Applications of Graphs in Statistical and Probabilistic Inference

Eric C. Anderson*

February 23, 2016

Abstract

This is a brief narrative that follows, roughly, my guest lecture to UCSC Math 115: Graph Theory, taught by Richard Montgomery, Winter 2016.

While this course will have steeped the students in the theoretical and algebraic aspects of graph theory, I will be providing a look into some applications of graphs in statistics and probability.

I enjoy talking about these applications of graph theory because it is not immediately obvious how graphs and probability might mix. By this, I mean that many engineering applications of graphs involve situations where the graphical representation is quite transparently a representation of a physical thing—for example paths between points that must be traversed by a traveling salesman (with the edge weights being distances), or routes upon which wire might be laid to provide an efficiently designed telephone network. In statistics, graphs are used to represent joint probability distributions and glean properties of those distributions on a local and a global scale. This is a little more abstract.

I am going to motivate this whole lecture with the example problem of computing probabilites of genetic data on pedigrees. We will consider just the simplest version of this problem—one genetic locus with two alleles, keeping in mind that things can get very hairy with multiple linked markers. We focus on pedigrees for a few reasons: 1) part of my own research currently involves the application of algorithms on graphs to infer unknown pedigrees using genetic data sampled from wild populations (of fish, birds, whales, you name it...); 2) the graphical structure of the problem follows the the lines of descent from parents to offspring, so it is somewhat tangible (it isn't a telephone wire network, but it is still somewhat tangible...); and 3) because the use of directed graphs in statistics has its origins in work on "path diagrams" by the famed geneticist Sewall Wright in the mid-1900's.

It is important to keep in mind, however, that these graphical concepts apply to probabilistic models and statistical inference much more widely than just to pedigrees. In fact, their general

*Fisheries Ecology Division, Southwest Fisheries Science Center, 110 Shaffer Road, Santa Cruz, CA 95060

utility in statistics began to be widely appreciated in the 1970's and it became clear that similar notions were in use in a wide variety of fields from statistical physics, to electrical engineering and computer science. In fact, one of the wonderful benefits of thinking about probability distributions in terms of graphs is that it has made it much easier for researchers in a wide variety of fields to appreciate just how similar many of the problems that they work on are, in terms of their essential probabilistic assumptions and calculations.

1 Overview

The lecture today will break down into the following sections:

- introduction to necessary genetics and probabilities and the idea of inference of genotypic state,
- the *acyclic directed* graphical model formalism and factorization into conditional probabilities,
- conditional independence and the *undirected graphical* model formalism, moralization, neighborhoods. (*Much, much more could be said here about Markov random fields, factorization over maximal clique potentials, graphical assessment of the computational complexity of inference, the treewidth of a graph, etc. but we will skip most of that.*)
- the *factor-graph* model formalism, the sum-product algorithm, and loopy belief propagation,
- pedigree inference.

This list of topics, treated in depth, could easily require a full quarter, so we will just be skimming the surface here. But what I hope that people will take home from this is an appreciation of the three different ways of representing probability distributions, graphically, and some of their applications, and the utility of graphical representations for algorithm development.

2 Simple Genetics and Probability

2.1 Basic genetics

With today's genetic technologies we can find a specific location in an individual's genome and read the DNA base (an A, C, G, or T) at exactly the same spot on a chromosome within or across individuals. Such a location is called a "locus" or a "SNP marker" and the specific base that is found there is called an *allele*. In a *diploid* there are two copies of the genome (one from the mother and one from the father) in an individual, which means that his/her *genotype* at the locus has two *allelic* values, like AA, or AG, or TT, *etc.* We typically do not know which copy came from the mother or the father, so we take the two allelic values in the genotype as *unordered*.

At most SNP markers, only two of the possible four DNA bases will be seen amongst all individuals in the population, thus, we can call one of the alleles ○ and the other one ●, and refer to the genotype as the number of ● alleles in the genotype. Thus genotypes can be 0, 1, or 2. For example, if we let ○ refer to the base A and ● refer to the base G, then the diploid genotypes can be named as follows

- AA = ○○ = 0
- AG or GA = ○● or ●○ = 1
- GG = ●● = 2

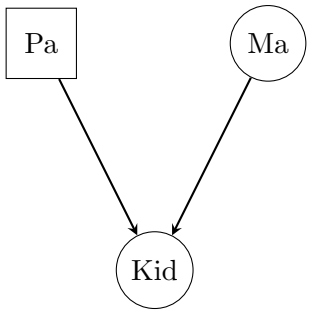
If you were to grab an individual from a population, the probability of his genotype, Y, can be determined by the frequency of the alleles in the population. Let q be the relative frequency of the ● allele. Then the genotype frequencies (assuming that they are drawn, independently, from the population into the individual) are

$$\begin{aligned} P(Y=0) &= (1-q)^2 \\ P(Y=1) &= 2q(1-q) \\ P(Y=2) &= q^2 \end{aligned}$$

2.2 Mendelian inheritance and conditional probabilities

Gregor Mendel (a monk in the 1800's with a penchant for growing and breeding pea plants, sipping tea, making observations, and thinking deeply about them) showed that when a diploid mother and father produce a child, each parent passes only one copy of a locus to its child. The copy that gets passed on is chosen randomly (probability $\frac{1}{2}$) within each parent and is independent of the copy that gets passed on by the other parent.

We can draw a picture of the production of an offspring like so:



This is a simple pedigree. Males are represented as squares and females as circles and lines of descent are depicted by directed edges. You might also recognize it as directed graph that has no directed cycles.

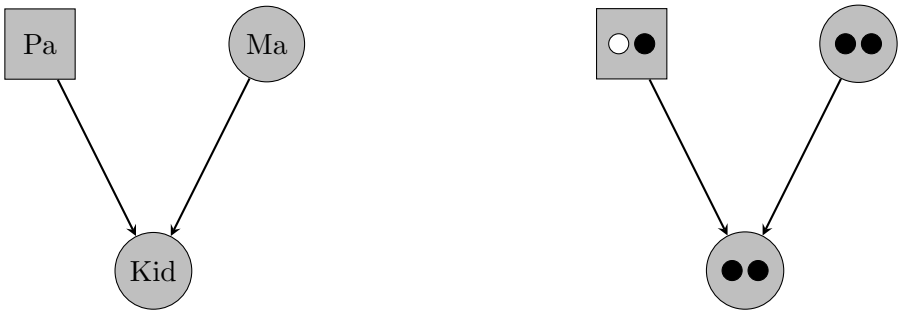
A natural thing to think about in this case is the *conditional probability* of Kid's genotype given the genotypes of its parents, *i.e.*, $P(Y_{\text{kid}}|Y_{\text{pa}}, Y_{\text{ma}})$. These probabilities are easily worked out from Mendel's laws. There are $3^3 = 27$ such conditional probabilities. Below are 9 of them for the case where $Y_{\text{pa}} = 1$, *i.e.*, ○●.

$$P(Y_{\text{kid}}|Y_{\text{pa}} = 1\,\bigcirc\!\!\bullet, Y_{\text{ma}}) = \begin{array}{c} Y_{\text{kid}}\downarrow \\ 0\,\bigcirc\!\bigcirc \\ 1\,\bigcirc\!\!\bullet \\ 2\,\bullet\!\bullet \end{array} \begin{array}{c} Y_{\text{ma}}\rightarrow \\ \\ \\ \\ \end{array} \begin{array}{ccc} 0\,\bigcirc\!\bigcirc & 1\,\bigcirc\!\!\bullet & 2\,\bullet\!\bullet \\ \hline \\ \end{array} \left(\begin{array}{ccc} \frac{1}{2} & \frac{1}{4} & 0 \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ 0 & \frac{1}{4} & \frac{1}{2} \end{array}\right)$$

Wow! That is super simple. We are just dealing with probabilities of 0, $\frac{1}{2}$, $\frac{1}{4}$; when does this get challenging? That is the fun part of these sorts of problems—when you build up lots of simple probabilities in a complex web of probabilistic dependence things can eventually get hairy.

2.3 Observed variables, joint probability

Here, let us introduce a little notational twist on the pedigree above: we will shade the vertices that are observed. In other words, in the pedigrees below the shading indicates we have observed whether the genotype of each individual is 0, 1, or 2. The pedigree on the right actually shows the observed genotypes.



If you are a statistician, one of the things you will love to do is to calculate the probability of all your observed data, as doing so forms the basis for using your data to learn about things that you might not have observed—the process called *inference* (more on that later).

In the right-hand pedigree above, the probability of all three individuals is a *joint* probability, which simply means it is a probability of two or more things (*i.e.*, random variables), considered together. We can write a joint probability like this $P(X_{\text{pa}} = 1, X_{\text{ma}} = 2, X_{\text{kid}} = 2)$. But how to compute it? Well, you can think about how the data would have been generated:

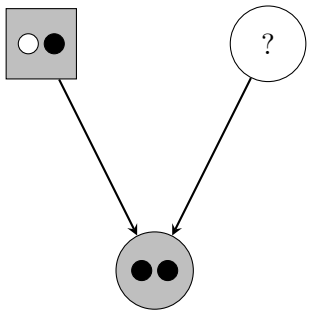
1. First Pa must have genotype 1, and Ma must have genotype 2. This happens with probabilities $2q(1-q)$ and q^2. If the parents' genotypes are independent (as they are) then the probability of both is just their product.
2. Then those two parents must have a kid with a genotype of 2, which happens with probability $\frac{1}{2}$.

So,

$$\begin{aligned} P(X_{\text{pa}} = 1, X_{\text{ma}} = 2, X_{\text{kid}} = 2) &= P(X_{\text{pa}} = 1)P(X_{\text{ma}} = 2)P(X_{\text{kid}} = 2|X_{\text{pa}} = 1, X_{\text{ma}} = 2) \\ &= 2q(1-q) \cdot q^2 \cdot \frac{1}{2} \end{aligned}$$

2.4 Inference, a simple example

Imagine that you have observed the genotype of Pa and Kid, but not Ma,



...so you would like to use all the information in the above figure to *infer* (as best you can) the genotype of Ma. We have two sources of information here:

1. If we never looked at Kid and Pa, but rather just plucked Ma out of the population we know that the probability of her genotype would be $(1-q)^2$ if ○○, $2q(1-q)$ if ○● or ●○, or q^2 if $(1-q)^2$ if ●●.
2. If we just focus on Ma's relationship with Kid and Pa, we can evaluate the evidence in favor of Ma's genotype by the relative size of $P(Y_{\text{kid}} = 2|Y_{\text{pa}} = 1, Y_{\text{ma}})$ considered as a function of Y_{ma}. This is sometimes called the "likelihood."

These two pieces of evidence are combined together using *Bayes' law* to compute the posterior probability. If $q = 0.6$ then the posterior probability of Ma's genotype is 0 for ○○, 0.4 for ●○ or ○●, and 0.6 for ●●. Left as an exercise if it is not already familiar to you.

Inference about an individual genotype in a pedigree (vertex in a graph) can be seen as a process of computing probabilites for an unobserved genotype (vertex in a graph) given that information from the observed genotypes (vertices in a graph) on the pedigree. This has many uses in biology, ecology, medicine, *etc.*

3 Acyclic directed graphs

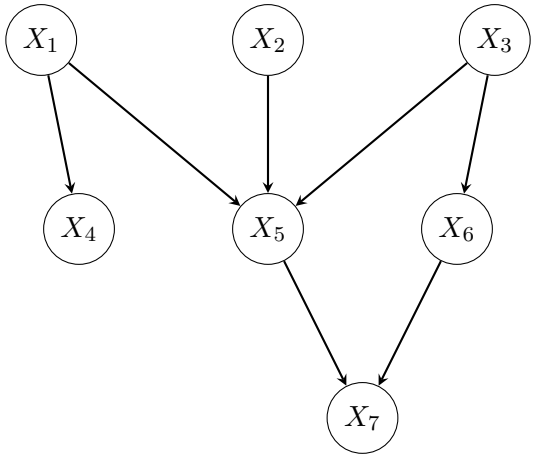
The pedigree above is an example of an *acyclic directed* graph, often called a DAG for short. The wonderful thing about DAGs: if you have a joint probability distribution that factorizes into a *product of conditional probabilities*, you can represent the factorization of the distribution as a DAG.

In general, a joint distribution that factorizes according to a DAG with vertices $X_1, \ldots, X_K$ representing random variables can be written as the product:

$$P(X_1, \ldots, X_K) = \prod_{i=1}^{K} P(X_i | \text{pa}(X_i))$$

where $\text{pa}(X_i)$ denotes the *parents* of X_i in the DAG, and we follow the convention that if $\text{pa}(X_i) = \emptyset$ then $P(X_i|\text{pa}(X_i)) = P(X_i)$—simply the "prior" probability of X_i.

For example, take a DAG that is not a pedigree:



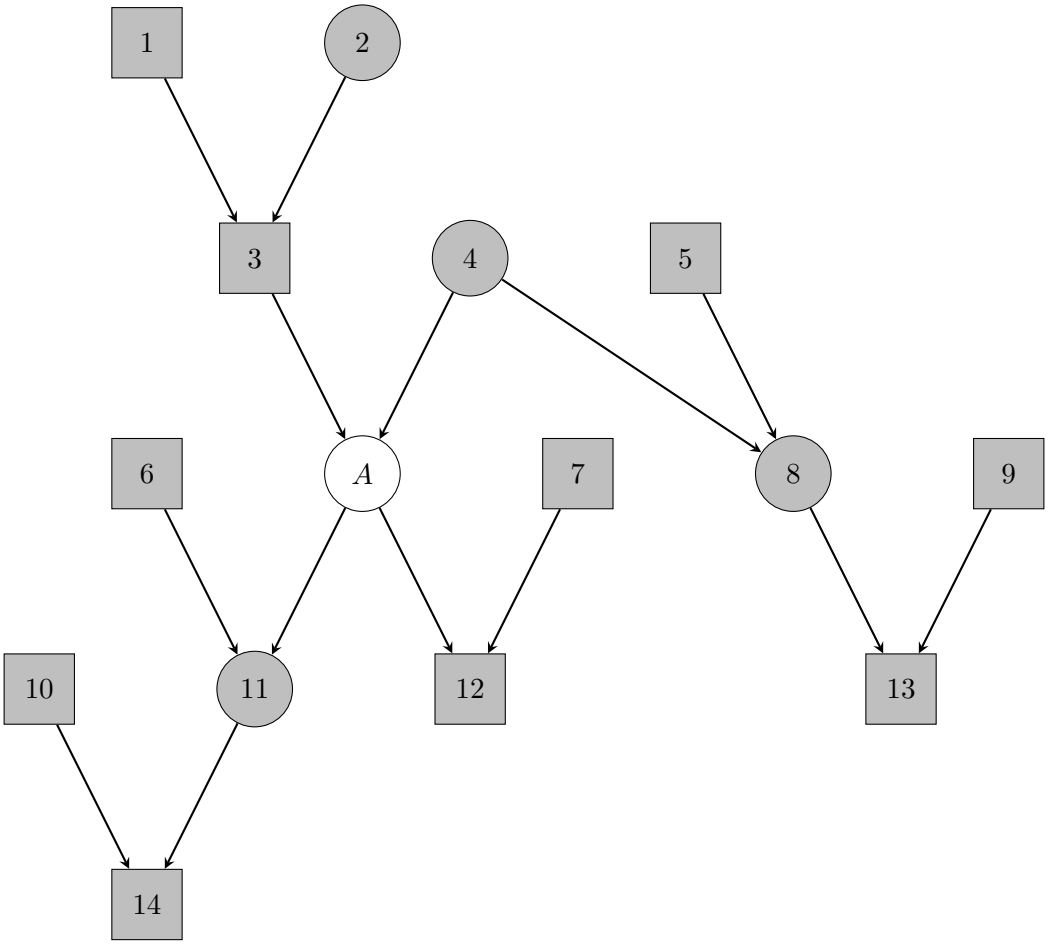
Writing down the factorization of a distribution that respects the above graph is left as an exercise. It is worth noting that, though the above graph contains a cycle, it is not a directed cycle, so it is still a DAG.

3.1 What good are DAGs?

This is a fair question. It appears to me that the primary value of DAGs in the field of probabilistic graphical models is that they can be used to represent the factorization of joint probability distributions into conditional probabilities. This makes a very nice way of visualizing the structure of complex probability models and especially of *Bayesian hierarchical models*. But beyond that, when it comes to tasks such as graphically assessing conditional independence, or designing algorithms for inference, other types of related graphs provide more suitable and useful representations. We illustrate this with the example in the following section.

3.2 Which relatives really matter?

Let us consider the pedigree below and ask ourselves the question, "If we want to do inference on the genotype of individual A given the observed genotypes of individuals 1–15, which of those observed individuals can we safely ignore and which must we include in our calculations?



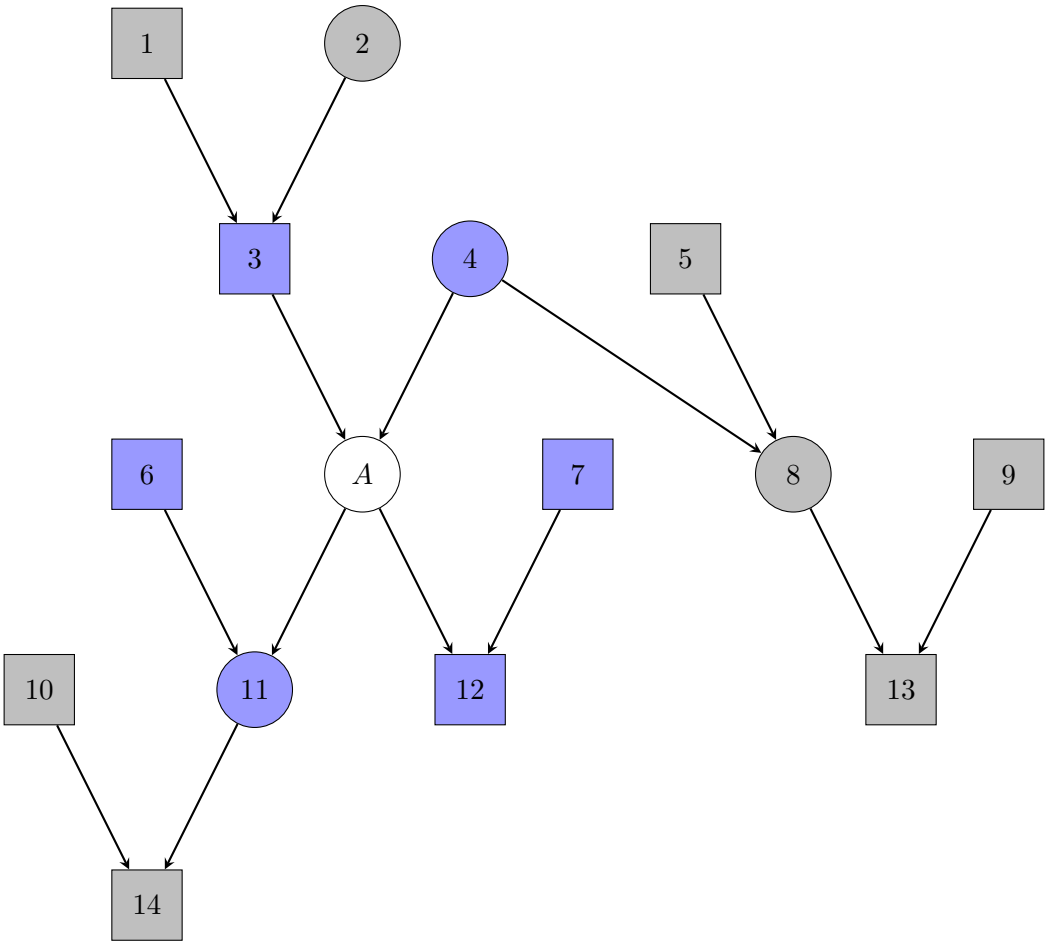
When doing computation or statistics, being able to ignore *irrelevant* data is always advantageous, but you don't want to toss out relevant data. So, some way of assessing which individual genotypes are relevant to inference of A's genotype is in order.

It is relatively easy to figure this out from the factorized joint density—any variables that appear (on either side of the "|") in a conditional probability density with Y_A are relevant. For example, the full joint probability of the pedigree above can be written:

$$\begin{aligned} P(\text{all}) &= P(Y_1)P(Y_2)P(Y_3|Y_1,Y_2)P(Y_4)P(Y_5) \\ &\times P(Y_6)P(Y_A|Y_3,Y_4)P(Y_7)P(Y_8|Y_4,Y_5)P(Y_9) \\ &\times P(Y_{10})P(Y_{11}|Y_6,Y_A)P(Y_{12}|Y_A,Y_7)P(Y_{13}|Y_8,Y_9) \\ &\times P(Y_{14}|Y_{10},Y_{11}) \end{aligned}$$

The only variables that can tell us something about Y_A (at least if everything is observed except for Y_A) are those that are muddled up inside conditional probabilities with Y_A in the joint probability. They are $(Y_3, Y_4, Y_6, Y_7, Y_{11}, Y_{12})$.

If you are excited about graphical representations of distributions you might like a nice, tidy, graphical way of visualizing the relatives who matter, perhaps by *adjacency* in the graph. Alas! That doesn't work with this DAG (or, in general, any DAG). Behold, below, that the relevant relatives are not actually all adjacent to A in the directed graph.



Specifically, there are not edges between A and 6 nor A and 7.

However, there is another class of graphical models used to represent probability distributions called *undirected* graphical models. These grew out of work in statistical physics where people were experimenting with trying to build up valid global probability distributions for many interacting particles by using models that only specified the local interactions between neighboring particles[1]. There is also a simple procedure called *moralization* to find the unique undirected graph associated with a DAG.

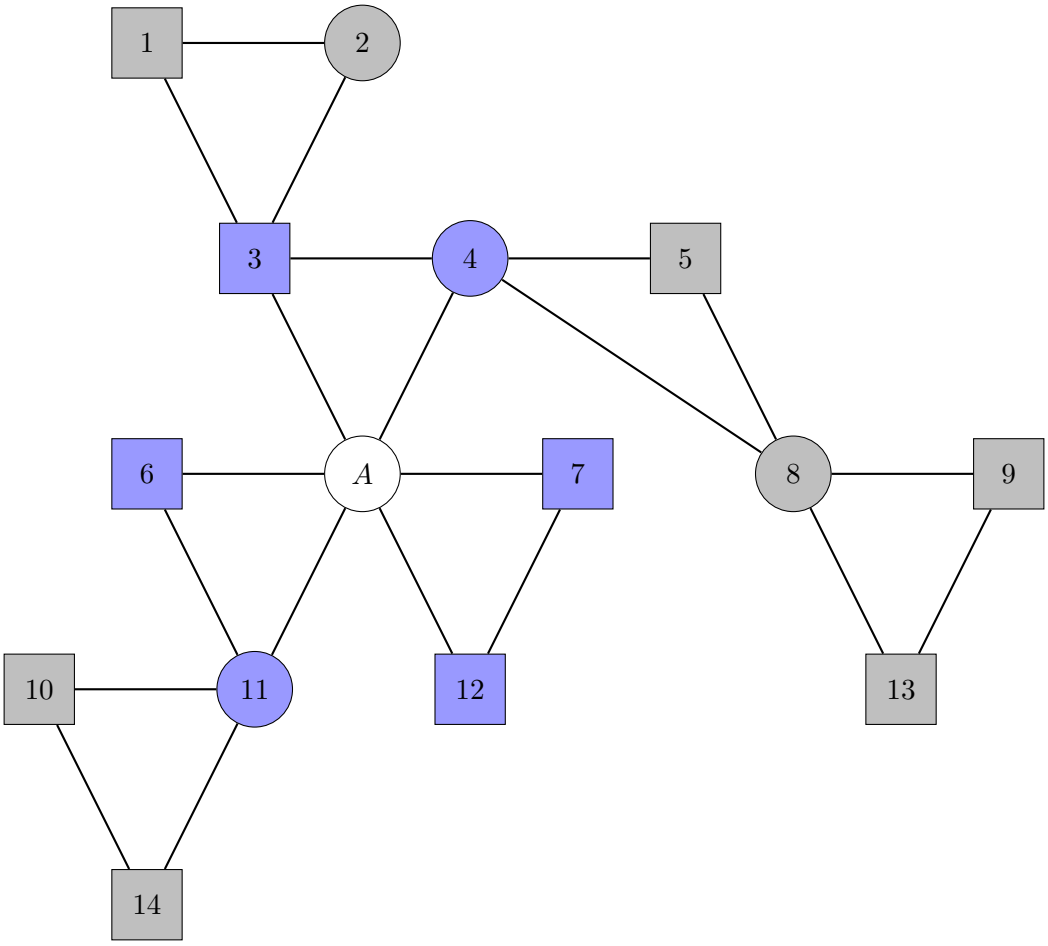
[1]We won't go too much into this area, but if you were interested in it, the key words are *Markov random field*, *Hammersley-Clifford theorem*, *Gibbs distributions*, and *Ising models*.

3.3 The moralized undirected representation of a DAG

Someone with a good sense of humor (or with a Puritan vision of romantic relations) named the process of converting a DAG into its associated undirected graph "moralization" because it involves first "marrying" parents that are "unmarried" in the DAG. In this case "unmarried" means that they are not connected together by an edge in the graph. Thus the two steps in moralizing a DAG are:

1. "marrying" unmarried parents by drawing undirected edges between all pairs of parents of the same vertex in the DAG.
2. converting all the directed edges in the original DAG into undirected edges.

This gives us a graph like the one below:



Notice that in this graph, all the blue vertices are indeed adjacent to A. The neighbors of any vertex V in an undirected graphical model are sometimes called the *Markov blanket* of V. The adjacency properties of an undirected graph determine a whole set of *Markov properties* that dictate

conditional independence relations between variables. V is conditionally independent of everything else in the graph given its Markov blanket.

4 Undirected graphical models

4.1 Factorization

Like DAGs, undirected graphical models represent factorizations of probability distributions but do so in terms of a product of *potentials* over *cliques* rather than a product over vertices of conditional probabilities. Additionally, the joint probabilities represented by undirected graphs are often unnormalized (*i.e.*, the normalizing constant is not known). A joint probability on variables $Y_1, \ldots, Y_K$ that respect an undirected graph G in which the vertices are $Y_1, \ldots, Y_K$ factorizes according to:

$$P(Y_1, \ldots, Y_K) = \frac{1}{Z} \prod_{C \in \mathrm{cl}(G)} \psi_C(X_C)$$

where $\mathrm{cl}(G)$ is the set of all cliques in G, X_C is a subset of variables $Y_1, \ldots, Y_K$ that form the clique C, ψ_C is a potential—a nonnegative function defined on X_C—and Z is a (possibly unknown) normalizing constant. This also holds when $\mathrm{cl}(G)$ is the set of all *maximal cliques*. We will leave it to the reader (and it is fairly clear) that since the cliques are Ma-Pa-Kid trios, the potentials can be defined to be the conditional probablities as well as any priors on individuals that have no parents in the pedigree, thus giving us the same factorization as we had with the directed graph.

4.2 Elimination algorithm, treewidth, etc.

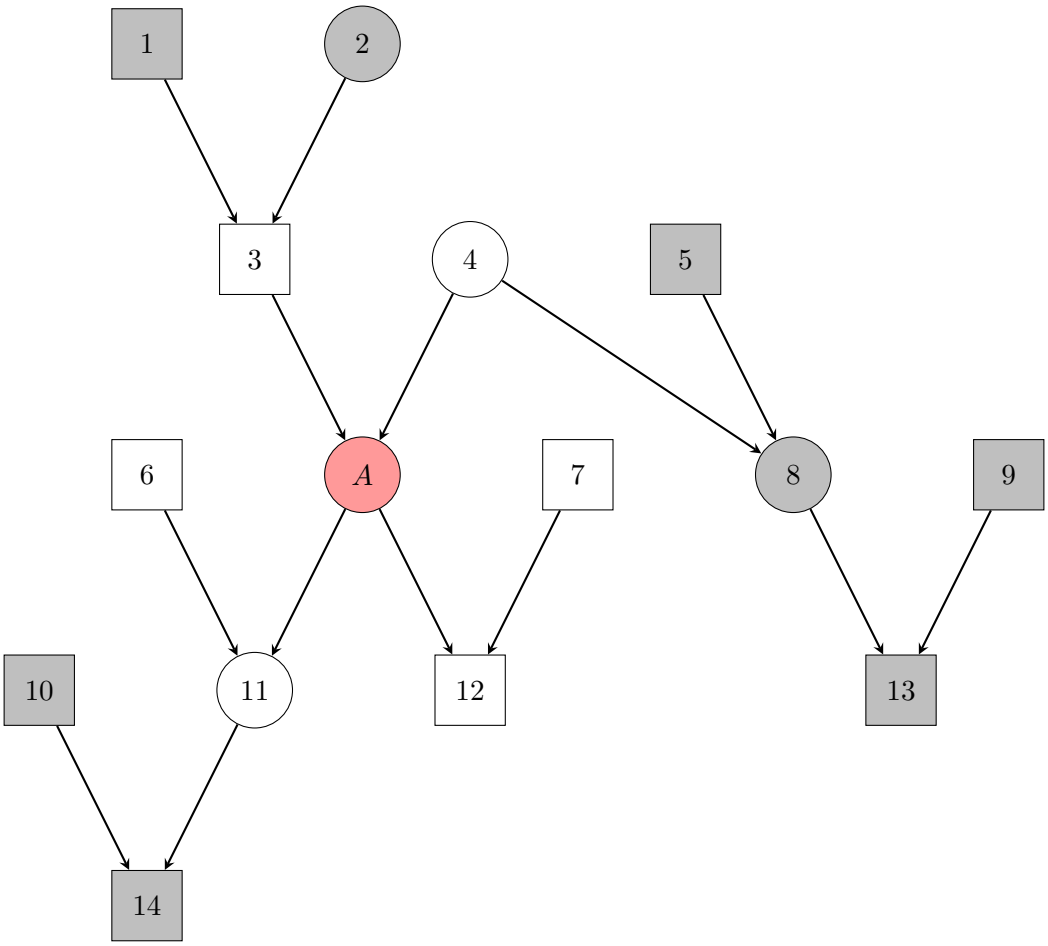
Undirected graphs play a useful role in understanding how computationally difficult it might be to do exact inference for an unobserved vertex in a graph. We won't be able to say much about that now, but for those interested, keywords are *elimination algorithm*, *tree decomposition of a graph*, *chordal graphs*, *treewidth*, and *junction tree algorithm*.

5 Inference in the presence of latent variables

A common problem in statistics involves the sitution where the joint probability of all the observed data cannot be easily computed because some data are missing. These are often called *missing data* problems or *latent variable* problems, and they can be understood graphically merely by the presence of unobserved variables in the graph, that lie between the observed variables and the one(s) you wish to make inference about.

To take a concrete example, let's go back to our directed graph with individual A, but consider what happens when individuals in A's Markov blanket are not actually observed. Note that, now,

A's more distant relatives are no longer irrelevant in inferring Y_A! The DAG is shown below with A highlighted in red.



Now, how do we do inference for Y_A, *i.e.*, how should we go about computing

$$P(Y_A|Y_1, Y_2, Y_5, Y_8, Y_9, Y_{10}, Y_{13}, Y_{14})?$$

The brute force method involves simply using the law of total probability [in other words, $P(A, B) = \sum_C P(A, B, C)$] to blindly sum the missing variables out of the joint distribution:

$$\sum_{Y_3}\sum_{Y_4}\sum_{Y_6}\sum_{Y_7}\sum_{Y_{11}}\sum_{Y_{12}} P(Y_A, Y_1, \ldots, Y_{14})$$

after which you can normalize that to sum to one over the three states Y_A can take and Voila! you have the conditional distribution you desire. The brute force method, however, is horribly inefficient, involving $3^6 = 729$ terms even though some parts of the sum don't vary when other parts do owing to the way that $P(Y_A, Y_1, \ldots, Y_{14})$ factorizes. Pursuing the brute force method would be akin to evaluating the sum

$$\sum_{x=1}^{1000}\sum_{y=1}^{1000} f(x)g(y)$$

by actually summing together 1,000,000 terms instead of noting that you could do it by summing 2,000 terms along with one multiplication:

$$\left(\sum_{x=1}^{1000} f(x)\right)\left(\sum_{y=1}^{1000} g(y)\right)$$

The key ingredient to figuring out how to do these types of sums efficiently on joint probabilities is the factorization of the joint probability. So, as you might expect, a graphical model interpretation can be useful here. In fact, one of the clearest ways of expressing these types of operations relies on another kind of graph that explicitly shows the factorization of a joint density (or any function). This type of graph (last kind of graph for today. . . I promise) is called a *factor graph*.

5.1 Factor graphs

A factor graph is a *bipartite* graph with two classes of nodes: *variable nodes* which represent variables (the same that vertices in a DAG do) and *factor nodes* which represent functions.

This type of graph forms the basis for the *sum-product algorithm* and *belief propagation*, both loopy and otherwise. From here on our we will be working from slides I made for a separate talk on the topic.