International Journal of Approximate Reasoning 68 (2016) 211–229

Contents lists available at ScienceDirect

International Journal of Approximate Reasoning

www.elsevier.com/locate/ijar

Equivalences between maximum a posteriori inference in Bayesian networks and maximum expected utility computation in influence diagrams

Denis Deratani Mauá

Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, Brazil

ARTICLE INFO

Article history:
Received 16 December 2014
Received in revised form 18 March 2015
Accepted 19 March 2015
Available online 30 March 2015

Keywords:
Bayesian networks
Maximum a posteriori inference
Influence diagrams
Maximum expected utility
Computational complexity

ABSTRACT

Two important tasks in probabilistic reasoning are the computation of the maximum posterior probability of a given subset of the variables in a Bayesian network (MAP), and the computation of the maximum expected utility of a strategy in an influence diagram (MEU). Both problems are $\mathrm{NP}^{\mathrm{PP}}$-hard to solve, and NP-hard to approximate when the treewidth of the underlying graph is bounded. Despite the similarities, researches on both problems have largely been conducted independently, with algorithmic solutions and insights designed for one problem not (trivially) transferable to the other one. In this work, we show constructively that these two problems are equivalent in the sense that any algorithm designed for one problem can be used to solve the other with small overhead. Moreover, the reductions preserve the boundedness of treewidth. Building on the known complexity of MAP on networks whose parameters are imprecisely specified, we show how to use the reductions to characterize the complexity of MEU when the parameters are set-valued. These equivalences extend the toolbox of either problem, and shall foster new insights into their solution.

1. Introduction

Bayesian networks are graphical representations of complex probabilistic situations. Nodes in a Bayesian network represent variables, and probabilistic (in)dependences can be deduced from the underlying graph in a computationally efficient way [1]. A Bayesian network provides a compact specification of a multivariate joint probability distribution, from which any inference on the domain can in principle be answered.

Maximum a posteriori inference (MAP) consists in finding a configuration of a certain subset of the variables that maximizes the posterior probability distribution induced by a Bayesian network [2].[1] MAP has applications, for example, in diagnostic systems and classification of relational and sequential data [3,4]. Solving MAP is computationally difficult, and the literature contains a plethora of approximate solutions with varying degrees of efficiency and accuracy [5–12].

Influence diagrams extend Bayesian networks with preferences and actions to cope with decision making situations [13, 14]. The maximum expected utility problem (MEU) is to select a mapping from observations to actions that maximizes the expected utility as defined by an influence diagram. MEU appears, for example, in troubleshooting and active sensing [14].

E-mail address: denis.maua@usp.br.

[1] We consider here the most general case of MAP inference, also known as Marginal MAP and Partial MAP inference.

http://dx.doi.org/10.1016/j.ijar.2015.03.007
0888-613X/

Although MEU is computationally difficult to solve, it counts with a large number of approximate solutions, ranging from local search to branch-and-bound approaches [15–24].

The (decision versions of the) MAP and MEU problems are complete for the complexity class NP^{PP} [2,18], which implies that any algorithm designed to solve one problem can *in principle* be used to solve the other.[2] Moreover, both problems are NP-complete when the treewidth of the underlying graph is assumed bounded [25–27].[3] In practice, however, these two problems have been investigated independently, with only a few similarities arising in the design of algorithms such as the use of clique-tree structures and message-passing for fast probabilistic inference [9–12,17,22].

In this work we provide *constructive* proofs of the equivalences between (the functional versions of) these two problems. We start by presenting background knowledge on graphs (Section 2), Bayesian networks (Section 3) and influence diagrams (Section 4), and formalizing the MAP and MEU problems. Then, we design a polynomial-time reduction that maps MAP problems into MEU problems (Section 5). We show that the reduction increases the treewidth of the underlying graphs by at most four, which makes the reduction closed in NP. We proceed to build two different polynomial-time reductions of MEU into MAP problems (Section 6). One reduction increases treewidth by at most five, while the other increases treewidth by at most three. Thus both reductions map MAP problem instances in NP into MEU problems also in NP. We evaluate empirically the performance of the reductions on MEU problem instances either directly by using state-of-the-art MEU solvers, or by using the reductions to create MAP instances which are subsequently solved by state-of-the-art MAP solvers (Section 8). The experiments show that MAP solvers can often solve the MEU problem more accurately and faster than MEU solvers; they also verify the practical feasibility of the reductions.

When data is scarce or conflicting, eliciting the conditional probabilities of the models can be difficult. Section 7 describes credal networks and credal influence diagrams, which extend, respectively, Bayesian networks and influence diagrams to cope with imprecision in the quantification of the model. De Campos and Cozman [28] showed that a variant of the MAP problem in credal networks of bounded treewidth is Σ_2^P-hard. By combining this result with the MAP to MEU reduction developed here, we show in Section 7.1 that a variant of the MEU problem in influence diagrams whose numerical parameters are set-valued is Σ_2^P-hard.

We conclude this document in Section 9 with an overview of the results, a brief discussion on some shortcomings of the reductions developed here, and a comment on possible extensions to this work.

We assume in the sequel that the reader is familiar with computational complexity theory, complexity classes such as NP, PP, the Polynomial Hierarchy (and in particular, the class Σ_2^P), and related concepts such as oracle languages, completeness and hardness.

2. Some useful concepts from graph theory

Consider a directed graph with nodes X and Y. A node X is a *parent* of Y if there is an arc going from X to Y, in which case we say that Y is a *child* of X. The *in-degree* of a node is the number of parents of it. We denote the parents of a node X by $\text{pa}(X)$ and its children by $\text{ch}(X)$. The family $\text{fa}(X)$ of a node X comprises the node itself and its parents. A *polytree* is a directed acyclic graph (DAG) which contains no undirected cycles. A DAG is *loopy* if it is not a polytree. Polytrees are important, as they are among the simplest structures, and probabilistic inference can be performed efficiently in polytree-shaped Bayesian networks of bounded in-degree.

The *moral graph* of a DAG is the undirected graph obtained by connecting nodes with a common child and dropping arc directions. The moral graph of a DAG might contain (undirected) cycles even when the DAG itself does not (e.g., any polytree with maximum in-degree greater than one).

A *tree decomposition* of an undirected graph G is a tree T such that

1. each node i associated to a subset $\mathcal{X}_i$ of nodes in G;
2. for every edge X–Y of G there is a node i of T whose associated node set $\mathcal{X}_i$ contains both X and Y;
3. for any node X in G the subgraph of T obtained by considering only nodes whose associated sets contain X is a tree.

The third property is known as the *running intersection property*. A *clique* is a set of pairwise connected nodes of an undirected graph. Any tree decomposition of a graph contains every clique of it included in some of the associated node sets [29]. The *treewidth* of a tree decomposition is the maximum cardinality of a node set minus one, that is, it equals $\max_i |\mathcal{X}_i| - 1$. The treewidth of a graph G is the minimum treewidth over all tree decompositions of it. The treewidth of a directed graph is the treewidth of its corresponding moral graph. Polytrees have treewidth given by the maximum in-degree of a node. We say that a class of graphs has bounded treewidth if there is a constant that upper bounds the treewidth of any graph in that class. For example, the class of polytrees with in-degree at most four has bounded treewidth, since any graph in the class has treewidth at most four.

The *removal* of a node X obtains a new graph where X and its incident edges are missing. The *elimination* of a node X from a graph G produces a graph G' by removing X and pairwise connecting all its neighbors. A node is *simplicial* if all its

[2] We assume here that the number of incoming arcs into any decision node in an influence diagram is logarithmically bounded by the number of variables, which limits the size of strategies to a polynomial in the input size.

[3] The treewidth of a graph is a measure of its similarity to a tree; we formalize this definition later.