Applications of graphs in Statistical and Probabilistic Inference

Eric C. Anderson

Fisheries Ecology Division
Southwest Fisheries Science Center
Santa Cruz, CA
USA

Guest lecture in Richard Montgomery's course

UCSC Math 115, Graph Theory

24 February 2016

Overview

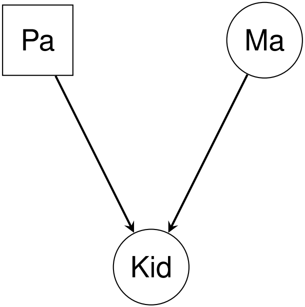
- Genetics and pedigrees
- DAGs, factorization, the idea of inference
- Undirected graphs (brief)
- Factor graphs and the sum-product algorithm
- Inference of pedigrees (brief)

Genotype nomenclature and probabilities

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

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

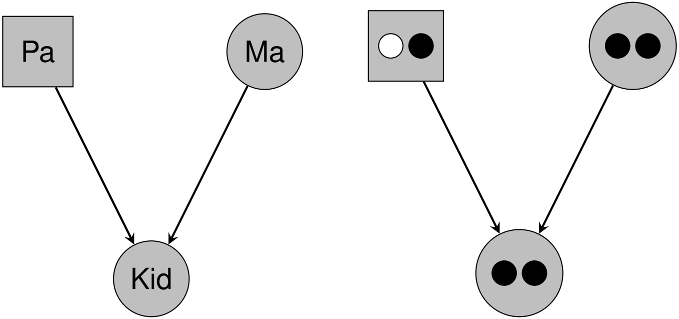
Simple 3-person pedigree



Conditional probabilities of inheritance

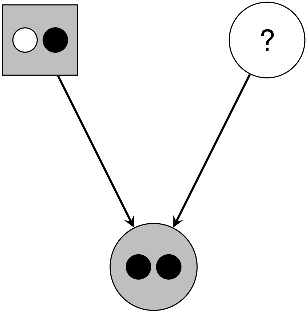
$$P(Y_{\text{kid}} | Y_{\text{pa}} = 1\ \text{○●}, Y_{\text{ma}}) = \begin{array}{c|ccc} Y_{\text{kid}} \downarrow \quad Y_{\text{ma}} \rightarrow & 0\ \text{○○} & 1\ \text{○●} & 2\ \text{●●} \\ \hline 0\ \text{○○} & \frac{1}{2} & \frac{1}{4} & 0 \\ 1\ \text{○●} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ 2\ \text{●●} & 0 & \frac{1}{4} & \frac{1}{2} \end{array}$$

Computing joint probabilities



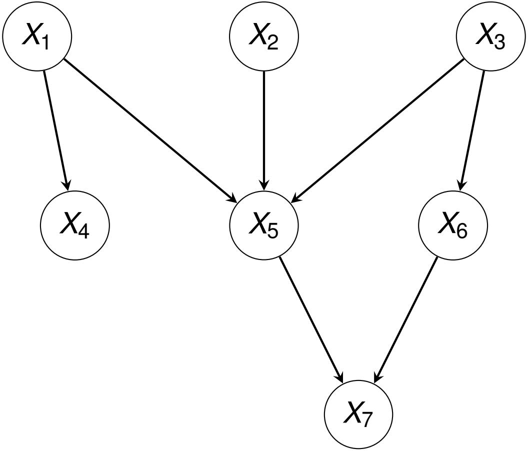
Inference, a simple example

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.

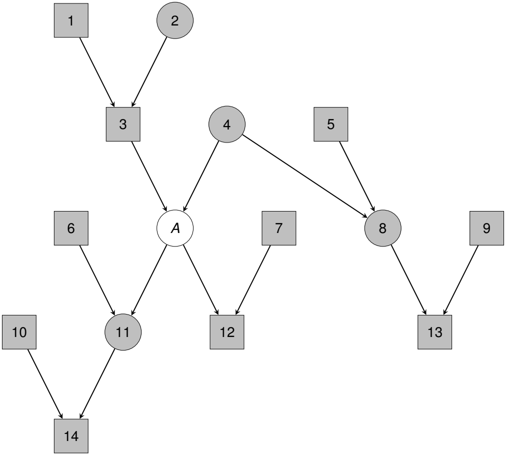
A DAG that is not a pedigree



Writing down the factorization of a distribution that respects the above graph is left as an exercise.

Which relatives really matter?

Imagine you wish to infer Y_A given everyone in the pedigree. Whose genotypes can you ignore?



Which relatives really matter?

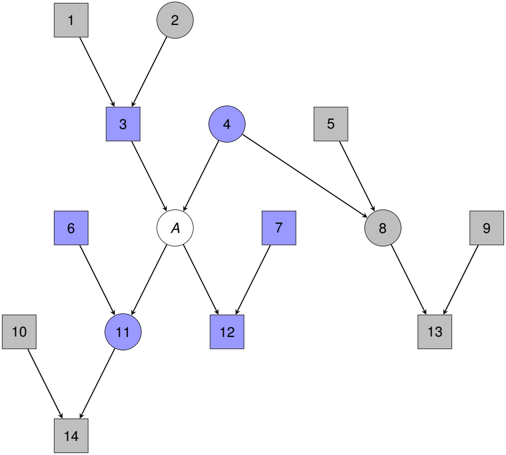
Look at who A is muddled up with in the terms of the joint probability:

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

They are $(Y_3, Y_4, Y_6, Y_7, Y_{11}, Y_{12})$.

Which relatives really matter?

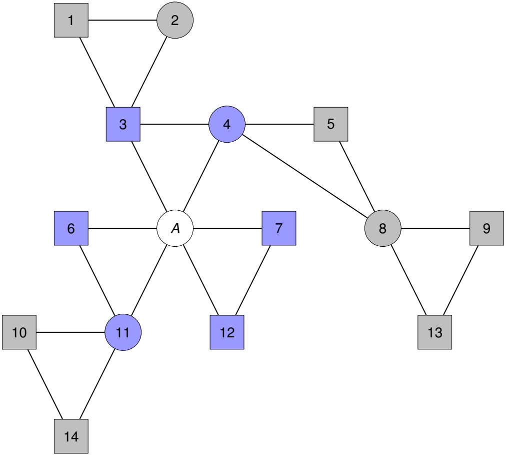
Behold! The relevant relatives are *not* all adjacent to A in the directed graph.



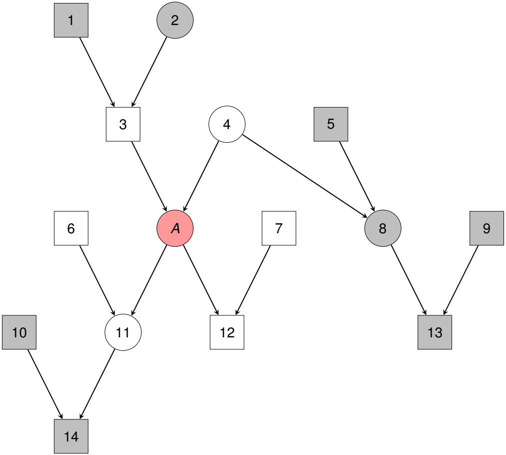
- Is this a moral question?

These are the people in your neighborhood

The *moralized undirected graph* associated with the DAG represents the *Markov blanket* of a vertex via adjacency.



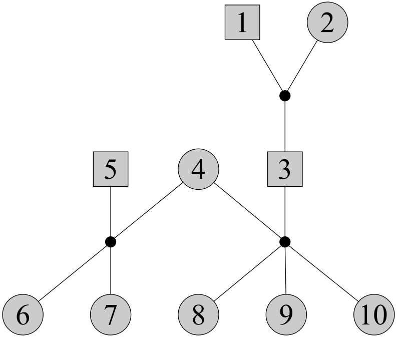
Inference with latent variables



How should we go about computing $P(Y_A | Y_1, Y_2, Y_5, Y_8, Y_9, Y_{10}, Y_{13}, Y_{14})$?

Genetic data on a simple pedigree

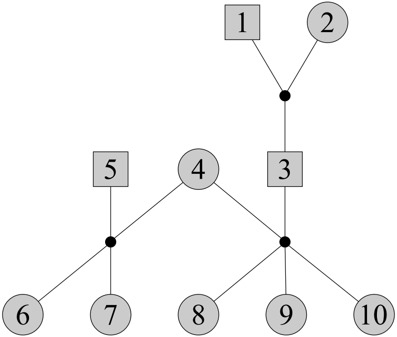
Genotypes $y = (y_1, \ldots, y_{10})$ observed with error, ϵ, from the true genotypes $x = (x_1, \ldots, x_{10})$. Founders' x_i drawn from allele frequencies θ.



$$
\begin{aligned}
p(x, y) = &\\
\times \quad & p(x_1|\theta)p(x_2|\theta)p(x_4|\theta)p(x_5|\theta)\\
\times \quad & p(x_3|x_1, x_2)p(x_6|x_5, x_4)p(x_7|x_5, x_4)\\
\times \quad & p(x_8|x_3, x_4)p(x_9|x_3, x_4)p(x_{10}|x_3, x_4)\\
\times \quad & \prod_{i=1}^{10} p(y_i|x_i, \epsilon)
\end{aligned}
$$

Genetic data on a simple pedigree

Genotypes $y = (y_1, \ldots, y_{10})$ observed with error, ϵ, from the true genotypes $x = (x_1, \ldots, x_{10})$. Founders' x_i drawn from allele frequencies θ.



$$p(x, y) =$$

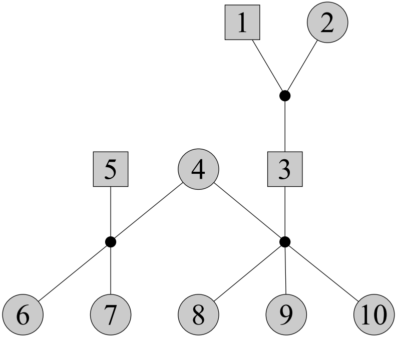
$$\times \quad p(x_1|\theta)p(x_2|\theta)p(x_4|\theta)p(x_5|\theta)$$

$$\times \quad p(x_3|x_1, x_2)p(x_6|x_5, x_4)p(x_7|x_5, x_4)$$
$$\times \quad p(x_8|x_3, x_4)p(x_9|x_3, x_4)p(x_{10}|x_3, x_4)$$

$$\times \quad \prod_{i=1}^{10} p(y_i|x_i, \epsilon)$$

Genetic data on a simple pedigree

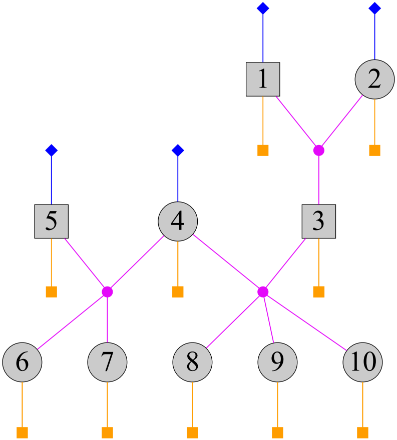
These probabilities fall into three different classes of functions of the x_i's: $f_p(x_i)$, $f_g(x_i)$, and $f_m(x_{\mathrm{pa}}, x_{\mathrm{ma}}, x_{\mathrm{kid},1} \dots, x_{\mathrm{kid},n})$, in which θ and y and ϵ are implicit (and fixed).



$$
\begin{aligned}
p(x,y) = & \\
\times & \quad f_p(x_1)f_p(x_2)f_p(x_4)f_p(x_5) \\
\times & \quad f_m(x_1,x_2,x_3)f_m(x_5,x_4,x_6,x_7) \\
\times & \quad f_m(x_3,x_4,x_8,x_9,x_{10}) \\
\times & \quad \prod_{i=1}^{10} f_g(x_i)
\end{aligned}
$$

A factor graph representation

$p(x,y)$ factorizes into a product over *factor nodes* of functions whose arguments are the adjacent *variable nodes*.



$$p(x,y) =$$

$$\times \quad f_p(x_1)f_p(x_2)f_p(x_4)f_p(x_5)$$

$$\times \quad f_m(x_1,x_2,x_3)f_m(x_5,x_4,x_6,x_7)$$

$$\times \quad f_m(x_3,x_4,x_8,x_9,x_{10})$$

$$\times \quad \prod_{i=1}^{10} f_g(x_i)$$

Sum Product Algorithm on *Singly-connected* Factor Graphs

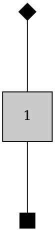
Kschischang *et al.* (2001) *IEEE Transactions on Information Theory*

- A message passing algorithm for the (marginal) conditional distribution of each x_i given all the y_i's.
  - Messages are potential functions of the individual *variable nodes* to or from which the messages are being sent.
  - Scheduling: a node can send an outgoing message on edge i only when the node has no other edges (apart from i) that have *not* received an incoming message.
  - An outgoing message from variable node v to factor node t on edge j is a simple product of all the incoming messages to v on edges other than j.
  - An outgoing message from factor node t to variable node v on edge i is the marginal of v given the function associated with t weighted by all incoming messages on edges other than i.

Bayes' Theorem via the Sum-Product Algorithm

A single SNP in a single diploid individual

$$\begin{aligned} f_p(x_1 = 0) &= 0.36 \\ f_p(x_2 = 1) &= 0.48 \\ f_p(x_3 = 2) &= 0.16 \end{aligned}$$

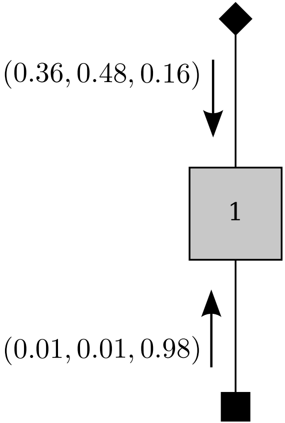


$$\begin{aligned} f_g(x_1 = 0) &= 0.01 \\ f_g(x_2 = 1) &= 0.01 \\ f_g(x_3 = 2) &= 0.98 \end{aligned}$$

- Consider a single SNP with two alleles, 0 and 1, with $\theta = q = 0.4$
- Hence three possible genotypes, 0, 1, and 2, with *a priori* probability $0.36, 0.48, 0.16$.
- Observe $y_1 = 2$, but allow a 2% chance of genotyping error that is equally likely to yield either of the two remaining genotypes, in error.

Bayes' Theorem via the Sum-Product Algorithm

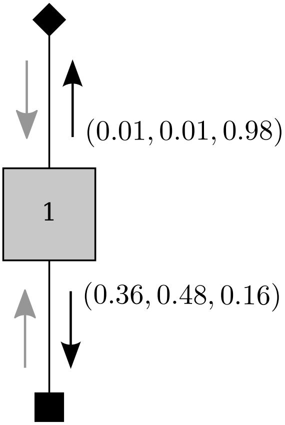
Sending the first round of messages



- The two factor nodes have only one edge, each
- Hence, they have no other edges with no incoming messages.
- So, they can send their messages to the variable node.

Bayes' Theorem via the Sum-Product Algorithm

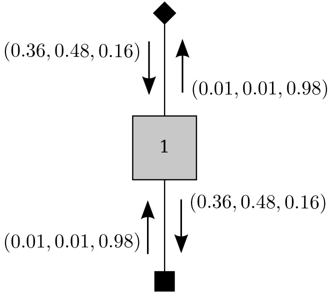
Sending the second round of messages



- Now, for each edge connected to the variable node, there are no other edges that have not received incoming messages.
- Hence the variable node can send outgoing messages on each edge.
- The message sent is the product of the incoming messages on all the other edges.

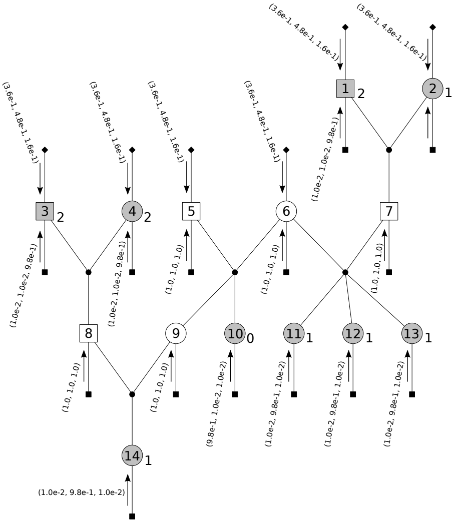
Bayes’ Theorem via the Sum-Product Algorithm

Computing quantities using those messages

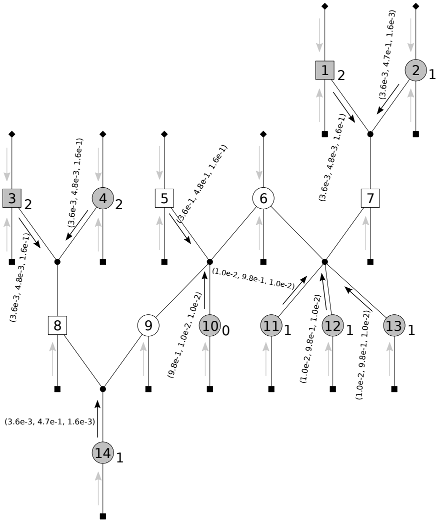


- Each edge has two messages going in different directions.
- The product of the two messages on an edge connected to a variable node:
  1. Gives the joint probability of x_i and y
  2. Normalizes to the probability of x_i given all the observed data and the allele freq in the population: $\approx (0.02, 0.03, 0.95)$

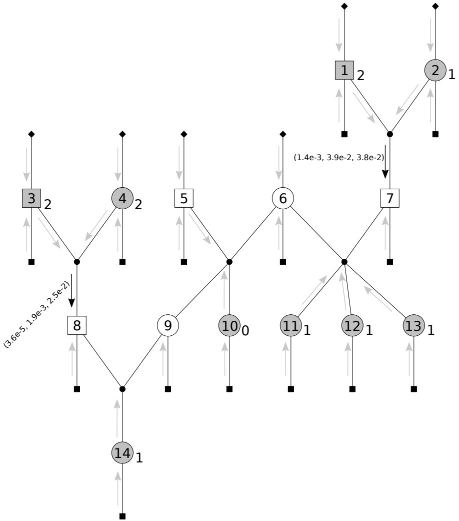
Sum-Product Algorithm on a Larger Pedigree



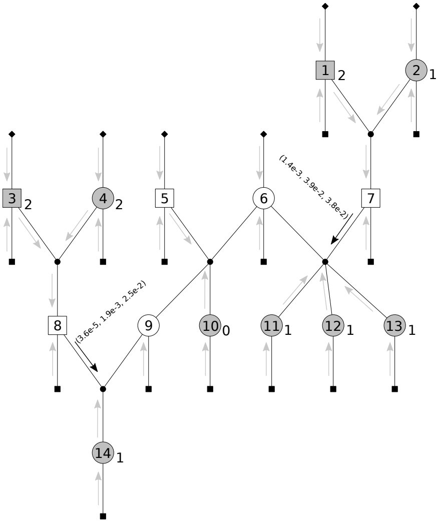
Sum-Product Algorithm on a Larger Pedigree



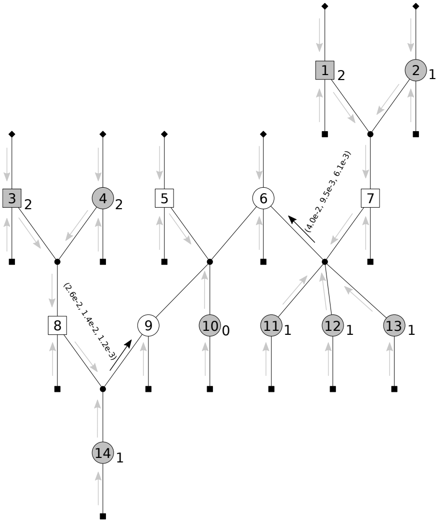
Sum-Product Algorithm on a Larger Pedigree



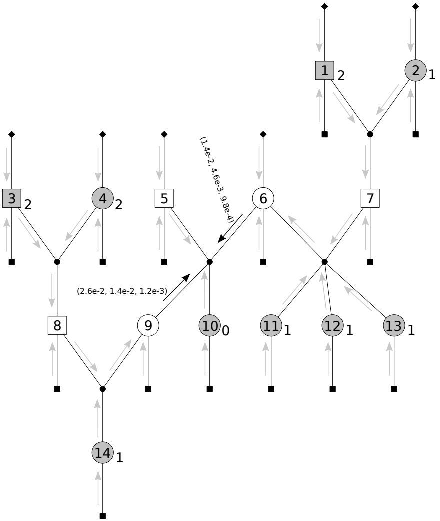
Sum-Product Algorithm on a Larger Pedigree



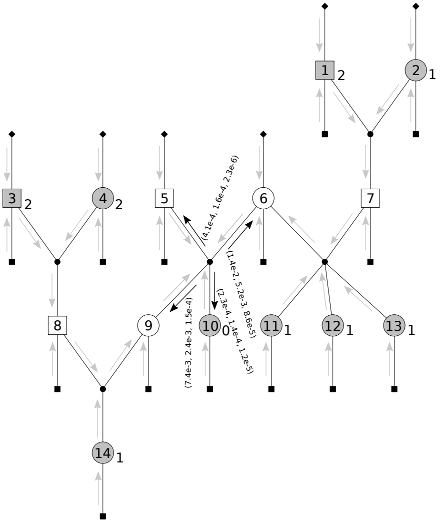
Sum-Product Algorithm on a Larger Pedigree



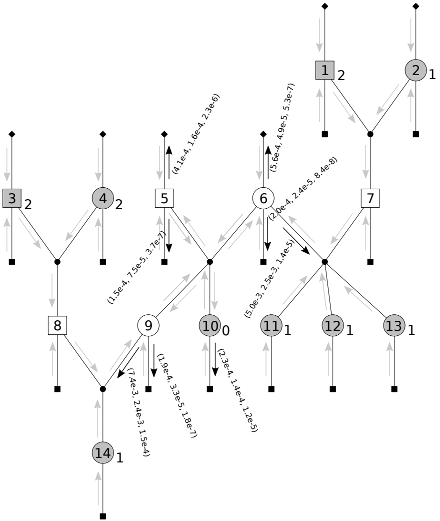
Sum-Product Algorithm on a Larger Pedigree



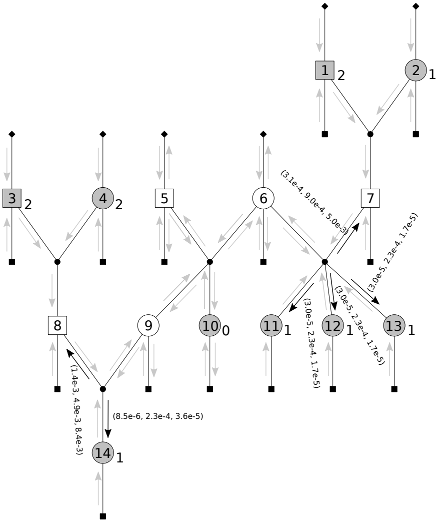
Sum-Product Algorithm on a Larger Pedigree



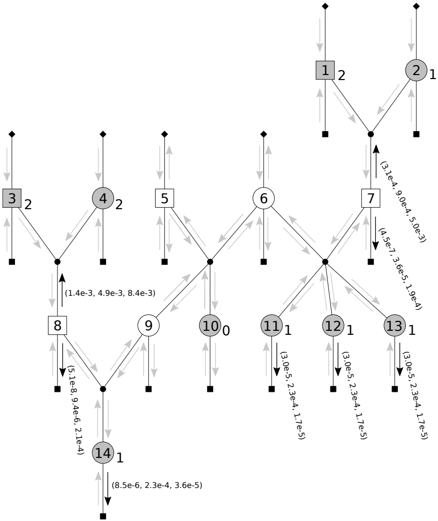
Sum-Product Algorithm on a Larger Pedigree



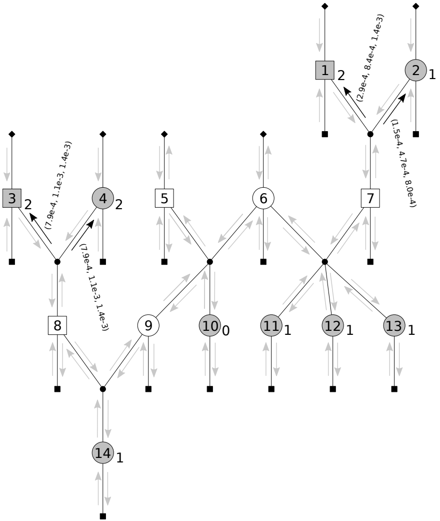
Sum-Product Algorithm on a Larger Pedigree



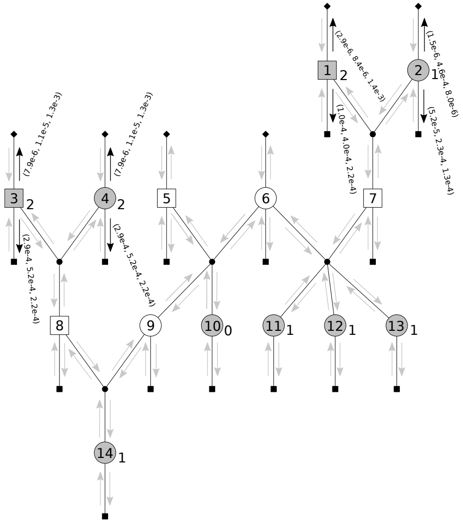
Sum-Product Algorithm on a Larger Pedigree



Sum-Product Algorithm on a Larger Pedigree



Sum-Product Algorithm on a Larger Pedigree



At the end of all that

- There are two messages in opposing directions along each edge
- The dot product of those two messages, on any edge, gives the joint probability of all the observed data.
- The normalized componentwise product of any to messages along an edge is the marginal conditional probability of the variable associated with the variable node to which the edge is incident.
- These relationships are used in Anderson & Ng (2016) to design and efficient Markov chain Monte Carlo sampler over the space of pedigrees to infer pedigrees from wild populations. (Demo).
- If there is time: what happens if the graph is not singly-connected?

Summary

- *acyclic directed graph*: factorization over vertices of conditional probability functions; good for representing complex, but normalized, joint probability functions.
- *undirected graph*: factorization over cliques of potential functions; good for representing Markov properties, conditional independence and unnormalized distributions defined from local interactions.
- *factor graph* explicit definition of factorization using factor nodes. Sum-product algorithm.
- Yay! Pedigree inference.

References, etc.

Lecture notes:

- Slides: `https://github.com/eriqande/graph-theory-guest-lecture-feb-2016/blob/master/tex/graphs-in-statistics-slides.pdf`
- Narrative notes: `https://github.com/eriqande/graph-theory-guest-lecture-feb-2016/blob/master/tex/graphs-in-statistics-narrative.pdf`

Anderson EC, Ng TC (2016) Bayesian pedigree inference with small numbers of single nucleotide polymorphisms via a factor-graph representation. *Theoretical population biology*, **107**, 39–51.

Kschischang FR, Frey BJ, Loeliger HA (2001) Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, **47**, 498–519.