Artificial Intelligence ••• (••••) •••–•••

Contents lists available at ScienceDirect

Artificial Intelligence

www.elsevier.com/locate/artint

# Integer Linear Programming for the Bayesian network structure learning problem

Mark Bartlett *, James Cussens

*Department of Computer Science, University of York, York, UK*

ARTICLE INFO

*Article history:*
Received in revised form 20 February 2015
Accepted 7 March 2015
Available online xxxx

*Keywords:*
Bayesian networks
Integer Linear Programming
Constrained optimisation
Cutting planes
Separation

ABSTRACT

Bayesian networks are a commonly used method of representing conditional probability relationships between a set of variables in the form of a directed acyclic graph (DAG). Determination of the DAG which best explains observed data is an NP-hard problem [1]. This problem can be stated as a constrained optimisation problem using Integer Linear Programming (ILP). This paper explores how the performance of ILP-based Bayesian network learning can be improved through ILP techniques and in particular through the addition of non-essential, implied constraints. There are exponentially many such constraints that can be added to the problem. This paper explores how these constraints may best be generated and added as needed. The results show that using these constraints in the best discovered configuration can lead to a significant improvement in performance and show significant improvement in speed using a state-of-the-art Bayesian network structure learner.



## 1. Introduction

Bayesian networks (BNs) use a directed acyclic graph (DAG) to represent conditional probability relationships between a set of variables. Each node in the network corresponds to one of the variables. Edges show conditional dependencies between these variables such that the value of any variable is a probabilistic function of the values of the variables which are its parents in the DAG.

While one can analytically create a BN from expert knowledge, there is considerable interest in learning Bayesian networks in which the relationship between the variables is not known. In this setting, multiple joint observations of the variables are first taken and then a BN structure that best explains the correlations in the data is sought. This is known as the Bayesian network structure learning problem. For any reasonably sized problem. the number of possible structures is far too large to evaluate each individually. Therefore a more intelligent alternative is needed.

In this paper, we tackle the Bayesian network structure learning problem using the score-and-search approach. Each possible parent set of each variable is first given a score based on the correlations between these variables in the observed data. A search algorithm is then used to determine which combination of these parent sets yields the DAG with the optimal overall score. As this search is an NP-hard problem [1], an intelligent search strategy is needed in order to efficiently optimise the BN structure for large numbers of variables.

* Corresponding author.
*E-mail addresses:* mark.bartlett@york.ac.uk (M. Bartlett), james.cussens@york.ac.uk (J. Cussens).
*URLs:* http://www.cs.york.ac.uk/~bartlett (M. Bartlett), http://www.cs.york.ac.uk/~jc (J. Cussens).

http://dx.doi.org/10.1016/j.artint.2015.03.003

The search for the best BN can be viewed as a constrained optimisation problem; select the parent sets for variables with the highest combined score subject to the constraint that these form an encoding of a DAG. Specifically, there are two constraints that must be respected. First, there must be exactly one parent set chosen for each variable. Second, there must be no (directed) cycles in the graph. Furthermore, it is possible to write the score which is to be optimised and both of these constraints as linear functions of binary variables, which means that the problem of learning the best BN can be formulated as an Integer Linear Programming (ILP) problem [2,3]. Formulating the problem in such a way means that highly optimised off-the-shelf ILP solvers can be used and that decades of research in ILP optimisation can be used to improve the speed of the search for the optimal BN.

Encoding the constraint that there is only one parent set for each node is straightforward. However, the constraint that there must be no cycles in the network is relatively complex to encode as a linear inequality and can either be enforced through the introduction of auxiliary variables and constraints [4] or through an exponential number of *cluster constraints* [2]. Previous work has revealed that these cluster constraints perform better in practice and so the current paper focuses on this encoding. As there are so many of these cluster constraints, we do not add them all initially, but rather add them to the problem as additional constraints as needed. That is to say, we first solve a relaxed version of the problem in which most of the acyclicity constraints are not present. We then identify some acyclicity constraints violated by this solution and add them to the problem before resolving. This process is called *separation*, as the added constraints separate the relaxed solution from the space of valid solutions, and the added constraints are known as *cuts* or *cutting planes* as they cut off a portion of the search space containing the relaxed solution. This process repeats until the solution found does not violate any additional acyclicity constraints. By so doing, we typically eliminate the need for most constraints which rule out cycles to ever be explicitly represented in the problem and so increase the solving speed of the problem and simultaneously reduce the memory needed.

In addition to the constraints necessary to define the problem, there are additional implied constraints that can also be added. Doing so may lead to an increase in performance through further constraining the search space or may prove detrimental by increasing the number of constraints that need to be generated and processed at each step.

The contribution of the current paper is to examine several extensions to the existing ILP based method which relate to improving the constraints generated and added during the search. The first extension examines the method by which we search for acyclicity constraints to add, the second introduces additional implied constraints of a different form, and the third attempts to ensure that the constraints found by other methods rule out greater invalid regions of the search space. In addition, the impact of several solver features is assessed.

The rest of this paper is arranged as follows. In Section 2, the problem of Bayesian network learning is addressed in more detail before looking at using Integer Linear Programming for this task. A software platform to carry out this learning is presented in Section 3. The novel contributions of this paper are presented in Section 4 before evaluation of these techniques are given in Section 5. Finally, Section 6 concludes.

## 2. Background

### 2.1. Bayesian network learning

There are two classes of methods for learning the structure of Bayesian networks. The first takes advantage of the fact that the structure of the network encodes information about conditional independence. One can perform multiple conditional independence tests on subsets of variables and use this information to infer what structure the BN should have.

The alternative method, and the one followed in this paper, is *score-and-search*. In this method, for each node, one computes scores for each possible set of parent nodes for the node and then uses some search algorithm to attempt to maximise a global score formed from the local scores, subject to the resulting network being acyclic.

There are many scores that have been proposed for learning BNs, for example BDeu [5], BIC [6], AIC [7]. These scores have the property of local decomposability, meaning that the global score can be found as a simple function of the score associated with each node. In the current paper, we restrict ourselves to consideration of the BDeu score, though we note that the software presented has been used to learn networks based on other scores [8–10].

Having produced local scores for the possible parent sets of each node, it is necessary to perform a search for the network with the maximum global score. This can be performed using any search method. These can be divided into heuristic methods that produce a high scoring network but cannot guarantee to produce the best one, and global searches that not only find the best network but also establish that no better network is possible. The work presented in this paper falls into this latter category, alongside recent approaches such as dynamic programming [11], A* search [12] and Branch-and-Bound [13]. As the quality of the learned network is identical for all exact methods, the primary challenge in this case is to produce a search algorithm that runs sufficiently quickly and is sufficiently scalable. Another recent approach [14,15] also uses Integer Linear Programming to find an optimal Bayesian network, but as this has the added constraint of bounded tree-width, this is not directly comparable to the results presented here.

If there are $n$ nodes in the BN, then the number of possible parent sets for each node is $2^{n-1}$. In practice, for even relatively modest $n$, this is much too large to even score each of them in a practicable time, and probably creates a search space that is too large to explore effectively. In most cases it is possible to show that many parent sets cannot occur in an optimal BN and so can be pruned [16]. This speeds up scoring considerably. However even after pruning there typically