Target Beliefs for SME-oriented, Bayesian Network-based Modeling

Robert Schrag
Haystax Technology
11210 Corsica Mist Ave
Las Vegas, NV 89135

Edward Wright, Robert Kerr, Robert Johnson
Haystax Technology
8251 Greensboro Dr, Suite 1000
McLean, VA 22102

Abstract

Our framework supporting non-technical subject matter experts' authoring of useful Bayesian networks has presented requirements for fixed probability soft or virtual evidence findings that we refer to as target beliefs. We describe exogenously motivated target belief requirements for model nodes lacking explicit priors and mechanistically motivated requirements induced by logical constraints over nodes that in the framework are strictly binary. Compared to the best published results, our target belief satisfaction methods are competitive in result quality and processing time on much larger problems.

1. INTRODUCTION

The variety of soft or virtual evidence finding on a Bayesian network (BN) node in which a specified probability distribution must be maintained during BN inference—called a fixed probability finding by (Ben Mrad, 2015) and called a target belief here—has received limited attention. Published results for inference algorithms respecting such findings have addressed small, artificial problems including at most 15 nodes (Peng et al., 2010; Zhang et al., 2008).

Our work on one real application has required addressing dozens of such findings in a BN comprising hundreds of nodes. In this context, target beliefs are motivated by modelers' need to address authoritative sources exogenous to the model itself, where beliefs should hold for selected non-BN root model nodes—i.e., nodes lacking explicit prior probability distributions (that otherwise might be used to achieve target beliefs directly).

For example, if a binary node Divorces appears deep in a person risk assessment network as an indicator of a top-level binary node Trustworthy, usually (without target beliefs or other node findings) the network's computed belief in Divorces will depend on the network's conditional probability tables (CPTs)[1]—not on a published statistic about the divorce rate in an intended subject population. To make our model's belief in Divorces agree with the exogenous statistic, a modeler can:

1. Adjust CPTs throughout the model to agree with the exogenous specification.
2. Invoke Jeffrey's rule (Jeffrey, 1983) to compute a likelihood finding on Divorces that achieves the specified belief.
3. Specify a target belief for Divorces and rely on target belief satisfaction machinery to achieve the target.

The first option is not entirely compatible with our modeling framework.[2] The modeler's manual effort under either of the first two options may be undermined as soon as s/he modifies the model again.[3] The last option offloads the work of target belief satisfaction to an automated process—at the expense of executing that process, as often as necessary. Execution time may be acceptable for a given use case if the model is small, if it is not modified often, or if model development is sufficiently simplified under this approach to enhance overall productivity. As we intend our framework to be subject matter expert- (SME-)friendly, this option is attractive. The more we can free a modeler to concentrate on higher-level decisions with greater domain impact, the more and better models s/he should be able to deliver.

[1] Including top-level node priors as a degenerate case.

[2] Our framework automatically computes CPTs (see section 2) to reflect a modeler's specified strength with which a child node (counter-)indicates its parent node. So, modifying CPTs is appropriate only when modifying these strengths is. Likewise, the representation would not naturally accommodate a conventional approach to machine learning of CPT entries.

[3] In principle, any of a large variety of modifications—including more invocations of this option to address additional exogenous probabilities—could affect computed belief in Divorces.

Our work adapting the framework to realize probabilistic argument maps for intelligence analysis (Schrag et al., 2016a; 2016b) has surfaced powerful representations (Logic constraints—see section 4) that can improve model clarity and correctness and that often require target beliefs.

In the following sections, we outline the framework, our large person risk assessment model, and the view of framework models as probabilistic argument maps. We explain how Logic constraints can improve arguments (models) and how target beliefs can support such constraints. We briefly review existing competitive target belief processing methods, then describe our own method and results.

2. SME-ORIENTED MODELING FRAMEWORK

We developed the framework to facilitate creation of useful BNs by non-technical SMEs. Faced with the challenge of operationalizing SMEs' policy-guided reasoning about person trustworthiness in a comprehensive risk model (Schrag et al., 2014), we first developed a model encoding hundreds of policy statements. The need for SMEs both to understand the model and to author its elements inspired us to develop and apply a technical approach using exclusively binary random variables (BN nodes) over the domain {true, false}. This led us to an overall representation that happens to extend standard argument maps (CIA, 2006) with Bayesian probabilistic reasoning (Schrag et al., 2016a; 2016b).

In the framework, every node (or argument map statement[4]) is a Hypothesis. Some Hypotheses are Logic nodes whose CPTs are deterministic. Connecting the nodes are links whose types are listed in Table 1. Argument maps' SupportedBy and RefutedBy links correspond to our IndicatedBy and CounterIndicatedBy links.

Table 1: Framework link types (center column). For the last two link types, the argument map-downstream statement (BN-downstream node) is a Logic node.

Argument map-downstream[5] statement	IndicatedBy	Argument map-upstream statement(s)
	CounterIndicatedBy	
	MitigatedBy	
	RelevantIf	
	OppositeOf	
	ImpliedByConjunction	
	ImpliedByDisjunction	

We encode strengths for non-Logic node-input links (first four rows of Table 1) using fixed odds ratios per Figure 1.

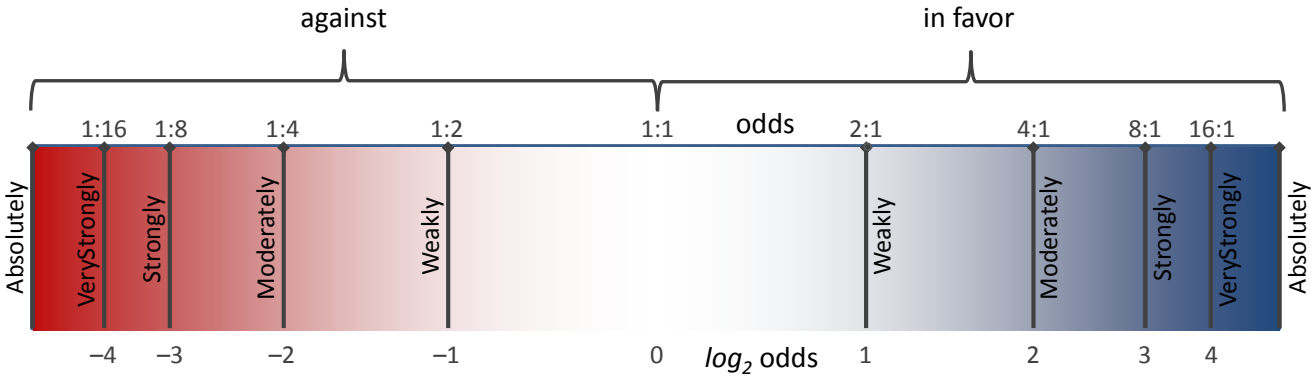

Figure 1: Odds ratios for discrete link strengths. Absolutely is intended as logical implication. We do not otherwise commit SMEs to absolute certainty.

[4] Our binary BN nodes correspond to propositions bearing truth values. In the argument map point of view, these propositions may be understood to be statements.

[5] Per argument map convention, "downstream" is left, "upstream" right in the left-flowing argument map of Figure 3. Except for Logic nodes, this is opposite of links' causal direction in BNs.

A framework process (Wright et al., 2015) converts specifications into corresponding BNs. The conversion process recognizes a pattern of link types incident on a given node and constructs an appropriate CPT reflecting specified polarities and strengths. The SME thus works in a graphical user interface (GUI) with an argument map representation (as if at a "dashboard"), and BN mechanics and minutiae all remain conveniently "under the hood."

The framework includes stock noisyOr and noisyAnd distributions (bearing a standard Leak parameter) for BN nodes with more than one parent. While these have so far been sufficient in our modeling efforts, we also could fall back to fine distribution specification. We have deliberately designed the framework to skirt standard CPT elicitation, which can tend to fatigue SMEs. Consider an indicator of h different Hypotheses, so with h BN parents and 2^h CPT rows. Suppose belief is discretized on a 7-point scale.[6] Then standard, row-by-row elicitation requires 2^h entries. With noisyOr or noisyAnd, we need only h entries bearing a polarity and strength for each parent, plus a Leak value for the distribution.

We are working to make modeling in the framework more accessible to SMEs, particularly via model editing capabilities in the GUI exhibited in Figure 3. (Schrag et al., 2016a) describes our framework encoding of an analyst's argument, favorable comparison of resulting modeled probabilities to analyst-computed ones, and favorable comparison of CPTs generated by the framework vs. elicited directly from analysts.

3. PERSON RISK MODEL WITH EXOGENOUS BELIEF REQUIREMENTS

Our person risk assessment application includes a core generic person BN accounting for interactions among beliefs about random variables representing different person attribute concepts like those in Figure 2.

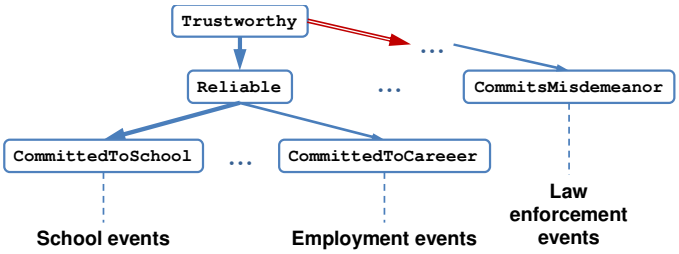

Figure 2: Partial generic person attribute concept BN (top), with related event categories (bottom). BN influences point (causally) from indicated concept hypothesis to indicating concept. Stronger indications have thicker arrows. A single negative indication has a red, double-lined arrow.

The framework processes a given person's event evidence to specialize this generic BN into a person-specific BN (Schrag et al., 2014).

We have specified target beliefs for some two dozen nodes in the generic person network. By processing the target beliefs in an event evidence-free context, we ensure that events have the effects intended, respecting both indication strengths and exogenous statistics.[7]

4. INTELLIGENCE ANALYSIS MODEL MOTIVATING REQUIRMENTS FROM LOGIC CONSTRAINTS

Figure 3 is a screenshot of a model addressing the CIA's Iraq retaliation scenario (Heuer, 2013)[8], where Iraq might respond to US forces' bombing of its intelligence headquarters by conducting major, minor, or no terror attacks, given limited evidence about Saddam Hussein's disposition and public statements, Iraq's historical responses, and the status of Iraq's national security apparatus. This model emphasizes Saddam's incentives to act. By setting a hard finding of false on the incentive-collecting node SaddamWins, we can examine computed beliefs under Saddam's worst-case scenario (and, by comparing this to his best-case scenario, determine that conducting major terror attacks is not his best move). See (Schrag et al., 2016a) for details.

[6] As (Karvetski et al., 2013) note, the inference quality of models developed this way usually rivals that of models developed with arbitrary-precision CPTs.

[7] Such a dividing line between generic model and evidence may not be so bright in a probabilistic argument map, where an intelligence analyst may enter both hypothesis and evidence nodes incrementally.

[8] See chapter 8, "Analysis of Competing Hypotheses."

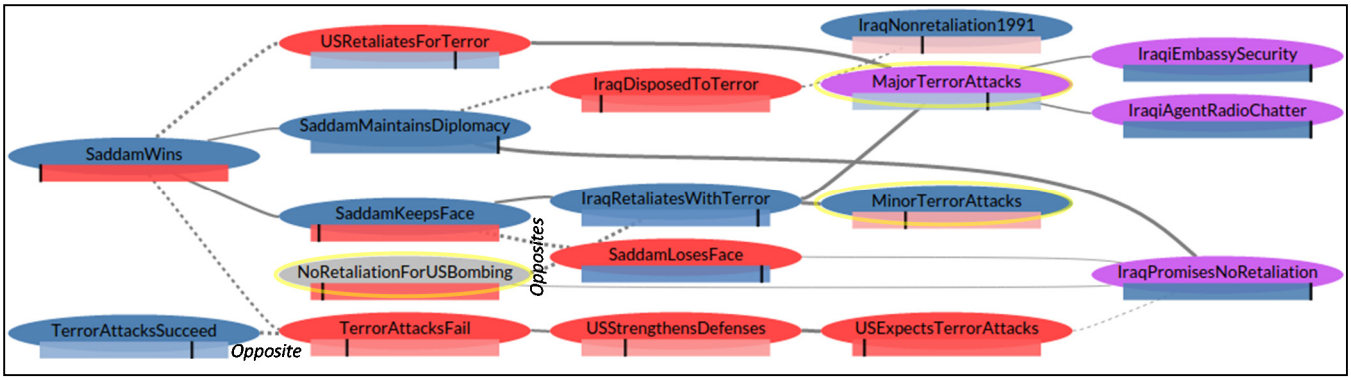

Figure 3: Statement nodes are connected by positive (solid grey line) and negative (dashed grey line) indication links of various strengths (per line thicknesses). Argument flow (from evidence to outcomes) is from right to left—e.g., SaddamWins is strongly indicated by SaddamKeepsFace. Outcome hypothesis nodes are circled in yellow. SaddamWins (hard finding false) captures Saddam's incentives to act or not. Belief bars' tick marks fall on a linear scale. Colors are explained in (Schrag et al., 2016a), also (Schrag et al., 2016b).

In developing the model in Figure 3, we identified some representation and reasoning shortcomings for which we are now implementing responsive capabilities (Schrag et al., 2016b). Relevant to our discussion here, TerrorAttacksFail (likewise TerrorAttacksSucceed) should be allowed to be true only when TerrorAttacks also is true.

We are working towards Logic nodes supporting any propositional expression using unary, binary, or higher arity operators[9]. When a Logic statement has a hard true finding[10], we refer to it as a Logic constraint, otherwise as a summarizing Logic statement.

We know that an attempted action can succeed or fail only if it occurs. By explicitly modeling (as Hypotheses) both the potential action results and adding a Logic constraint[11], we can force zero probability for every excluded truth value combination, improving the model. See Figure 4. The constraint node (left, in right model fragment) ensures that the model will believe in attack success/failure only when an attack actually occurs. Setting the hard true finding on this node turns the summarizing Logic statement (left, in the left fragment) into the Logic constraint—but also distorts the model's computed probabilities for the three Hypotheses. Presuming these probabilities have been deliberately engineered by the modeler, our framework must restore them. It does so by implementing (bottom fragment) a target belief (per the ConstraintTBC node) on one of the Hypotheses.

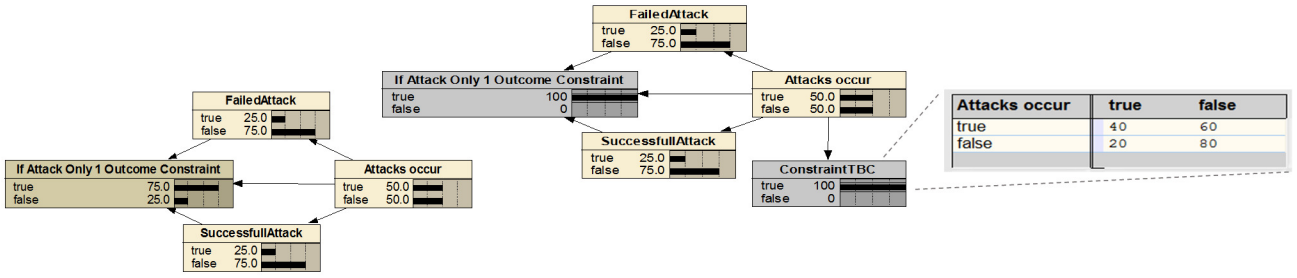

Figure 4: Logic constraints can help ensure sound reasoning.

[9] See, e.g., https://en.wikipedia.org/wiki/Truth_table.

[10] A likelihood finding could be used to implement a soft constraint.

[11] This constraint can be rendered (abbreviating statement names) as (or (and occur (xor succeed fail)) (and (not Occurs) (nor Succeeds Fails))) or more compactly via an if-then-else logic function (notated ite) as (ite Occurs (xor Succeeds Fails) (nor Succeeds Fails))—if an attack occurs, it either succeeds or fails, else it neither succeeds nor fails.

We implement a target belief either (depending on purpose) using a BN node like ConstraintTBC or (equivalently) via a likelihood finding on the subject BN node. The GUI does not ordinarily expose an auxiliary node like ConstraintTBC to a SME/analyst-class user.

This example is for illustration. We can implement this particular BN pattern without target beliefs. We also could implement absolute-strength IndicatedBy links as simple implication Logic constraints. However, this would not naturally accommodate one of these links' key properties—the ability to specify degree of belief in the link's upstream node when the downstream node is true—relevant because we can infer nothing about P given $P \Rightarrow Q$ and knowing Q to be true. It also demands two target belief specs that tend to compete. We are working to identify more Logic constraint patterns that can be implemented without target beliefs and to generalize specification of belief degree for any underdetermined entries in a summarizing Logic statement's CPT.

5. TARGET BELIEF PROCESSING

Ben Mrad et al. (2015) survey BN inference methods addressing fixed probability findings—our target beliefs. The most recent published results (Peng et al., 2010) address problems with no more than 15 nodes (all binary). Apparently, earlier approaches materialized full joint distributions—these authors anecdotally reported late-breaking results using a BN representation, with dramatically improved efficiency. Mrad et al. report related capabilities in the commercial BN tools Netica and BayesiaLab. Netica's "calibration" findings are concerned with comparing predictions to real data and could help identify where target beliefs were needed, however would do nothing to satisfy them. We have not experimented with BayesiaLab. While our performance results may similarly be construed as anecdotal—we have not systematically explored a relevant problem space—we have addressed a much larger problem. Our person risk assessment BN includes over 600 nodes and 26 target beliefs.

The basic scheme of our target belief processing approach is to interleave applications of Jeffrey's rule[12] with standard BN inference. Intuitively, each iteration—or "fitting step" (Zhang, 2008)—measures the difference between affected nodes' currently computed beliefs and specified target beliefs, makes changes to bring one or more nodes closer to target, and propagates these changes in BN inference. We continue iterating until a statistic over computed-vs.-target belief differences meets a desired criterion, or until reaching a limit on iterations, in which case we report failure. Just as for hard findings and likelihood findings, not all sets of target beliefs can be achieved simultaneously. In our intended incremental model development concept of operations (CONOPS), the framework's report that a latest-asserted target belief induces unsatisfiability should be taken as a signal that a modeling issue requires attention—much as would the similar report about a latest-asserted CPT.

We have implemented the following refinements to this basic scheme, improving performance.

1. Measure beliefs on a (modified) log odds scale.
2. Conservatively[13] apply Jeffrey's rule to all affected nodes in early iterations/fitting steps, then in late steps select for adjustment just the node with greatest difference between computed and target beliefs.
3. Save the work from previous target belief processing for a given model (e.g., under edit) to support fast incremental operation.

5.1 MODIFIED LOG ODDS BELIEF MEASUREMENT

Calculating the differences between beliefs measured on a scale in the log odds family, vs. on a linear scale, better reflects differences' actual impacts. We use the function depicted in Figure 5—a variation on log odds in which each factor of 2 less than even odds (valued at 0) loses one unit of distance that we refer to as a bit. So, for belief = 0.125 we calculate –2 bits.

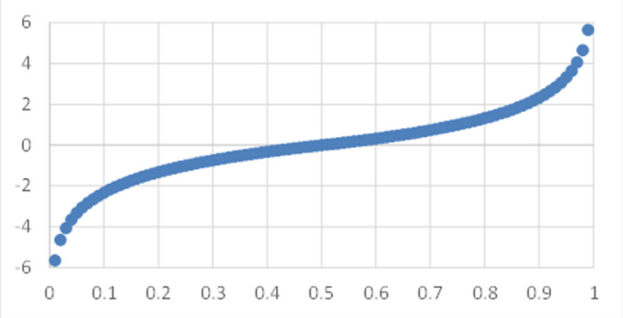

Figure 5: Belief transformation function (modified log odds) used in calculating computed-vs.-target belief differences

We express differences between beliefs in terms of such bits. So, difference(0.999, 0.87) = 7.02 bits and difference(0.87, 0.76) = 0.90 bits, whereas both pairs of untransformed beliefs (that is, (0.999, 0.87) and (0.87, 0.76)) have the same ratio, 1.14.[14] The transformation

[12] See (Jeffrey, 1983), as mentioned in section 1.

[13] See section 5.2.

[14] This difference metric is more conservative than the Kullback-Leibler distance or cross-entropy metric used in (Peng et al., 2010)'s I-divergence calculation. The absolute value of this function also has the advantage of being symmetric.

seems to inhibit oscillations among competing target beliefs.

5.2 MULTIPLE ADJUSTING IN ONE FITTING STEP

Moving all affected nodes all the way to their target beliefs in one fitting step is too aggressive in this model. We can get closer to a solution by adjusting more conservatively. We found that applying Jeffrey's rule to take affected variables {½ , 1/3, ¼, ...} of the way toward their target beliefs in successive fitting steps worked better than scaling calculated differences by any fixed proportion. This trick seems to be advantageous just for the first two or three fitting steps, after which single-node adjustments become more effective.

Incorporating both this refinement and the preceding one and running with a maximum belief difference of 0.275 bits for any node (yielding adequate model fidelity for our application), we complete target belief processing in 19 seconds (running inside a Linux virtual machine on a 2012-vintage Dell Precision M4800 Windows laptop).[15] That's not necessarily GUI-fast, but this is a larger model than many of our SME users may ever develop. Fitting steps took a little less than one second on average, with each step's processing dominated by the single call to BN inference.

These results remain practically anecdotal, as we have so far developed in our framework only this one large model including many target beliefs. Experience with different models may lead to more generally useful values for run-time parameters.

5.3 INCREMENTAL OPERATION

Under incremental operation, we execute only single-node fitting steps, as individual model edits usually have limited effect on overall target belief satisfaction. So far, we have experimented with incremental operation only for our person risk model.

Over two runs (with target beliefs processed in original input order vs. reversed):

- Average processing times per affected node were 2.1 and 2.3 seconds, respectively. Individual target beliefs processed in about 1.1 seconds or less about half the time. Figure 6 plots processing times for the first run, by affected node number, including a 4-node moving average.
- The least number of fitting steps was 0, the greatest 17 (taking from 0 to 8.7 seconds).
- Total run times were 54 and 59 seconds, respectively. So, batch (vs. incremental) processing can be advantageous, depending on CONOPS and use case.

[15] We found that tightening tolerance by a factor of 6.6 increased run time by a factor of 3.0.

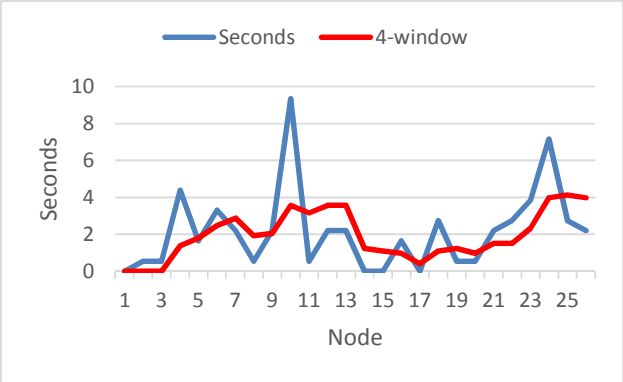

Figure 6: Run-time by affected node increment, with 4-node moving average window

6. CONCLUSION

Target beliefs have an important place in our SME-oriented modeling framework, where their processing is supported effectively by our methods described here. We might reduce or eliminate requirements for exogenous target beliefs by pushing SMEs towards arbitrary-precision link strengths (see Schrag et al. 2016b), but we are counting on target belief machinery to implement Logic constraints that make the SMEs' accessible modeling representation more expressive and versatile—ultimately more powerful. We expect target belief processing to be well within GUI response times for small models, including, per (Burns, 2015), the vast majority of intelligence analysis problems amenable to our argument mapping approach. We anticipate further work, especially to develop theory and practice for efficient implementation of different Logic constraint patterns.

Acknowledgements

We gratefully acknowledge the stimulating context of broader collaboration we have shared with other co-authors of (Schrag et al., 2016a; 2016b).

References

Ali Ben Mrad, Veronique Delcroix, Sylvain Piechowiak, Philip Leicester, and Mohamed Abid (2015), "An explication of uncertain evidence in Bayesian networks: likelihood evidence and probabilistic evidence," *Applied Intelligence,* published online 20 June 2015.

Kevin Burns (2015), "Bayesian HELP: Assisting Inferences in All-Source Intelligence," *Cognitive Assistance in Government,* Papers from the AAAI 2015 Fall Symposium, 7–13.

CIA Directorate of Intelligence, "A Tradecraft Primer: The Use of Argument Mapping," *Tradecraft Review* 3(1), Kent Center for Analytic Tradecraft, Sherman Kent School, 2006.

Richards J. Heuer, Jr. (2013), *Psychology of Intelligence Analysis,* Central Intelligence Agency Historical Document. https://www.cia.gov/library/center-for-the-study-of-intelligence/csi-publications/books-and-monographs/psychology-of-intelligence-analysis (Posted: Mar 16, 2007 01:52 PM. Last Updated: Jun 26, 2013 08:05 AM.)

R. Jeffrey (1983), *The Logic of Decision, 2nd Edition,* University of Chicago Press.

Christopher W. Karvetski, Kenneth C. Olson, Donald T. Gantz, and Glenn A. Cross (2013), "Structuring and analyzing competing hypotheses with Bayesian networks for intelligence analysis," *EURO J Decis Process* 1:205–231.

Yun Peng, Shenyong Zhang, Rong Pan (2010), "Bayesian Network Reasoning with Uncertain Evidences," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems,* 18(5):539–564.

Robert Schrag, Edward Wright, Robert Kerr, and Bryan Ware (2014), "Processing Events in Probabilistic Risk Assessment," *9th International Conference on Semantic Technologies for Intelligence, Defense, and Security (STIDS).*

Robert Schrag, Joan McIntyre, Melonie Richey, Kathryn Laskey, Edward Wright, Robert Kerr, Robert Johnson, Bryan Ware, and Robert Hoffman (2016a), "Probabilistic Argument Maps for Intelligence Analysis: Completed Capabilities," *16th Workshop on Computational Models of Natural Argument*.

Robert Schrag, Edward Wright, Robert Kerr, Robert Johnson, Bryan Ware, Joan McIntyre, Melonie Richey, Kathryn Laskey, and Robert Hoffman (2016b), "Probabilistic Argument Maps for Intelligence Analysis: Capabilities Underway," *16th Workshop on Computational Models of Natural Argument*.

Edward Wright, Robert Schrag, Robert Kerr, and Bryan Ware (2015), "Automating the Construction of Indicator-Hypothesis Bayesian Networks from Qualitative Specifications," Haystax Technology technical report, https://labs.haystax.com/wp-content/uploads/2016/06/BMAW15-160303-update.pdf.

Shenyong Zhang, Yun Peng, and Xiaopu Wang (2008), "An Efficient Method for Probabilistic Knowledge Integration," In *Proceedings of The 20th IEEE International Conference on Tools with Artificial Intelligence (ICTAI),* November 3–5, vol 2. Dayton, pp 179–182).