

of Bayesian networks. Exact inference algorithms for pedigree analysis, resembling variable elimination, were developed in the 1970s (Cannings *et al.*, 1978). Bayesian networks have been used for identifying human genes by reference to mouse genes (Zhang *et al.*, 2003), inferring cellular networks (Friedman, 2004), genetic linkage analysis to locate disease-related genes (Silberstein *et al.*, 2013), and many other tasks in bioinformatics. We could go on, but instead we'll refer you to Pourret *et al.* (2008), a 400-page guide to applications of Bayesian networks. Published applications over the last decade run into the tens of thousands, ranging from dentistry to global climate models.

Judea Pearl (1985), in the first paper to use the term “Bayesian networks,” briefly described an inference algorithm for general networks based on the cutset conditioning idea introduced in Chapter 6. Independently, Ross Shachter (1986), working in the influence diagram community, developed a complete algorithm based on goal-directed reduction of the network using posterior-preserving transformations.

Pearl (1986) developed a clustering algorithm for exact inference in general Bayesian networks, utilizing a conversion to a directed polytree of clusters in which message passing was used to achieve consistency over variables shared between clusters. A similar approach, developed by the statisticians David Spiegelhalter and Steffen Lauritzen (Lauritzen and Spiegelhalter, 1988), is based on conversion to an undirected form of graphical model called a **Markov network**. This approach is implemented in the HUGIN system, an efficient and widely used tool for uncertain reasoning (Andersen *et al.*, 1989).

The basic idea of variable elimination—that repeated computations within the overall sum-of-products expression can be avoided by caching—appeared in the symbolic probabilistic inference (SPI) algorithm (Shachter *et al.*, 1990). The elimination algorithm we describe is closest to that developed by Zhang and Poole (1994). Criteria for pruning irrelevant variables were developed by Geiger *et al.* (1990b) and by Lauritzen *et al.* (1990); the criterion we give is a simple special case of these. Dechter (1999) shows how the variable elimination idea is essentially identical to **nonserial dynamic programming** (Bertele and Brioschi, 1972).

Nonserial dynamic programming

This connects Bayesian network algorithms to related methods for solving CSPs and gives a direct measure of the complexity of exact inference in terms of the tree width of the network. Preventing the exponential growth in the size of factors in variable elimination can be done by dropping variables from large factors (Dechter and Rish, 2003); it is also possible to bound the error introduced thereby (Wexler and Meek, 2009). Alternatively, factors can be compressed by representing them using algebraic decision diagrams instead of tables (Gogate and Domingos, 2011).

Exact methods based on recursive enumeration (see Figure 13.11) combined with caching include the recursive conditioning algorithm (Darwiche, 2001), the value elimination algorithm (Bacchus *et al.*, 2003), and AND-OR search (Dechter and Mateescu, 2007). The method of weighted model counting (Sang *et al.*, 2005; Chavira and Darwiche, 2008) is usually based on a DPLL-style SAT solver (see Figure 7.17 on page 234). As such, it is also performing a recursive enumeration of variable assignments with caching, so the approach is in fact quite similar. All three of these algorithms can implement a complete range of space/time tradeoffs. Because they consider variable assignments, the algorithms can easily take advantage of determinism and context-specific independence in the model. They can also be modified to use an efficient linear-time algorithm whenever the partial assignment makes the remaining network a polytree. (This is a version of the method of **cutset conditioning**, which was described