International Journal of Approximate Reasoning 52 (2011) 935–947

Contents lists available at SciVerse ScienceDirect

International Journal of Approximate Reasoning

journal homepage: www.elsevier.com/locate/ijar

Fusing multiple Bayesian knowledge sources

Eugene Santos Jr. [a], John T. Wilkinson [a,*], Eunice E. Santos [b]

[a] Thayer School of Engineering, Dartmouth College, Hanover, NH, USA
[b] Department of Computer Science, University of Texas, El Paso, TX, USA

ARTICLE INFO

Article history:
Available online 12 February 2011

Keywords:
Bayesian Knowledge Bases
Knowledge fusion
Probability theory

ABSTRACT

We address the problem of information fusion in uncertain environments. Imagine there are multiple experts building probabilistic models of the same situation and we wish to aggregate the information they provide. There are several problems we may run into by naively merging the information from each. For example, the experts may disagree on the probability of a certain event or they may disagree on the direction of causality between two events (e.g., one thinks *A* causes *B* while another thinks *B* causes *A*). They may even disagree on the entire structure of dependencies among a set of variables in a probabilistic network. In our proposed solution to this problem, we represent the probabilistic models as Bayesian Knowledge Bases (BKBs) and propose an algorithm called *Bayesian knowledge fusion* that allows the fusion of multiple BKBs into a single BKB that retains the information from all input sources. This allows for easy aggregation and de-aggregation of information from multiple expert sources and facilitates multi-expert decision making by providing a framework in which all opinions can be preserved and reasoned over.

1. Introduction

There are many situations when information fusion can be useful, from fusion of low level sensor data to the aggregation of higher level models built by human experts. One simple example is the case of multiple medical diagnosis expert systems, each containing rules from different experts. One expert may believe that the causal link between disease *A* and symptom *B* is strong while another may think it is weak; or one may think that condition *A* causes condition *B* while another thinks that *B* causes *A*. In the former case, we could take the max of the two conditional probabilities, or the average, or some other function of the two, but in doing so we would likely lose some information. In the latter case, we would end up with a cyclic causal relationship, violating the rules of some knowledge representation schemes like the widely used Bayesian Networks [1]. The ability to aggregate knowledge from mutiple sources in light of these difficulties is crucial in decision making scenarios involving mutiple subject matter experts and perspectives.

In addition to combining the opinions of multiple experts, another major goal of this work is to allow the easy aggregation and de-aggregation of pieces of information. This allows for reasoning over only the relevant pieces of the knowledge base. It also alleviates the problem of the high computational cost of reasoning in very large knowledge bases and allows for focused source-based analyses. Ease of aggregation is also ideal when dealing with a probabilistic model of a dynamic or evolving situation where new knowledge is being gained and added to the knowledge base incrementally and old knowledge may be removed if it is no longer applicable.

In this paper, we use Bayesian Knowledge Bases (BKBs) [2–4] as our knowledge representation framework. A BKB is a generalization of a Bayesian Network (BN) in which dependencies are specified between states of random variables instead

* Corresponding author. Tel.: +1 603 646 9945.
E-mail address: john.t.wilkinson@dartmouth.edu (J.T. Wilkinson).

0888-613X/$ - see front matter
doi:10.1016/j.ijar.2011.01.008

of between random variables, directed cycles are allowed, and the probability distribution specified need not be complete. We show how to take a collection of BKBs and merge them into one single BKB. In the context of the fusion process, we refer to the input BKBs as Bayesian Knowledge Fragments (BKFs) since each represents one fragment of knowledge that we wish to fuse into a larger knowledge base. Thus each BKF contains one chunk of knowledge and the fusion process allows them to be composed into larger, semantically meaningful models. These fragments can be easily stored, retrieved, and aggregated.

In related work, others have tried to use BNs for knowledge representation, but techniques based on BNs run into problems when it comes to tractable manipulation and reasoning. These techniques fall victim to exponential run time complexity, the need for complete specification of probability distributions, and overly rigid rules regarding conditional dependence structure. Included in this class are BN variants such as object-oriented BNs [5], multi-entity BNs [6], and Bayesian blackboards [7].

Our fusion algorithm is based on the addition of new nodes to the input fragments called *source nodes*. These nodes indicate which rules in the knowledge base came from which fragments. A *reliability index* is given to each fragment indicating the trustworthiness of the source of the knowledge contained in the particular fragment. With these source nodes and their corresponding reliabilities, the inference process on the fused BKB can consider information from multiple sources when constructing explanations for any evidence observed. Thus the constructed explanation may contain elements from several different sources.

In the next section, we will describe BKBs in more detail. Then we will present the fusion algorithm and some examples of it in use. After that, we will further investigate the properties of the fused BKB, and then we will end the paper with a conclusion and description of future directions for this research.

2. Bayesian Knowledge Bases

A BKB is a collection of conditional probability rules (CPRs) of the form *if* $A_1 = a_1, A_2 = a_2, \ldots, A_n = a_n$, *then* $B = b$ *with probability p*, that satisfies conditions of normalization and mutual exclusivity that will be formally defined shortly. In these rules, A_i and B are random variables and a_i and b are instantiations, or states, of those random variables. BKBs subsume Bayesian Networks (BNs) as all BNs are representable as BKBs (one can form a BKB from a BN by making all the conditional probability table entries in the BN into CPRs in the BKB). In contrast to BNs, and similar to probabilistic decision graphs [8], BKBs allow for independence to be specified between states of random variables instead of between the random variables themselves.

BKBs also do not require that the probability distribution specified by the rules be complete; inference is carried out on only those rules that are specified. This allows for reasoning in the face of incompleteness and alleviates the burden of filling in large conditional probability tables which can be especially problematic when all the probabilities for these tables are not readily available. Others such as Druzdzel and van der Gaag [9] have done some work in allowing underspecified probability distributions by defining the distribution in terms of a set of (in)equalities on the probabilities that effectively constrain the space of distributions that are possible. However, in this approach, inconsistencies between experts must be addressed by modifying the knowledge base, unlike in our approach in which the opinion of each expert is preserved.

Similarly to a BN, a BKB can be represented graphically, but the graph is not required to be acyclic as with BNs. Other graphical models do allow cyclicity but with various associated drawbacks. The well known Markov random fields allow cyclicity but are undirected and cannot represent some dependencies that a directed model such as a BN can. Another model

1. Clinton downplayed MLK = yes, Race = Black $\xrightarrow{0.75}$ Support = Obama
2. Race = White, Like Bill Clinton = yes, Hillary more electable = yes $\xrightarrow{0.80}$ Support = Clinton
3. Support = Obama $\xrightarrow{0.15}$ Campaign for Obama = yes
4. Support = Clinton $\xrightarrow{0.70}$ Clinton downplayed MLK = no

Fig. 1. A BKB fragment from a knowledge base about the South Carolina Democratic Primary election as a set of CPRs.

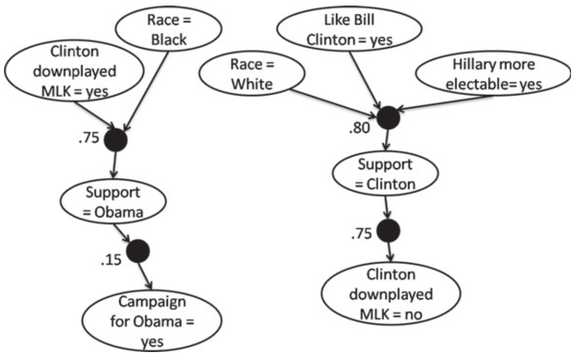

Fig. 2. A BKB fragment from a knowledge base about the South Carolina Democratic Primary election as a directed graph.

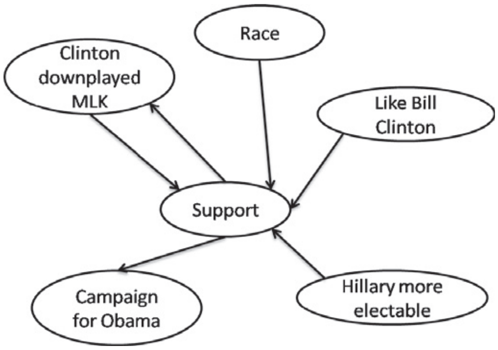

Fig. 3. Underlying random variable relationships in Fig. 2.

allowing cycles is the dependency network model of Heckerman et al. [10], but these are equivalent in expressive power to Markov networks and are not useful in specifying causal relationships.

There are two types of nodes in a graphically depicted BKB, I-nodes representing the different instantiations of the random variables, and S-nodes, or "support nodes". An example of a BKB from work on the South Carolina Democratic Primary election is shown in rule form in Fig. 1 and in graph form in Fig. 2. Each solid black circle in Fig. 2 is an S-node and corresponds to exactly one of the CPRs from Fig. 1. The number next to the S-node is its weight. Each text-filled oval is an I-node corresponding to one state of one of the random variables. The dependencies shown in the BKB would result in a circular directed graph at the random variable level (Fig. 3) and thus would not be a valid BN. The circular relationship results because black voters felt that Hillary Clinton had downplayed Martin Luther King Jr.'s role in the civil rights movement and this caused them to support Obama [11]. However, some white voters who liked the Clinton family and supported Hillary may have been skeptical that she had any ill intentions when making the remark about Martin Luther King Jr. Similar to d-separation in BNs, it is possible to determine independence semantics from the graph induced by a BKB [12].

We now give the formal definition of the graphical representation of a BKB from [13]:

Definition. A *correlation-graph* is a directed graph $G = (I \cup S, E)$ in which $I \cap S = \emptyset$, $E \subset \{I \times S\} \cup \{S \times I\}$, and $\forall q \in S$, there exists a unique $\alpha \in I$ such that $(q, \alpha) \in E$. If there is a link from $q \in S$ to $\alpha \in I$, we say that *q supports* α.

An edge $(a, b) \in E$ will be denoted as $a \rightarrow b$ throughout the rest of the paper. For each S-node q in a correlation graph G, we denote the set of all I-nodes that point to q as $Tail_G(q)$, i.e. $Tail_G(q) = \{\alpha \in I | \alpha \rightarrow q \in E\}$. Similarly the $Head_G(q)$ is the I-node that q points to in G, i.e. the α such that $q \rightarrow \alpha \in E$.

Two sets of I-nodes, I_1 and I_2 are said to be *mutually exclusive* if there is an I-node $(R_1 = v_1)$ in I_1 and an I-node $(R_2 = v_2)$ in I_2 with $R_1 = R_2$ and $v_1 \neq v_2$. Intuitively, mutual exclusivity is the condition that the two sets I_1 and I_2 cannot be satisfiable at one time (i.e. there must be some random variable that is given a contradictory assignment in each of the sets). Similarly, two S-nodes q_1 and q_2 are called *mutually exclusive* if $Tail(q_1)$ and $Tail(q_2)$ are mutually exclusive. S-nodes and I-node sets that are not mutually exclusive are called *compatible*. A *state* is a set of I-nodes θ such that there is at most one instantiation of each random variable in θ.

Definition. A *Bayesian Knowledge Base* (BKB) is a tuple $K = (G, w)$ where $G = (I \cup S, E)$ is a correlation-graph, and $w : S \rightarrow [0, 1]$ such that

(1) $\forall q \in S$, $Tail_G(q)$ contains at most one instantiation of each random variable.
(2) For distinct S-nodes $q_1, q_2 \in S$ that support the same I-node, $Tail_G(q_1)$ and $Tail_G(q_2)$ are mutually exclusive.
(3) For any $Q \subseteq S$ such that (i) $Head_G(q_1)$ and $Head_G(q_2)$ are mutually exclusive, and (ii) $Tail_G(q_1)$ and $Tail_G(q_2)$ are not mutually exclusive for all distinct q_1 and q_2 in Q,

$$\sum_{q \in Q} w(q) \leq 1.$$

So a BKB is a correlation graph along with a weight function (specifying the probabilities in the CPRs) satisfying the above conditions. The first condition states that no rule can have two contradictory assignments in its tail. The second ensures that all S-nodes pointing to the same I-node are mutually exclusive, and the last condition ensures normalization of the probabilities.

Reasoning with BKBs takes on two main forms we refer to as belief revision and belief updating, similar to reasoning with BNs [1]. Belief revision is similar to computing the maximum a posteriori (MAP) or most probable explanation (MPE) in a BN. In belief revision we have some evidence in the form of assignments of random variables to states, e.g. assume we have a BKB with n variables and k pieces of evidence $A_1 = a_1, A_2 = a_2, \ldots, A_k = a_k$. The goal of belief revision is to find the assignment of $A_{k+1}, A_{k+2}, \ldots, A_n$ to states such that $P(A_{k+1} = a_{k+1}, \ldots, A_n = A_n | A_1 = a_1, \ldots, A_k = a_k)$ is maximized.

Intuitively, this can be thought of as finding the most probable state of the world that contains all the evidence. In belief updating, the goal is to find the probability of a state of a random variable given the evidence. For example, assuming again that $A_1 = a_1, A_2 = a_2, \ldots, A_k = a_k$, we may want to find $P(B = b | A_1 = a_1, A_2 = a_2, \ldots, A_k = a_k)$ for some instantiation b of a random variable B.

In BKBs however, we are often interested in partial states of the world instead of complete states, especially because we may have a BKB specifying an incomplete probability distribution in which it is not possible to perform a traditional MAP computation as described above. Thus, in revision, our goal becomes to compute the most likely partial state of the world (also called the most probable *inference* supported by the evidence, as will be described later). In updating, we can find the marginal probability of a partial state of the world in addition to that of a single state of a random variable or a complete state of the world. Given that BKBs support incomplete specification of knowledge, we may have to make these calculations in the face of this incompleteness. In these situations, only the information that has been specified by the knowledge engineer is used in a given computation, no assumptions are made to "fill in" the portions of the probability space that have not been specified.

In carrying out both types of reasoning, we make use of subgraphs called *inferences* that capture the joint probability of the I-nodes in the subgraph. Intuitively, an inference is a partial state of the world.

Definition. Let $K = (G, w)$ be a BKB with correlation graph $G = (I \cup S, E)$. A subgraph $\tau = (I' \cup S', E')$ of G is called an *inference* over K if

(1) τ is acyclic.
(2) (*Well-supported*) $\forall \alpha \in I', \exists q \in S', q \rightarrow \alpha \in E'$
(3) (*Well-founded*) $\forall q \in S', Tail_\tau(q) = Tail_G(q)$
(4) (*Well-defined*) $\forall q \in S', Head_\tau(q) = Head_G(q)$
(5) There is at most one I-node corresponding to any given random variable in I'.

The properties above state that in an inference, (2) each I-node must have an S-node pointing to it, (3) if an S-node in the inference has any I-nodes pointing to it in the correlation graph they must also be in the inference, (4) there cannot be S-nodes in the inference that do not point to any I-nodes in the inference, and (5) there cannot be mutually exclusive I-nodes in an inference. The joint probability of the set of I-nodes in an inference is simply the product of the weights of all S-nodes in the inference, i.e. for an inference $\tau = (I' \cup S', E'), P(\tau) = \prod_{q \in S'} w(q)$.

3. The fusion algorithm

In this section, we will discuss related work on model fusion and then define an algorithm that takes multiple Bayesian Knowledge Fragments as input and fuses them into one larger BKF. Each BKF is assumed to come from a distinct expert source. There has been prior work on fusing probabilistic models such as that of Matzkevich and Abramson [14], but the drawback to their approach seems to be that they resolve conflicts in the fusion of multiple Bayesian Networks by forming a new network representing the "consensus" of the experts who created the input networks instead of preserving the knowledge encoded by each expert as in our approach. Jiang et al. [15] also propose a framework for fusing multiple BNs and integrating new ones over time, but their approach can require non-trivial changes to the input BNs such as reversal of arcs to avoid the creation of cycles and an ad-hoc addition of variables in order to get the conditional probability tables in a state in which they can be combined to form a single set of CPTs. There is also a body of work from Laskey and Mahoney [16–18] on an object-oriented approach to managing and aggregating fragments of BNs for specific situations. Related work by Lu and Druzdzel using structural equations to incrementally build a model can be found in [19].

In our model, the reliability and probability of correctness of each source is assessed to produce a *reliability index* for each fragment, allowing more reliable sources to be given more weight in the fused BKB. As long as the input fragments are valid BKBs, the algorithm assures that the fusion of these BKBs is still a valid BKB. A BKF can thus be described as a BKB along with a source and its reliability index, i.e., it can be described as a tuple $K = (G, w, \sigma, r(\sigma))$ where G and w define a valid BKB, σ is the source of the BKB, and $r(\sigma)$ is the reliability index, a non-negative real number denoting the reliability of the source.

The idea behind the algorithm is to first take the union of all input fragments. Then, for each random variable A in the fused BKF, we create a new random variable S_A representing the source of a rule supporting an I-node representing a state of A. Each S-Node that points to a state of A has an I-node added to its tail called a *source node* whose random variable is S_A and whose state is σ, the source of the fragment the S-node came from. The reliability of this source node is $r(\sigma)$, the reliability of the fragment that the S-node came from. So if a rule q supporting an I-node $(A = a)$ is from source σ, we would add the source I-node $(S_A = \sigma)$ to the tail of q in the fused BKF. We also add an S-node q' to the fused BKF whose head is $(S_A = \sigma)$. The weight of this S-node is proportional to the reliability index $r(\sigma)$ for the source, normalized so that the weights of all sources for a given random variable do not exceed 1. Intuitively, adding a source node, as in the algorithm, turns a rule that says "If A is true then B is true with probability p" into a rule that says "If source σ is correct and A is true, then B is true with probability p".

The algorithm can be described more formally as follows. The input is a set of n BKFs, $\{K_1, K_2, \ldots, K_n\}$ where $K_i = (G_i, w_i, \sigma_i, r(\sigma_i))$ and $G_i = (I_i \cup S_i, E_i)$. The output is a new BKB, $K' = (G', w')$ with $G' = (I' \cup S', E')$, that is the fusion of the n input BKFs. For an I-node α in some fragment, let R_α be the random variable that α is an instantiation of. For a source node $s = (S_{R_\alpha} = \sigma_i)$ let the reliability of s be $r(s) = r(\sigma_i)$.

Bayesian-Knowledge-Fusion($K_1, K_2, \ldots, K_n$)

Let $G' = (I', S', E')$ be an empty correlation graph and w' a weight function
$I' \leftarrow \cup_{i=1}^{n} I_i$
$S' \leftarrow \cup_{i=1}^{n} S_i$
$E' \leftarrow \cup_{i=1}^{n} E_i$
for all fragments K_i with $i \leftarrow 1$ **to** n
    for all S-nodes $q \in S_i$
        Let $\alpha \leftarrow Head_{G_i}(q)$
        Let the source I-node for q be $s = (S_{R_\alpha} = \sigma_i)$
        Add s to I' and add a new S-node q_s to S'
        Add the edges $q_s \rightarrow s$ and $s \rightarrow q$ to E'
        Let $w'(q) \leftarrow w_i(q)$
for all source variables S_{R_α}
    Let $\Lambda = \{s \mid s$ is a source node which is a state of $S_{R_\alpha}\}$
    Let $\rho \leftarrow \sum_{s \in \Lambda} r(s)$
    for each $s \in \Lambda$, let q_s be the S-node such that $q_s \rightarrow s \in E'$
        Let $w'(q_s) \leftarrow r(s)/\rho$
return $K' = (G', w')$

4. Common knowledge fusion problems

Although the algorithm presented seems fairly simple, it allows us to solve a wide variety of problems that may crop up when fusing knowledge from multiple sources. The first problem we will discuss is of experts who disagree about the weight of a relationship. In this scenario, we have one expert who believes that A causes B with some probability p_1 and a different expert who believes that A causes B with some probability $p_2 \neq p_1$. In more traditional systems, these rules would need to be combined into a single rule, with one weight, that could be stored. However, we argue that in doing so, valuable information provided by an expert may be lost. Instead of two probabilities provided by independent experts, we now have only one value which we hope is better than either of the two we started with. Unfortunately, it may be the case that one probability was much better than another. It may also be the case that the fact that there was disagreement about this value is significant, but by reducing the knowledge stored to a single rule, we lose this information as well.

Using our knowledge fusion algorithm, this problem is avoided. In our system, the rules from both of the experts would be stored in the system and available for reasoning. The rules can be appropriately weighted based on the reliability of each of the experts with respect to the situation being modeled. A hypothetical example related to work performed with Dr. Felicia Pratto of the University of Connecticut is shown in Fig. 4 (Pratto, personal communication). Here, two experts have both constructed a rule indicating the likelihood that the Palestinian people will support Hamas if they believe that the Palestinian Authority has corruption problems. From the figure it is clear that one expert believes that the perception of corruption has a significant impact on support for Hamas while the other expert believes that the relationship is less certain and places a much lower probability on the rule.

In order to use the Bayesian Knowledge Fusion algorithm, the experts need to be assigned reliabilities. In this case, one of the experts happens to be Dr. Pratto, while the other is one of her students. Naturally, we assign a much higher reliability to the rule created by Dr. Pratto (0.93) and a much lower probability to the one provided by her student (0.03). This results

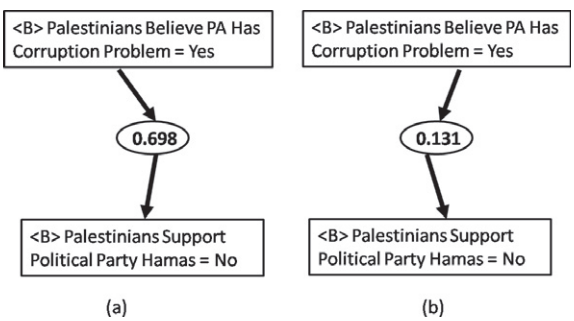

Fig. 4. The same rule created by two different experts, but with different weights.

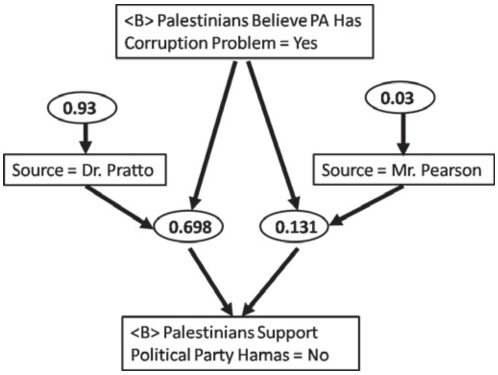

Fig. 5. Result of the fusion of the rules from Fig. 4 after sources were assigned weights and the algorithm applied.

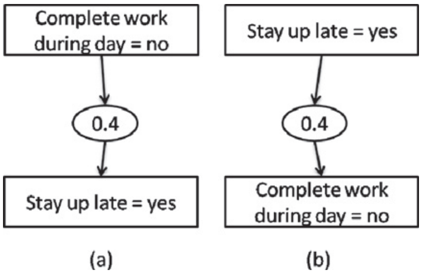

Fig. 6. Two rules regarding academic life that disagree about direction of causality.

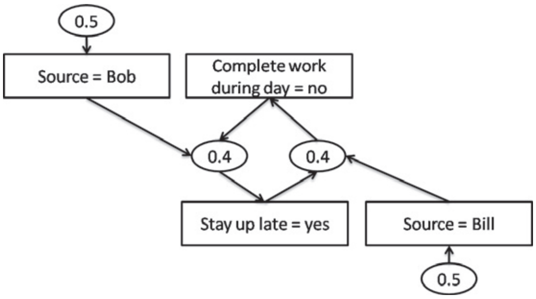

Fig. 7. Fusion of the two rules from Fig. 6.

in the fused fragment shown in Fig. 5 which has each of the original rules, along with their weights, fused into a single probabilistically valid knowledge base.

Another problem that is handled well by our framework is that of cyclicity. In traditional probabilistic knowledge representations, cyclic knowledge is not allowed. For instance, one of the important properties of a Bayesian Network is that it must be an *acyclic* graph. However, when dealing with multiple sources of knowledge, it is quite possible that they will disagree about the direction of causality of a rule in the knowledge base. This could lead to the situation where one expert believes that *A* implies *B* with some probability while another expert believes that *B* implies *A* with some probability. If we were to draw this in the form of a Bayesian Network, a cycle would be present. Normally, an attempt would be made to determine the "correct" direction of the dependence and only this direction would be stored in the knowledge base. However, in our framework, cyclic knowledge is allowed and does not pose any problems. The sources would each be assigned a reliability and each of their rules would be included in the knowledge base without modification.

A simple example drawn from academic life is shown in Fig. 6. These fragments were created based on a comic from the popular PhD Comics website by Cham [20]. The comic deals with the vicious cycle many students end up in with regard to their sleep schedule. Often students do not get any work done during the day, so they stay up late to try to finish some work and make up for it. Then since they stayed up late doing work, they are tired the next day and do not get any work done during the day as a result. Different experts, call them Bill and Bob, may create two different rules from this comic (as in Fig. 6) that are essentially the same but with the direction of causality reversed. One has concluded that staying up late causes no work to be completed during the day, while the other thinks that not completing work during the day causes one to stay up late because they want to make up for procrastination during the daytime.

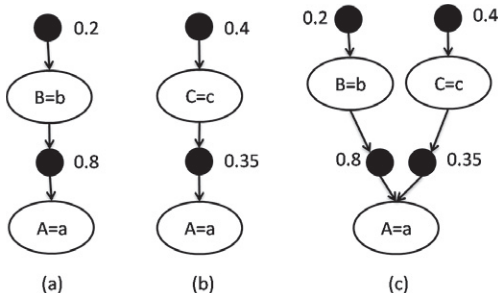

Fig. 8. Naive union of two fragments results in a violation of mutual exclusivity.

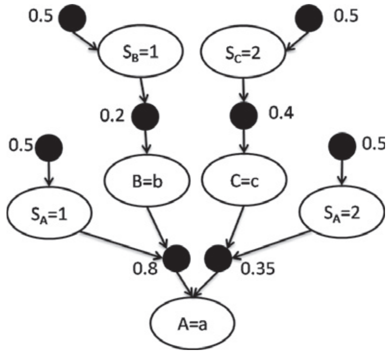

Fig. 9. Fusion of the two fragments from Fig. 8 using our Bayesian knowledge fusion algorithm solves mutual exclusivity problem.

As in the previous example, trying to use both of these rules in a Bayesian Network would result in problems. Specifically it would introduce a cycle, which is not allowed. In our framework however, the rules can be fused without issue. As before, the source of each of the rules must be given a reliability. In this case, as we have no reason to think one source more reliable than the other, we assign equal weight to each of the sources. The result of applying the Bayesian Knowledge Fusion algorithm to the rules from Fig. 6 is shown in Fig. 7.

A final problem that can arise in the context of multiple source knowledge fusion is a violation of the properties required by the definition of a Bayesian Knowledge Base, in particular, the mutual exclusivity requirement. This principle requires that all I-nodes pointing to the same S-node must be pairwise mutually exclusive, i.e. for any two S-nodes pointing to the same I-node, there must be some I-node in the tail of each of the S-nodes representing the same random variable with a different instantiation of that variable. Intuitively, the S-nodes supporting some I-node each represent a different way to prove the truth of the I-node. If all of the I-nodes supporting some S-nodes can be shown to be true, then it can be inferred that head of the S-node is true as well. Mutual exclusivity requires that in any possible world, there is only one valid way to prove the truth of a given I-node, i.e. for any I-node, only one S-node supporting it is allowed to be true in any possible world. A similar condition is present implicitly in Bayesian Networks. When the conditional probability table (CPT) is filled out for a node in a BN, it is easy to see that the lines in the CPT are similar to the S-nodes in a BKB, and any pair of lines in a CPT is guaranteed to contain a contradictory assignment to some random variable.

In our framework, we assume that all knowledge fragments that serve as input to our algorithm are valid BKBs. Thus we are guaranteed that each of these fragments satisfies the mutual exclusion condition in the definition of a BKB. It is vitally important that after the fusion algorithm completes there are no violations of this condition in the resulting fused knowledge base, or else we cannot guarantee that a valid BKB will be produced by the algorithm. It may seem at first that given a set of BKBs, we could just take their union in order to fuse them into a larger BKB, but this presents problems. Consider the two fragments in Fig. 8(a) and (b) whose naive union is shown in Fig. 8(c). We can see that this is not a valid BKB because the two S-nodes supporting $(A = a)$ are not mutually exclusive. If we add in source nodes as in Fig. 9, we can see that the S-nodes are now mutually exclusive because the source nodes in the tails of the random variables cannot both be true at the same time. Thus, in addition to allowing us to keep track of sources in the model, the source nodes also resolve problems that could occur if we decided to just take the simple union of several fragments.

The algorithm also solves normalization problems that may occur after fusion. There is a danger of violating property (3) in the definition of a BKB and ending up with probabilities that can sum to more than 1 using a naive fusion approach, but, as we will show in the next section, the source nodes prevent this.

As a note, the fusion algorithm will work on Bayesian Networks as well as BKBs, but when you fuse BNs, you will most likely end up with a BKB, not a BN, after the fusion. First, cycles may be introduced violating the acyclicity of BNs. More troublesome is the fact that the original conditional probability tables from the BNs will need to be revised so that they remain valid. For example, one may want to fuse the two BNs in Fig. 10 into one BN. The probabilities for $C = F$ were not

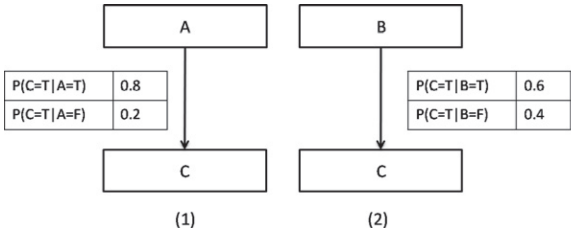

Fig. 10. Two Bayesian Networks that need to be fused.

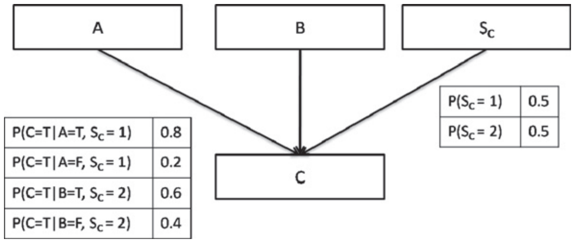

Fig. 11. BNs from Fig. 10 fused using the Bayesian Knowledge Fusion algorithm resulting in an invalid BN, but a valid BKB.

included in the figure but can be computed from the probabilities listed, and the prior probabilities for *A* and *B* are not listed because they are not important to the example. The algorithm will add a source variable, called S_C here, that points to *C* and will then create probabilities for *C* conditioned on this variable. There are two possible states for S_C, each representing a single source (in this case called simply 1 and 2). $P(S_C = 1)$ corresponds to the reliability of source 1.

Fig. 11 shows the result of the fusion algorithm. A CPT is created for S_C that corresponds to the reliabilities of each of the two sources (which are given equal weight here). Then the rules shown to the left of *C* are created. The problem here is that the rules to the left of *C* do not constitute a valid CPT for a BN. Since *C* depends on *A*, *B*, and *C*, it is necessary that the CPT entries contain probabilities for all possible combinations of these three variables. However, the entries generated are conditioned on only two of the variables and do not represent an exhaustive list of all possible combinations. Although it is not a standard BN, reasoning can still be performed using techniques such as IB-MAP [21]. On the other hand, this is still a valid BKB and is in fact probabilistically complete in that there will exist a set of mutually exclusive inferences whose probabilities sum to one.

5. Algorithm properties

As long as the fragments that are input into the algorithm are themselves valid BKBs, the output of the algorithm will also be a valid BKB. To prove this we need to show that the conditions in the definition of a BKB are satisfied.

Theorem 1. *The output* $K' = (G', w')$ *of* Bayesian-Knowledge-Fusion *is a valid BKB.*

Proof. To complete the proof, we need to show that K' satisfies all properties of a BKB given in the earlier definition. Thus G' must be a correlation graph, w' a valid weight function, and properties 1–3 from the earlier definition must be satisfied.

G' is a correlation graph because only S-nodes and I-nodes were added to K' with no edges from S-node to S-node or I-node to I-node. For each S-node that was added, a source I-node was also added for it to support. w' is a valid weight function from S' to [0, 1] because weights of S-nodes from input fragments are preserved and weights of new S-nodes are normalized so that they are in [0, 1]. Now we will show that each of the three properties from the definition are satisfied.

(1) $\forall q \in S'$, $Tail_G(q)$ contains at most one instantiation of each random variable because we added only one I-node (the source node) to the tail of any S-node and the the source node is an instantiation of a random variable that did not previously exist in the fragments.
(2) Now we must show that for distinct S-nodes $q_i, q_j \in S$ that support the same I-node, $Tail_G(q_i)$ and $Tail_G(q_j)$ are mutually exclusive. This is true by construction of G'. To see this, let q_i be any S-node in G' and assume it came from input fragment K_i and has head $(R = v)$. If there are any other S-nodes from K_i with the same head, they are mutually exclusive with q since K_i is itself a BKB and S-nodes with the same head in a BKB must be mutually exclusive. If there is an S-node q_j from $K_j, j \neq i$, that has the same head as q_i, then it is mutually exclusive with q_i because the source nodes for q_i and q_j are of the same variable with different instantiations. i.e. $(S_R = i)$ is in the tail of q_i and $(S_R = j)$ is in the tail of q_j.

(3) Finally we need to show that for any $Q \subseteq S'$ such that (i) $Head_{G'}(q_1)$ and $Head_{G'}(q_2)$ are mutually exclusive, and (ii) $Tail_{G'}(q_1)$ and $Tail_{G'}(q_2)$ are not mutually exclusive for all distinct q_1 and q_2 in Q,

$$\sum_{q \in Q} w(q) \le 1.$$

Assume $Q \subseteq S'$ and satisfies conditions (i) and (ii) above. Then all the I-nodes that members of Q point to must be from the same random variable. The reason is that the members of Q would not be mutually exclusive if this was not the case. To prove this, assume that there exist distinct S-nodes $q_1, q_2 \in Q$ such that $h_1 = Head_{G'}(q_1) = (R_1 = v_1)$ and $h_2 = Head_{G'}(q_2) = (R_2 = v_2)$ with $R_1 \ne R_2$. Then h_1 is not mutually exclusive with h_2 because mutually exclusive I-nodes have the same random variable with a different instantiation of that variable. Thus we have a contradiction, and so all I-nodes that S-nodes in Q point to are of the same random variable.

Now, assume the random variable pointed to by the members of Q is a source random variable. Then, since the source nodes are normalized so that their probabilities sum to less than 1 by the fusion algorithm, $\sum_{q \in Q} w(q) \le 1$.

Now assume that the random variable pointed to by members of Q is not a source random variable. Then we claim that all members of Q must be from the same input fragment. The reason is that source nodes were added to all S-nodes by the fusion algorithm that have a state which indicates the fragment they came from. If two S-nodes point to an instantiation of the same random variable, then either they are from the same fragment, or they are from different fragments and they have a source I-node in their tail with different states (corresponding to the source fragment). Therefore two S-nodes pointing to an I-node of the same random variable either come from the same fragment or have tails that are mutually exclusive, which is not allowed by the definition of the set Q. Therefore all members of Q are from the same input fragment. Since that input fragment was a valid BKB, it must be that $\sum_{q \in Q} w(q) \le 1$ (or else the input fragment would be proven an invalid BKB and we would have a contradiction).

These are sufficient to satisfy the definition of a BKB, so the output of the fusion algorithm is indeed a valid BKB. □

Now that we know the result of the fusion algorithm is a valid BKB if all input fragments are valid BKBs, it is interesting to ask what other properties of the input fragments are preserved by the algorithm. One property that we know is preserved is *groundedness*.

Definition. A node $a \in I \cup S$ in a BKB is said to be *grounded* if there exists an inference τ over the BKB such that a is in τ.

Groundedness is a desirable property because if an S-node remains grounded during changes to the knowledge base, the initial probability semantics assigned to it is preserved [3].

Theorem 2. *If all S-nodes in the input fragments are grounded in their respective fragments, then all S-nodes in the fusion of these fragments, K', are grounded in K'.*

Proof. Let $q \in S'$ be an S-node originally from fragment K_i, then there is a subgraph τ that is an inference over K_i such that $q \in \tau$. We will construct a new subgraph $\tau' \supset \tau$ that is an inference over K'. Start with $\tau' = \tau$, and then for each S-node q in τ, there is a source node s such that there is an edge from s to q in E', and there is an S-node q' pointing to s. Thus add $q, q', s, s \to q$, and $q' \to s$ to K'. So τ' consists of τ, the source nodes for S-nodes in τ, the S-nodes supporting these source nodes, and the corresponding edges.

Now it needs to be shown that τ' is an inference over K'. τ' is acyclic because τ was acyclic and all that was added to τ was the source nodes and their supports. Let a be a node from τ, then there is no path from a to itself in τ' because there is no path from a to itself in τ, and it is impossible to reach nodes in $\tau' - \tau$ from a. This is because the only nodes in this set are the source nodes and their supports, the only input to a source node is its support, and there are no inputs to the supports. From this argument it also follows that there is no path from a node in $\tau' - \tau$ to itself, and so τ' is acyclic.

τ' is well-supported as well. Let $\alpha \in I'$. If $\alpha \in \tau$ then it is supported by the same S-node that supported it in τ. If $\alpha \in \tau' - \tau$, then it is a source node and it has a support in τ' by construction.

Now it will be shown that τ' is well-founded. Let $q \in S'$ be from fragment K_i originally. Then $Tail_{G'}(q)$ is $Tail_{G_i}(q)$ with the source node added to it. Also $Tail_{\tau'}(q)$ is $Tail_{\tau}(q)$ with the source node added to it. Since $Tail_{G_i}(q) = Tail_{\tau}(q)$ by the well-foundedness of τ, we have that $Tail_{G'}(q)$ and $Tail_{\tau'}(q)$ are both equal to $Tail_{\tau}(q)$ with the source node added. Therefore they are equal to each other and τ' is well-founded.

τ' is also well-defined. To see this, let $q \in \tau$ be from fragment K_i, then $Head_{\tau'}(q) = Head_{\tau}(q)$. Since τ is an inference, $Head_{\tau}(q) = Head_{G_i}(q)$. Then since the algorithm does not modify the head of an S-node, $Head_{G'}(q) = Head_{G_i}(q)$. Therefore $Head_{\tau'}(q) = Head_{G'}(q)$, and τ' is well-defined.

Finally it needs to be shown that there is at most one instantiation of any random variable in τ'. Then let α_1 and α_2 be I-nodes in τ', we need to show that $R_{\alpha_1} \ne R_{\alpha_2}$. We will look at three different cases. First let $\alpha_1 \in \tau$ and $\alpha_2 \in \tau' - \tau$, then $R_{\alpha_1} \ne R_{\alpha_2}$ because τ has only random variables from the original fragments and $\tau' - \tau$ has only random variables corresponding to sources which were created by the algorithm.

So now assume that $\alpha_1, \alpha_2 \in \tau$. Then $R_{\alpha_1} \ne R_{\alpha_2}$ or else τ would not be a valid inference.

Now assume that $\alpha_1, \alpha_2 \in \tau' - \tau$ (i.e. they are source nodes). Then $R_{\alpha_1} \neq R_{\alpha_2}$ because all sources in τ' were inserted by the algorithm while processing fragment K_i (the fragment that inference τ came from). The algorithm creates one unique random variable for each unique S-node in the original fragments. Since source nodes must all point to an S-node that is supporting a member of τ, both α_1 and α_2 are in the tail of some S-node(s) that support members of τ. Since τ is an inference, it has at most one instantiation of each random variable in it, and each I-node has only one S-node supporting it. The algorithm creates a source node for every S-node in an input fragment and the random variable of an S-node is determined by the I-node it points to. In this case, α_1 and α_2 point to different S-nodes in τ which must point to different I-nodes in τ. Therefore $R_{\alpha_1} \neq R_{\alpha_2}$.

Thus τ' is an inference that contains q and q is grounded. □

In addition to groundedness of S-nodes, we are also interested in whether I-nodes or sets of I-nodes participate in any inferences.

Corollary 3. *If all S-nodes in the input BKBs are grounded in their respective BKBs, then all I-nodes in K' are grounded in K'.*

Proof. Assume α is an I-node in K', then it must be in either the head or the tail of some S-node q. By the previous result, q is in some inference τ' in the fused BKB. Since τ' is well-founded and well-defined, the head and tail of q in K' must also be in τ'. Therefore α is in τ' and so α is grounded. □

This establishes that single I-nodes remain grounded in the fused BKB, but what about sets of I-nodes?

Definition. A state θ is *well-represented* in a BKB K if there exists an inference $\tau = (I' \cup S', E)$ over K whose I-node set I' coincides with θ, i.e. if $\theta \subseteq I'$.

Similar to groundedness, we can show that the fusion algorithm also preserves well-representedness:

Corollary 4. *All well-represented states in the input fragments are well-represented in K'.*

Proof. If a state θ was well-represented in some fragment K_i then it was in an inference τ in K_i. After the fusion process it is also in the inference τ' discussed in proof of the previous theorem. □

Finally it would be nice to know how much larger the output of our fusion algorithm is than would be the output of a naive union of the inputs. i.e. how many nodes are we adding through the course of the algorithm?

Theorem 5. *Assume there are n input fragments to the fusion algorithm and fragment $K_1, \ldots, K_n$ where fragment K_i has m_i S-nodes. Let $M = \sum_{i=1}^{n} m_i$, then the fusion algorithm adds $O(M)$ more nodes to the fused output than would result from a naive union of all the fragments.*

Proof. This result follows from the specification of the algorithm. One source I-node is added to each S-node in the algorithm. Then one S-node is added to support this source node. Thus a constant number of nodes are added to the BKB for each S-node in an input fragment. □

This also begs the question of how much complexity is being added to reasoning algorithms that operate on the fused BKB. The complexity in reasoning arises when there are more alternate paths formed through the network that can be used to form an inference graph. Since the source nodes and their supports that are added to the fused BKB do not have anything pointing to them, the additional paths they introduce are very short, and thus the penalty in terms of computational time from adding these nodes should not be too bad. Coming up with specific bounds to characterize the additional complexity and determining what conditions may result in more or less of a penalty will be an important component of future research on the fusion algorithm.

6. Reasoning example

Now, an example of how the fusion process can be used to aggregate information from multiple experts will be given in this section. Assume there are two doctors asked to provide rules to a diagnosis system. They do not have much time to help so they each provide only a few simple rules. The provided fragments are shown on the left of Fig. 12. The fusion of these two fragments is shown on the right of the same figure. In the fragments, random variables A, B, and C correspond to symptoms, and *disease* is a random variable representing the possible diseases a patient may have.

If we are presented with a patient who has symptom A, we can perform belief revision on the fragments to infer a likely disease. From fragment 1 we would conclude the patient had d_1. From fragment 2 we would not be able to make any conclusions, but from the fusion of the two fragments we would see that d_2 was most likely. This would be done by performing belief revision on each BKB to find the most probable inference containing the evidence that ($A = yes$). The result of belief revision on the fused BKB in Fig. 12 is shown in Fig. 13. The setting of each random variable in the most

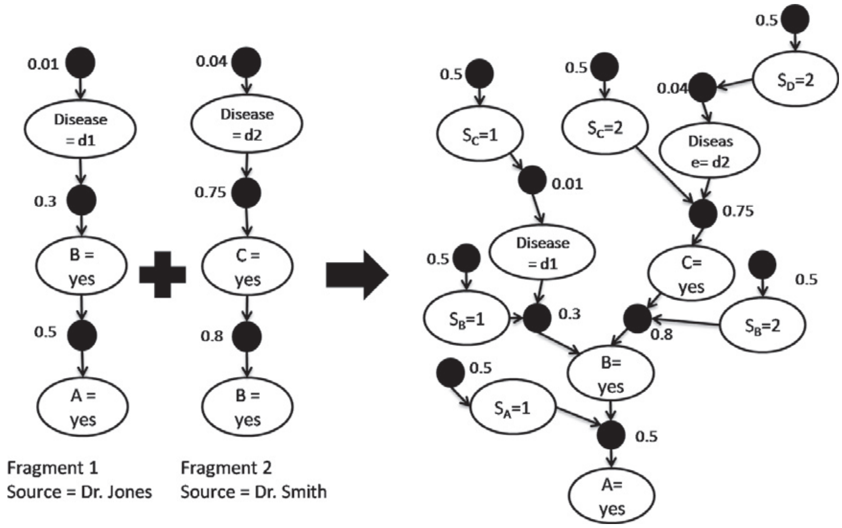

Fig. 12. Fragments from two different doctors on the left. The BKB resulting from their fusion is shown on the right.

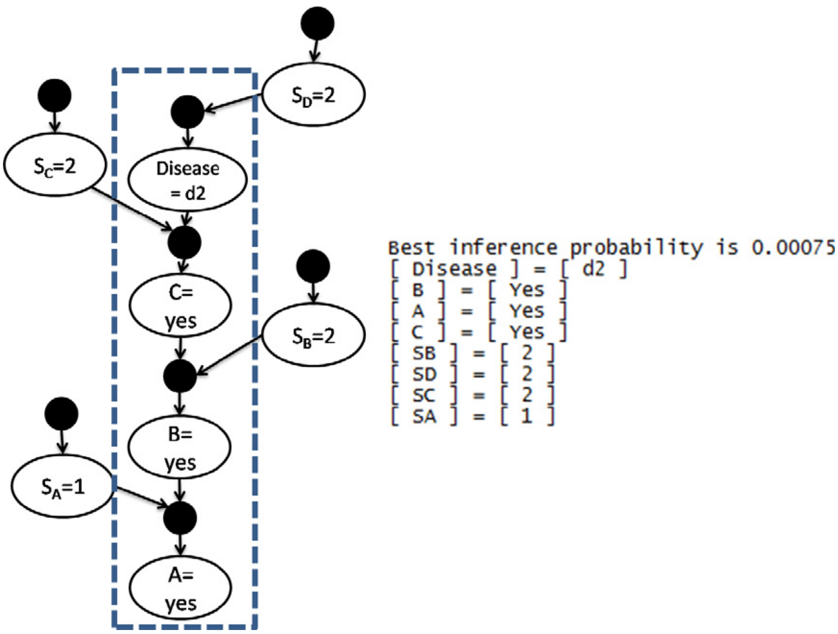

Fig. 13. Most probable inference from BKB in Fig. 12 with evidence ($A = yes$) contains the original nodes (center) and the source nodes that were used to support them.

probable inference is shown. Each source random variable is also assigned to a state indicating which source was used to support the truth of the random variable. The output can be interpreted as follows: The most likely symptoms are *A*, *B*, and *C*. The source of the rule supporting *A* was 1, and the source of the rule supporting *B*, *C*, and the disease was 2 in this inference.

The authors have used the fusion process in modeling multiple real world scenarios. For instance, the 2008 South Carolina Democratic primary election was modeled [22]. In this case, 17 different sources were used such as news reports, blogs, and expert analysis. A fragment was created from each of the sources and then they were fused for analysis. Each fragment averaged roughly five to ten I-nodes and up to 20 S-nodes. The fusion algorithm proved especially helpful in two regards. First, it was difficult for the knowledge engineer to deal with a network the size of final fused BKB. The sheer number of variables and rules that were present made it challenging to check for errors and visualize the relevant portions of the model during editing. It was much easier to work with each of the fragments separately and ensure that any modifications made sense in the context of the single fragment. Second, there were often multiple violations of the mutual exclusivity condition between any two fragments neccesitating either a change in the fragments or an algorithm such as ours in order to aggregate them.

7. Benefits of source nodes

The ability to set and modify source reliabilities as well as include sources as nodes in the probabilistic representation presents an opportunity for extensive analysis that has not been tapped as of yet. For instance, given the case when multiple sources are used repeatedly, their reliability can be tracked and updated over time as the knowledge they provide is tested against the real world. Reliability with respect to a particular topic can also be computed. For instance, when a repeat source presents a new knowledge fragment, we could compare the similarity of this fragment to all other fragments they have provided in the past (perhaps using topic detection or graph comparison algorithms) to get an idea of how reliable the source has been when modeling similar situations in the past.

In prior work [22], the authors developed an algorithm to compute the *contribution* of a variable to the truth of some other variable in a BKB. Given a target I-node q and some other I-node r, the algorithm computes the *contribution* of node r to the probability of q. The algorithm works roughly as follows. First, we note that the probability of q can be found by looking at all possible "worlds" sanctioned by the BKB and summing the probabilities of all worlds in which q is true. Here, a world is simply an assignment of all variables to a state. Let the set of all the worlds in which q is true be W. Then, in order to find the contribution of r to q, we look at the subset $W' \subseteq W$ in which r is also true. The contribution of r to q is the sum of the probabilities of the worlds in W'. Since we are using a BKB and not a BN, we do not always have complete worlds to deal with. The corollary of a world in a BKB is an inference. Thus, to compute contributions, we actually sum over inferences instead of summing over complete states of the world.

Fortunately, the same algorithm can be used to calculate the contribution of a *source* to some variable in a BKB since sources are included as a node in the knowledge base just as the rest of the variables in the model are. Thus it is possible to see the contribution of each source to the truth of each variable in order to, for instance, determine if a decision is being unduly influenced by one or a small set of sources instead of taking a broader look at the situation.

There is also the opportunity to ask "what-if" questions by manipulating the sources. For instance, what if I removed source A? What would the result of reasoning be then? In this case, it could be good to target sources that have a very high contribution to the answers being produced. This could give a decision maker the chance to see if source A is really skewing things, or if the answer would still be the same without the input of this source.

8. Conclusion and future work

In this paper, we introduced an algorithm to fuse several Bayesian Knowledge Fragments into one BKB. This allows the easy aggregation and de-aggregation of chunks of information from multiple sources. The fusion was shown to produce a valid BKB that preserves the rules in the input fragments and their groundedness. The fusion process also allows for the tracking of sources and construction of explanations that contain rules from multiple experts, forming an explanation with greater likelihood than one formed from rules taken from a single expert in isolation. The algorithm allows for disagreement and even circularity in the rules provided by different sources.

Future research in this area will further investigate the properties of fused knowledge bases including the added complexity of reasoning resulting from the addition of source nodes. We will also identify more possible application areas in which to apply the fusion algorithm to demonstrate its usefulness. Another particular problem that will be addressed is the validation of fused BKBs. Previous work has looked at how to validate the correctness of a single BKB with regard to a set of test cases [23,13]. Given a set of fragments that all pass their own individual test cases, we will investigate under what conditions it can be guaranteed that the fused BKB will also be able to pass these test cases. In particular, if the fused BKB does not pass all test cases, is there a way of adjusting the reliabilities of the input fragments so that it does pass? This would provide a way of training the knowledge base without modifying the input fragments themselves, only the weights assigned to each of their sources.

Additionally, it will be interesting to look at the transportability of fragments from one situation to another. It would be ideal if we could build a large library of fragments from which a subset could be drawn and fused to address a particular situation. This raises the question of how easily one can construct fragments that are generalizable and how well they would perform in different application areas. Work by Druzdzel and Díez [24] suggests that it is not easy to see when information from multiple sources can be combined in new situations without introducing some bias that can skew results by orders of magnitude.

Acknowledgements

A preliminary version of this paper can be found in [25]. The authors would like to thank the reviewers for their comments which helped to greatly improve the paper.

This work was supported in part by Defense Threat Reduction Agency Grant No. HDTRA1-07-1-0003, Air Force Office of Scientific Research Grant Nos. FA9550-07-1-0050 and FA9550- 06-1-0169, and Office of Naval Research Grant No. N00014-06-C-0020. Special thanks to the Air Force Research Laboratory Information Directorate and Securboration, Inc. for their continued support and transition.

References

[1] J. Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Morgan Kaufmann, San Mateo, CA, 1988.
[2] E. Santos Jr., E.S. Santos, A framework for building knowledge-bases under uncertainty, Journal of Experimental and Theoretical Artificial Intelligence 11 (2) (1999) 265–286.
[3] E. Santos Jr., E.S. Santos, S.E. Shimony, Implicitly preserving semantics during incremental knowledge base acquisition under uncertainty, International Journal of Approximate Reasoning 33 (1) (2003) 71–94.
[4] T. Rosen, S.E. Shimony, E. Santos Jr., Reasoning with BKBs algorithms and complexity, Annals of Mathematics and Artificial Intelligence 40 (3–4) (2004) 403–425.
[5] D. Koller, A. Pfeffer, Object-Oriented Bayesian Networks, Morgan Kaufmann, 1997., pp. 302–313.
[6] K.B. Laskey, MEBN: a language for first-order Bayesian knowledge bases, Artificial Intelligence 172 (2–3) (2008) 140–178.
[7] C. Sutton, C. Morrison, P.R. Cohen, J. Moody, J. Adibi, A Bayesian blackboard for information fusion, in: Proceedings of the 7th International Conference on Information Fusion, 2004.
[8] M. Jaeger, Probabilistic decision graphs-combining verification and ai techniques for probabilistic inference, International Journal of Uncertainty Fuzziness and Knowledge-Based Systems 12 (Supp. 1) (2004) 19–42., http://dx.doi.org/10.1142/S0218488504002564.
[9] M. Druzdzel, L. van der Gaag, Elicitation of probabilities for belief networks: combining qualitative and quantitative information, in: Proceedings of the Eleventh Annual Conference on Uncertainty in Artificial Intelligence, 1995, pp. 141–148.
[10] D. Heckerman, D.M. Chickering, C. Meek, R. Rounthwaite, C. Kadie, Dependency networks for inference, collaborative filtering, and data visualization, Journal of Machine Learning Research 1 (2000) 49–75.
[11] T. Shipman, Obama supporter warns Clinton over 'racism', The Daily Telegraph (January 19). Available from: <http://www.telegraph.co.uk/news/worldnews/1576021/Obama-supporter-warns-Clinton-over-racism.html>.
[12] S.E. Shimony, E. Santos Jr., T. Rosen, Independence semantics for BKBs, Proceedings of the Thirteenth International Florida Artificial Intelligence Research Society Conference, AAAI Press, 2000, pp. 308–312.
[13] E. Santos Jr., H.T. Dinh, On automatic knowledge validation for Bayesian knowledge bases, Data and Knowledge Engineering 64 (1) (2008) 218–241.
[14] I. Matzkevich, B. Abramson, The topological fusion of bayes nets, UAI '92: Proceedings of the Eighth Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1992, pp. 191–198.
[15] C.A. Jiang, T.-Y. Leong, K.-L. Poh, Pgmc: a framework for probabilistic graphic model combination, AMIA Annu. Symp. Proc., 2005, pp. 370–374.
[16] K.B. Laskey, S.M. Mahoney, Network fragments: representing knowledge for constructing probabilistic models, Proceedings of Thirteenth Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann, 1997, pp. 334–341.
[17] S.M. Mahoney, K.B. Laskey, Constructing situation specific belief networks, Proceedings of Fourteenth Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann, 1998, pp. 370–378.
[18] S.M. Mahoney, K.B. Laskey, Representing and combining partially specified cpts, Proceedings of Fifteenth Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann, 1999, pp. 391–400.
[19] T.-C. Lu, M.J. Druzdzel, Interactive construction of graphical decision models based on causal mechanisms, European Journal of Operational Research 199 (3) (2009) 873–882., doi:10.1016/j.ejor.2009.01.056.
[20] J. Cham, The vicious cycle, Piled Higher and <DeeperHttp://www.phdcomics.com/comics/archive.php?comicid=1015>.
[21] S.E. Shimony, The role of relevance in explanation I: irrelevance as statistical independence, International Journal of Approximate Reasoning 8 (4) (1993) 281–324., doi:10.1016/0888-613X(93)90027-B.
[22] E.E. Santos, E. Santos Jr., J.T. Wilkinson, H. Xia, On a framework for the prediction and explanation of changing opinions, SMC'09: Proceedings of the 2009 IEEE International Conference on Systems, Man and Cybernetics, IEEE Press, Piscataway, NJ, USA, 2009, pp. 1446–1452.
[23] E. Santos Jr., Verification and validation of Bayesian knowledge-bases, Data and Knowledge Engineering 37 (3) (2001) 307–329., doi:http://dx.doi.org/10.1016/S0169-023X(01)00011-8.
[24] M.J. Druzdzel, F.J. Díez, Combining knowledge from different sources in causal probabilistic models, The Journal of Machine Learning Research 4 (2003) 295–316.
[25] E. Santos Jr., J.T. Wilkinson, E. Santos, Bayesian knowledge fusion, Proceedings of the Florida Artificial Intelligence Research Society Conference Twenty-Second International FLAIRS Conference, AAAI Press, 2009, pp. 559–564.