Hierarchical Bayesian Networks: A Probabilistic Reasoning Model for Structured Domains

Elias Gyftodimos GYFTODIM@CS.BRIS.AC.UK
Peter A. Flach PETER.FLACH@BRISTOL.AC.UK
Machine Learning group, Computer Science department, University of Bristol, UK

Abstract

Bayesian Networks are being used extensively for reasoning under uncertainty. Inference mechanisms for Bayesian Networks are compromised by the fact that they can only deal with propositional domains. In this work, we introduce an extension of that formalism, Hierarchical Bayesian Networks, that can represent additional information about the structure of the domains of variables. Hierarchical Bayesian Networks are similar to Bayesian Networks, in that they represent probabilistic dependencies between variables as a directed acyclic graph, where each node of the graph corresponds to a random variable and is quantified by the conditional probability of that variable given the values of its parents in the graph. What extends the expressive power of Hierarchical Bayesian Networks is that a node may correspond to an aggregation of simpler types. A component of one node may itself represent a composite structure; this allows the representation of complex hierarchical domains. Furthermore, probabilistic dependencies can be expressed at any level, between nodes that are contained in the same structure.

1. Introduction

Bayesian Networks [Pear88] are a useful tool for probabilistic inference, but are unable to deal with non-propositional domains. Aggregate data types can be represented either using a flattened representation, which results in loss of expressivity, or by representing the aggregate type as a single node, disregarding possible probabilistic dependencies between the components of the structure.

In this paper we introduce an extension of Bayesian Networks, namely *Hierarchical Bayesian Networks*, which are a representation formalism for probabilistic independencies between variables that belong to structured domains. Probabilistic inference mechanisms of standard Bayesian Networks can generally be extended for Hierarchical Bayesian Networks as well. We begin with a very brief presentation of Bayesian Networks in the next section. In section 3 we provide an insight to Hierarchical Bayesian Networks, followed by basic definitions of the model and operations we apply. Section 4 contains an overview of inference in Bayesian Networks and some preliminary ideas about how existing methods may be adapted for the hierarchical case. We conclude by giving an outline of our current track of work.

2. Bayesian networks

2.1. Basic theory

A Bayesian network is a directed acyclic graph where nodes correspond to random variables and arcs between nodes represent probabilistic dependencies. From a simplifying perspective, an arc pointing from node A to node B can be perceived as A *causing* or *influencing* B.

Each node in the network is annotated with a conditional probability table, that represents the conditional probability of the variable given the values of its parents in the graph. For nodes that have no parents, the corresponding table will simply contain the prior probabilities for that variable.

A Bayesian Network is a compact representation of the full joint probability of the random variables in the graph. The joint probability of n variables can be expressed using a table whose size is of order $O(2^n)$. In a Bayesian Network, the joint probability is expressed in factorised form, and conditional independencies are exploited in order to simplify the posterior probability expressions. Hence, it is enough to store n separate tables of size $O(2^k)$, where k is the maximum number

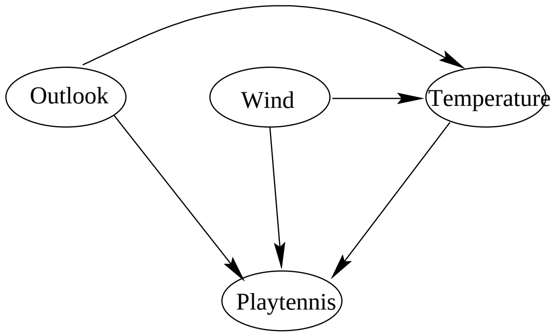


Figure 1. The PlayTennis decision example.

of incoming arrows in a node.

The main property of Bayesian Networks can be loosely expressed in the following way: A node is conditionally independent of its non-descendents, given the values of its immediate parents. A strict definition of the independencies that a Bayesian Network encodes uses the following criterion of *d-separation* [Pear88]:

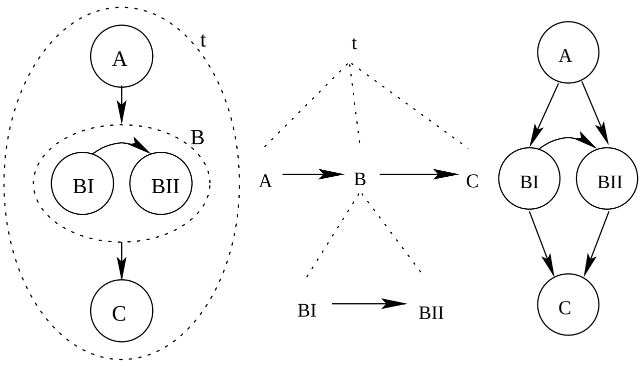
Definition 2.1 (d-separation) *Let X, Y, Z be three disjoint set of vertices in a directed acyclic graph. Z is said to* d-separate *X from Y if for every undirected path between any $x \in X$ and $y \in Y$ there exists a node w on that path, such that either of the following holds:*

- *w does not have converging arrows (i.e., along the path connecting x and y) and w is in Z*
- *w has converging arrows and neither w or any of its descendants in the graph is in Z*

For X, Y, Z being three sets of nodes in a Bayesian Network, if Z d-separates X from Y then $P(X|Y,Z) = P(X|Z)$, i.e. X is independent of Y given Z.

2.2. Example: PlayTennis

Let us consider the following example: We wish to describe whether or not a particular day is appropriate for a specific individual to enjoy her favourite sport, depending on weather conditions. Assume that we observe that the value of the boolean variable PlayTennis is correlated to sky outlook, wind and temperature (which all have discrete domains), and also we observe a strong correlation between sky outlook and temperature, as well as between wind and temperature. This dependency model can be decomposed as the Bayesian Network shown in Figure 1.



(a) (b) (c)

P(A)	a1	a2
	0.3	0.7

P(BII|A,BI)	bII1	bII2	bII3
a1bI1	0.3	0.4	0.3
a1bI2	0.1	0.5	0.4
a2bI1	0.7	0.1	0.2
a2bI2	0.9	0.1	0

P(BI|A)	bI1	bI2
a1	0.4	0.6
a2	0.8	0.2

P(C|BI,BII)	c1	c2	c3
bI1bII1	0.2	0.6	0.2
bI1bII2	0.3	0.4	0.3
bI1bII3	0.2	0.2	0.6
bI2bII1	0.5	0.3	0.2
bI2bII2	0.6	0.2	0.2
bI2bII3	0.7	0.2	0.1

(d)

Figure 2. A simple Hierarchical Bayesian Network. (a) Nested representation of the network structure. (b) Tree representation of the network structure. (c) Standard Bayesian Network expressing the same dependencies. (d) Probabilistic part.

3. Hierarchical Bayesian Networks

3.1. Intuition

Hierarchical Bayesian Networks are a generalisation of standard Bayesian Networks, where a node in the network may be an aggregate data type. This allows the random variables of the network to represent arbitrarily structured types. Within a single node, there may also be links between components, representing probabilistic dependencies among parts of the structure.

Hierarchical Bayesian Networks encode conditional probability dependencies the same way as standard Bayesian Networks. Hierarchical Bayesian Networks can express further knowledge about variable structure and use that knowledge to build more realistic probabilistic models.

A Hierarchical Bayesian Network consists of two parts.

The *structural part* contains the variables of the network and describes the *part-of* relationships and the *probabilistic dependencies* between them. The *part-of* relationships in a structural part may be illustrated either as nested nodes (Fig. 2(a)) or as a tree hierarchy (Fig. 2(b)). The second part of a Hierarchical Bayesian Network, the *probabilistic part*, contains the conditional probability tables that quantify the links introduced at the structural part.

3.2. Example: PlayGolf

We will demonstrate how a Hierarchical Bayesian Network can express dependencies in structured domains. Consider a random variable *PlayGolf* that expresses the probability that a particular day is appropriate for a given individual to exercise her hobby. That decision is independently influenced by the weather and by the individual's mood according to business affairs. Furthermore, assume that *Weather* is a triple of random variables *(Outlook, Temperature, Wind)* and *Business* is a pair of random variables *(Market, Meeting)* where *Market* consists itself of the pair *(Currency, Shares)*.

A hypothetical configuration of the probabilistic dependencies between those variables is illustrated as a Hierarchical Bayesian Network (structural part only shown) in Figure 3(a).

We can observe that the Hierarchical Bayesian Network structure is much more informative than a standard Bayesian Network mapping the same independencies, shown in Figure 3(b). Additionally, the assumption that components of *Weather* and *Business* are independent to each other is shown explicitely the structure. The structure can easily be extended (adding more components inside *Business* composite node) or refined (transforming a leaf node into a composite one) without having to examine dependencies with the separate *Weather* components (as would be the case in a standard Bayesian Network without any domain information).

The network structure can easily be manipulated to express further assumptions that we might want to assert when performing inference. E.g., consider the Naive Bayes assumption, according to which the values of different attributes are independent given the value of the class attribute. This assumption can be viewed as a 2-step transformation of the original network. First, we need *PlayGolf* to be a parent node of the other attributes (so that it will appear in the conditional part of the probabilistic expressions). In the resulting network *Weather* and *Business* nodes will be linked, since they are not independent given the value of *PlayGolf* (if the individual would not play golf and

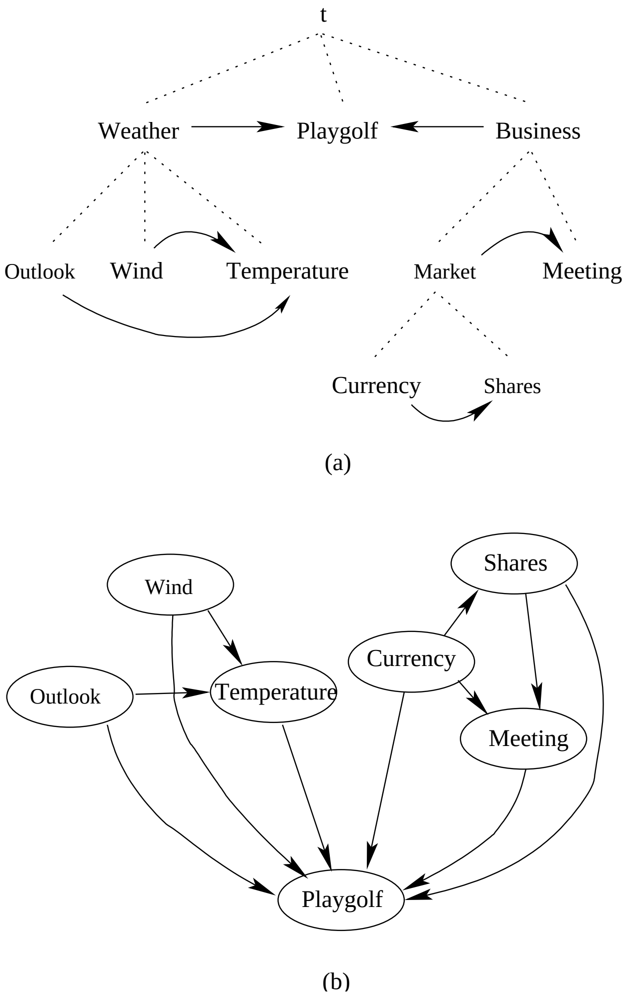


Figure 3. The Playgolf example structure. (a) Hierarchical Bayesian Network structure. (b) Standard Bayesian Network mapping the same independencies

the weather is fine, there must be something going bad about business). The "naive" assumption then lies in removing that link. That higher-level link is in essence the set of all links from each component of the one composite node to each component of the other one. Grouping these as a single link in the level of composite nodes makes the application of similar assumptions much more straightforward.

3.3. Definitions

We will now provide more formal definitions of the notion of Hierarchical Bayesian Networks. We begin by introducing hierarchical type aggregations, over which Hierarchical Bayesian Networks are defined. Currently, the only aggregation operator that we allow for composite types is the Cartesian product, but we plan to extend composite types to include aggregations such as lists and sets. This will demand a proper definition of probability distribution over these constructs.

Definition 3.1 (Composite type) *Let $\{\tau_1, \tau_2, ..., \tau_n\}$ be a set of atomic types (domains). The Cartesian product*

$$\tau = \tau_1 \times \tau_2 \times ... \times \tau_n$$

is a composite type. *The types $\tau_1, \tau_2, ..., \tau_n$ are called the* component types *of τ.*

Definition 3.2 (Type structure) *The type structure corresponding to a type τ is a tree t such that:*

- *If τ is an atomic type, t is a single leaf labelled τ.*
- *If τ is composite, t has root τ and children the type structures that correspond to the components of τ.*

Definition 3.3 (HBN-tree structure) *Let τ be an atomic or composite type, and t its corresponding type structure. An* HBN-tree structure *T over the type structure t, is a triplet $< R, \mathcal{V}, \mathcal{E} >$ where*

- *R is the* root *of the structure, and corresponds to a random variable of type τ.*
- *$\mathcal{V}$ is a set of* HBN-tree structures *called the* t-children *of R. If τ is a* simple type *then this set is empty, otherwise it is the* set of HBN-tree structures over the component-types of τ. *R is also called the* t-parent *of the elements of $\mathcal{V}$.*
- *$\mathcal{E} \subset \mathcal{V}^2$ is a set of directed edges between elements of $\mathcal{V}$ such that the resulting graph contains no directed cycles. For $(v, v') \in \mathcal{E}$ we say that v and v' participate in a* p-relationship, *or more specifically that v is a* p-parent *of v' and v' is a* p-child *of v.*

If τ is an atomic type, an *HBN-tree structure* over t will be called an *HBN-variable.* We will use the term *HBN-variable* to refer also to the random variable of type τ that the root of the structure is associated to. Referring to the Hierarchical Bayesian Network of Figure 2, there are four HBN-variables: A, BI, BII and C. The t-parent of BII is B, and its only p-parent is BI.

Given an *HBN-tree structure* $T =< R, \mathcal{V}, \mathcal{E} >$ and a *t-child* of R, $< R', \mathcal{V}', \mathcal{E}' > \in \mathcal{V}$, then for any v_P, v_C, v_i such that $(v_P, v) \in \mathcal{E}, (v, v_C) \in \mathcal{E}, v_i \in \mathcal{V}'$, we say that v_P is an *higher-level parent* of v_i, and that v_i is an *higher-level parent* of v_C. Furthermore, if v_{HLP} is a *higher-level parent* of v, then v_{HLP} is also a *higher-level parent* of v_i, and if v is a *higher-level parent* of v_{HLC} then v_i is also a *higher-level parent* of v_{HLC}. Back to the example in Figure 2, the higher-level parents of BII are A and BI, and its only higher-level child is C.

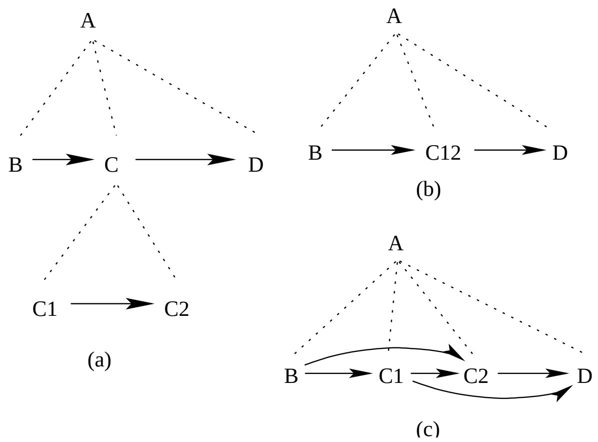


Figure 4. (a) A part of an HBN-tree structure. (b) Result of pruning the structure on node C. (c) Result of flattening the structure on C.

Definition 3.4 *The* HBN-Probabilistic Part *related to an HBN-structure T consists of:*

- *A probability table for each HBN-variable in T that does not have any p-parents or higher-level parents.*
- *A conditional probability table for each other HBN-variable, given the values of all HBN-variables that are its p-parents or higher-level parents.*

Definition 3.5 *A* Hierarchical Bayesian Network *is a triplet $< T, \mathcal{P}, t >$ where*

- *t is a type structure*
- *$T =< R, \mathcal{V}, \mathcal{E} >$ is an HBN-tree structure over t*
- *$\mathcal{P}$ is the HBN-Probabilistic Part related to T*

The key property underlying a Hierarchical Bayesian Network is that the value of a variable is conditionally independent of the nodes that are not its descendants, given the values of its direct or higher-level parents.

3.4. Pruning and Flattening

We now define the operations of *pruning* and *flattening* a Hierarchical Bayesian Network on a given node

v (Fig. 4). Informally, pruning a structure on a composite node disregards the additional structure underneath, replacing the whole sub-tree with a single node, that will be an aggregation of all the atomic types contained in it. Flattening also disregards the structure, but retains the individual nodes (and the p-relationships between them). Using pruning and flattening we can employ various useful transformations on a Hierarchical Bayesian Network, such as deriving a standard Bayesian Network that maps the same probabilistic independencies between the atomic types.

Definition 3.6 (Pruning) *Let t be a type structure with root τ and $T =< R, \mathcal{V}, \mathcal{E} >$ an HBN-tree structure over t.*

P_t is a closed operator on type structures, defined as follows:

- *If τ is an atomic type, $P_t(t) = t$.*
- *Otherwise, $P_t(t) = t'$, where t' is a leaf type structure, corresponding to τ.*

That is, pruning a composite type simply yields the equivalent atomic type, ignoring the internal structure.

Assuming $\mathcal{V}$ is not empty, let $v \in \mathcal{V}$ be a t-child of R, i.e. an HBN-tree structure over t', where t' is a child of t.

P_h is a closed operator on HBN-tree structures, defined as follows: $P_h(v)$ is an HBN-tree structure over $P_t(t')$, i.e. an HBN-variable.

The operation of pruning an HBN-tree structure T under a node v, *which we notate $P_{HBN}(T, v)$, results in an HBN-tree structure $T' =< R, \mathcal{V}', \mathcal{E}' >$, where $P_h(v)$ replaces v in $\mathcal{V}, \mathcal{E}$ to give $\mathcal{V}', \mathcal{E}'$ respectively.*

Definition 3.7 (Type structure flattening) *Let t be a type structure with root τ and $t_i, i = 1, 2, ...n$ the children of τ.*

The operation of flattening the type structure t under t_i, *denoted $F_t(t, t_i)$, is defined as follows:*

- *If t_i is a leaf, $F_t(t, t_i) = t$.*
- *Otherwise, if t_i has children $t'_j, j = 1, 2, ..., m$, and $t'' = F_t(...(F_t(Ft(t_i, t'_1), t'_2), ...), t'_m)$ (i.e., t'' is t' flattened under all its children), and $t''_k, k = 1, 2, ..., r$ are the children of t'', then $F_t(t, t_i)$ is a type structure with root t and children*

  $$\{t_1, t_2, ..., t_n\} \setminus \{t_i\} \cup \{t''_1, t''_2, ..., t''_r\}$$

  (i.e., t_i is replaced by the children of t'').

Definition 3.8 (HBN-tree structure flattening) *Let $T =< R, \mathcal{V}, \mathcal{E} >$ be an HBN-tree structure over a type structure t, $T' =< R', \mathcal{V}', \mathcal{E}' >$ an HBN-tree structure over a type structure t' with $T' \in \mathcal{V}$, and τ, τ' the roots of t, t' respectively.*

The operation of flattening the HBN-tree structure T under T', *denoted $F_h(T, T')$, is an HBN-tree structure over the type structure $F_t(t, t')$ defined as follows:*

- *If T' is an HBN-variable, $F_h(T, T') = T$.*
- *Otherwise, if $V' = \{v'_1, v'_2, ..., vn'\}$, $T'' = F_h(...(F_h(Fh(T', v'_1), v'_2), ...), v'_n)$ is the result of flattening T under all its t-children and $\mathcal{V}'' = \{v''_1, v''_2, ..., v''_m\}$ are the t-children of T'', then $F_h(T, T') =< R, \mathcal{V}_1, \mathcal{E}_1 >$ such that*
  - *$\mathcal{V}_1 = \mathcal{V} \setminus \{T'\} \cup \mathcal{V}''$, i.e. T' is replaced by the children of T'' in the HBN-tree structure*
  - *$\mathcal{E}_1$ is similar to $\mathcal{E}$, except for each $(T', v) \in= \mathcal{E}$ being replaced by $\{(v''_i, v) | i = 1, 2, ..., m\}$ in $\mathcal{E}_1$, and each $(v, T') \in= \mathcal{E}$ being replaced by $\{(v, v''_i) | i = 1, 2, ..., m\}$ in $\mathcal{E}_1$. I.e., all v_i participate in the former p-relationships of T'.*

Definition 3.9 (Corresponding Bayesian Network) *Let H be a Hierarchical Bayesian Network structure with HBN-probabilistic part P associated to it, and $H' =< R, \mathcal{V}, \mathcal{E} >$ the Hierarchical Bayesian Network that results after flattening H successively under all its t-children. The* corresponding Bayesian Network *of H is a Bayesian Network whose graphical part is the graph $< \mathcal{V}, \mathcal{E} >$ and probabilistic part is P.*

4. Inference in Hierarchical Bayesian Networks

Inference algorithms used for standard Bayesian Networks can also be applied to the Hierarchical Bayesian Network model. Backward reasoning algorithms [Russ85] and message passing algorithms [Pear88] can be used if we restrict our focus on the corresponding Bayesian Networks. The additional information in the Hierarchical Bayesian Network may serve towards a better interpretation of the resulting probability distributions.

Standard inference algorithms are not directly applicable to networks that contain loops (a loop is a closed path in the underlying undirected graph structure). One technique of coping with loops is to merge groups of variables into compound nodes, eliminating the circles in the graph. In the case of Hierarchical Bayesian Networks, knowledge of the hierarchical structure can serve as a guideline for which nodes to merge, in such

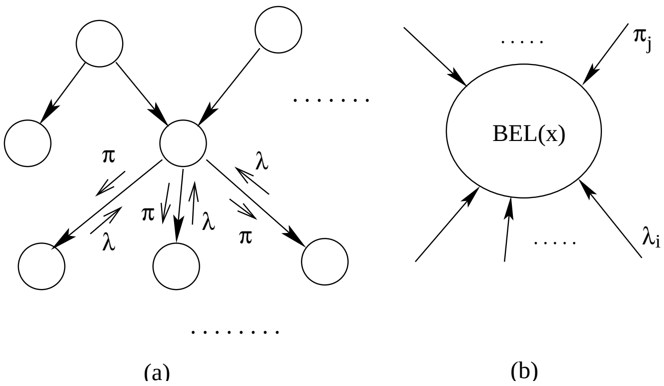


Figure 5. Belief propagation in polytrees. (a) Message passing in polytree structures. (b) Local belief update.

a way that the resulting network would allow a more meaningful interpretation.

4.1. Message Propagation Algorithm

A very efficient method of calculating beliefs in polytree-structured standard Bayesian Networks (i.e., that do not contain any undirected closed path) is the message-passing algorithm described in [Pear88]. Every node in the network is associated to a *belief state*, that is a vector whose elements sum up to one and correspond to the proportionate beliefs that each value in the domain may be the variable's value, given all available knowledge. The belief state of every node can be directly retrieved given the belief states of its parents and children. Whenever a change in a node's belief state occurs, either forced by some direct observation or indirectly, due to a change of the state of a neighbour, the node calculates its new belief state and propagates the relevant information to its parents and children (Fig. 5). The algorithm then repeats until the network reaches an equilibrium.

Belief calculation is performed locally in three independent steps:

Belief updating: Calculating a node's belief vector, using the latest information available from its neighbours.

Bottom-up propagation: Computing the λ *messages* that will be sent to parent nodes.

Top-down propagation: Computing the π *messages* that will be sent to children nodes.

The locality of the above algorithm is based on probabilistic independencies that result from the assumption that the network does not contain any undirected circles. If a network does contain loops, either an equilibrium cannot be reached, or, if it can be, it will not necessarily represent the actual joint probability distribution.

This algorithm may be directly applied to Hierarchical Bayesian Networks in the trivial case where the corresponding Bayesian Network does not contain undirected cycles. Even if individual composite nodes contain no loops, these will occur in the corresponding Bayesian Network in any case where some composite node participates in more than one p-relationship. The only case where loops will not occur is if we allow for polytree-like structures, where only leaf nodes may be composite, under the further restriction of not containing any p-links.

4.2. Structures containing loops

In the case where probabilistic dependencies form loops in the network infrastructure (i.e., the undirected network) the above algorithm cannot be applied directly. We define a Hierarchical Bayesian Network to contain loops if its corresponding Bayesian Network contains loops. There are several approaches that can be used to perform inference on a network containing loops [Pear88, Henr86]. As a trivial case, these methods can be applied to any Hierarchical Bayesian Network after flattening it to a standard Bayesian Network. Here we will restrict our discussion on an existing clustering method and show how it can be specifically adapted to the Hierarchical Bayesian Network case.

Clustering methods eliminate loops by grouping together clusters of two or more vertices into composite nodes. Different cluster selections may yield different polytrees when applied to a given Bayesian Network. As an extreme case, all non-leaf nodes may be grouped in a single cluster (Fig. 6(b)).

One popular method, described in [Pear88], is based on the construction of *join trees*. Briefly, the technique consists in building a triangulated undirected graph G (i.e., a Markov Network) that represents independency relations similar to the original Bayesian Network, and then linking the maximal cliques of G to form a tree structure. The advantage of this method is that the resulting directed acyclic structure is a tree, making the application of message propagation highly efficient. The trade-off is that the common parents of a node in the original Bayesian Network, along with the node itself, will be grouped into a single cluster, so information about independencies between them will be lost (Fig. 6(c,d)).

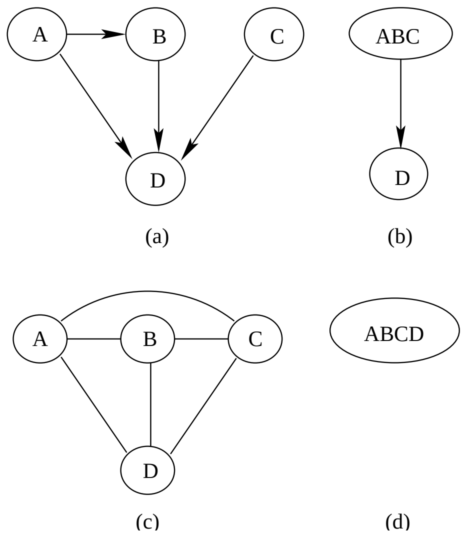


Figure 6. Coping with loops. (a) A Bayesian Network containing the loop ABDA. (b) Grouping all non-leaf variables in a single cluster. (c) Equivalent Markov network. (d) Join tree algorithm results in a single cluster.

The same algorithm can be applied directly on any fully flattened Hierarchical Bayesian Network. However, we can make use of the additional information that a Hierarchical Bayesian Network structure contains to arrive to more informative structures. The method we introduce is the following algorithm:

Algorithm 1 (HBN-decycling algorithm) *To* decycle *a node v:*

- *If v is a non-leaf node:*
  - *Decycle all components of v*
  - *If v participates in two or more p-edges, prune the network structure on v.*
  - *If v has exactly one p-parent (or p-child), flatten the structure on v replacing every p-connected subset of the t-children of v by a single cluster.*
  - *If v has no p-parents or p-children, flatten the structure on the node v.*
- *If v is a leaf node, leave v unchanged*

By applying the *HBN-decycling* algorithm on a Hierarchical Bayesian Network and then retrieving its corresponding Bayesian Network, we arrive at a *polytree* Bayesian Network. The application of the inference algorithm for Bayesian Networks is not as efficient for polytrees as it is for standard trees, but this is balanced by the smaller size of individual nodes.

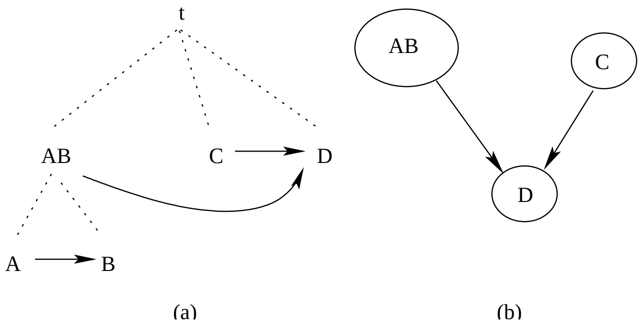


Figure 7. (a) A Hierarchical Bayesian Network containing the loop ABDA. (b) Result of the *HBN-decycling* algorithm.

In figure 7(a) we see an HBN-tree structure containing a loop. The structure is similar to the Bayesian Network in figure 6(a), with nodes A and B additionally forming a composite node. The result of the decycling algorithm (Fig. 7(b)) retains much more structural information from the original structure.

4.3. Stochastic simulation methods

Complexity of exact inference algorithms for Bayesian Networks is known to be exponential to the number of nodes. Approximate methods can be useful in many cases, especially when large networks are concerned. In *stochastic simulation* [Russ85, Henr86] a network is used to stochastically generate a large set of value assignments to variables, and the probability of any particular event is approximated by calculating the relative frequency that it occurs with.

In Hierarchical Bayesian Networks, we can apply a similar procedure using the partial ordering of nodes that the network represents. In order to generate a value assignment, we begin by stochastically choosing a value for each variable node with no p-parents, according to the given prior probabilities. Then, we proceed by choosing values for variable nodes whose p-parents have already been considered, with respect to the relevant conditional probabilities. When all variable nodes have values assigned to them, we store that sample and iterate.

5. Conclusions and further work

We have introduced an extension of Bayesian Networks that deals with structured data. Our approach takes advantage of existing Bayesian Network methods, and is an elegant way of incorporating into them specific domain knowledge.

Our current work is focused on implementing inference methods for Hierarchical Bayesian Networks, aiming to be tested against the performance of standard Bayesian Networks. Further on, we plan to introduce more aggregation operators for types, such as lists and sets. This will demand somewhat more sophisticated probability decomposition methods than the Cartesian product [Flac00], and will allow the application of the model to structures of arbitrary form and length, such as web pages or DNA-sequences. Presently, we are also investigating learning methods for networks given an observed set of data. Assuming that a type structure is known, the goal is to find the configuration for probabilistic dependency links and conditional probability values that fit best the given data. This problem may be addressed with Bayesian techniques (see [Coop92, Heck95]) where the likelihood of a model given the observed evidence is computed through the probability of the evidence having occurred given the model.

Acknowledgements

The authors wish to thank the anonymous reviewers for their useful comments. This work is funded by the EPSRC, as part of the "Efficient probabilistic models for inference and learning" research project at the University of Bristol.

References

[Coop92] Cooper, Gregory F. and Herskovits, Edward, "A Bayesian Method for the Induction of Probabilistic Networks from Data", in *Machine Learning*, 9, pp. 309–347, 1992.

[Flac00] Flach, Peter A. and Lachiche, Nicolas, "Decomposing probability distributions on structured individuals", in: James Cussens and Alan Frisch, editors, *Work-in-Progress Reports of the 10th International Conference on Inductive Logic Programming*, pages 96–106, London, July 2000.

[Heck95] Heckerman, David, Geiger, Dan and Chickering, David M., "Learning Bayesian Networks: The Combination of Knowledge and Statistical Data", in *Machine Learning*, 20, p. 197, 1995.

[Henr86] Henrion, M., "Propagating uncertainty by logic sampling in Bayes' networks", Technical report, Department of Engineering and Public Policy, Carnegie-Mellon University, 1986.

[Pear88] Pearl, Judea, "Probabilistic Reasoning in Intelligent Systems — Networks of Plausible inference", Morgan Kaufmann, 1988.

[Russ85] Russell, Stuart J. and Norvig, Peter, "Artificial intelligence, a modern approach", 2nd ed, Prentice Hall 1995.