \mid x_3; y) \\ & \cdot p(x_3 \mid x_2) \\ & \cdot p(x_2), \end{aligned}$$

and then represent that variable with its own node. To avoid any ambiguity, however, we'll need to decorate this node differently from the component nodes. I like to use dashed borders (Figure 10)

6.3.3 Conditioned Nodes

Some applications are less concerned with a given joint distribution than certain conditional distributions derived from that joint distribution when we bind particular component variables to particular values.



$$
\begin{aligned}
&p(x_1, x_2, x_3, x_4; y) \\
&\quad = \; p(x_4 \mid y = f(x_1, x_3)) \\
&\qquad \cdot p(x_1 \mid x_3) \\
&\qquad \cdot p(x_3 \mid x_2) \\
&\qquad \cdot p(x_2)
\end{aligned}
$$

Figure 10: Dashed, circular nodes represent the output of functions that depend on components variables. We can interpret these nodes as modeling the pushforward distribution along the intermediate function.

Once we have a graphical model for the joint distribution, this conditioning process is straightforward to visualize by adding a distinct decoration to the nodes corresponding to the bound variables. A common convention is to fully color these nodes.

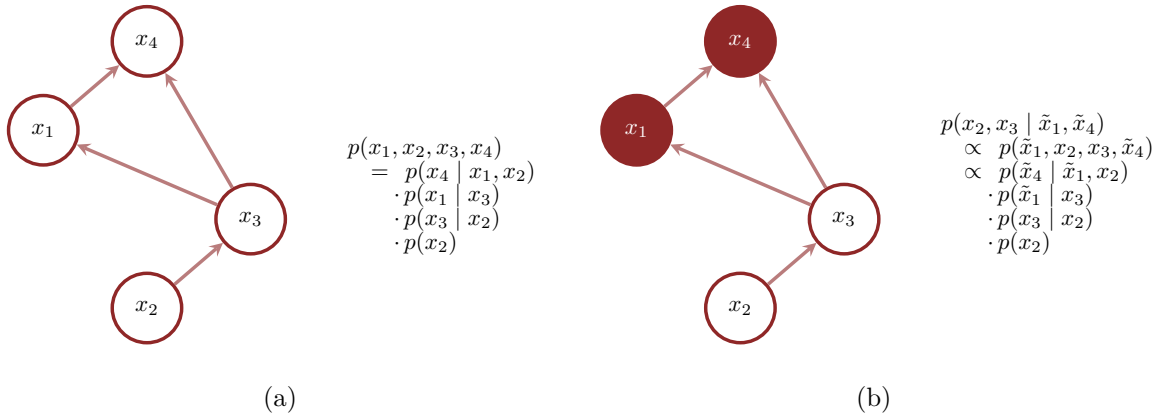
For example, let's say that we start with the joint distribution defined by the probability density functions (Figure 11a)

$$\begin{aligned} p(x_1, x_2, x_3, x_4) = \; & p(x_4 \mid x_1, x_3) \\ & \cdot p(x_1 \mid x_3) \\ & \cdot p(x_3 \mid x_2) \\ & \cdot p(x_2), \end{aligned}$$

but we are interested in the conditional distribution defined by probability density functions

$$p(x_2, x_3 \mid \tilde{x}_1, \tilde{x}_4) = \frac{p(\tilde{x}_1, x_2, x_3, \tilde{x}_4)}{p(\tilde{x}_1, \tilde{x}_4)}.$$

We can visualize this particular conditioning by coloring in the nodes representing X_1 and X_4 (Figure 11b).



(a) (b)

Figure 11: The various conditional distributions that we can derive from a joint distribution can be represented by (a) taking the graphical model for the joint distribution and (b) coloring in the nodes representing the components being bound to particular values.

6.3.4 Plate Notation

Some patterns of conditional dependencies are sufficiently common that it's useful to introduce graphical shorthands that allow for more compact visualizations.

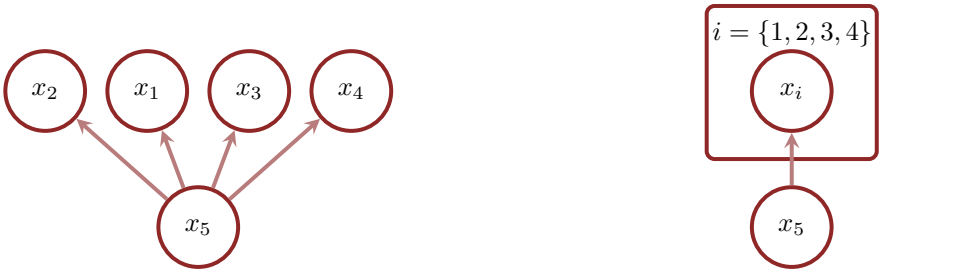
Consider, for example, the product space $X = X^{1:5}$, and the conditional decomposition

$$\begin{pmatrix} (\,(1), \{(5)\}\,), \\ (\,(2), \{(5)\}\,), \\ (\,(3), \{(5)\}\,), \\ (\,(4), \{(5)\}\,), \\ (\,(5), \{(5)\}\,) \end{pmatrix}$$

or, equivalently, the sequence of probability density functions

$$\begin{aligned} p(x_1, x_2, x_3, x_4, x_5) = \; & p(x_1 \mid x_5) \\ & \cdot p(x_2 \mid x_5) \\ & \cdot p(x_3 \mid x_5) \\ & \cdot p(x_4 \mid x_5) \\ & \cdot p(x_5). \end{aligned}$$

This particular conditional structure results in a fan-like graphical model (Figure 12a), which can take up a lot of visual space. We can make the graphical model a bit more compact, however, by packing the first four nodes together into a single **plate** that represents their common dependence on the fifth node.



$$p(x_1, x_2, x_3, x_4, x_5) = \left[\prod_{i=1}^{4} p(x_i \mid x_5)\right] p(x_5)$$

(a)

$$p(x_1, x_2, x_3, x_4, x_5) = \left[\prod_{i=1}^{4} p(x_i \mid x_5)\right] p(x_5)$$

(b)

Figure 12: (a) Conditional dependencies where many subsequences of active components depend on one, and only one, other subsequence of components, result in fan-like graphical models. (b) These graphical models can be visually compressed by packing the dependent nodes into a single plate.

This plate notation is especially useful when we have an arbitrary number of component spaces whose behavior conditionally depends on the same component spaces. For instance, we might

have the product space $X = X^{1:I} \times X_{I+1}$ and a joint distribution built up from the conditional decomposition

$$p(x_1, \ldots, x_{I+1}) = \left[\prod_{i=1}^{I} p(x_i \mid x_{I+1}) \right] p(x_{I+1}).$$

We can try to construct heuristic visualizations to represent the arbitrary number of nodes (Figure 13a), but the plate notation allow us to visualize this structure exactly (Figure 13b).

$p(x_1, \ldots, x_i, x_{I+1})$
$= \left[\prod_{i=1}^{I} p(x_i \mid x_{I+1})\right] p(x_{I+1})$

(a)

$p(x_1, \ldots, x_I, x_{I+1})$
$= \left[\prod_{i=1}^{I} p(x_i \mid x_{I+1})\right] p(x_{I+1})$

(b)

Figure 13: (a) Graphical models with an arbitrary number of nodes are awkward to visualize. (b) When applicable, plates avoid this awkwardness entirely.

Plates can also be nested to accommodate *hierarchical conditional dependencies*

6.4 Subtleties and Limitations

Directed graphical models can be a powerful tool for communicating the structure of conditional decompositions in practice, especially as a complement to explicit probability density functions. That said, directed graphical models are not without subtleties and limitations that require some care.

For instance, directed graphical models communicate the structure of a particular *conditional decomposition*, not a particular *joint distribution*. The same joint distribution under different partitions of the component indices will result in different graphical models. At the same time, different joint distributions with the same conditional dependencies will correspond to the same graphical model, just with differences in the particular conditional distributions and/or marginal distribution.

In other words, joint distributions alone do not define unique graphical models, and graphical models alone do not completely define joint distributions. In order to communicate the details

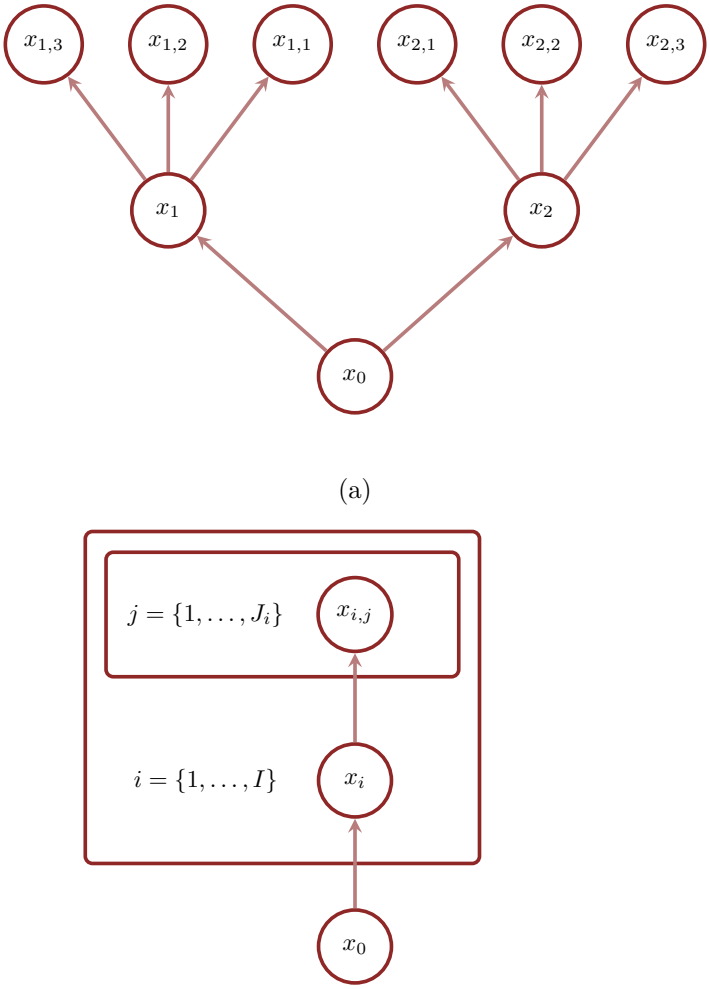


Figure 14: (a) Nested hierarchies of conditional dependencies can also be compactly visualized by (b) nested plates within other plates.

of a particular joint distribution, we need to complement a graphical model with additional information.

Another potential issue is that transforming a joint distribution will, in general, transform the corresponding graphical model. Transformations that preserve the product structure of a space will, by definition, also preserve the structure of conditional decompositions, and hence the structure of graphical models. For instance, we can freely transform individual component spaces without affecting the graphical modeling representation. On the other hand, transformations that reorder the component spaces or mix them together will change conditional decompositions, and hence the corresponding graphical models.

One of the more frustrating aspects of directed graphical models is that they can struggle to effectively communicate certain operations. Consider, for example, the joint distribution defined by the density function decomposition

$$\begin{aligned} p(x_1, \ldots, x_6) = \; & p(x_1 \mid x_4 - x_6) \\ & \cdot p(x_2 \mid x_6 - x_5) \\ & \cdot p(x_3 \mid x_5 - x_4) \\ & \cdot p(x_4)\, p(x_5)\, p(x_6). \end{aligned}$$

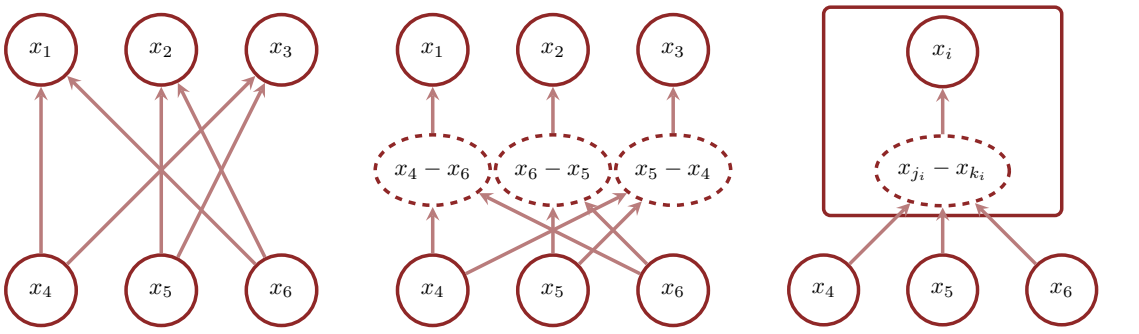
A graphical model using only six nodes, one for each X_i, can be confused with a more general conditional dependencies (Figure 15a),

$$\begin{aligned} p(x_1, \ldots, x_6) = \; & p(x_1 \mid x_4, x_6) \\ & \cdot p(x_2 \mid x_5, x_6) \\ & \cdot p(x_3 \mid x_4, x_5) \\ & \cdot p(x_4)\, p(x_5)\, p(x_6). \end{aligned}$$

We can capture the conditional dependence on the differences with pushforward nodes (Figure 15a), but the overlapping arrows can be difficult to parse. Moreover, the parsing becomes only more difficult as we add more nodes that also depend on differences between x_4, x_5, and x_6.

Another possibility is to employ plate notation (Figure 15c). The problem with this approach, however, is that each y_i doesn't depend on x_4, x_5, and x_6 all at the same time. Consequently, this plate diagram is ambiguous as well.

All of this is to say that graphical models are not always the best tool for communicating conditional dependencies. Sometimes a probability density function decomposition is just more effective. Other times we might need to reach for other representations, such as *probabilistic programming*. This will be a topic for a future chapter.



$$p(x_1, x_2, x_3, x_4, x_5, x_6) = p(x_1 \mid x_4, x_6) \cdot p(x_2 \mid x_5, x_6) \cdot p(x_3 \mid x_4, x_5) \cdot p(x_4)\, p(x_5)\, p(x_6)$$

(a)

$$p(x_1, x_2, x_3, x_4, x_5, x_6) = p(x_1 \mid x_4 - x_6) \cdot p(x_2 \mid x_6 - x_5) \cdot p(x_3 \mid x_5 - x_4) \cdot p(x_4)\, p(x_5)\, p(x_6)$$

(b)

$$p(x_1, x_2, x_3, x_4, x_5, x_6) = \prod_{i=1}^{3} p(x_i \mid x_{j_i} - x_{k_i}) \cdot p(x_4)\, p(x_5)\, p(x_6)$$

(c)

Figure 15: Some conditional structures are just awkward to to represent with graphical models. Consider, for example, a joint density function built up from the conditional density functions $p(x_1 \mid x_4 - x_6)$, $p(x_2 \mid x_6 - x_5)$, and $p(x_3 \mid x_5 - x_4)$. (a) The immediate graphical model doesn't communicate the particular dependence on the differences. (b) Introducing pushforward nodes helps, but at the expense of a much busier graphical model. (c) Using plates results in a more elegant graphical model, but requires introducing conditional indices j_i and k_i whose behavior is left ambiguous.

7 Conclusion

Directed graphical models are a powerful tool for not only communicating the properties of joint distributions, but also developing appropriate joint distributions in the first place. Specifically, they allow us to outline the basic structure of a joint distribution from the perspective of a given partition of the component spaces, without having to consider particular probability density functions at all. We will make extensive use of directed graphical models once we move on to probabilistic modeling.

Acknowledgements

I thank jd for helpful comments.

A very special thanks to everyone supporting me on Patreon: Alessandro Varacca, Alex D, Alexander Noll, Amit, Andrea Serafino, Andrew Mascioli, Andrew Rouillard, Andrés Castro Araújo, Ara Winter, Ari Holtzman, Austin Rochford, Aviv Keshet, Avraham Adler, Ben Matthews, Ben Swallow, Benoit Essiambre, boot, Brendan Galdo, Bryan Chang, Cameron Smith, Canaan Breiss, Cat Shark, Cathy Oliveri, Charles Naylor, Chase Dwelle, Chris Jones, Christina Van Heer, Christopher Mehrvarzi, Colin Carroll, Colin McAuliffe, Damien Mannion, dan mackinlay, Dan W Joyce, Dan Waxman, Dan Weitzenfeld, Daniel Hammarström, Danny Van Nest, David Burdelski, Dr. Jobo, Dr. Omri Har Shemesh, Dylan Maher, Dylan Spielman, Ebriand, Ed Cashin, Eric LaMotte, Erik Banek, Eugene O'Friel, Felipe González, Felipe Vaca, Fergus Chadwick, Francesco Corona, Geoff Rollins, Glenn Williams, Granville Matheson, Guilherme Marthe, Hamed Bastan-Hagh, haubur, Hector Munoz, Horace Guy, hs, Hugo Botha, Håkan Johansson, Ian Costley, idontgetoutmuch, Ignacio Vera, Ilaria Prosdocimi, iris pisscauldron, Isaac Vock, jacob pine, Jair Andrade, James C, James Hodgson, James Wade, Janek Berger, Jarrett Byrnes, Jason Pekos, Jason Wong, jd, Jeff Burnett, Jeff Dotson, Jeff Helzner, Jeffrey Erlich, Jerry Lin , Jesper Fischer Ehmsen, Jessica Graves, Joe Sloan, John Flournoy, Jonathan H. Morgan, Jonathon Vallejo, Josh Knecht, JU, Julian Lee, Justin Bois, Karim Naguib, Karim Osman, Karsten Skogsholm, Konstantin Shakhbazov, Kristian Gårdhus Wichmann, Kádár András, Lars Barquist, lizzie , LOU ODETTE, Mads Christian Hansen, Marek Kwiatkowski, Mark Donoghoe, Markus P., Daniel Edward Marthaler, Matthieu LEROY, Mattia Arsendi, Matěj, Maurits van der Meer, Max, Michael Colaresi, Michael DeJesus, Michael DeWitt, Michael Dillon, Michael Lerner, Mick Cooney, MisterMentat , Márton Vaitkus, N Sanders, Nathaniel Burbank, Nicholas Cowie, Nick S, Octavio Medina, Ole Rogeberg, Olivier Ma, Paolo, Pat LS, Patrick Kelley, Patrick Boehnke, Pau Pereira Batlle, Pieter van den Berg , ptr, quasar, Ramiro Barrantes Reynolds, Raúl Peralta Lozada, Rex, Riccardo Fusaroli, Richard Nerland, Rob Davies, Robert Frost, Robert Goldman, Robert kohn, Robin Taylor, Ryan Gan, Ryan Grossman, Ryan Kelly, S Hong, Sean Wilson, Sergiy Protsiv, Seth Axen, shira, Simon Duane, Simon Lilburn, Simon Steiger, Simone, Spencer, sssz, Stefan Lorenz, Stephen Lienhard, Steve Forrest, Steve Harris, Steven Forrest, Stew Watts, Stone Chen, Stoyan Georgiev,

Susan Holmes, Svilup, Tate Tunstall, Tatsuo Okubo, Teresa Ortiz, Theodore Dasher, Thomas Siegert, Thomas Vladeck, Tobychev , Tomas Capretto, Tony Wuersch, Virgile Andreani, Virginia Fisher, Vitalie Spinu, Vladimir M, VO2 Maximus Decius, Will Farr, Will Lowe, Will Wen, William A Vauter, yolhaj, yureq, and Zach A.

References

Bishop, Christopher M. 2006. *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer, New York.

Knopfmacher, A, and M. E. Mayes. 2005. "A Survey of Factorization Counting Functions." *International Journal of Number Theory* 01 (04): 563–81.

License