Belief Networks, Hidden Markov Models, and Markov Random Fields: a Unifying View*

Padhraic Smyth
Information and Computer Science Department
University of California, Irvine
CA 92697-3425.
smyth@ics.uci.edu

March 20, 1998

Abstract

The use of graphs to represent independence structure in multivariate probability models has been pursued in a relatively independent fashion across a wide variety of research disciplines since the beginning of this century. This paper provides a brief overview of the current status of such research with particular attention to recent developments which have served to unify such seemingly disparate topics as probabilistic expert systems, statistical physics, image analysis, genetics, decoding of error-correcting codes, Kalman filters, and speech recognition with Markov models.

1 Introduction

Let $\mathbf{U} = \{X_1, \ldots, X_N\}$ be a set of random variables, representing for example, symptoms and diseases in a medical diagnosis context, features and classes in a pattern recognition problem, or properties of individual particles in a statistical physics problem. Let $p(\mathbf{U})$ represent the joint distribution for $\mathbf{U}$. In this paper we will use the term *graphical models* to refer to a family of techniques which exploit a duality between graph structures and probability models.

The central idea behind graphical models is to represent the independence structure in $p(\mathbf{U})$ by an *annotated graph*. The nodes of the graph are in one-to-one correspondence with the variables in $\mathbf{U}$ and the edges of the graph reflect the independence structure (if any) in $p(\mathbf{U})$. Thus, for example, a probability model with no independence structure (namely, every variable depends directly on every other variable) is represented by a completely connected graph. Conversely, a model $p(\mathbf{U})$ where all variables are independent of each other is represented by a graph with no edges between any of the nodes. Of more usual interest are the families of probability models which lie between these extremes. Annotation of the graph is achieved by factoring the underlying probability model $p(\mathbf{U})$ into conditional probability tables (for directed graphs) or potential functions (for undirected

*To appear in *Pattern Recognition Letters*, 1998

graphs). These factors are stored as tables or simple functions at the individual nodes. These local tables and functions represent the numerical specification of local dependencies and are the vehicle for efficient calculations using the graph formalism.

It is well known that for moderately large N, specification and manipulation of $p(\mathbf{U})$ directly is intractable unless there exists considerable structure in the probability model. For example, with N binary variables, a model with no independence structure requires the specification of $O(2^N)$ probability values. Furthermore, calculations of particular posterior probabilities given observed evidence will also tend to scale exponentially in N, rendering such models useless in practice. This intractability has been well-known in different disciplines for some time and there has been considerable, and often independent, work in different areas on exploiting independence structure to achieve tractability.

In statistics, the use of graphical frameworks to represent and manipulate multivariate probability distributions is by now well-established (Whittaker (1990), Lauritzen (1996)). In an artificial intelligence (AI) context, Pearl (1988) independently developed a substantial body of theory for constructing and manipulating conditional independence relations using directed graphical models called belief networks. In statistical physics, there is a long tradition of performing efficient probability calculations on lattice systems of large numbers of particles whose probability distributions have certain Markov properties (Kinderman and Snell, 1990). This work, linked with related ideas in statistics (Isham, 1981), motivated a whole sub-discipline of image analysis based on Markov random fields (Geman and Geman, 1984) and related work in neural network modeling using Boltzmann machines (Hinton and Sejnowski, 1986). The fact that all of these models are closely related is relatively well-known although not always explicitly referred to in the literature.

Less well known are recent realizations that the "extended family" of graphical models also encompasses some very well-known and widely used techniques in engineering. Specifically, hidden Markov models (including the forward-backward algorithm) as used in speech recognition can be viewed as special cases of graphical models (Smyth, Heckerman and Jordan, 1997), Kalman filtering equations and models can be profitably viewed from a graphical model context (Levy, Benveniste, and Nikoukhah, 1996), and a variety of well-known algorithms for decoding error-correcting codes turn out to be special cases of more general graphical model algorithms (MacKay, McEliece, and Cheng, in press).

The purpose of this paper is to briefly review some of these connections. The paper does not discuss the details of graphical models in any depth: for a recent introductory exposition see Jensen (1996) and for a more mathematical viewpoint see Lauritzen (1996). The paper by Smyth, Hecker-

man, and Jordan (1997) discusses links between different forms of graphical models used in statistics, AI, physics, and engineering. The primary goal of this paper is to point the reader to the relevant literature on the topic and promote the viewpoint that graphical models provide a unified and useful framework for a large class of problems involving probabilistic inference.

2 A Brief Introduction to Graphical Models

Graphical models fall into two general classes, those based on acyclic directed graphs (ADGs)[1] and those based on undirected graphs (UGs). There is a third category based on mixed graphs which are beyond the scope of this paper. Both ADG and UG representations rely on the notion of decomposing the underlying multivariate probability distribution into a factored form. For ADGs the factors are local conditional probabilities, for UGs they are local clique functions (non-negative functions related to probabilities). In this context, ADGs are easier to construct and interpret since they have a clearer probabilistic semantics than UGs in terms of the numerical specification of the probability model. ADGs and UGs can each efficiently represent probability distributions which the other cannot represent in efficient form. The directed ADG formalism is primarily used in AI and statistics where cause-effect relationships are important in modeling and can be made explicit by the use of directed arcs in the graph. The undirected UG formalism is popular in the statistical physics and image processing communities where associations between variables (particles or pixels) are considered correlational rather than causal. UGs under various guises are variously referred to in the literature as Markov random fields, Markov networks, Boltzmann machines, and log-linear models. ADGs are often referred to as Bayesian networks, belief networks, or recursive graphical models, and less frequently as causal networks, directed Markov networks, and probabilistic (causal) networks.

A graphical model contains both *structure* and *parameters*. The structure of the model consists of the specification of a set of *conditional independence relations* for the probability model $p(\mathbf{U})$, represented as a set of *missing edges* in the graph for the graphical model. If variable X_i does not depend directly on variable X_j, then there is no edge between them. The precise semantic implications differ between ADGs and UGs, but the central concept is the same: a node is connected to those other nodes on which it directly depends. Note that a graph structure implies a *set* of probability models which are constrained to obey the independence assumptions as represented by the connectivity of the graph. Conversely, the independence relations which are implicit in the

[1] ADGs are also often referred to as *directed acyclic graphs* (DAGS); however, the term ADG is more precise, since the term DAG implies a *directed* version of an *acyclic graph*, which is not well-defined.

probability model $p(\mathbf{U})$, constrain the possible corresponding graphical structures.

The parameters of a graphical model consist of the specification of the joint probability distribution $p(\mathbf{U})$. This specification is in factored form and the factors are defined locally on the nodes of the graph. *Inference* is the problem is of calculating posterior probabilities for variables of interest given observed data and given a specification of the probabilistic model. Typical inference problems include calculating the probability of a class variable given observed features (in classification) and calculating the probability of observed data under various different models (as in speech recognition). The related task of *maximum a posteriori (MAP) identification* is the determination of the most likely state of a set of unobserved variables, given observed data and the probabilistic model. The *learning* or *estimation* problem is that of determining the parameters (and possibly structure) of the probabilistic model from data.

3 Why use Graphical Models?

A key point is that the analysis and manipulation of multivariate models involving independence relations can be considerably facilitated by exploiting the relationship between probability models and graphs. The major advantages to be gained are in *model description* and *computational efficiency*.

3.1 Model Description

Graphs are a natural medium for representing information in a compact form which humans can grasp, understand, and use. In particular, the *structure* of a graphical model clarifies the conditional independencies in the implied probability models, allowing model assessment and revision. Whittaker (1990, chapter 3) provides a number of examples which clearly demonstrate that even with relatively few variables it is much easier to reason about independence relations using a graph than it is without. In addition, the fact that the graphical model forces the modeller to explicitly encode and confront independence assumptions can be extremely useful in model-building. This can be particularly useful for example in areas such as AI, statistical modeling in the social and medical sciences, and time-series modeling.

3.2 Computational Efficiency

Graphical models are a powerful basis for specifying efficient algorithms for computing quantities of interest in the probability model, e.g., calculation of the probability of observed data given the model. Computational inference methods are often based on undirected representations (Lauritzen and Spiegelhalter, 1988). ADGs can be reduced to an equivalent UG structure in a relatively

straightforward manner, although the corresponding UG may be less efficient at representing the same probability distribution as the original ADG, i.e., have more edges. The "canonical" graphical form for computation is the "clique tree" (Jensen, 1996), which is constructed from the UG representation via triangulation. Inference simply consists of local message passing in the clique tree. The "clique tree inference algorithms" (Jensen, 1996) are quite general and subsume the earlier more specialized inference algorithms such as those proposed by Pearl (1988). The complexity of the local inference algorithms scale as the sum of the sizes of the clique state-spaces (where a clique state-space is equal to the product over each variable in the clique of the number of states of each variable). Thus, local clique updating can reduce the complexity of exact inference and MAP calculations on $\mathbf{U}$ from $O(m^N)$ to $O(m^K)$, where N is the total number of variables, K is the number of variables in the largest clique, and all variables are assumed to take m discrete values. For dense graphs, exact computations are intractable (K becomes very large) and a variety of approximation schemes exist, largely based on sampling techniques which have evolved from statistical physics methodologies for intractable lattice-type graphs.

A key feature of computation in graphical models is that these inference algorithms can be specified automatically (in effect, "compiled") once the initial structure of the graph is determined. Note that the graphical model framework provides no panacea for avoiding the combinatorial parameter explosion which can result when one tries to build more realistic models. Rather, it allows one to identify an efficient inference procedure in an automatic manner, *if the structure of the model permits efficient inference.*

4 Relationships between Specific Classes of Graphical Models

4.1 Belief networks as ADGs

In AI the best known family of graphical models are belief networks. Belief networks are ADGs which were developed originally by Pearl (1988) for probabilistic reasoning or "probabilistic expert systems." Pearl (1988, p.125) notes that the origins of such models can be traced to the work of Wright (1921) in genetics. Belief networks have gained widespread acceptance and application within AI in areas such as diagnosis, planning, robotics, computer vision, and so forth (see, for example, Heckerman, Wellman, and Mamdani, 1995). From an AI perspective the well-defined semantics of an ADG, where each node is a direct descendant of its "causal" parents, provide a useful and practical language for knowledge elicitation. In addition, belief networks provide a sound and efficient

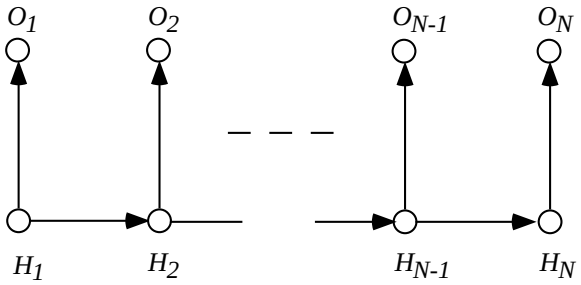

Figure 1: An ADG for a first-order HMM. The H_i and the O_i are the hidden state variables and the observable variables, respectively, $1 \leq i \leq N$.

framework for dealing with uncertainty, in contrast to earlier attempts within AI to handle uncertain reasoning. More recently there has been significant interest and progress in *learning* or *estimating* both the parameters and structure of belief networks from data, thus broadening their application to problems where large data sets are available and perhaps relatively little or no prior knowledge in the form of available experts (Buntine, 1996; Heckerman, Geiger, and Chickering, 1995). In a related context, learning of multivariate regression and classification models such as neural networks, can also be treated profitably within a graphical model framework (Buntine, 1994).

4.2 Hidden Markov models as ADGs

The well-known "first-order" hidden Markov model (HMM) (as widely used in speech recognition) is a particularly simple probability model and has a direct representation as a graphical model (Figure 1). Speech recognition systems take advantage of the fact that there exist efficient algorithms (linear in the length of the Markov chain) for solving the inference and MAP problems associated with recognition. Inference is solved by the forward-backward algorithm and the MAP problem is handled by the Viterbi algorithm (Rabiner, 1989). Since we can represent a HMM as a simple graphical model, it follows that the inference and MAP problems can be solved by the standard algorithms developed by Pearl (1988), Lauritzen and Spiegelhalter (1988), and subsequent refinements (Jensen, Lauritzen and Olesen (1990). Smyth, Heckerman, and Jordan (1997) show that the forward-backward algorithm and Viterbi algorithms are in fact directly equivalent to the Pearl et al algorithms, i.e., these algorithms had been developed completely independently in both communities. The equivalence is not surprising once one realizes that a HMM is a relatively simple graphical model. Of much greater significance is the fact that the graphical model algorithms are perfectly general and can thus handle arbitrary extensions to the standard "first-order" model. No additional effort is required in terms of deriving new inference procedures for more complicated models, since

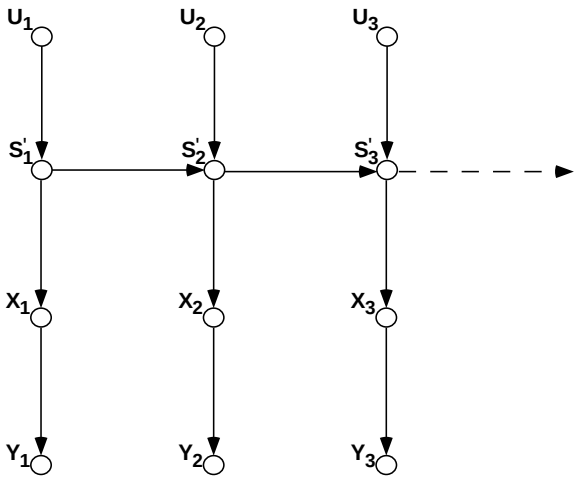

Figure 2: An ADG for a simple convolutional encoder/decoder. The U_i are the input information bits, the S_i are the state variables of the convolutional encoder, the X_i are the code words, and the Y_i are the received noisy codewords $1 \leq i \leq N$. Note that all of the dependencies in the model are deterministic except the dependence of Y_i on X_i, which models a memoryless noisy channel. The inference problem (which is equivalent to decoding) is to determine the most likely values for the information bits U_i given the observed noise codewords Y_i.

the inference algorithms follow directly from the general specifications of Pearl et al. Examples of more complex HMM structures to account for coarticulation in speech and multiple hidden chains to couple audio and video signal inputs are discussed in Smyth, Heckerman and Jordan (1997).

4.3 Decoding Algorithms for Error Correcting Codes as ADGs

In error correcting coding applications a sequence of "information bits" is converted into a sequence of codewords which is transmitted over a noisy channel. At the receiving end of the channel, a decoder must try to estimate the original information sequence, given only the noisy codeword sequence. It has recently been realized that the decoding process is well-modeled as inference on a graphical model. The inference problem is that of calculating the probability of the input sequence given the observed codewords. The graphical model for the problem arises from the coupling of the deterministic mapping of inputs to codewords with the noisy channel process mapping codewords into noisy observations. Typically the resulting graph is highly structured: Figure 2 shows the ADG for a convolutional decoder. The algorithms developed in the coding literature for decoding tend to be very similar to the forward-backward algorithm and Viterbi algorithms used for HMMs. Thus, it is not surprising that one can recreate these algorithms as special cases of the more general graphical model inference algorithms.

While this direct equivalence of existing algorithms is interesting (as with HMMs), the more

significant aspect is the capability which graphical models provide for synthesizing new decoding algorithms using more complex structures. From a graphical model viewpoint, extensions to deal with channels with memory, multiple interleaved codes, iterative decoding for approximate solutions, and so forth, can all be handled in a straightforward and systematic manner. The power of graphical models in this context has only recently been realized by the coding community and there is currently significant research activity on decoders based on graphical models (e.g., MacKay, McEliece and Cheng, in press; Kschischang and Frey, in press).

4.4 Kalman Filter and Related Algorithms as ADGs

Kalman filters, and related linear models for dynamical systems, are essentially very similar to HMMs but where the hidden state variables are real-valued rather than discrete. Thus, it should not be surprising to the reader at this point to learn that such models can also be represented within the graphical model family, again as ADGs due to the causal nature of temporal processes (Kenley, 1986). More recently there have been significant extensions which have proceeded by showing the direct equivalence of the standard Kalman prediction/smoothing equations to graphical model inference algorithms, and by then exploiting the generality of graphical models to propose novel extensions to standard Kalman filters within a unified framework (Levy, Benveniste, and Nikoukah, 1996). In the context of more general time-series modeling, graphical models can also play a useful role. For example, Berzuini and Larizza (1996) describe a complex medical application treated within a graphical model framework

4.5 Markov Random Fields as UGs

As mentioned earlier, Markov random fields (MRFs) are the most well-known undirected graphical model formalism and were originally developed in statistical physics to model systems of particles interacting in a 2d or 3d lattice (Kinderman and Snell, 1980). More recently MRFs have been widely applied to problems in image analysis, where pixels or voxels play the role of particles in the physical system. The resulting UGs have many loops, resulting in exponential complexity in N (the number of nodes) for exact solutions to the inference problem. A wide variety of techniques have been developed for approximating the exact solution. Physicists and statisticians have developed elaborate techniques based on iterative sampling (Monte-Carlo) ideas which are guaranteed to converge under fairly general conditions (Gilks, Richardson, and Spiegelhalter (1996)). Closed form approximations to the exact solution have also been popular. For example, the use of "mean-field" approximations are motivated by physical arguments on the nature of cumulative long-range particle interactions,

and the popular Iterative Conditional Modes algorithm for image analysis relies on greedy local maximization of the posterior probability of the pixel labels given the observed data (Besag, 1986).

It is also worth noting that many *directed* graphical models of practical interest have sufficiently dense structure to not admit efficient exact solutions. Thus, since inference with an ADG is typically carried out by inference on a related UG, there is increasing interest and utility in exploring the UG approximations for applications involving ADGs (see for example, Saul and Jordan (1996)).

5 Conclusion

There has been a recent convergence of ideas relating probability models and graph structures. The graph formalism is an effective and efficient representation for multivariate independence structure, both for model construction and for inference. The ability to view seemingly different algorithms for seemingly different problems within a unified graphical model framework can provide powerful insights. More important is the fact that the graphical model framework enables the construction and application of novel and relatively complex multivariate models in a straightforward and systematic manner.

Acknowledgements

The author gratefully acknowledges discussions on this topic with Wray Buntine, David Heckerman, Michael Jordan, and Paul Stolorz. The research described in this paper was supported in part by the California Institute of Technology and the Air Force Office of Scientific Research under grant no. F49620-97-1-0313, and was carried out in part by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

References

Berzuini, C. and Larizza, C. (1996). A unified approach for modeling longitudinal and failure time data, with application in medical monitoring. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 18(2), 109–123.

Besag, J. (1986). On the statistical analysis of dirty pictures. *J. R. Statist. Soc. B*, 48(3), 259–302.

Buntine, W. (1994). Operations for learning with graphical models. *Journal of Artificial Intelligence Research*, 2, 159–225.

Buntine, W. (1996). A guide to the literature on learning probabilistic networks from data. *IEEE Trans. on Knowledge and Data Engineering*, 8(2), 195–210.

Dawid, A. P. (1992). Applications of a general propagation algorithm for probabilistic expert systems. *Statistics and Computing*, 2, 25–36.

Geman, S. and Geman, D. 1984. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. Pattern Analysis and Machine Intelligence*, **6**, 721-741.

Gilks, W. R., Richardson, S., and Spiegelhalter, D. J. (editors) (1996). *Markov Chain Monte Carlo in Practice*, Chapman and Hall, London.

Heckerman, D., Geiger, D., and Chickering, D. (1995). Learning Bayesian networks: The combination of knowledge and statistical data, *Machine Learning*, 20, 197-244.

Heckerman, D., Wellman, M. and Mamdani A. (editors) (1995). *Communications of the ACM: Special Issue on Uncertainty in AI*, 38(3).

Hinton, G. E. and Sejnowski, T. J. (1986). Learning and relearning in Boltzmann machines. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, Rumelhart D. E., McClelland J. L., and the PDP Research Group, editors, Cambridge, MA: MIT Press, v.1, ch. 7.

Jensen, F. V. (1996). *An Introduction to Bayesian Networks*, University College London Press, London.

Jensen, F. V., Lauritzen, S. L. and Olesen, K. G. (1990). Bayesian updating in recursive graphical models by local computations. *Computational Statistical Quarterly*, **4**, 269–282.

Kenley, C. R. (1986). *Influence Diagram Models with Continuous Variables*. PhD. Thesis Disseration, Department of Engineering-Economic Systems, Stanford University.

Kindermann, R., and Snell, J. L. (1980). *Markov Random Fields and their Applications*, American Mathematical Society.

Kschischang, F. R. and Frey, B. J. (in press). Iteratve decoding of compound codes by probability propagation in graphical models. *IEEE Journal on Selected Areas in Communications*.

Levy, B. C., Benveniste, A., Nikoukhah, R. (1996). High-level primitives for recursive maximum likelihood estimation. *IEEE Transactions on Automatic Control*, 41(8), 1125-1145.

Lauritzen, S. L. (1996). *Graphical Models*, Clarendon Press, Oxford.

Lauritzen, S. L. and Spiegelhalter D. J. (1988). Local computations with probabilities on graphical structures and their application to expert systems (with discussion). *J. Roy. Statist. Soc. Ser. B*, **50**, 157–224.

MacKay, D. J. C, McEliece, R. J., and Cheng, J-F. (in press). Turbo-decoding as an instance of Pearl's belief propagation algorithm. *IEEE Journal on Selected Areas in Communications.*

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann, San Mateo, CA.

Rabiner, L. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, **77**, 257-285.

Saul, L. K., and Jordan, M. I. (1996). Exploiting tractable substructures in intractable networks. In D. S. Touretzky, M. C. Mozer, & M. E. Hasselmo (eds.), *Advances in Neural Information Processing Systems 8*, MIT Press, Cambridge MA.

Smyth, P., Heckerman, D., Jordan, M. I. (1997). Probabilistic independence networks for hidden Markov probability models. *Neural Computation*, 9(2), 227–269.

Whittaker, J. (1990). *Graphical Models in Applied Multivariate Statistics*, John Wiley and Sons: Chichester, UK.

Wright, S. (1921). Correlation and causation. *Journal of Agricultural Research*, 20:557–85.