Machine Intelligence and Pattern Recognition
Volume 8

Uncertainty in Artificial Intelligence 3

Edited by

Laveen N. KANAL

Tod S. LEVITT

John F. LEMMER

Machine Intelligence and Pattern Recognition

Volume 8

Series Editors

L.N. KANAL

and

A. ROSENFELD

University of Maryland
College Park, Maryland, U.S.A.

NORTH-HOLLAND
AMSTERDAM • NEW YORK • OXFORD • TOKYO

PREFACE

This Volume contains selected papers from the third Uncertainty in Artificial Intelligence (AI) Workshop held July 10-12, 1987 at the University of Washington in Seattle, Washington. It also contains several interesting papers not presented at the workshop. Many of the workshop papers have been expanded or modified to include updated results and references to other papers in this volume.

Because representation and reasoning under uncertainty are still poorly enough understood that implementation choices and tradeoffs are best understood in specific applications, most of the papers in this volume address multiple issues in uncertain reasoning. We have classified the papers in this volume, according to their major focus, into one of four categories:

- Interpretation and comparison of uncertainty calculi
- Representation and computation in Bayesian Inference
- Structure and control for systems reasoning under uncertainty
- Learning and explanation

The papers in this volume on interpreting and comparing uncertainty calculi reflect significant progress in this area. In contrasting papers from the first volume of this series to those of the current volume, it is possible to note a much more ecumenical spirit in the interpretation and comparison. The tone of religious fervor about "competing calculi" evident in the first volume has given way to reasoned comparisons on such AI issues as representation,semantics, knowledge acquisition, complexity, control, and explanation, as coherence, and completeness. Most authors now agree that any uncertainty calculus can probably "do it all" in some sort of Turing sense, but admit each calculus has its own peculiar advantages and disadvantages.

The special attention to Bayesian techniques represents the relative maturity of Bayesian methods relative to other calculi, rather than a general consensus about the superiority of Bayesian methods. This maturity results from <u>over</u> 200 years of research in probability and is evidenced by the existence of a well developed decision theory. Because of this maturity, probabilities are often the choice of application developers, which in turn results in a large number of papers on Bayesian techniques.

Structure and control are a major topic in uncertain reasoning for the same reasons that they are in any AI system: the process of uncertain reasoning is simply too complex to compute everything. For the same reasons that a chess playing program cannot examine all possible moves, an uncertain reasoning system cannot examine every facet of every possible conclusion.

Learning and explanation are equally important and probably even more difficult in uncertain reasoning systems than they are in any other AI system. An uncertain reasoning system must learn not only the facts that must also be learned by other AI systems, but also must learn the uncertainty associated with these facts. Because of the uncertainties involved in such systems, their actions, inferences, decisions and recommendations are even harder to understand than in other AI systems. This makes explanation even more critical.

It is clear from the papers presented here and in the previous volumes of this series that there are still many fundamental open issues in the area of uncertain reasoning. It is our hope that this collection of papers will provide motivation and assistance to those who wish to explore these open issues in an area whose recognized importance continues to grow.

Laveen N. Kanal, College Park, MD
John F. Lemmer, Rome, N.Y.
Tod S. Levitt, Mt. View, CA

CONTRIBUTORS

Tejwansh S. Anand, Philips Laboratories,
Briarcliff Manor, N.Y. 10520

Leonard J. Bertrand, Strategic Decisions Group,
Menlo Park, CA 94025

Thomas O. Binford, Stanford University,
Stanford, VA 94305

Gautam Biswas, Vanderbilt University,
Nashville, TN 37235

Piero P. Bonissone, GE Corporate Research & Development,
Schenectady, N.Y. 12301

Lashon B. Booker, Naval Research Laboratory,
Washington, DC 20375

Wray L. Buntine, S.O.C.S. - U.T.S.,
Australia, 2007

Richard A. Caruana, Philips Laboratories,
Briarcliff Manor, N.Y. 10520

Homer L. Chin, Stanford University,
Stanford, CA 94305

Paul Cohen, University of Massachusetts,
Amherst, MA 01003

Gregory F. Cooper, Stanford University,
Stanford, CA 94305

Bruce D'Ambrosio, Department of Computer Science,
Oregon State University

Norman Dalkey, UCLA Computer Science Department,
Los Angeles, CA 90024

David M. Eddy, Duke University,
Durham, N.C. 27706

Christopher Elsaesser, The Mitre Corporation,
McLean, VA 22102

Greg Hager, University of Pennsylvania,
Philadelphia, PA 19104

Vic Hasselblad, Duke University,
Durham, N.C. 27706

David Heckerman, Stanford Medical Center,
Menlo Park, CA 94025

Gavin Hemphill, Defense Research Establishment Atlantic,
Dartmouth, N.S., Canada B2Y3Z7

Max Henrion, Carnegie Mellon University,
Pittsburgh, PA 15217

Eric J. Horvitz, Stanford University,
Stanford, CA 94305

Naveen Hota, JAYCOR,
Vienna, VA 22180

Holly Jimison, Stanford Medical Center,
Stanford, CA 94305

Henry E. Kyburg, Jr., University of Rochester,
Rochester, N.Y. 14627

Tod S. Levitt, Advanced Decision Systems,
Mountain View, CA 94305

Wallace B. Mann, Stanford University,
Stanford, CA 94305

Mary McLeish, University of Guelph,
Guelph, Ontario, Canada

Max Mintz, University of Pennsylvania,
Philadelphia, PA 19104

Eric M. Neufeld, University of New Brunswick,
Fredericton, N.B. E3B 5A3

Judea Pearl, UCLA Computer Science Department,
Los Angeles, CA 90024

Bruce M. Perrin, McDonnell Douglas Astronautics Co.,
St. Louis, MO 63166

David Poole, University of British Columbia,
Vancouver, Canada T6G 2E1

George Rebane, UCLA Computer Science Department,
Los Angeles, CA 90024

Ross D. Shachter, Stanford University,
Stanford, CA 94305

Glen Shafer, University of Kansas,
Lawrence, Kansas 66045

Thomas B. Slack, PAR Government Systems Corporation,
New Hartford, N.Y. 13413

David F. Spiegelhalter, MRC Biostatistics Unit,
Cambridge CB2 2BW

Spenser Star, Carnegie Mellon University,
Pittsburgh, PA 15213

David S. Vaughn, McDonnell Douglas Research Laboratory,
St. Louis, MO 63166

Ben P. Wise, Thayer School of Engineering, Dartmouth College,
Hanover, N.H. 03755

Robert Wolpert, Duke University,
Durham, N.C. 27706

Nancy C. Wood, GE Corporate Research & Development,
Schenectady, N.Y. 12301

Lei Xu, Tsinghua University,
Beijing, China

Robert M. Yadrick, McDonnell Douglas Research Laboratory,
St. Louis, MO 63166

John Yen, USC/Information Science Institute,
Marina del Ray, CA 90292

CONTENTS

reface v

Contributors xi

I. INTERPRETATION AND COMPARISON

An Algorithm for Computing Probabilistic Propositions
G.F. Cooper 3

Higher Order Probabilities
H.E. Kyburg, Jr. 15

Nilsson's Probabilistic Entailment Extended to Dempster-Shafer Theory
M. McLeish 23

Towards Solving the Multiple Extension Problem: Combining Defaults and Probability
E. Neufeld and D. Poole 35

Satisfaction of Assumptions is a Weak Predictor of Performance
B.P. Wise 45

Evaluation of Uncertain Inference Models III: The Role of Tuning
B.P. Wise, B.M. Perrin, D.S. Vaughan, and R.M. Yadrick 55

Can Evidence be Combined in the Dempster-Shafer Theory 63
J. Yen

II. REPRESENTATION AND COMPUTATION IN BAYESIAN INFERENCE

Bayesian Inference in Model-Based Machine Vision
T.O. Binford, T.S. Levitt, and W.B. Mann 73

Computing Belief Commitments Using Tensor Products
L.B. Booker, N. Hota, and G. Hemphill 97

Decision Tree Induction Systems: A Bayesian Analysis
W. Buntine 109

Bayesian Belief Network Inference Using Simulation
H.L. Chin and G.F. Cooper 129

A Bayesian Perspective on Confidence
D. Heckerman and H. Jimison 14

Some Practical Issues in Constructing Belief Networks
M. Henrion 161

The Recovery of Causal Poly-Trees from Statistical Data
G. Rebane and J. Pearl 175

A Heuristic Bayesian Approach to Knowledge Acquisition: Application to the Analysis of Tissue-Type Plasminogen Activator
R.D. Shachter, D.M. Eddy, V. Hasselblad, and R. Wolpert 183

Advantages and a Limitation of Using LEG Nets in a Real Time Problem
T.B. Slack 191

A Unified Approach to Imprecision and Sensitivity of Beliefs in Expert Systems
D.J. Spiegelhalter 199

Structuring Causal Tree Models with Continuous Variables
L. Xu and J. Pearl 209

III. STRUCTURE AND CONTROL

Using the Dempster-Shafer Scheme in a Mixed-Initiative Expert System Shell
G. Biswas and T.S. Anand 223

T-Norm Based Reasoning in Situation Assessment Applications
P.P. Bonissone and N.C. Wood 241

Steps Towards Programs that Manage Uncertainty
P.R. Cohen 257

A Hybrid Approach to Reasoning Under Uncertainty
B. D'Ambrosio 267

Estimation Procedures for Robust Sensor Control
G. Hager and M. Mintz 285

Reasoning about Beliefs and Actions Under Computational Resource Constraints
E.J. Horvitz 301

Efficient Inference on Generalized Fault Diagrams
R.D. Shachter and L.J. Bertrand 325

Implementing Evidential Reasoning in Expert Systems
J. Yen 333

IV. LEARNING AND EXPLANATION

The Automatic Training of Rule Bases that Use Numerical Uncertainty Representations
R.A. Caruana 347

Modifiable Combining Functions
P.R. Cohen, G. Shafer, and P.P. Shenoy 357

The Inductive Logic of Information Systems
N.C. Dalkey 375

Explanation of Probabilistic Inference
C. Elsaesser 387

Theory-Based Inductive Learning: An Integration of Symbolic and Quantitative Methods
S. Star 401

I

INTERPRETATION AND COMPARISON

Uncertainty in Artificial Intelligence 3
L.N. Kanal, T.S. Levitt, and J.F. Lemmer (Editors)

AN ALGORITHM FOR COMPUTING PROBABILISTIC PROPOSITIONS

Gregory F. Cooper

Medical Computer Science Group
Medical School Office Building
Stanford University
Stanford, California 94305-5479

A method for computing probabilistic propositions is presented. It assumes the availability of a single external routine for computing the probability of one instantiated variable*, given a conjunction of other instantiated variables. In particular, the method allows belief network algorithms to calculate general probabilistic propositions over nodes in the network. Although in the worst case the time complexity of the method is exponential in the size of a query, it is polynomial in the size of a number of common types of queries.

1. Introduction

This paper presents an algorithm for computing the probability of a propositional logic sentence in the context of a set of known probabilities. Probability inference algorithms have typically been developed to calculate $P(S_1 \mid S_2)$, where S_1 is either a single instantiated variable or a conjunction of instantiated variables, and S_2 is a conjunction of instantiated variables [1, 2, 3, 4, 5]. When S_1 is a single instantiated variable, we call these *single variable* (SV) *algorithms*.

We will extend probability queries to the case where S_1 and S_2 are well-formed formulas in propositional logic (propositions). Thus, it will be possible to apply disjunction, conjunction, and negation to variables in both the conditioning and the conditioned part of a probability query.

* The term *instantiated variable* is used to denote a variable with a known, assigned value.

The Propositional Probability Query (PPQ) algorithm for performing these type calculations is a simple procedure based on calls to an SV algorithm. Thus, implementations of SV algorithms, such belief network algorithms [3, 4, 5, 6], can be easily augmented to answer more general propositional probability queries. Furthermore, PPQ can usually answer queries much more efficiently than a brute-force technique that explicitly sums over the entire joint probability space of the model variables.

PPQ is a specialization of general probabilistic logic inference algorithms [7]. It handles only propositional logic rather than first order predicate logic. PPQ also assumes that there is sufficient probabilistic information for an SV algorithm to compute a unique (point) probability, rather than an upper and lower bound (i.e., interval) probability.

2. The Principal Steps in the Algorithm

There are four main steps underlying PPQ, as shown below.* None of the steps assumes that variables are binary. Thus, PPQ can answer queries that contain multi-valued variables.

Step 1. Convert a conditional query into two marginal queries.
$P(S_1 \mid S_2)$ can be expressed as $P(S_1 \wedge S_2)/P(S_2)$. Therefore, the task of computing conditional propositional probability queries is readily decomposed into computing two marginal propositional probabilities.

Step 2. Eliminate disjunctions.
The next step is to transform a marginal propositional probability P(S) into a form P(S') in which S' contains only conjunctions and negations. That is, disjunctions are transformed into equivalent expressions that contain only conjunctions and negations. This can be done by successive applications of de Morgan's law, namely, $X_1 \vee X_2 \vee \cdots \vee X_n \Rightarrow \overline{\overline{X_1} \wedge \overline{X_2} \wedge \ldots \wedge \overline{X_n}}$.

* In all four steps we use the symbol $\wedge$ to represent conjunction, $\vee$ to represent disjunction, and an overbar to represent logical negation.

Step 3. Eliminate negations.
In the third step, we simplify S' to a conjunction of instantiated variables by recursively removing all negation operators in S'. For example, suppose $S' = \overline{S'_1 \wedge S'_2} \wedge S'_3$, where S'_1, S'_2, and S'_3 are propositions. In order to simplify this expression, the negation operator spanning S'_1 and S'_2 can be removed as follows:

$$P(S') = P(\overline{S'_1 \wedge S'_2} \wedge S'_3)$$

$$= P(\overline{S'_1 \wedge S'_2} \mid S'_3)\, P(S'_3)$$

$$= (1 - P(S'_1 \wedge S'_2 \mid S'_3))\, P(S'_3)$$

$$= (1 - \frac{P(S'_1 \wedge S'_2 \wedge S'_3)}{P(S'_3)})\, P(S'_3)$$

$$= P(S'_3) - P(S'_1 \wedge S'_2 \wedge S'_3)$$

The terms $P(S'_3)$ and $P(S'_1 \wedge S'_2 \wedge S'_3)$ in the last line above are the only terms which must be computed; the derivation merely clarifies how the final line was obtained. Thus, the key to Step 3 is the use of a simple elementary probability relation, namely, $P(\overline{X} \wedge Y) = P(Y) - P(X \wedge Y)$, where X and Y are arbitrary propositions. (In the special case that Y = T, it follows that $P(\overline{X} \wedge Y) = P(\overline{X}) = 1 - P(X)$.) In the current example, recursive application of this simplification rule to $P(S'_3)$ and $P(S'_1 \wedge S'_2 \wedge S'_3)$ ultimately yields probabilities with terms that consist only of a conjunction of instantiated variables. We can then apply Step 4 to determine the value of each of these probability terms and thus the value of P(S').

Step 4. Compute joint probabilities.
If S' consists only of a conjunction of n instantiated variables of the form $X_1 \wedge X_2 \wedge \ldots \wedge X_n$, then by application of the chain-rule of conditional probabilities we know that:

$$P(S') = P(X_1 \mid X_2 \wedge \ldots \wedge X_n)\, P(X_2 \mid X_3 \wedge \ldots \wedge X_n) \ldots P(X_{n-1} \mid X_n)\, P(X_n)$$

Note that each of these terms can be computed by an SV algorithm.

3. An Example

As an example, consider a query to calculate $P(X_1 \vee X_2 \mid X_3 \vee \overline{X_4})$. For simplicity, we will assume these are binary variables. The steps PPQ uses to answer this query are shown in Figure 1. Because the query variables in the example are all binary, it is acceptable to stop applying Step 3 when the only negation operators that remain in an expression are those that scope over single variables. However, for the more general case, we would apply Step 3 until no negation operators remain within *any* expression. Notice that the expression $P(\overline{X_3} \mid X_4)\, P(X_4)$ occurs twice in Figure 1. This demonstrates that caching some of the subcalculations may improve the efficiency of the algorithm. Caching would be most efficient if it occurred among expressions higher in the tree, in order to prune redundant calculations before they terminate at the leaf-node stage. For example, by storing the value of $P(\overline{X_3} \wedge X_4)$ after it is first calculated, we can eliminate the need to calculate this probability a second time, as is done in Figure 1.

4. Time Complexity Analysis

4.1. Worst Case Analysis

The worst-case computational time complexity of PPQ is $O(m\ 2^{2m}\, g(n))$, where m is the number of variable references in a query, n is the total number of variables in the knowledge base, and g(n) is the worst-case time complexity for a given SV algorithm to compute an SV probability using a knowledge base of size n. In the worst-case, the SV algorithm calculation, as reflected in g(n), can be quite expensive. For example, for belief networks, this inference problem is known to be NP-hard [8]. On the other hand, there are SV algorithms with an O(n) time complexity for special belief network topologies, such as networks with only one pathway between any two nodes [5]. However, the main issue we consider here in the analysis of PPQ is the time complexity incurred due to calculations other than those of the SV algorithm. This complexity is reflected in the term $m\ 2^{2m}$. Therefore, the focus of our complexity analysis will be on the derivation of $O(m\ 2^{2m})$ as a worst-case result for the number of calls to an SV algorithm, for a query of size m.

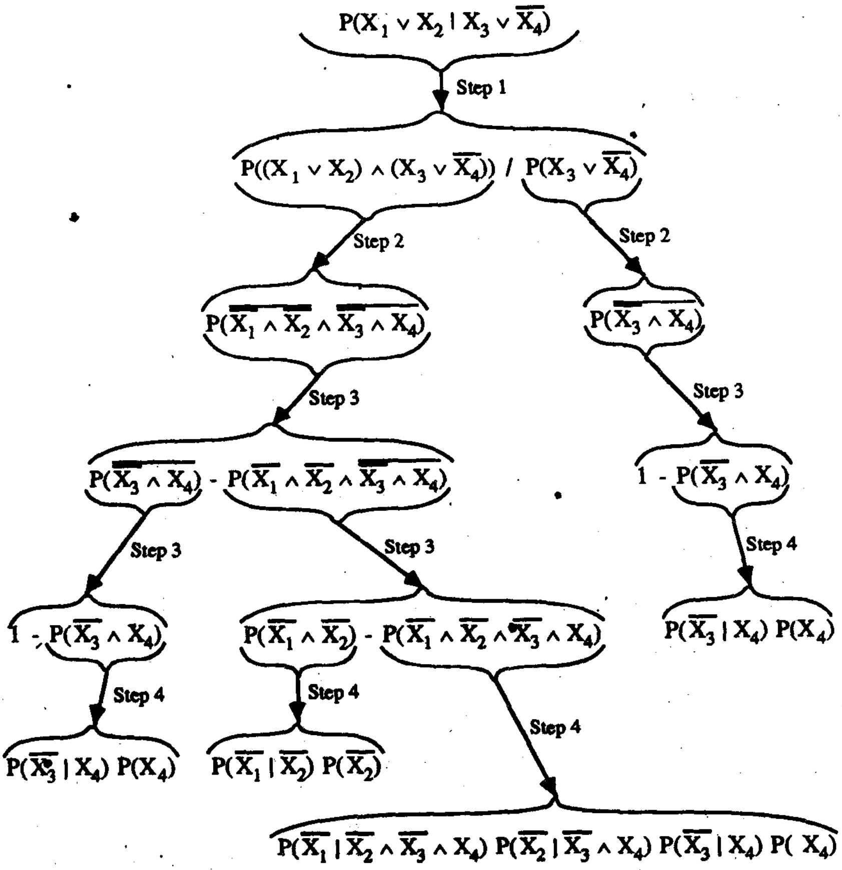

Figure 1

Application of the four steps in PPQ to decompose a query at the root node into queries at the leaf nodes that can be computed using an SV algorithm.