PNPACK: Algorithm and Software Development for Computing with Probabilities
Draft White Paper

December 31, 1996

Summary

We propose a multi-year, multi-investigator project that will significantly advance the state of the art in computing with probabilistic information, building on dramatic technical advances in the last decade. The project has three basic components:

1. Basic research to develop more powerful representations for probabilistic information and more efficient algorithms for computing with those representations.
2. Development and distribution of a suite of efficient, portable, extensible software—PNPACK. The suite will embody the latest developments in probabilistic reasoning technology, and will be structured so as to allow easy incorporation of the new algorithms and representations that will be developed by the basic research component.
3. Development of new prototype applications using PNPACK as a platform and taking advantage of the new algorithms and representations.

The project is modelled on the successful LAPACK project, which has revolutionized the quality, efficiency, and ease of development of scientific computing applications.

1 Introduction and objectives

It is hard to think of real-world applications of AI that do not involve some form of uncertainty. Uncertainty arises from the following sources:

- Inexact and incomplete "inputs"—this includes ambiguity in natural language; occlusions, noise, and systematic errors in vision systems; ambiguity in speech recognition systems; inexact tests, ambiguous symptoms, and missing information in medical diagnosis, etc.
- Inexact and incomplete knowledge of the underlying mechanisms and processes in the application domain—for example, connections between diseases and manifestations in medicine; processes leading to bankruptcies among credit customers; behaviour of drivers on freeways, etc.

- Intractability of using exact models even when available.

There is a growing consensus that systems based on probability, particularly in the Bayesian tradition, provide rigorous and effective tools for handling uncertainty in a wide variety of applications. The development of mathematical tools for decomposing complex problems into structured probability representations has led to rapid improvements in inference technology and application performance.

For example, probabilistic networks (PNs)[1] dominate the "Uncertainty in AI" and "AI in Medicine" communities, where they have led to deployment of several successful medical diagnosis systems as well as expert assistants incorporated in the Windows 95 operating system (see the recent special issue of CACM [?]). They also have a significant following in the natural language processing, information retrieval, and computer vision communities. Decision networks (DNs)[2], which combine utility information with probabilistic networks, are widely used by decision analysts. *Dynamic* probabilistic and decision networks (DPNs and DDNs) can be used to model, monitor, and control stochastic, partially observable, temporal processes in a wide variety of applications.

Other structured probability models include Markov random fields (MRFs), hidden Markov models (HMMs), and hierarchical mixtures of experts (HMEs). Within their respective range of applicability, each of these representations dominates competing approaches. For example, HMMs are currently the technology of choice for automated speech recognition and DNA modelling. Finally, the neural network community has come to recognize the value of probabilistic interpretations of neural networks (NNs) and has begun to adopt probabilistic techniques for NN training.

In summary, AI, computer science, and statistics have made very significant strides in the past decade in the area of computing with probabilities. Several thousand researchers in universities and industry are now engaged in investigating the application and further development of structured probabilistic models.

One very serious obstacle to further progress is the significant cost of developing or acquiring usable probabilistic inference software. Because the mathematics and algorithmic details are quite complex, it is often not feasible for AI researchers to develop their own code. On the other hand, commercial software such as HUGIN from Aalborg University in Denmark [Andersen *et al.*, 1989] is both expensive ($6500 academic license) and inflexible because source code is not available. The most comprehensive available public domain software is the IDEAL system, which is in use at about 150 sites [?]. IDEAL was designed for simplicity, with pedagogy in mind, is written in Lisp, and runs approximately 100 to 500 times slower than necessary. These problems mean that recent and ongoing developments in representations and algorithms for probabilistic information cannot be incorporated and distributed to researchers worldwide.

A second problem with existing software systems is the lack of integration of the various types of structured probabilistic models. No system currently allows for more than one kind of representation, despite the fact that the underlying semantics is identical—probability distributions over random variables. Furthermore, even within the communities working on specific models, such as probabilistic networks, interoperability is hindered by the lack of a common representation.[3]

In addition to the problems for researchers, the lack of efficient, integrated software with source code availability also causes serious difficulties in the area of education and training. Although textbooks are available that stress probabilistic methods [Pearl, 1988; Russell and Norvig, 1995], the lack of software suitable for educational purposes has proved to be a big problem for instructors, and is slowing down the process of training new generations of researchers and practitioners in this vital technology.

[1] Also known as causal networks, Bayesian networks, and belief networks [Pearl, 1988].

[2] Also called influence diagrams.

[3] The recent announcement by Microsoft of a standard file format for belief networks is a positive development on which we will build.

The objectives to be achieved by the proposed research are as follows:

1. Extend the scope of problems that can be represented and solved using structured probabilistic models, and develop more flexible and efficient algorithms for probability computations.
2. Enhance the impact of our basic research, and improve the productivity of the research and development commmunity in general, through the development and distribution of a suite of efficient, portable, extensible software—PNPACK. The suite will embody the latest developments in probabilistic reasoning technology, and will be structured so as to allow easy incorporation of the new algorithms and representations that will be developed by the basic research component and by other researchers.
3. Advance the state of the art and stress our software design through the development of new prototype applications using PNPACK as a platform and taking advantage of the new algorithms and representations.

Subsequent sections explain how the three objectives will be achieved. Appendix A describes the basic terminology and methods used in structured probability models. Appendices B and C describe the personnel, organization, and proposed budget of the project.

2 Proposed plan of work

The following subsections describe in more detail the work we will carry out to achieve the three main objectives stated above. We specify the planned completion point on a rough four-year timeline.

2.1 Research on probabilistic systems

We begin with a brief discussion of existing techniques that have not yet been incorporated into existing packages. These present challenges of integration and implementation. We then then describe a broad set of research topics.

2.1.1 Integration and efficient implementation of existing methods (Yr.1)

Current packages do not include the latest developments in inference and decision algorithms for probabilistic networks. Four principal developments are missing from both commercial and public domain packages:

- *Hybrid networks* containing both discrete and continuous variables [Lauritzen and Wermuth, 1989]. Continuous variables are essential for modelling many real-world domains.
- *Stochastic simulation* methods such as likelihood weighting [Shachter and Peot, 1989]. These are currently the only feasible techniques for very large networks or for general networks with continuous variables.
- *Dynamic probabilistic networks* (DPNs) for handling domains that change over time [Dean and Kanazawa, 1988; Kjaerulff, 1992].
- *Learning methods* for constructing networks from data and prior knowledge [Heckerman *et al.*, 1994; Lauritzen, 1995; Russell *et al.*, 1995].

2.1.2 Development of new methods (Yrs.1–3)

Each of the following research projects will result in the creation and distribution of new code in later releases of the system:

- *Real-time monitoring using dynamic networks* (Yr.1).
  Dynamic probabilistic networks (DPNs) represent and update probability distributions on the state of a partially observable system over time. They can be used for monitoring and controlling arbitrary stochastic processes. We will refine and implement methods for real-time monitoring based on the ER/SOF and temporal invariance algorithms reported in [Forbes *et al.*, 1995; Kanazawa *et al.*, 1995].

- *Compilation techniques for efficient, stand-alone inference* (Yr.1).
  Knowledge base interpreters, including most expert system shells as well as probabilistic network packages, solve queries as if the evidence pattern and knowledge base were completely new. Compilation techniques such as QDAG [?] convert a network and evidence pattern into straight-line C code that can then be compiled into a compact, stand-alone binary that executes very efficiently in a target machine without requiring the presence of the entire package. As with compilation of logic programs, speedups of three orders of magnitude can be expected. We will develop compilation techniques for all major inference methods including stochastic simulation and DPN inference. Such methods will be vital in speeding up learning methods, where each iteration uses the same evidence pattern.

- *Parallel algorithms* (Yr.2).
  Parallization of stochastic simulation algorithms, with the exception of Monte Carlo Markov Chain, is straightforward and offers potentially massive speedup. Clustering and conditioning methods offer coarse-grain parallelism at the network node level and fine-grain parallelism in matrix computations. We anticipate providing a "network supercomputer" Probabilistic Computation Facility using Berkeley's NOW system, which will be available as an Internet resource.

- *New methods for stochastic simulation* (Yr.2).
  We will develop and analyze new algorithms such as "Survival of the Fittest" (SOF) [Kanazawa *et al.*, 1995], which offer faster convergence than existing methods.

- *Hierarchical probabilistic models* (Yr.3).
  Complex systems such as computers, vehicles, cities, transportation systems, and corporations can often be modelled successfully using hierarchical representations, in which complexity is reduced exponentially through multiple levels of abstraction. We will investigate both state abstraction and temporal abstraction (for process modelling and planning), studying efficient inference algorithms for hierarchical models as well as interfaces for constructing and using hierarchical models of complex systems.

- *Handling multiple goals* (Yr.3).
  Many real problems involve multiple, conflicting objectives, which can be handled formally by multi-attribute utility theory. The decomposition of a utility function can be described by a directed graph in much the same way that probability distributions are represented by probabilistic networks. We will therefore investigate the design of interfaces and representations for utility functions and efficient algorithms for optimal planning under uncertainty with multiple objectives.

- *Estimation and propagation of second-order probabilities*(Yr.2).
  Probabilistic models acquired from experts or estimated from data are often subject to uncertainty in their numerical parameters. This uncertainty impacts reliability and sensitivity and can guide the

acquisition of further data and expertise. We will investigate methods for representing such *second-order* probabilistic information and for propagating it as part of inference on the underlying model, building on recent work at Berkeley [?].

- *Representation and inference with continuous variables* (Yr.3).
  Hybrid systems involving mixtures of discrete and continuous variables are common in the real world (for example, a car's position is continuous but its turn signal is discrete). Very little is known about appropriate representations for typical relationships between continuous and discrete variables. We will investigate canonical models for hybrid systems with uncertainty using techniques such as decomposition into systems of soft thresholds, as well as stochastic simulation algorithms designed specifically for inference in hybrid systems.

- *Knowledge-based construction of probability models* (Yr.3).
  More powerful representation languages such as *first-order probabilistic logic* allow for the creation of general knowledge bases from which standard probability models can be constructed and solved for specific situations. We will investigate languages and methods for knowledge-based model construction, in particular for monitoring and controlling processes over time where the structure of the model may need to change dynamically (for example, as new objects enter the system).

- *Qualitative probability* (Yr.2).
  Qualitative probabilistic networks [Wellman, 1990] allow specification of non-numeric constraints on probabilities (e.g., "higher impact velocity increases likelihood of damage"). We will extend the representation and inference schemes for probability models to include qualitative constraints.

- *New learning algorithms* (Yr.2).
  Existing methods for learning probabilistic networks from data [Russell *et al.*, 1995] will be extended to handle continuous and hybrid models, functional representations of conditional probability distributions, DPNs and the automatic creation and modification of network structure including hidden variables.

- *Local structure* (Yr.2).
  For many applications, explicit tabulation of conditional distributions is unwieldy or impossible (e.g., with continuous variables). We will extend our software and representations to allow specification and learning of models in which component distributions are specified from canonical families. Examples include noisy-OR models, logistic functions, and multilayer perceptrons. Experiments have shown that the consequent reduction in the number of parameters results in much faster learning as well as the discovery of additional causal structure.

- *Causal analysis* (Yr.3).
  Causal networks extend standard probabilistic netowrks ... [[pearl's stuff]]

2.1.3 Development of new applications (Yrs.2–4)

In order to validate the applicability of our overall software design and the component algorithms and representations, we will udnertake a number of application projects. These will depend on specific area/industry collaborations. Examples include the following:

- *Applications of adaptive dynamic probabilistic networks* (Yrs.2–3).
  Adaptive DPNs can be used to create models of stochastic processes; because they decompose the state representation into a set of state variables, DPN models can be exponentially more compact than

corresponding HMMs and hence, as we have found experimentally, they can often learn faster from much less data. We will address high-impact applications including speech recognition and identification of DNA coding regions. We expect DPNs and related models to become a powerful new weapon in the armory of empirical scientific investigation.

- *Computer vision* (Yrs.3–4).
  Computer vision and image processing have used probabilistic methods, usually based on Markov random fields, for many years, but only for low-level processing [Geman and Geman, 1984]. We propose that low-level and high-level vision can be integrated successfully by adopting probabilistic models for object, scene, and behaviour recognition, as demonstrated for traffic scenes in [?]. We will develop vision systems using such techniques, including hierarchical component models and inference systems combining top-down and bottom-up recognition.

- *Economic and financial modelling* (Yrs.2–3).
  Many financial and economic models use very simple, parameterized, quasilinear representations such as logistic regression whereas in fact the underlying processes are often complex and time-varying. At the same time, large amounts of data are available from insurance and credit databases and federal government surveys. We will investigate the application of our modelling and learning techniques to tasks such as commodity supply/demand forecasting, insurance and credit risk analysis, and development of longitudinal personal finance models.

2.2 Basic software design

The project will use Java as the top-level implementation language, to take advantage of portability, robustness, easy of GUI development, and the enormous ongoing investment in development systems, compilers, and runtime technology. We will develop a class hierarchy for representations of the basic elements (prior and conditional probability distributions, data sets) such that algorithms can be written at a level of generality that allows them to apply to a wide variety of specific types of representations (e.g., Markov networks, probabilistic networks, HMMs, etc.). For example, the EM algorithm can be used to learn models under any representation.

The system will also include compilation of models to "object code," which may be in C or Java, so that specific models can run efficiently and in stand-alone mode for applications and embedded systems. In addition, we may use efficient libraries such as LAPACK and BLAS for high-performance inference.

We will include comprehensive code for testing, so that any system updates or extensions can be verified for correctness before release. Timing code will be provided to allow for extensive benchmarking of the software and proposed additions. Documentation standards will be specified and enforced. We will [[Java, class hierarchy, compiling/code generation, C, C++, library functions (LAPACK/BLAS/runtime), testing and timing, coding and doc standard, i/o representation standards, GUI, extension language]]

References

[Andersen *et al.*, 1989] S. K. Andersen, K. G. Olesen, F. V. Jensen, and F. Jensen. HUGIN—a shell for building Bayesian belief universes for expert systems. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence (IJCAI-89)*, volume 2, pages 1080–1085, Detroit, Michigan, August 1989. Morgan Kaufmann.

[Dean and Kanazawa, 1988] Thomas Dean and Keiji Kanazawa. Probabilistic temporal reasoning. In *Proceedings of the Seventh National Conference on Artificial Intelligence (AAAI-88)*, pages 524–528, St. Paul, Minnesota, 1988. American Association for Artificial Intelligence.

[Forbes *et al.*, 1995] Jeff Forbes, Tim Huang, Keiji Kanazawa, and Stuart Russell. The BATmobile: Towards a Bayesian automated taxi. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI-95)*, Montreal, Canada, August 1995. Morgan Kaufmann.

[Geman and Geman, 1984] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 6(6):721–741, 1984.

[Heckerman *et al.*, 1994] D. Heckerman, D. Geiger, and M. Chickering. Learning Bayesian networks: The combination of knowledge and statistical data. Technical Report MSR-TR-94-09, Microsoft Research, Redmond, Washington, 1994.

[Kanazawa *et al.*, 1995] Keiji Kanazawa, Daphne Koller, and Stuart Russell. Stochastic simulation algorithms for dynamic probabilistic networks. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence (UAI-95)*, Montreal, 1995. Morgan Kaufmann.

[Kjaerulff, 1992] U. Kjaerulff. A computational scheme for reasoning in dynamic probabilistic networks. In *Proceedings of the Eighth Conference on Uncertainty in Artificial Intelligence*, pages 121–129, 1992.

[Lauritzen and Spiegelhalter, 1988] Steffen L. Lauritzen and David J. Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society*, B 50(2):157–224, 1988.

[Lauritzen and Wermuth, 1989] S. L. Lauritzen and N. Wermuth. Graphical models for associations between variables, some of which are qualitative and some quantitative. *Annals of Statistics*, 17:31–57, 1989.

[Lauritzen, 1995] S. L. Lauritzen. The EM algorithm for graphical association models with missing data. *Computational Statistics and Data Analysis*, 19:191–201, 1995.

[Pearl, 1988] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann, San Mateo, California, 1988.

[Pradhan *et al.*, 1994] M. Pradhan, G. M. Provan, B. Middleton, and M. Henrion. Knowledge engineering for large belief networks. In *Proceedings of Uncertainty in Artificial Intelligence*, Seattle, Washington, 1994. Morgan Kaufmann.

[Russell and Norvig, 1995] Stuart J. Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach.* Prentice-Hall, Englewood Cliffs, New Jersey, 1995.

[Russell *et al.*, 1995] Stuart Russell, John Binder, Daphne Koller, and Keiji Kanazawa. Local learning in probabilistic networks with hidden variables. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI-95)*, pages 1146–52, Montreal, Canada, August 1995. Morgan Kaufmann.

[Shachter and Peot, 1989] R. D. Shachter and M. A. Peot. Simulation approaches to general probabilistic inference on belief networks. In *Proceedings of the Fifth Conference on Uncertainty in Artificial Intelligence (UAI-89)*, Windsor, Ontario, 1989. Morgan Kaufmann.

[Wellman, 1990] Michael P. Wellman. Fundamental concepts of qualitative probabilistic networks. *Artificial Intelligence*, 44(3):257–303, August 1990.

Appendix A Background

The central concept in probability theory is that of a *random variable*, that is, an aspect of the world whose exact state is unknown. For example, this might be the *Disease* from which a patient is suffering, the *Weather* in some geographical region, or the *CrudeOilPrice* at some future time. Variables range over a set of *values* that can be discrete (e.g., *Disease*) or continuous (e.g., *CrudeOilPrice*). A *probability distribution* $P(X)$ for a variable X specifies the probability for each possible value. *Evidence* specifies observed values for a set of variables. The most typical inference problem involves computing the *conditional* probability distribution $P(X|E)$ for a variable X given some evidence E.

A complex situation is modelled by a set of random variables. Any particular configuration is described by an assignment of values to all the random variables. A *joint probability distribution* for a set of variables specifies a probability for each possible complete assignment. The joint probability distribution contains all the information required to answer any query about the domain.

Since the number of possible complete assignments to a set of variables is exponential in the number of variables (or infinite in the case of continuous variables), the key problem is to find *compact* representations of the joint probability distribution that nonetheless reflect the actual properties of the domain, and to find efficient algorithms for manipulating these representations to solve problems. *Structured probability models* provide a variety of techniques for doing this. The following paragraphs provide a brief introduction to probabilistic networks. For details of HMMs, see Rabiner [?]. For HMEs, see Jordan [?].

Probabilistic networks derive their power from the ability to represent conditional independences among variables, which allows them to take advantage of the "locality" of causal influences. Intuitively, a variable is independent of its indirect causal influences given its direct causal influences. In Figure ??, for example, the outcome of the X-ray does not depend on whether the patient is a smoker given that we know that the patient has lung cancer. If each variable has at most k other variables that directly influence it, then the total number of required parameters is linear in n and exponential in k. This enables the representation of quite large problems. For example, the CPCS network [Pradhan *et al.*, 1994] contains 448 variables and compares well with the world's leading diagnosticians in internal medicine.

Formally, a probabilistic network is defined by a directed acyclic graph together with a conditional probability table (CPT) associated with each node (see Figure ??).[4] Each node represents a random variable. The CPT associated with variable X specifies the conditional distribution $\mathbf{P}(X \mid Parents(X))$. The arcs encode probabilistic dependence in the sense that each variable must be conditionally independent of its non-descendants in the graph, given its parents. This constraint implies that the network provides a complete representation of the joint distribution through the following equation:

$$P(x_1 \ldots x_n) = \prod_{i=1}^{n} P(x_i \mid Parents(X_i)). \tag{1}$$

where $P(x_1 \ldots x_n)$ is the probability of a particular combination of values for $X_1, \ldots, X_n$. Provided each node has a bounded number of parents, the total amount of information needed to specify the joint probability distribution is now only linear in the number of variables.

Once a network has been constructed, inference algorithms operate on it to calculate probabilities for query variables given values for evidence variables. It is important to note that the distinction between evidence and query variables is entirely flexible—any variable can be set and any variable can be queried. The best exact

[4] We have described the simplest form of network. Networks can include continuous as well as discrete variables, provided the representation of the conditional density function is finite. CPTs can be represented implicitly by parameterized functions instead of explicit tables.

inference algorithms typically use a transformation to Markov random fields [Lauritzen and Spiegelhalter, 1988]. Stochastic approximation algorithms using Monte Carlo simulation have also been developed [Pearl, 1988]. Although the general inference problem is likely to be of exponential complexity in the worst case, large networks are often solvable in practice.

B Project personnel

C Project organization

Timeline, management, budget, deliverables, distribution plan