Chapter 20

Graphical Models

We have spent a lot of time looking at ways of figuring out how one variable (or set of variables) depends on another variable (or set of variables) — this is the core idea in regression and in conditional density estimation. We have also looked at how to estimate the joint distribution of variables, both with kernel density estimation and with models like factor and mixture models. The later two show an example of how to get the joint distribution by combining a conditional distribution (observables given factors; mixture components) with a marginal distribution (Gaussian distribution of factors; the component weights). When dealing with complex sets of dependent variables, it would be nice to have a general way of composing conditional distributions together to get joint distributions, and especially nice if this gave us a way of reasoning about what we could ignore, of seeing which variables are irrelevant to which other variables. This is what **graphical models** let us do.

20.1 Conditional Independence and Factor Models

The easiest way into this may be to start with the diagrams we drew for factor analysis. There, we had observables and we had factors, and each observable depended on, or loaded on, some of the factors. We drew a diagram where we had nodes, standing for the variables, and arrows running from the factors to the observables which depended on them. In the factor model, all the observables were conditionally independent of each other, given all the factors:

$$p(X_1, X_2, \ldots X_p | F_1, F_2, \ldots F_q) = \prod_{i=1}^{p} p(X_i | F_1, \ldots F_q) \tag{20.1}$$

But in fact observables are also independent of the factors they do not load on, so this is still too complicated. Let's write loads(i) for the set of factors on which the observable X_i loads. Then

$$p(X_1, X_2, \ldots X_p | F_1, F_2, \ldots F_q) = \prod_{i=1}^{p} p(X_i | F_{\text{loads}(i)}) \tag{20.2}$$

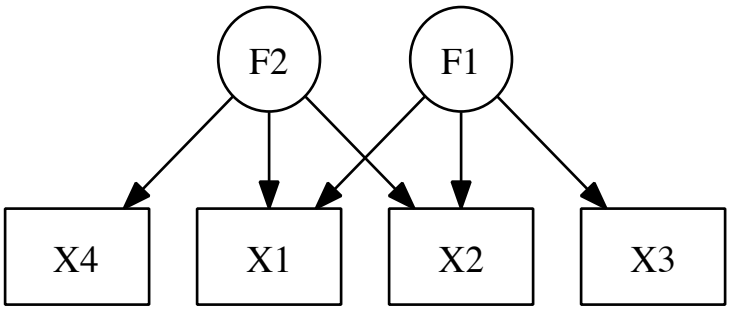

FIGURE 20.1: *Illustration of a typical model with two latent factors (F_1 and F_2, in circles) and four observables (X_1 through X_4).*

Consider Figure 20.1. The conditional distribution of observables given factors is

$$p(X_1, X_2, X_3, X_4|F_1, F_2) = p(X_1|F_1, F_2)p(X_2|F_1, F_2)p(X_3|F_1)p(X_4|F2) \tag{20.3}$$

X_1 loads on F_1 and F_2, so it is independent of everything else, given those two variables. X_1 is unconditionally dependent on X_2, because they load on common factors, F_1 and F_2; and X_1 and X_3 are also dependent, because they both load on F_1. In fact, X_1 and X_2 are still dependent given F_1, because X_2 still gives information about F_2. But X_1 and X_3 are independent given F_1, because they have no other factors in common. Finally, X_3 and X_4 are unconditionally independent because they have no factors in common. But they become dependent given X_1, which provides information about both the common factors.

None of these assertions rely on the detailed assumptions of the factor model, like Gaussian distributions for the factors, or linear dependence between factors and observables. What they rely on is that X_i is independent of *everything else*, given the factors it loads on. The idea of graphical models is to generalize this, by focusing on relations of direct dependence, and the conditional independence relations implied by them.

20.2 Directed Acyclic Graph (DAG) Models

We have a collection of variables, which to be generic I'll write $X_1, X_2, \ldots X_p$. These may be discrete, continuous, or even vectors; it doesn't matter. We represent these visually as nodes in a graph. There are arrows connecting some of these nodes. If an

arrow runs from X_i to X_j, then X_i is a **parent** of X_j. This is, as the name "parent" suggests, an anti-symmetric relationship, i.e., X_j cannot also be the parent of X_i. This is why we use an arrow, and why the graph is **directed**[1]. We write the set of all parents of X_j as parents(j); this generalizes the notion of the factors which an observable loads on to. The joint distribution "decomposes according to the graph":

$$p(X_1, X_2, \ldots X_p) = \prod_{i=1}^{p} p(X_i | X_{\text{parents}(i)}) \tag{20.4}$$

If X_i has no parents, because it has no incoming arrows, take $p(X_i|X_{\text{parents}(i)})$ just to be the marginal distribution $p(X_i)$. Such variables are called **exogenous**; the others, with parents, are **endogenous**. An unfortunate situation could arise where X_1 is the parent of X_2, which is the parent of X_3, which is the parent of X_1. Perhaps, under some circumstances, we could make sense of this and actually calculate with Eq. 20.4, but the general practice is to rule it out by assuming the graph is **acyclic**, i.e., that it has no cycles, i.e., that we cannot, by following a series of arrows in the graph, go from one node to other nodes and ultimately back to our starting point. Altogether we say that we have a **directed acyclic graph**, or **DAG**, which represents the direct dependencies between variables.[2]

What good is this? The primary virtue is that if we are dealing with a DAG model, the graph tells us all the dependencies we need to know; those are the conditional distributions of variables on their parents, appearing in the product on the right hand side of Eq. 20.4. (This includes the distribution of the exogeneous variables.) This fact has two powerful sets of implications, for probabilistic reasoning and for statistical inference.

Let's take inference first, because it's more obvious: all that we have to estimate are the conditional distributions $p(X_i|X_{\text{parents}(i)})$. We do not have to estimate the distribution of X_i given *all* of the other variables, unless of course they are all parents of X_i. Since estimating distributions, or even just regressions, conditional on many variables is hard, it is extremely helpful to be able to read off from the graph which variables we can *ignore*. Indeed, if the graph tells us that X_i is exogeneous, we don't have to estimate it conditional on anything, we just have to estimate its marginal distribution.

20.2.1 Conditional Independence and the Markov Property

The probabilistic implication of Eq. 20.4 is perhaps even more important, and that has to do with conditional independence. Pick any two variables X_i and X_j, where X_j is not a parent of X_i. Consider the distribution of X_i conditional on its parents *and* X_j. There are two possibilities. (i) X_j is not a descendant of X_i. Then we can see that X_i and X_j are conditionally independent. This is true *no matter what* the actual conditional distribution functions involved are; it's just implied by the joint

[1] See Appendix K for a brief review of the ideas and jargon of graph theory.
[2] See §20.6 for remarks on undirected graphical models, and graphs with cycles.

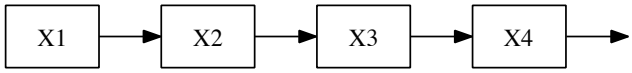

FIGURE 20.2: *DAG for a discrete-time Markov process. At each time t, X_t is the child of X_{t-1} and the parent of X_{t+1}.*

distribution respecting the graph. (ii) Alternatively, X_j is a descendant of X_i. Then in general they are not independent, even conditional on the parents of X_i. So the graph implies that certain conditional independence relations will hold, but that others in general will *not* hold.

As you know from your probability courses, a sequence of random variables $X_1, X_2, X_3, \ldots$ forms a **Markov process**[3] when "the past is independent of the future given the present": that is,

$$X_{t+1} \perp\!\!\!\perp (X_{t-1}, X_{t-2}, \ldots X_1) | X_t \tag{20.5}$$

from which it follows that

$$(X_{t+1}, X_{t+2}, X_{t+3}, \ldots) \perp\!\!\!\perp (X_{t-1}, X_{t-2}, \ldots X_1) | X_t \tag{20.6}$$

which is called the **Markov property**. DAG models have a similar property: if we take any collection of nodes I, it is independent of its non-descendants, given its parents:

$$X_I \perp\!\!\!\perp X_{\text{non-descendants}(I)} | X_{\text{parents}(I)} \tag{20.7}$$

This is the **directed graph Markov property**. The ordinary Markov property is in act a special case of this, when the graph looks like Figure 20.2[4].

On the other hand, if we condition on one of X_i's children, X_i will generally be dependent on any other parent of that child. If we condition on multiple children of X_i, we'll generally find X_i is dependent on all its co-parents. It should be plausible, and is in fact true, that X_i is independent of everything else in the graph if we condition on its parents, its children, and its children's other parents. This set of nodes is called X_i's **Markov blanket**.

20.3 Conditional Independence and *d*-Separation

It is clearly very important to us to be able to deduce when two sets of variables are conditionally independent of each other given a third. One of the great uses of

[3] After the Russian mathematician A. A. Markov, who introduced the theory of Markov processes in the course of a mathematical dispute with his arch-nemesis, to show that probability and statistics could apply to dependent events, and hence that Christianity was not *necessarily* true (I am not making this up: Basharin *et al.*, 2004).

[4] To see this, take the "future" nodes, indexed by $t+1$ and up, as the set I. Their parent consists just of X_t, and all their non-descendants are the even earlier nodes at times $t-1$, $t-2$, etc.

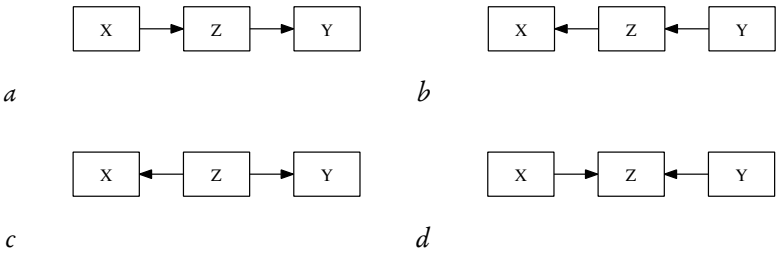

FIGURE 20.3: *Four DAGs for three linked variables. The first two (a and b) are called* **chains***; c is a* **fork***; d is a* **collider***. If these were the whole of the graph, we would have* $X \not\perp\!\!\!\perp Y$ *and* $X \perp\!\!\!\perp Y|Z$*. For the collider, however, we would have* $X \perp\!\!\!\perp Y$ *while* $X \not\perp\!\!\!\perp Y|Z$*.*

DAGs is that they give us a fairly simple criterion for this, in terms of the graph itself. *All* distributions which conform to a given DAG share a common set of conditional independence relations, implied by the Markov property, no matter what their parameters or the form of the distributions.

Our starting point is that when we have a single directed edge, we can reason from the parent to the child, or from the child to the parent. While (as we'll see in Part IV) it's reasonable to say that *influence* or *causation* flows one way, along the direction of the arrows, *statistical information* can flow in either direction. Since dependence is the presence of such statistical information, if we want to figure out which variables are dependent on which, we need to keep track of these information flows.

While we can do inference in either direction across any one edge, we do have to worry about whether we can propagate this information further. Consider the four graphs in Figure 20.3. In every case, we condition on X, which acts as the source of information. In the first three cases, we can (in general) propagate the information from X to Z to Y — the Markov property tells us that Y is independent of its non-descendants given its parents, but in none of those cases does that make X and Y independent. In the last graph, however, what's called a **collider**[5], we cannot propagate the information, because Y has no parents, and X is not its descendant, hence they are independent. We learn about Z from X, but this doesn't tell us anything about Z's other cause, Y.

All of this flips around when we condition on the intermediate variable (Z in Figure 20.3). The chains (Figures 20.3*a* and *b*), conditioning on the intermediate variable blocks the flow of information from X to Y — we learn nothing more about Y from X and Z than from Z alone, at least not along this path. This is also true of the **fork** (Figure 20.3*c*) — conditional on their common cause, the two effects are uninformative about each other. But in a collider, conditioning on the common effect Z makes X and Y dependent on each other, as we've seen before. In fact, if we don't condition on Z, but do condition on a descendant of Z, we also create dependence between Z's parents.

[5]Because two incoming arrows "collide" there.

We are now in a position to work out conditional independence relations. We pick our two favorite variables, X and Y, and condition them both on some third set of variables S. If S **blocks** every undirected path[6] from X to Y, then they must be conditionally independent given S. An unblocked path is also called **active**. A path is active when every variable along the path is active; if even one variable is blocked by S, the whole path is blocked. A variable Z along a path is active, conditioning on S, if

1. Z is a collider along the path, and in S; or,
2. Z is a descendant of a collider, and in S; or
3. Z is not a collider, and not in S.

Turned around, Z is blocked or de-activated by conditioning on S if

1. Z is a non-collider and in S; or
2. Z is collider, and neither Z nor any of its descendants is in S

In words, S blocks a path when it blocks the flow of information by conditioning on the middle node in a chain or fork, and doesn't create dependence by conditioning on the middle node in a collider (or the descendant of a collider). Only *one* node in a path must be blocked to block the whole path. When S blocks *all* the paths between X and Y, we say it **d-separates** them[7]. A collection of variables U is d-separated from another collection V by S if every $X \in U$ and $Y \in V$ are d-separated.

In every distribution which obeys the Markov property, d-separation implies conditional independence[8]. It is not *always* the case that the reverse implication, the one from conditional independence to *d*-separation, holds good. We will see in Part IV, that when the distribution is "faithful" to a DAG, causal inference is immensely simplified. But going from d-separation to conditional independence is true in any DAG, whether or not it has a causal interpretation.

20.3.1 D-Separation Illustrated

The discussion of d-separation has been rather abstract, and perhaps confusing for that reason. Figure 20.4 shows a DAG which might make this clearer and more concrete.

If we make the conditioning set S the empty set, that is, we condition on nothing, we "block" paths which pass through colliders. For instance, there are three exogenous variables in the graph, X_2, X_3 and X_5. Because they have no parents, any path from one to another must go over a collider (Exercises 1 and 2). If we do not condition on anything, therefore, we find that the exogenous variables are d-separated and thus independent. Since X_3 is not on any path linking X_2 and X_5,

[6] Whenever I talk about undirected paths, I mean paths without cycles.

[7] The "d" stands for "directed"

[8] We will not prove this, though I hope I have made it plausible. You can find demonstrations in Spirtes *et al.* (2001); Pearl (2000); Lauritzen (1996).

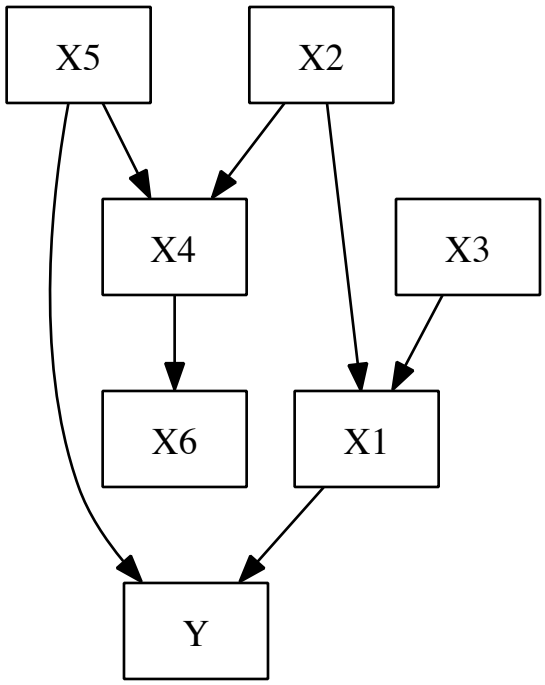

FIGURE 20.4: *Example DAG used to illustrate d-separation.*

or descended from a node on any such path, if we condition only on X_3, then X_2 and X_5 are still d-separated, so $X_2 \perp\!\!\!\perp X_5 | X_3$. There are two paths linking X_3 to X_5: $X_3 \rightarrow X_1 \leftarrow X_2 \rightarrow X_4 \leftarrow X_5$, and $X_3 \rightarrow X_1 \rightarrow Y \leftarrow X_5$. Conditioning on X_2 (and nothing else) blocks the first path (since X_2 is part of it, but is a fork), and also blocks the second path (since X_2 is not part of it, and Y is a blocked collider). Thus, $X_3 \perp\!\!\!\perp X_5 | X_2$. Similarly, $X_3 \perp\!\!\!\perp X_2 | X_5$ (Exercise 4).

For a somewhat more challenging example, let's look at the relation between X_3 and Y. There are, again, two paths here: $X_3 \rightarrow X_1 \rightarrow Y$, and $X_3 \rightarrow X_1 \leftarrow X_2 \rightarrow X_4 \leftarrow X_5 \rightarrow Y$. If we condition on nothing, the first path, which is a simple chain, is open, so X_3 and Y are d-connected and dependent. If we condition on X_1, we block the first path. X_1 is a collider on the second path, so conditioning on X_1 opens the path there. However, there is a second collider, X_4, along this path, and just conditioning on X_1 does not activate the second collider, so the path as a whole remains blocked.

$$Y \not\perp\!\!\!\perp X_3 \tag{20.8}$$
$$Y \perp\!\!\!\perp X_3 | X_1 \tag{20.9}$$

To activate the second path, we can condition on X_1 and either X_4 (a collider along that path) or on X_6 (a descendant of a collider) or on both:

$$Y \not\perp\!\!\!\perp X_3 | X_1, X_4 \tag{20.10}$$
$$Y \not\perp\!\!\!\perp X_3 | X_1, X_6 \tag{20.11}$$
$$Y \not\perp\!\!\!\perp X_3 | X_1, X_4, X_6 \tag{20.12}$$

Conditioning on X_4 and/or X_6 does not activate the $X_3 \rightarrow X_1 \rightarrow Y$ path, but it's enough for there to be one active path to create dependence.

To block the second path again, after having opened it in one of these ways, we can condition on X_2 (since it is a fork along that path, and conditioning on a fork blocks it), or on X_5 (also a fork), or on both X_2 and X_5. So

$$Y \perp\!\!\!\perp X_3 | X_1, X_2 \tag{20.13}$$
$$Y \perp\!\!\!\perp X_3 | X_1, X_5 \tag{20.14}$$
$$Y \perp\!\!\!\perp X_3 | X_1, X_2, X_5 \tag{20.15}$$
$$Y \perp\!\!\!\perp X_3 | X_1, X_2, X_4 \tag{20.16}$$
$$Y \perp\!\!\!\perp X_3 | X_1, X_2, X_6 \tag{20.17}$$
$$Y \perp\!\!\!\perp X_3 | X_1, X_2, X_5, X_6 \tag{20.18}$$

etc., etc.

Let's look at the relationship between X_4 and Y. X_4 is not an ancestor of Y, or a descendant of it, but they do share common ancestors, X_5 and X_2. Unconditionally, Y and X_4 are dependent, both through the path going $X_4 \leftarrow X_5 \rightarrow Y$, and through that going $X_4 \leftarrow X_2 \rightarrow X_1 \rightarrow Y$. Along both paths, the exogenous variables are forks, so *not* conditioning on them leaves the path unblocked. X_4 and Y become d-separated when we condition on X_5 and X_2.

X_6 and X_3 have no common ancestors. Unconditionally, they should be independent, and indeed they are: the two paths are $X_6 \leftarrow X_4 \leftarrow X_2 \rightarrow X_1 \leftarrow X_3$, and

$X_6 \leftarrow X_4 \leftarrow X_5 \rightarrow Y \leftarrow X_1 \leftarrow X_3$. Both paths contain a single collider (X_1 and Y, respectively), so if we do not condition on them the paths are blocked and X_6 and X_3 are independent. If we condition on either Y or X_1 (or both), however, we unblock the paths, and X_6 and X_3 become d-connected, hence dependent. To get back to d-separation while conditioning on Y, we must also condition on X_4 or X_5, or both. To get d-separation while conditioning on X_1, we must also condition on X_4, or on X_2, or on X_4 and X_2. If we condition on both X_1 and Y and want d-separation, we could just add conditioning on X_4, or we could condition on X_2 and X_5, or all three.

If the abstract variables are insufficiently concrete, consider reading them as follows:

$$
\begin{aligned}
Y &\Leftrightarrow \text{Grade in this class}\\
X_1 &\Leftrightarrow \text{Effort spent on this class}\\
X_2 &\Leftrightarrow \text{Enjoyment of statistics}\\
X_3 &\Leftrightarrow \text{Workload this term}\\
X_4 &\Leftrightarrow \text{Quality of work in linear regression class}\\
X_5 &\Leftrightarrow \text{Amount learned in linear regression class}\\
X_6 &\Leftrightarrow \text{Grade in linear regression}
\end{aligned}
$$

Pretending, for the sake of illustration, that this is accurate, how heavy your workload is this semester (X_3) would predict, or rather retrodict, your grade in linear regression last semester (X_6), once we control for how much effort you put into this class (X_1). Changing your workload this semester would not, however, reach backwards in time to raise or lower your grade in regression.

20.3.2 Linear Graphical Models and Path Coefficients

We began our discussion of graphical models with factor analysis as our starting point. Factor models are a special case of linear (directed) graphical models, a.k.a. path models[9] As with factor models, in the larger class we typically center all the variables (so they have expectation zero) and scale them (so they have variance 1). In factor models, the variables were split into two sets, the factors and the observables, and all the arrows went from factors to observables. In the more general case, we do not necessarily have this distinction, but we still assume the arrows from a directed acyclic graph. The conditional expectation of each variable is a linear combination of the values of its parents:

$$\mathbb{E}\left[X_i \,|\, X_{\text{parents}(i)}\right] = \sum_{j \in \text{parents}(i)} w_{ji} X_j \tag{20.19}$$

just as in a factor model. In a factor model, the coefficients w_{ji} were the factor loadings. More generally, they are called **path coefficients**.

[9]Some people use the phrase "structural equation models" for linear directed graphical models exclusively.

The path coefficients determine all of the correlations between variables in the model. If all of the variables have been standardized to mean zero and variance 1, and the path coefficients are calculated for these standardized variables, we can find the correlation between X_i and X_j as follows:

- Find all of the undirected paths between X_i and X_j.
- Discard all of the paths which go through colliders.
- For each remaining path, multiply all the path coefficients along the path.
- Sum up these products over paths.

These rules were introduced by the great geneticist and mathematical biologist Sewall Wright in the early 20th century (see further reading for details). These "Wright path rules" often seem mysterious, particularly the bit where paths with colliders are thrown out. But from our perspective, we can see that what Wright is doing is finding all of the *unblocked* paths between X_i and X_j. Each path is a channel along which information (here, correlation) can flow, and so we add across channels.

It is frequent, and customary, to assume that all of the variables are Gaussian. (We saw this in factor models as well.) With this extra assumption, the joint distribution of all the variables is a multivariate Gaussian, and the correlation matrix (which we find from the path coefficients) gives us the joint distribution.

If we want to find correlations conditional on a set of variables S, $\mathrm{corr}(X_i, X_j|S)$, we still sum up over the unblocked paths. If we have avoided conditioning on colliders, then this is just a matter of dropping the now-blocked paths from the sum. If on the other hand we have conditioned on a collider, that path *does* become active (unless blocked elsewhere), and we in fact need to modify the path weights. Specifically, we need to work out the correlation induced between the two parents of the collider, by conditioning on that collider. This can be calculated from the path weights, and some fairly tedious algebra[10]. The important thing is to remember that the rule of d-separation still applies, and that conditioning on a collider can create correlations.

Path Coefficients and Covariances If the variables have not all been standardized, but Eq. 20.19 still applies, it is often desirable to calculate covariances, rather than correlation coefficients. This involves a little bit of extra work, by way of keeping track of variances, and in particular the variances of "source" terms. Since many references do not state the path-tracing rules for covariances, it's worth going over them here.

To find the *marginal* covariance between X_i and X_j, the procedure is as follows:

1. Find all of the unblocked paths between X_i and X_j (i.e., discard all paths which go through colliders).
2. For each remaining path:

[10] See for instance Li *et al.* (1975).

   (a) multiply all the path coefficients along the path;
   (b) find the node along that path which is the ancestor of all the other nodes along that path[11], and call it the path's source;
   (c) multiply the product of the coefficients by the variance of the source.
3. Sum the product of path coefficients and source variances over all remaining paths.

(Notice that if all variables are standardized to variance 1, we don't have to worry about source variances, and these rules reduce to the previous ones.)

To find the *conditional* covariance between X_i and X_j given a set of variables S, there are two procedures, depending on whether or not conditioning on S opens any paths between X_i and X_j by including colliders. If S does not contain any colliders or descendants of colliders (on paths between X_i and X_j),

1. For each unblocked path linking X_i and X_j:
   (a) multiply all the path coefficients along the path;
   (b) find the source of each path[12];
   (c) multiply the product of the coefficients by the variance of the source.
2. Sum the product of path coefficients and source variances over all remaining paths.

If, on the other hand, conditioning on S opens paths by conditioning on colliders (or their descendants), then we would have to handle the consequences of conditioning on a collider. This is usually too much of a pain to do graphically, and one should fall back on algebra. The next sub-section does however say a bit about what *qualitatively* happens to the correlations.

[[TODO: In final revision, write out full graphical rules for completeness]]

20.3.3 Positive and Negative Associations

We say that variables X and Y are **positively associated** if increasing X predicts, on average, an increase in Y, and vice versa[13]; if increasing X predicts a decrease in Y, then they are **negatively associated**. If this holds when conditioning out other variables, we talk about positive and negative partial associations. Heuristically, positive association means positive correlation in the neighborhood of any given x, though the magnitude of the positive correlation need not be constant. Note that not all dependent variables have to have a definite sign for their association.

We can multiply together the signs of positive and negative partial associations along a path in a graphical model, the same we can multiply together path coefficients in a linear graphical model. Paths which contain (inactive!) colliders should be neglected. If all the paths connecting X and Y have the same sign, then we know

[11] Showing that such an ancestor exists is Exercise 3a.

[12] Showing that the source of an unblocked, collider-free path cannot be in S is Exercise 3b.

[13] I.e., if $\frac{d\mathbb{E}[Y|X=x]}{dx} \geq 0$

that over-all association between X and Y must have that sign. If different paths have different signs, however, then signs alone are not enough to tell us about the over-all association.

If we are interested in conditional associations, we have to consider whether our conditioning variables block paths or not. Paths which are blocked by conditioning should be dropped from consideration. If a path contains an activated collider, we need to include it, but we reverse the sign of one arrow into the collider. That is, if $X \stackrel{+}{\rightarrow} Z \stackrel{+}{\leftarrow} Y$, and we condition on Z, we need to replace one of the plus signs with a $-$ sign, because the two parents now have an over-all negative association.[14] If on the other hand one of the incoming arrows had a positive association and the other was negative, we need to flip one of them so they are both positive or both negative; it doesn't matter which, since it creates a positive association between the parents[15].

[[TODO: Write out formal proofs as appendix]]

20.4 Independence, Conditional Independence, and Information Theory

[[TODO: Move to planned appendix on information theory]]

Take two random variables, X and Y. They have some joint distribution, which we can write $p(x,y)$. (If they are both discrete, this is the joint probability mass function; if they are both continuous, this is the joint probability density function; if one is discrete and the other is continuous, there's still a distribution, but it needs more advanced tools.) X and Y each have marginal distributions as well, $p(x)$ and $p(y)$. $X \perp\!\!\!\perp Y$ if and only if the joint distribution is the product of the marginals:

$$X \perp\!\!\!\perp Y \Leftrightarrow p(x,y) = p(x)p(y) \tag{20.20}$$

We can use this observation to measure how dependent X and Y are. Let's start with the log-likelihood ratio between the joint distribution and the product of marginals:

$$\log \frac{p(x,y)}{p(x)p(y)} \tag{20.21}$$

This will always be exactly 0 when $X \perp\!\!\!\perp Y$. We use its average value as our measure of dependence:

$$I[X;Y] \equiv \sum_{x,y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)} \tag{20.22}$$

(If the variables are continuous, replace the sum with an integral.) Clearly, if $X \perp\!\!\!\perp Y$, then $I[X;Y] = 0$. One can show[16] that $I[X;Y] \geq 0$, and that $I[X;Y] = 0$ implies

[14] If both smoking and asbestos are positively associated with lung cancer, and we know the patient does not have lung cancer, then high levels of smoking must be compensated for by low levels of asbestos, and vice versa.

[15] If yellow teeth are positively associated with smoking and negatively associated with dental insurance, and we know the patient does not have yellow teeth, then high levels of smoking must be compensated for by excellent dental care, and conversely poor dental care must be compensated for by low levels of smoking.

[16] Using the same type of convexity argument ("Jensen's inequality") we used §19.2.1 for understanding why the EM algorithm works.

$X \perp\!\!\!\perp Y$. The quantity $I[X;Y]$ is clearly symmetric between X and Y. Less obviously, $I[X;Y] = I[f(X);g(Y)]$ whenever f and g are invertible functions. This **coordinate-freedom** means that $I[X;Y]$ measures *all forms* of dependence, not just linear relationships, like the ordinary (Pearson) correlation coefficient, or monotone dependence, like the rank (Spearman) correlation coefficient. In information theory, $I[X;Y]$ is called the **mutual information**, or **Shannon information**, between X and Y. So we have the very natural statement that random variables are independent just when they have no information about each other.

There are (at least) two ways of giving an operational meaning to $I[X;Y]$. One, the original use of the notion, has to do with using knowledge of Y to improve the efficiency with which X can be encoded into bits (Shannon, 1948; Cover and Thomas, 2006). While this is very important — it's literally transformed the world since 1945 — it's not very statistical. For statisticians, what matters is that if we test the hypothesis that X and Y are independent, with joint distribution $p(x)p(y)$, against the hypothesis that they dependent, with joint distribution $p(x,y)$, then the mutual information controls the error probabilities of the test. To be exact, if we fix any power we like (90%, 95%, 99.9%, ...), the size or type I error rate α_n, of the best possible test shrinks exponentially with the number of IID samples n, and the rate of exponential decay is precisely $I[X;Y]$ (Kullback, 1968, §4.3, theorem 4.3.2):

$$\lim_{n\to\infty} -\frac{1}{n}\log \alpha_n \leq I[X;Y] \tag{20.23}$$

So positive mutual information means dependence, and the magnitude of mutual information tells us about how detectable the dependence is[17].

Suppose we conditioned X and Y on a third variable (or variables) Z. For each realization z, we can calculate the mutual information,

$$I[X;Y|Z=z] \equiv \sum_{x,y} p(x,y|z)\log\frac{p(x,y|z)}{p(x|z)p(y|z)} \tag{20.24}$$

And we can average over z,

$$I[X;Y|Z] \equiv \sum_{z} p(z) I[X;Y|Z=z] \tag{20.25}$$

This is the **conditional mutual information**. It will not surprise you at this point to learn that $X \perp\!\!\!\perp Y|Z$ if and only if $I[X;Y|Z] = 0$. The magnitude of the conditional mutual information tells us how easy it is to detect conditional dependence.

[17]Symmetrically, if we follow the somewhat more usual procedure of fixing a type I error rate α, the type II error rate β_n (= 1−power) also goes to zero exponentially, and the exponential rate is $\sum_{x,y} p(x)p(y)\log\frac{p(x)p(y)}{p(x,y)}$, a quantity called the "lautam information" (Palomar and Verdú, 2008). (For proofs of the exponential rate, see Palomar and Verdú (2008, p. 965), following Kullback (1968, §4.3, theorem 4.3.3).)

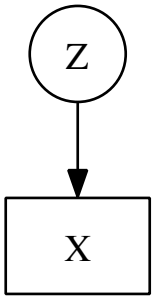

FIGURE 20.5: *DAG for a mixture model. The latent class Z is exogenous, and the parent of the observable random vector X. (If the components of X are conditionally independent given Z, they could be represented as separate boxes on the lower level.*

20.5 Examples of DAG Models and Their Uses

Factor models are examples of DAG models (as we've seen). So are mixture models (Figure 20.5) and Markov chains (see above). DAG models are considerably more flexible, however, and can combine observed and unobserved variables in many ways.

Consider, for instance, Figure 20.6. Here there are two exogeneous variables, labeled "Smoking" and "Asbestos". Everything else is endogenous. Notice that "Yellow teeth" is a child of "Smoking" alone. This does not mean that (in the model) whether someone's teeth get yellowed (and, if so, how much) is a function of smoking alone; it means that whatever other influences go into that are independent of the rest of the model, and so unsystematic that we can think about those influences, taken together, as noise.

Continuing, the idea is that how much someone smokes influences how yellow their teeth become, and also how much tar builds up in their lungs. Tar in the lungs, in turn, leads to cancer, as does by exposure to asbestos.

Now notice that, in this model, teeth-yellowing will be unconditionally dependent on, i.e., associated with, the level of tar in the lungs, because they share a common parent, namely smoking. Yellow teeth and tarry lungs will however be conditionally independent given that parent, so if we control for smoking we should not be able to predict the state of someone's teeth from the state of their lungs or vice versa.

On the other hand, smoking and exposure to asbestos are independent, at least in this model, as they are both exogenous[18]. Conditional on whether someone has

[18]If we had two variables which in some physical sense were exogenous but dependent on each other,

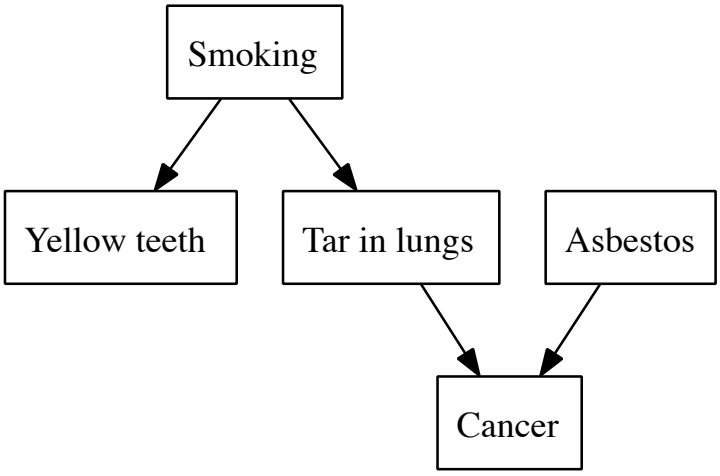

FIGURE 20.6: *DAG model indicating (hypothetical) relationships between smoking, asbestos, cancer, and covariates.*

cancer, however, smoking and asbestos will become *dependent*.

To understand the logic of this, suppose (what is in fact true) that both how much someone smokes and how much they are exposed to asbestos raises the risk of cancer. Conditional on not having cancer, then, one was probably exposed to little of either tobacco smoke or asbestos. Conditional on both not having cancer and having been exposed to a high level of asbestos, one probably was exposed to an unusually low level of tobacco smoke. Vice versa, no cancer plus high levels of tobacco tend to imply especially little exposure to asbestos. We thus have created a negative association between smoking and asbestos by conditioning on cancer. Naively, a regression where we "controlled for" cancer would in fact tell us that exposure to asbestos keeps tar from building up in the lungs, prevents smoking, and whitens teeth.

More generally, conditioning on a third variable can *create* dependence between otherwise independent variables, when what we are conditioning on is a common *descendant* of the variables in question.[19] This conditional dependence is *not* some kind of finite-sample artifact or error — it's really there in the joint probability distribution. If all we care about is prediction, then it is perfectly legitimate to use it. In the world of Figure 20.6, it really is true that you can predict the color of someone's teeth from whether they have cancer and how much asbestos they've been exposed to, so if that's what you want to predict[20], why not use that information? But if you want to do more than just make predictions without understanding, if you want to understand the structure tying together these variables, if you want to do *science*, if you don't want to go around telling yourself that asbestos whitens teeth, you really do need to know the graph.[21]

20.5.1 Missing Variables

Suppose that we do not observe one of the variables, such as the quantity of tar in the lungs, but we somehow know all of the conditional distributions required by the graph. (Tar build-up in the lungs might indeed be hard to measure for living people.) Because we have a joint distribution for *all* the variables, we could estimate the conditional distribution of one of them given the rest, using the definition of conditional probability and of integration:

$$p(X_i|X_1,X_2,X_{i-1},X_{i+1},X_p) = \frac{p(X_1,X_2,X_{i-1},X_i,X_{i+1},X_p)}{\int p(X_1,X_2,X_{i-1},x_i,X_{i+1},X_p)dx_i} \tag{20.26}$$

We could in principle do this for *any* joint distribution. When the joint distribution comes from a DAG model, however, we can simplify this considerably. Recall from

we would represent them in a DAG model by either a single *vector-valued* random variable (which would get only one node), or as children of a latent unobserved variable, which was truly exogenous.

[19]Economists, psychologists, and other non-statisticians often repeat the advice that if you want to know the effect of X on Y, you should not condition on Z when Z is endogenous. This is bit of folklore is a relic of the days of ignorance, when our ancestors groped towards truths they could not grasp. If we want to know whether asbestos is associated with tar in the lungs, conditioning on the yellowness of teeth is fine, even though that is an endogenous variable.

[20]Maybe you want to guess who'd be interested in buying whitening toothpaste.

[21]We return to this example in §24.2.2.

§20.2.1 that X_i is independent of all the other variables given its Markov blanket, i.e., its parents, its children, and the other parents of its children. We can therefore drop from the conditioning everything which isn't in the Markov blanket. Actually *doing* the calculation then boils down to a version of the EM algorithm.[22]

If we observe only a subset of the other variables, we can still use the DAG to determine which ones actually matter to estimating X_i, and which ones are superfluous. The calculations then however become much more intricate.[23]

20.6 Non-DAG Graphical Models: Undirected Graphs and Directed Graphs with Cycles

This section is optional, as, for various reasons, we will not use these models in this course.

20.6.1 Undirected Graphs

There is a lot of work on probability models which are based on *undirected* graphs, in which the relationship between random variables linked by edges is completely symmetric, unlike the case of DAGs[24]. Since the relationship is symmetric, the preferred metaphor is not "parent and child", but "neighbors". The models are sometimes called **Markov networks** or **Markov random fields**, but since DAG models have a Markov property of their own, this is not a happy choice of name, and I'll just call them "undirected graphical models".

The key Markov property for undirected graphical models is that any set of nodes I is independent of the rest of the graph given its neighbors:

$$X_I \perp\!\!\!\perp X_{\text{non-neighbors}(I)} | X_{\text{neighbors}(I)} \tag{20.27}$$

This corresponds to a factorization of the joint distribution, but a more complex one than that of Eq. 20.4, because a symmetric neighbor-of relation gives us no way of *ordering* the variables, and conditioning the later ones on the earlier ones. The trick turns out to go as follows. First, as a bit of graph theory, a **clique** is a set of nodes which are all neighbors of each other, and which cannot be expanded without losing that property. We write the collection of all cliques in a graph G as cliques(G). Second, we introduce **potential functions** ψ_c which take clique configurations and return non-negative numbers. Third, we say that a joint distribution is a **Gibbs dis-**

[22] Graphical models, especially directed ones, are often called "Bayes nets" or "Bayesian networks", because this equation is, or can be seen as, a version of Bayes's rule. Since of course it follows directly from the definition of conditional probability, there is nothing distinctively Bayesian here — no subjective probability, or assigning probabilities to hypotheses.

[23] There is an extensive discussion of relevant methods in Jordan (1998).

[24] I am told that this is more like the idea of causation in Buddhism, as something like "co-dependent origination", than the asymmetric one which Europe and the Islamic world inherited from the Greeks (especially Aristotle), but you would really have to ask a philosopher about that.

tribution[25] when

$$p(X_1, X_2, \ldots X_p) \propto \prod_{c \in \text{cliques}(G)} \psi_c(X_{i \in c}) \tag{20.28}$$

That is, the joint distribution is a product of factors, one factor for each clique. Frequently, one introduces what are called **potential functions**, $U_c = \log \psi_c$, and then one has

$$p(X_1, X_2, \ldots X_p) \propto e^{-\sum_{c \in \text{cliques}(G)} U_i(X_{i \in c})} \tag{20.29}$$

The key correspondence is what is sometimes called the **Gibbs-Markov theorem**: a distribution is a Gibbs distribution with respect to a graph G if, and only if, it obeys the Markov property with neighbors defined according to G.[26].

In many practical situations, one combines the assumption of an undirected graphical model with the further assumption that the joint distribution of all the random variables is a multivariate Gaussian, giving a **Gaussian graphical model**. An important consequence of this assumption is that the graph can be "read off" from the inverse of the covariance matrix Σ, sometimes called the **precision matrix**. Specifically, there is an edge linking X_i to X_j if and only if $(\Sigma^{-1})_{ij} \neq 0$. (See Lauritzen (1996) for an extensive discussion.) These ideas sometimes still work for non-Gaussian distributions, when there is a natural way of transforming them to be Gaussian (Liu *et al.*, 2009), though it is unclear just how far that goes.

20.6.2 Directed but Cyclic Graphs

Much less work has been done on directed graphs with cycles. It is very hard to give these a causal interpretation, in the fashion described in the next chapter. Feedback processes are of course very common in nature and technology, and one might think to represent these as cycles in a graph. A model of a thermostat, for instance, might have variables for the set-point temperature, the temperature outside, how much the furnace runs, and the actual temperature inside, with a cycle between the latter two (Figure 20.7).

Thinking in this way is however simply sloppy. It always takes *some* time to traverse a feedback loop, and so the cycle really "unrolls" into an acyclic graph linking similar variables at *different* times (Figure 20.8). Sometimes[27], it is clear that when people draw a diagram like Figure 20.7, the incoming arrows really refer to

[25] After the American physicist and chemist J. W. Gibbs, who introduced such distributions as part of **statistical mechanics**, the theory of the large-scale patterns produced by huge numbers of small-scale interactions.

[26] This theorem was proved, in slightly different versions, under slightly different conditions, and by very different methods, more or less simultaneously by (alphabetically) Dobrushin, Griffeath, Grimmett, and Hammersley and Clifford, and almost proven by Ruelle. In the statistics literature, it has come to be called the "Hammersley-Clifford" theorem, for no particularly good reason. In my opinion, the clearest and most interesting version of the theorem is that of Griffeath (1976), an elementary exposition of which is given by Pollard (`http://www.stat.yale.edu/~pollard/Courses/251.spring04/Handouts/Hammersley-Clifford.pdf`). (On the other hand, Griffeath was one of my teachers, so discount accordingly.) Calling it the "Gibbs-Markov theorem" says more about the content, and is fairer to all concerned.

[27] As in Puccia and Levins (1985), and the `LoopAnalyst` package based on it (Dinno, 2009).

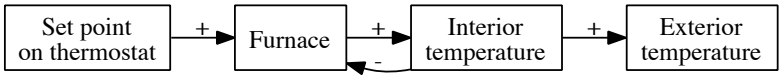

FIGURE 20.7: *Directed but cyclic graphical model of a feedback loop. Signs (+, − on arrows are "guides to the mind". Cf. Figure 20.8.*

the change, or rate of change, of the variable in question, so it is merely a visual short-hand for something like Figure 20.8.

Directed graphs with cycles are thus primarily useful when measurements are so slow or otherwise imprecise that feedback loops cannot be unrolled into the actual dynamical processes which implement them, and one is forced to hope that one can reason about equilibria instead[28]. If you insist on dealing with cyclic directed graphical models, see Richardson (1996); Lacerda *et al.* (2008) and references therein.

20.7 Further Reading

The paper collection Jordan (1998) is actually extremely good, unlike most collections of edited papers; Jordan and Sejnowski (2001) is also useful. Lauritzen (1996) is thorough but more mathematically demanding. The books by Spirtes *et al.* (1993, 2001) and by Pearl (1988, 2000, 2009b) are deservedly classics, especially for their treatment of causality, of which much more in Part IV. Glymour (2001) discusses applications to psychology.

While I have presented DAG models as an outgrowth of factor analysis, their historical ancestry is actually closer to the "path analysis" models introduced, starting around 1918, by the great geneticist and mathematical biologist Sewall Wright to analyze processes of development and genetics. Wright published his work in a series of papers which culminated in Wright (1934). That paper is now freely available online, and worth reading. (See also `http://www.ssc.wisc.edu/soc/class/soc952/Wright/wright_biblio.htm` for references to, and in some cases copies of, related papers by Wright.) Path analysis proved extremely influential in psychology and sociology. Loehlin (1992) is user-friendly, though aimed at psychologists who know less math anyone taking this course. Li (1975), while older, is very enthusiastic and has many interesting applications in biology. Moran (1961) is a very clear treatment of the mathematical foundations, extended by Wysocki (1992) to the case where each variable is itself multi-dimensional vector, so that path "coefficients" are themselves matrices.

Markov random fields where the graph is a regular lattice are used extensively in spatial statistics. Good introductory-level treatments are provided by Kindermann

[28] Economists are fond of doing so, generally without providing any rationale, based in economic theory, for supposing that equilibrium *is* a good approximation (Fisher, 1983, 2010).

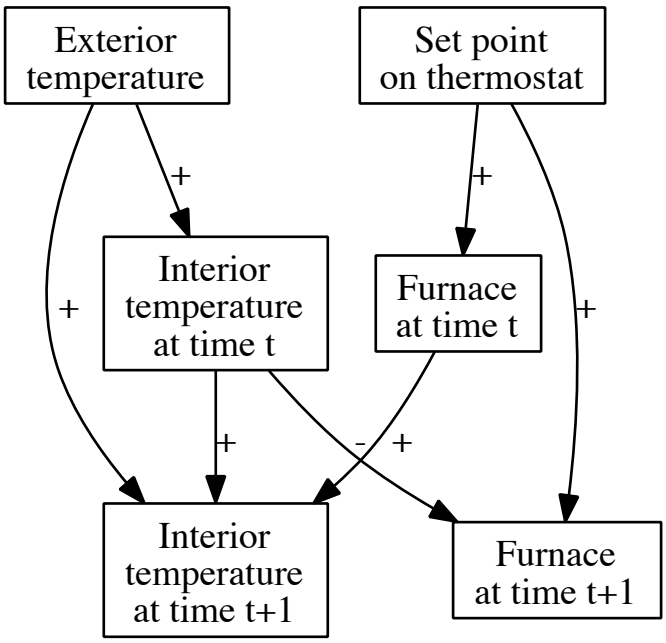

FIGURE 20.8: *Directed, acyclic graph for the situation in Figure 20.7, taking into account the fact that it takes time to traverse a feedback loop. One should imagine this repeating to times $t+2$, $t+3$, etc., and extending backwards to times $t-1$, $t-2$, etc., as well. Notice that there are no longer any cycles.*

and Snell (1980) (the full text of which is free online), and by Guttorp (1995), which also covers the associated statistical methods. Winkler (1995) is also good, but presumes more background in statistical theory. (I would recommend reading it after Guttorp.) Griffeath (1976), while presuming more probability theory on the part of the reader, is extremely clear and insightful, including what is simultaneously one of the deepest and most transparent proofs of the Gibbs-Markov theorem. Lauritzen (1996) is a mathematically rigorous treatment of graphical models from the viewpoint of theoretical statistics, covering both the directed and undirected cases.

If you are curious about Gibbs distributions in, so to speak, their natural habitat, the book by Sethna (2006), also free online, is the best introduction to statistical mechanics I have seen, and presumes very little knowledge of actual physics on the part of the reader. Honerkamp (2002) is less friendly, but tries harder to make connections to statistics. If you already know what an exponential family is, then Eq. 20.29 is probably extremely suggestive, and you should read Mandelbrot (1962).

On information theory (§20.4), the best book is Cover and Thomas (2006) by a large margin. References specifically on the connection between *causal* graphical models and information theory are given in Chapter 24.

20.8 Exercises

1. Find all the paths between the exogenous variables in Figure 20.4, and verify that every such path goes through at least one collider .

2. Is it true that in any DAG, every path between exogenous variables must go through at least one collider, or descendant of a collider? Either prove it or construct a counter-example in which it is not true. Does the answer change we say "go through at least one collider", rather than "collider or descendant of a collider"? .

3. (a) Take any two nodes, say X_1 and X_2, which are linked in a DAG by a path which does not go over colliders. Prove that there is a unique node along the path which is an ancestor of all other nodes on that path. (Note that this shared ancestor may in fact be X_1 or X_2.) *Hint:* do exercise 2.

   (b) Take any two nodes which are linked in a DAG by a path which remains open when conditioning on a set of variables S containing no colliders. Prove that for every open path between X_1 and X_2, there is a unique node along the path which is an ancestor of all other nodes on that path, and that this ancestor is not in S.

4. Prove that $X_2 \perp\!\!\!\perp X_3 | X_5$ in Figure 20.4.