Complexity Reduction of Influence Nets Using Arc Removal

Sajjad Haider[a,*] and Syed Ali Raza[b]

[a]*Artificial Intelligence Lab, Faculty of Computer Science, Institute of Business Administration, City Campus, Garden/Shaheed Kiyani Rd., Karachi, Pakistan*

[b]*Center of Quantum Computation and Intelligent System, Faculty of Engineering and Information Technology, University of Technology, Sydney, 81 Broadway, Ultimo, Sydney, NSW, Australia*

Abstract. The model building of Influence Nets, a special instance of Bayesian belief networks, is a time-consuming and labor-intensive task. No formal process exists that decision makers/system analyst, who are typically not familiar with the underlying theory and assumptions of belief networks, can use to build concise and easy-to-interpret models. In many cases, the developed model is extremely dense, that is, it has a very high link-to-node ratio. The complexity of a network makes the already intractable task of belief updating more difficult. The problem is further intensified in dynamic domains where the structure of the built model is repeated for multiple time-slices. It is, therefore, desirable to do a post-processing of the developed models and to remove arcs having a negligible influence on the variable(s) of interests. The paper applies sensitivity of arc analysis to identify arcs that can be removed from an Influence Net without having a significant impact on its inferencing capability. A metric is suggested to gauge changes in the joint distribution of variables before and after the arc removal process. The results are benchmarked against the KL divergence metric. An empirical study based on several real Influence Nets is conducted to test the performance of the sensitivity of arc analysis in reducing the model complexity of an Influence Net without causing a significant change in its joint probability distribution.

Keywords: Bayesian Networks, Influence Nets, Sensitivity Analysis, Model Building

1. Introduction

Bayesian networks (BNs) have become the tool of choice for reasoning in uncertain domains. They have been successfully applied in several domains ranging from medical diagnosis to anti-terrorism and from information fusion to system troubleshooting. Mittal and Kassim [25] and Pourret et al. [28] provide an extensive list of areas where BNs have been applied. Mathematically, a BN is a graphical representation of a joint probability distribution. It consists of two components: (a) a directed acyclic graph in which each node represents a random variable while arcs between pairs of nodes represent certain conditional independence properties and (b) a collection of parameters that describes the conditional probability of each variable given its parent in the graph. Together, these two components represent a unique probability distribution [27].

A lot of research in the field of Bayesian belief networks is driven by their two major limitations: (a) difficulty of knowledge elicitation and (b) intractability of belief updating algorithms. The former is related to the exponential number of parameters required to fill the conditional probability table (CPT) of a variable while the latter is related to the non-polynomial time required to do belief updating in a BN [8]. Different schemes and approximate/simulation-based algorithms have been suggested that aim to tackle the belief updating issue [1, 7, 12-14, 29]. For knowledge elicitation, schemes such as Noisy-Or [27, 28] and the CAST logic [6, 29] have been proposed that ask a linear number of parameters from a subject matter expert and convert them into

* Corresponding author. E-mail: sajjad.haider@khi.iba.edu.pk; Tel.: (+92-21) 111-677-677 (Ext. 1323)

conditional probability tables (having exponential number of parameters). The variant of Bayesian networks that uses the CAST logic for knowledge elicitation is referred to as Influence Nets [32]. The primary purpose of building an IN is to connect a set of actionable events to a desired effect through chains of cause and effect relationships. The IN is then used to identify a course of action that has the highest probability of achieving the desired effect. Influence Nets and their extensions, Timed Influence Nets and Dynamic Influence Nets, have been experimentally used in Effects-based Operations [11, 16-19, 21, 30-35].

Despite several enhancements on knowledge elicitation and belief updating fronts, the model construction of Bayesian belief networks and its variants (such as Influence Nets) in large and complex domains is an extremely difficult and a pain-stacking task. Many times a subject matter expert building such models is not aware of the underlying conditional independence assumption and builds a very dense model (where each node is connected to almost every other node in the network). This not only complicates the already intractable task of belief updating but also reduces the readability of the model. Thus, there is a need for a support tool that can aid in reducing the complexity of a belief network without causing a significant change in the underlying joint probability distribution. This complexity reduction would also be very helpful when the belief network is used in dynamic domains where its structure is repeated for multiple time slices.

The approach presented in this paper uses sensitivity of arc analysis as a heuristic for complexity reduction of an Influence Net. It takes advantage of the local parameter specification mechanism used by the CAST logic. Instead of asking complete conditional probability tables for each node in an IN, the logic allows a subject matter expert to specify the influence of a parent node on the child node while ignoring all other influences on the child node. This unique feature of INs is exploited in this paper and, together with the sensitivity of arc analysis, it aids in removing insignificant arcs from the modeled IN. The approach changes the CAST logic parameters to their extreme values and then analyze the impact of these changes on the desired effect to decide whether the corresponding arc can be removed from the network or not.

It is worth mentioning that efforts have been made within the Bayesian network domain to apply sensitivity analysis to assess the impact of change in a single parameter on the variable(s) of interest. This change in parameter can be in the form of obtaining soft/hard evidence on certain variables or by changing values in the conditional probability table of a variable [2, 3, 10, 36]. Efforts have also been made to analyze the impact of changing multiple parameters at a time although the process itself is computationally expensive [5, 23]. When done in consultation with domain experts and as a model refinement tool, the sensitivity analysis helps in refining the parameters of a BN according to experts' satisfaction [9]. Chan [4] provides a good discussion on the issues related to sensitivity analysis in Bayesian belief networks.

Before opting for any scheme (be the ones presented in this paper or any other) for removing insignificant arcs from an Influence Net, it is important to make sure that the arc removal process does not result in changing the joint distribution of the model variables in a significant way. The exact way of gauging this change is to compare the joint probability distribution of the modeled variables before and after the arc removal process. This, however, requires generating and comparing an exponential number of parameters (and that too of a very small magnitude as the sum should add up to 1) and is therefore an intractable process. The paper presents an approximate way of gauging the impact of arc removal process on the probability of variable(s) of interests. The metric makes use of the sets of actions finder (SAF) algorithm [20, 22]. As stated earlier, INs are typically built to determine the cause and effect relationships among actionable events and a desired effect connected through chains of uncertain variables. Thus, the primary variable of interest in an Influence Net is the desired effect and how it is impacted by different states of actionable events. The SAF algorithm aims to find this impact. It runs in quadratic time and provides a reasonable mechanism to gauge changes in the joint probability distribution before and after the arc removal process. It is benchmarked against Kullback-Leiber (KL) divergence [24] method for comparing distance between the two probability distributions.

The rest of the paper is organized as follows. Section 2 provides an overview of Influence Nets along with the CAST logic. The proposed heuristic, sensitivity of arc analysis, to remove insignificant arcs from an IN is explained in Section 3. KL-divergence and the SAF algorithm based metric, to estimate the changes in joint probability distributions before and after arc removal process, are discussed in Section 4. Experimental design, models used in the experiments, and the results are described in Section 5. Finally, Section 6 concludes the paper and provides the future research directions.

2. Influence Nets

Influence Nets are Directed Acyclic Graphs (DAGs) where nodes in the graph represent random variables, while the edges between pairs of variables represent causal relationships. The modeling of the causal relationships is accomplished by creating a series of cause and effect relationships among variables representing a set of actionable events and a desired effect. The actionable events are drawn as root nodes (nodes without incoming edges), while the desired effect is modeled as a leaf node (node without outgoing edges). Typically, the root nodes are drawn as rectangles while the non-root nodes are drawn as rounded rectangles.

Fig. 1 shows a portion of an Influence Net developed to model the political crisis that occurred in East Timor during the late 90s [33]. The model was developed as a prototype for the Decision Support System for Coalition Operations developed by SPAWAR Systems Center-San Diego to support the Operations Planning Team of the Commander in Chief, U.S. Pacific Command. As mentioned above, the leaf node "Rebels decide to avoid violence" is the desired effect, while the root nodes, such as "US president declares resolve to keep peace in Indonesia", "Coalition forces deploys forces to Indonesia", etc., represent actionable events. The directed edge with an arrowhead between two nodes shows the parent node promoting the chances of a child node being true, while the roundhead edge shows the parent node inhibiting the chances of a child node being true.

The strength of the positive or negative influence of the presence/absence of each event on its child event is captured through the CAST logic parameters. These parameters are linear in terms of number of arcs in the network and thus relieve the experts from specifying an exponential number of conditional probability values – a daunting task when each node has many parents (incoming arcs). The conditional probabilities values are internally generated by the CAST logic algorithm. The Influence Nets are therefore appropriate for the following situations: i) for modeling situations in which it is difficult to fully specify all conditional probability values, and/or ii) the estimates of conditional probabilities are subjective, and iii) estimates for the conditional probabilities cannot be obtained from empirical data, e.g., when modeling potential human reactions and beliefs [31]. The following items characterize an IN [20]:

1. A set of random variables that makes up the nodes of an IN. All the variables in the IN have binary states.
2. A set of directed links that connect pairs of nodes.
3. Each link has associated with it a pair of CAST logic parameters that shows the causal strength of the link (usually denoted as h and g values).
4. Each non-root node has an associated CAST logic parameter (denoted as the baseline probability), while a prior probability is associated with each root node.

2.1. CAusal STrength (CAST) Logic

The CAusal STrength (CAST) logic [6, 29] was developed with the aim of overcoming the knowledge acquisition intractability of BNs. Instead of asking an exponential number of parameters to specify the CPT of a node, the logic asks a linear number of parameters and transforms these numbers into CPTs. The logic is an extension of the Noisy-Or approach [1, 8] and requires two parameters for each link in an Influence Net. The first parameter represents the impact of parent node being true on the child node while the second represents the impact of parent node being false on the child node. The negative values show a negative influence of an event on its child event while the positive values show a positive influence of an event on its child event. Fig. 2 shows an IN with four nodes, namely, A, B, C, and X. On each arc in the IN, two causal strengths are specified. For instance, the arc between B and X has values –0.33 and 0.9. The first value, referred to as h, states that if B is true, then this will cause X to be false with probability 0.33, while the second value, referred to as g, states that if B is false, then this will cause X to be true with probability 0.9. Both h and g can take values in the interval (-1, 1). All non-root nodes are assigned a baseline probability, which is similar to the "leak" probability in the Noisy-Or approach. This probability is the user-assigned assessment that the event would occur independently of the modeled influences in a net.

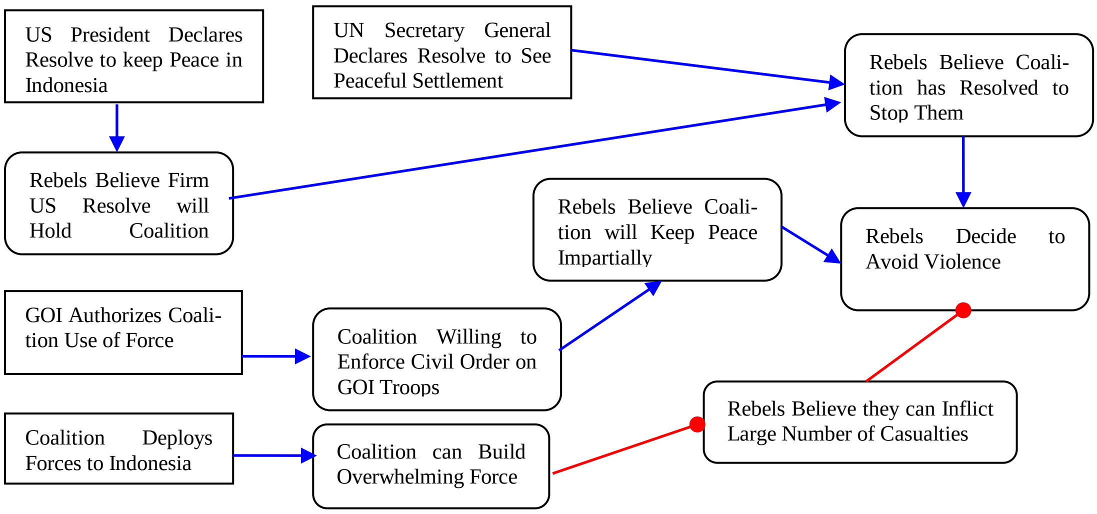

Fig. 1. A Sample Influence Net

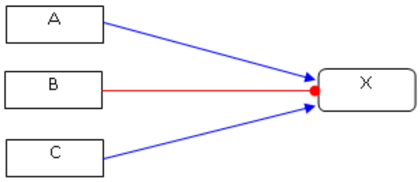

Fig. 2. An Influence Network with CAST Logic Parameters

Once a user specifies all the g and h values and the baseline probabilities, these parameters are converted into CPTs of the corresponding nodes in an IN and are then used during probability propagation and belief updating. The conversion process consists of following major steps:

– Aggregate positive causal strengths
– Aggregate negative causal strengths
– Combine the positive and negative causal strengths, and
– Derive conditional probabilities

In the Influence Net of Fig. 2, the CPT of node X has eight entries. This includes $P(X \mid A, B, C)$, $P(X \mid A, B, \neg C)$, ……., and $P(X \mid \neg A, \neg B, \neg C)$. The four steps listed above are used to calculate each of these eight conditional probabilities. For instance, to calculate the probability $P(X \mid A, B, \neg C)$, the h values on the arcs connecting A and B to X and the g value on the arc connecting C to X are considered. Hence, the set of causal strengths is {0.9, -0.33, -0.66}.

Aggregate the Positive Causal Strengths:
In this step, the set of causal strengths with positive influence are combined. They are aggregated using the equation

$$PI = 1 - \Pi_i (1 - C_i) \qquad \forall C_i > 0$$

where Ci is the corresponding g or h value having positive influence and PI is the combined positive causal strength. For our example

$$PI = 1 - (1 - 0.9) = 0.9$$

Aggregate the Negative Causal Strengths:
In this step, the causal strengths with negative values are combined. The equation used for aggregation is

$$PI = 1 - \Pi_i (1 - |C_i|) \qquad \forall C_i < 0$$

where Ci is the corresponding g or h value having negative influence and NI is the combined negative causal strength. Using the above equation, the aggregate negative influence is found to be:

$$NI = 1 - (1 - 0.33)(1 - 0.66) = 0.77$$

Combine Positive and Negative Causal Strengths:

In this step, aggregated positive and negative influences are combined to obtain an overall net influence. Mathematically,

If PI > NI

AI = (PI – NI) / (1 – NI)

If NI > PI

AI = (NI – PI) / (1 – PI)

Thus, the overall influence for the current example is

AI = (0.9 – 0.77) / (1 – 0.77) = .56

Derive Conditional Probabilities:

In the final step, the overall influence is used to compute the conditional probability value of a child for the given combination of parents.

P(child | jth state of parent states) =

*baseline + (1 – baseline) * AI   when PI > NI*

*baseline – baseline * AI   when PI < NI*

Using the above equation, *P (X | A, B, ¬C)* is obtained as:

*P (X | A, B, ¬C) = 0.5 + 0.5 * 0.56 = .78*

The steps explained above are repeated for the remaining seven entries of node X's CPT. It should be noted that if the experts had sufficient time and knowledge of the influences then they could have directly provided the CPT for each node in the IN instead of providing g and h values. Furthermore, after estimating the CPTs if some entries do not satisfy the experts then those entries can be modified.

3. Sensitivity of Arc Analysis

The paper presents a sensitivity of arc analysis (SAA) based heuristic for arc removal in an IN. Influence Nets are primarily used to model the impact of actionable events on a desired effect. Thus, any sort of sensitivity analysis involves changing one parameter at a time and observing its impact on the desired effect. The presented approach changes the CAST logic parameters to their extreme values and observes the impact of this change on the probability of achieving the desired effect. The arcs having an insignificant impact are removed. The algorithm is presented as Algorithm 1. It is worth mentioning that the extreme values suggested in the table (such as 0.99 and -0.99) are arbitrary and can be replaced by 0.9/-0.9 or 0.999/-0.999 (any absolute number less than 1.0). The step 3 of the CAST logic (combining positive and negative causal strengths) requires the absolute maximum strength to be less than 1.0 to avoid division by zero.

```
Given L, E
  where L = Set of Links
        E = Desired Effect

Compute P(E). Set P_Original (E) = P(E).
Iterate ∀ I where I ∈ L
    Set h_temp = h_I // h value associated with I
    If h_I > 0
            Set h_I to 0.99 // set h to maximum value
            Compute P(E). Set delta_hmax = P(E) – P_Original(E).
            Set h_I to 0 // set h to minimum value
            Compute P(E). Set delta_hmin = P(E) – P_Original((E).
    Else
            Set h_I to 0 // set h to maximum value
            Compute P(E). Set delta_hmax = P(E) – P_Original(E).
            Set h_I to -0.99 // set h to minimum value
            Compute P(E). delta_hmin = P(E) – P_Original((E).
    Set h_difference = |delta_hmax – delta_hmin|.
    Set h_I = h_temp //restore the original value of h_I

    Set g_temp = g_I // g value associated with I
    If g_I > 0
            Set g_I to 0.99 // set g to maximum value
            Compute P(E). delta_gmax = P(E) – P_Original(E).
            Set g_I to 0 // set g to minimum value
            Compute P(E). delta_gmin = P(E) – P_Original(E).
    Else
            Set g_I to 0 // set g to maximum value
            Compute P(E). delta_gmax = P(E) – P_Original(E).
            Set g_I to -0.99 // set g to minimum value
            Compute P(E). delta_gmin = P(E) – P_Original(E).
    Set g_difference = |P(g_max) – P(g_min)|.
    Set g_I = g_temp //restore the original value of h_I
```

Algorithm 1. Sensitivity of Arc Analysis Algorithm

Considering Fig. 2, there are three links in the IN and hence three pairs of g and h values. Suppose the prior probabilities of actionable events A, B, and C are 0.5, 0.6, and 0.4, respectively, which would result in the marginal probability of X being 0.71. The algorithm changes each of the g and h values to their extreme values and observes the impact on the desired effect (in this case, X). For instance, the link between A and X has a positive value of h (0.9) and a negative value of g (-0.66). The algorithm first changes the value of h to 0.99 and computes the new marginal probability of X. It then changes the value of h to 0 and re-computes the marginal probability of X. The difference between the two marginal probability values of X is also stored. The h parameter is now restored to its original value and the same process is repeated for the g value which is first changed to 0 and then to -0.99. The difference in the marginal probability of X due to these changes is recorded. The process is repeated for the other arcs in the Influence Net and the results are presented in Table 1.

Table 1

Results of Applying SAA on IN of Figure 2

Link	A → X	B → X	C → X
delta_hmin	0.56	0.41	0.61
delta_hmax	0.74	0.73	0.73
|delta_hmax–delta_hmin|	0.18	0.33	0.12
delta_gmax	0.50	0.62	0.44
delta_gmin	0.77	0.75	0.78
|delta_gmax –delta_gmin|	0.27	0.13	0.34

The sensitivity of arc analysis gives insight to a system analyst/decision maker about the significance of each link and how it impacts the probability of achieving the desire effect. The paper suggests the use of this analysis as a heuristic to report arcs that have insignificant impact on the desired effect. The system analyst/decision maker can then decide to either remove such arcs or make changes in their CAST logic parameters to capture the desired impact. The subject is further discussed in Section 5.

4. Metrics to Gauge Changes in the Joint Distribution

The sensitivity of arc analysis is primarily a change in one parameter at a time heuristic to identify the significance/insignificance of an arc in an IN. The underlying joint distribution of an IN, however, is likely to contain nonlinearities and thus a decision based on this one parameter at a time heuristic may produce misleading results in certain cases. It is important, therefore, to identify if the arc removal process has caused a significant change in the joint distribution of the modeled variables. The exact way of gauging this change is by computing the joint probability distributions of the original and reduced models (before and after the arc removal process) and comparing them against each other. The joint probability distribution of N variables (X_1,…..X_N) in a Bayesian network (or in an Influence Nets) is computed with the help of the chain rule:

$$P(X_1,\ldots\ldots,X_n) = \Pi\, P(X_i \mid pa(X_i))$$
$$\text{where } i = 1\ldots.N$$

The computation, however, requires generating an exponential number of parameters for each IN and then comparing them. For instance, if there are 100 variables in an Influence Net, the process requires computing 2 x 2100 values for both the original and the reduced INs and then comparing them. The exponential number of steps required to generate and compare the joint distribution makes this process an intractable one. Furthermore, the computed number would be extremely small as the sum of 2100 values, representing the joint probability distribution, would add up to 1. Thus, there is a need for metrics which can assist in (an approximate) comparison of the joint distribution of the two INs. The paper uses the sets of actions finder (SAF) algorithm [20, 22] as a metric to gauge changes in the joint probability distribution of an IN due to the arc removal process. It is important to remember that the primary purpose of building an IN is to model the impact of actionable events on the desired effect. Thus, if there is not a significant difference in the results produced by the SAF algorithm, then it could be safe to assume that the joint distribution modeled by the IN has not changed substantially. The proposed SAF based metric is benchmarked against the KL (Kullback-Leiber) divergence method for estimating the distance between two probability distributions. The following subsections provide a brief description of both metrics.

4.1. Sets of Actions Finder (SAF) Algorithm

```
Given A, E, S, t
      where  A = Set of Actions
             S = Set of Selected Actions
             E = Desired Effect
             t = Threshold

  1. Initialize S = null.
  2. Iterate ∀ I where I ∈ A.
      Set P(I) = 0 //set the prior probability of all actionable
events to zero
  3. Compute P(E). Set P(E_start) = P(E).
  4. Iterate ∀ I where I ∈ A
      Set P(I) = 1 where I ∈ A
      Iterate ∀ J where J ∈ A \ {I}
        Set P(J) = 0
      Compute P(E)
      Set Diff(I) = P (E) – P(E_start)
  5. Select the highest Diff(I) obtained above.
      If Diff(I) > 0 OR if the corresponding P(E) > t
          Remove I from A
          Insert I into S
          Set P(E_start) = P(E)
          Go to Step 5
      Else Stop
```

Algorithm 2. The SAF Algorithm

The sets of actions finder (SAF) algorithm [20, 22] is a heuristic approach to determine the sets of actions that cause the probability of the desired effect to be above (below) a given probability threshold. The algorithm achieves this task in significantly less time than what is required for an exhaustive examination of the actions' search space, which is exponential in terms of the number of actions. The algorithm runs in quadratic time and uses a greedy approach to identify the best (or close-to-best) sets of actions. The algorithm is presented as Algorithm 2 below. It starts with a single action which, when considered individually, causes the highest increase (for a maximization problem) in the probability of the desired effect being true. This is followed by the selection of a second action from the remaining set of actions that together with the first action cause the highest increase in the probability of the desired effect. Other actions are added iteratively in a similar manner. The process stops at a point where (i) the inclusion of an action causes the probability of the objective node to decrease and to fall below the given probability threshold or (ii) there are no more actions to add. Once alternative sets of actions are obtained, they can be grouped together to form more general sets of actions.

4.2. Kullback-Leiber Divergence

The most common method of comparing two probability distributions is the Kullback-Leiber divergence [24]. The quality of the reduced Influence net, caused by removing insignificant arcs, is measured in terms of divergence of its probability distribution from the probability distribution of the original network. Suppose a random variable A with n states. Let Pa represents its initial probability distribution and Pa' represents its posterior probability distribution. The KL-divergence is then computed as:

$$KL(Pa, Pa') = \sum_{a=1}^{n} P(a) log \frac{P(a)}{P(a')}$$

The KL-divergence only takes into account the probabilities of a target variable. For instance, considering Fig. 2, suppose variable X is our target variable and let X' denotes its posterior state. The KL-divergence for variable X will then be given as:

KL(X,X') = P(X) * log(P(X)/P(X')) + P(~X) * log(P(~X) / P(~X'))

5. Design of Experiments and Results

This section discusses the design and results of experiments/simulations conducted to test the validity of the sensitivity of arc analysis based heuristics. The two metrics that are used to gauge the changes in the joint probability distributions of the INs are given below:

(a) SAF Algorithm based Metric:

Absolute difference between the maximum probability to achieve the desired effect in the original and the reduced INs

(b) Kullback-Leiber Divergence:

The divergence between the probability distributions before and after the arc removal process.

The first metric records the absolute difference between the maximum probability of achieving the desired effect, as produced by the SAF algorithm, in the original and reduced INs. The second metric, on the other hand, computes the KL divergence in the probability distributions of the desired effect in the original and the reduced Influence Nets. As suggested in Section 3, SAA suggests the removal of an arc if changing its CAST logic parameters (g and h values) to their extreme values does not change the probability of achieving the desired effect above a certain cutoff point. Three distinct cutoff points (shown in Table 2) are tested in the experiments. It is important to note that the presented approach only serves as a decision support tool to aid a system modeler in making a better sense of the model. Thus, the choice of a specific cut-off point is arbitrarily and rests with the system modeler. The experiments below, however, justify the use of the proposed metric and show that it provides a consistent method for model simplification.

Table 2

Cutoff Points

S. #	SAAPerturb's Cutoff Points
1	|hmax - hmin| < 0.01 and |gmax - gmin| < 0.01
2	|hmax - hmin| < 0.005 and |gmax - gmin| < 0.005
3	|hmax - hmin| < 0.001 and |gmax - gmin| < 0.001

Five unique INs are used in the simulations. Each IN is run 100 times with different and randomly generated sets of CAST logic parameters. Thus, conclusions are drawn based on the results of 500 INs. A brief description (such as number of nodes, number of links, number of actionable events, etc.) of the 5 INs is presented in Table 3 while the INs are shown in Fig. 3. The IN models are built in Pythia, a software tool to model and analyze Influence Nets. The software was developed at the System Architectures Lab, George Mason University. Few of the INs have already been published and analyzed in detail. For instance, M1 was developed by Wagenhals and Wentz [35] to combat the insider threat in an information security scenario. M2 was developed by DeGregario et al. [12] to model certain aspect of the first Gulf war. M5 was developed by Wagenhals et al. [34] to model the political crisis that occurred in East Timor during the final years of the previous decade. A small por-

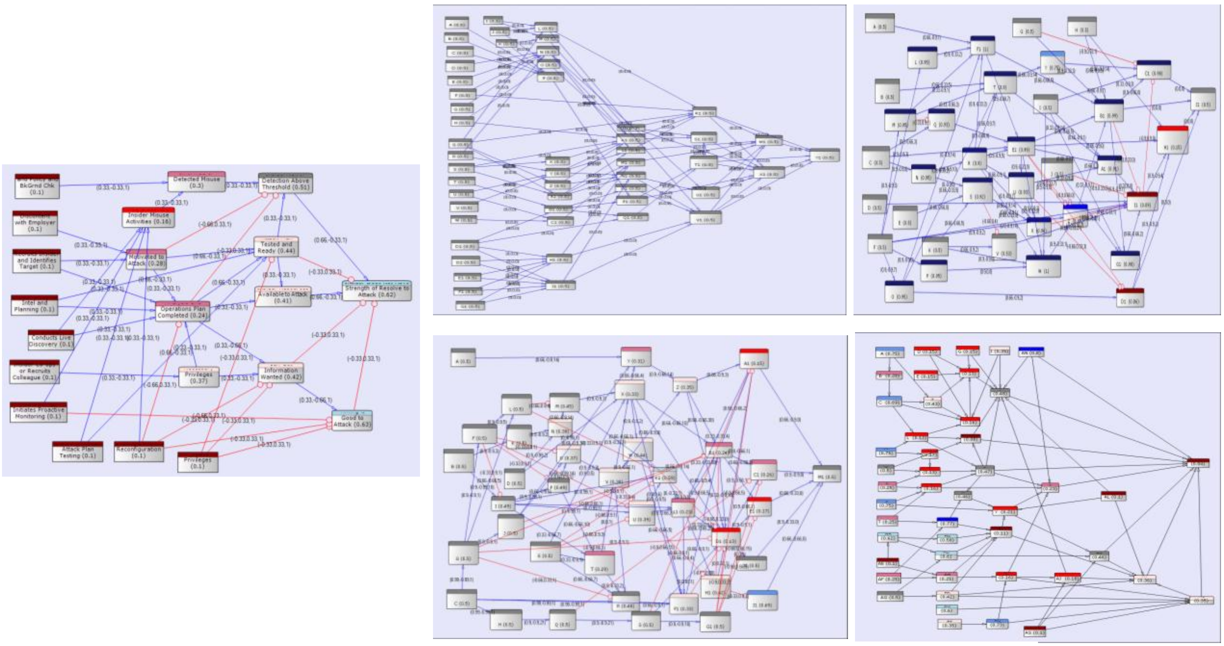

Fig. 3. Influence Nets Models (M1, M2, M3, M4, M5) Used in the Simulation

tion of this net was also shown in Fig. 1. Even though M2 and M4 were not published, they nevertheless were built by domain experts to capture certain anti-terrorism activities.

Table 3

Description of Models Used in the Experiments

Model	# of Nodes	# of Links	Node-to-Link Ratio	# of Actionable Events
M1	21	39	1.86	10
M2	52	96	1.85	23
M3	36	78	2.17	11
M4	39	98	2.51	5
M5	48	89	1.89	18

The steps involved in the simulation are presented in Table 4. During each iteration, an IN is initialized with a set of randomly generated CAST logic parameters. After converting these parameters into conditional probability tables, the SAF algorithm is run to find the best probability of achieving the desired effect (Step 3). Sensitivity of arc analysis is run next and arcs having insignificant impact, based on the given cutoff point, on the probability of the desired effect are removed from the model (Step 4). Step 6 runs the SAF algorithm on the reduced model obtained after the arc removal process. Finally Step 7 computes metrics based on SAF and KL-divergence using the results obtained from the original and the reduced INs.

Table 4

Working of the Simulation

Given ***M***, ***T***

where ***M*** is an Influence Net Model

T is the cutoff point

Iterate 100 times

1. Randomly initialize the CAST logic parameters (i.e., g and h values) of ***M***
2. Convert the CAST logic parameters into corresponding conditional probability tables
3. Run the SAF algorithm on M and store its result
4. Run Sensitivity of arc analysis and remove those arcs from ***M*** having insignificant impact (based on ***T***) on the probability of the desired effect
5. Let ***R*** be the reduced Influence Net Model
6. Run the SAF algorithm on ***R*** and store its results
7. Compute SAF-based metric and KL-divergence using the results obtained in Steps 3 and 6.

The averages of 100 simulations for each of the five models under cutoff points 0.001, 0.005, and 0.01 are given in Table 5, 6, and 7, respectively. For instance, the first row of Table 5 shows that the average decrease (over 100 simulations) in the link-to-node ratio of M1 after the arc removal process is

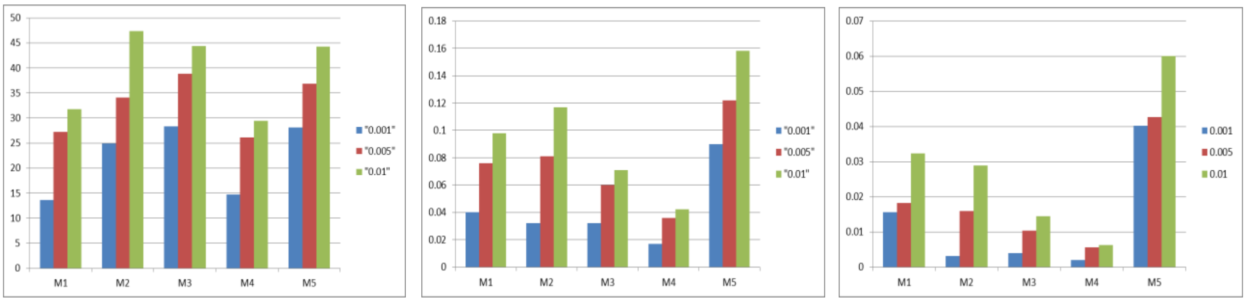

Fig. 4: Graphs of Metrics for Different Cut-off Points. From left to right (a), (b), and (c)

13.59%. The two metrics based on SAF algorithm and KL-divergence have values 0.04 and 0.016, respectively. The simulation was run under the cutoff point of 0.001, that is, an arc was removed if changing the g and h values to their extreme values only cause a change of less than 0.001 in the probability of the desired effect. Similarly, the second last row of Table 6 shows a drop of 26.15% in the link-to-node ratio for M4 under the cutoff point of 0.005. The SAF based metric indicates a little change in the joint probability distribution of the original and the reduced INs. The metric scores 0.036 while the KL-divergence based metric scores 0.006. Other entries of Tables 5-7 can be read in a similar manner.

Table 5

SAA Simulation Results When Cut-Off is less than 0.001

Model	Avg. % Reduction in Node-to-Link Ratio	Avg. Change in Best Probability by SAF	Avg. KL Divergence
M1	13.59	0.040	0.016
M2	24.95	0.032	0.003
M3	28.346	0.032	0.004
M4	14.77	0.017	0.002
M5	28.10	0.09	0.040

Table 6

SAA Simulation Results When Cut-Off is less than 0.005

Model	Avg. % Reduction in Node-to-Link Ratio	Avg. Change in Best Probability by SAF	Avg. KL Divergence
M1	27.26	0.076	0.018
M2	34.04	0.081	0.016
M3	38.81	0.060	0.010
M4	26.15	0.036	0.006
M5	36.88	0.122	0.042

The aggregate statistics over all the INs for each cut-off point are presented in Table 8. For instance, the last row of Table 8 suggests that under the cutoff point 0.01, the average reduction in link-to-node ratio is 39.41% (Column 2), the average absolute change in the best probability produced by SAF is 0.097 (Column 3) and the average KL-divergence is 0.028. It can be seen from the table that as the cutoff point increases, the reduction in link-to-node ratio also increases. The same is true about the change in the best probabilities produced by the SAF algorithm as well as the KL-divergence scores. This simply indicates that as fewer arcs are removed from the original IN, the changes in the joint distribution of the modeled variables are also small. A similar finding can be obtained from the graphs shown in Fig. 4. For instance, Fig. 4(a) plots the decrease in link-to-node ratio of all five models under the three cut-off points. The pattern is consistent for all the models, that is, as the cut-off point increases; the reduction in link-to-node ratio also increases. Similar statements can also be made about the SAF-based metric and the KL-divergence metric shown in Fig. 4(b) and Fig. 4(c), respectively. Based on these results, it is safe to say that the SAF algorithm based metric provides a good heuristic to gauge changes in the joint probability distributions of the modeled variables before and after the arc removal process. It is, however, difficult to suggest a fixed cutoff point for any arbitrary Influence Net as there is no ideal cutoff point and all depends upon the discretion of the group of subject matter experts and knowledge engineers involved in the model construction. The heuristics presented in this paper, nevertheless, provide a good decision support tool to ease this laborious and scrupulous model construction process.

Table 7

SAA Simulation Results When Cut-Off is less than 0.01

Model	Avg. % Reduction in Node-to-Link Ratio	Avg. Change in Best Probability by SAF	Avg. KL Divergence
M1	31.72	0.098	0.032
M2	47.31	0.117	0.029
M3	44.32	0.071	0.014
M4	29.48	0.042	0.006
M5	44.22	0.158	0.060

Table 8

SAAPerturb Aggregated Results of All Five Influence Nets

Cut-Off Points	Reduced Node-to-Link %	Avg. Change in SAF Probabilities	Avg. KL Divergence
0.001	21.95	0.042	0.013
0.005	32.63	0.075	0.019
0.01	39.41	0.097	0.028

6. Conclusions

The model building of Influence Nets is a labor intensive and painstaking process. No formal process exist that can guide subject matter experts, who at times are not quite familiar with the underlying theory of probabilistic graphical models, to make an Influence Net more readable and less dense. The paper presented an approach to reduce the model complexity of Influence Nets using sensitivity of arc analysis. The approach tests how sensitive a desired effect is to the strength of each influence. If the probability of achieving a desired effect is indifferent to changes in the causal strength of each arc then that arc can be suggested for permanent removal from the IN. However, due to nonlinearities typically present in the joint probability distribution of the modeled variables, it is important to know how this arc removal process affects the joint distribution. Considering the intractable amount of time required to generate and to compare the joint probability distributions before and after the arc removal process; a metric, based on the SAF algorithm, is suggested to gauge this change. The metric is benchmarked against the classical KL-divergence method.

The approach presented in this paper works as a decision support tool to subject matter experts/knowledge engineers during the model building phase. It identifies arcs which are present in an IN but have no significant impact on the overall inference capabilities of the IN. The process aids the group of experts in developing a consistent model of the situation which is easy to interpret and is able to capture the intended situation with less complexity. It must be stated, however, that the Influence Net's construction is an incremental process and the reduced network can be further revised (addition/deletion of more arcs and nodes) until the group of experts, involved in its construction, are happy with the final look of it.

Acknowledgements

The authors are grateful to Dr. Lee Wagehals and Dr. Raymond Janssen for proving the five real Influence Nets models that are used in the experiments

References

[1] Cano, J. E., Hern'andez, L. D., and Moral, S., "Importance Sampling Algorithms for the Propagation of Probabilities in Belief Networks," International Journal of Approximate Reasoning," 15, pp. 77-92. 1996.

[2] Castillo, E., Gutierrez, J., and Hadi, A., "Sensitivity analysis in discrete Bayesian networks," IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, vol. 27, 1997, pp. 412-423.

[3] Chan, H., "Sensitivity Analysis of Bayesian Networks and Its Application for Service Engineering," Soft Computing and Pattern Recognition, International Conference of, Los Alamitos, CA, USA: IEEE Computer Society, 2009, pp. 551-556.

[4] Chan, H., Sensitivity Analysis of Probabilistic Graphical Models: Theoretical Results and Their Applications on Bayesian Network Modeling and Inference, VDM Verlag, 2009.

[5] Chan, H. and Darwiche, A., "Sensitivity analysis in Bayesian networks: from single to multiple parameters," Proceedings of the 20th conference on Uncertainty in artificial intelligence, Banff, Canada: AUAI Press, 2004, pp. 67-75.

[6] Chang, K. C., Lehner, P. E., Levis, A. H., Zaidi, S. A. K., and Zhao, X., "On Causal Influence Logic", Technical Report, George Mason University, Center of Excellence for C3I, Dec. 1994.

[7] Cheng, J. and Druzdzel, M. J., "AIS-BN: An Adaptive Importance Sampling Algorithm for Evidential Reasoning in Large Bayesian Networks," Journal of Artificial Intelligence Research, 13, pp. 155-188, 2000.

[8] Cooper, G. F., "The Computation Complexity of Probabilistic Inference Using Bayesian Belief Networks," Artificial Intelligence, 42(2-3), pp. 393-405, 1990.

[9] Coupé, V.M.H., Gaag, L.C.V.D., and Habbema, J.D.F., "Sensitivity analysis: an aid for belief-network quantification," Knowl. Eng. Rev., vol. 15, 2000, pp. 215-232.

[10] Coupé, V.M.H., Peek, N., Ottenkamp, J., and Habbema, J.D.F., "Using sensitivity analysis for efficient quantification of a belief network," Artificial Intelligence in Medicine, vol. 17, Nov. 1999, pp. 223-247.

[11] DeGregorio, E. A., Janssen, R. A., Wagenhals, L. W., and Messier, R. H., "Integrating Effects-Based and Attrition-Based Modeling", In Proceedings of the 9th International

Command and Control Research and Technology Symposium, San Diego, June 2004.
[12] Doucet, A., de Freitas, N., Murphy, K. and Russell, S., “Rao-Blackwellised Particle Filtering for Dynamic Bayesian Networks”, In Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence, 2000.
[13] Fung, R. and Chang, K. C., "Weighting and Integrating Evidence for Stochastic Simulation in Bayesian Networks," In Proceedings of the 5th Conference on Uncertainty in Artificial Intelligence, pp. 209-220, 1990.
[14] Fung, R. and Favero, B. d., "Backward simulation in Bayesian networks," In Proceedings of the 10th Conference on Uncertainty in Artificial Intelligence, San Francisco CA, pp. 227-234, 1994.
[15] Galan, S. F., Figueroa, G. A., Diez, F. J., and Sucar, L. E., “Comparative Evaluation of Temporal Nodes Bayesian Networks and Networks of Probabilistic Events in Discrete Time,” MICAI, pp. 498-507, 2004.
[16] Haider, S., “From Dynamic Influence Nets to Dynamic Bayesian Networks: A Transformation Algorithm, International Jounral of Intelligent Systems, In Press
[17] Haider, S. and Levis, A. H., "Finding Effective Courses of Action using Particle Swarm Optimization," In Proceedings of the World Congress on Computational Intelligence, Hong Kong, pp. 1135-1140, 2008.
[18] Haider, S. and Levis, A. H., "Modeling Time-Varying Uncertain Situations using Dynamic Influence Nets," International Journal of Approximate Reasoning, 49 (2), pp. 488-502, 2008
[19] Haider, S. and Levis, A. H., "Effective Courses-of-Action Determination to Achieve Desired Effects," IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans, vol. 37 (6), Nov. 2007.
[20] Haider, S., Zaidi, A. K. and Levis, A. H., “Identification of Best Set of Actions in Influence Nets,” International Journal of Hybrid Intelligent Systems (IJHIS), 5 (1), pp. 19-29, 2008.
[21] Haider, S., Zaidi, A. K. and Levis, A. H., “On Temporal Analysis of Timed Influence Nets using Point Graphs,” In Proceedings of the 18th International FLAIRS Conference, Clearwater Beach, FL, pp. 629-634, 2005.
[22] Haider, S., Zaidi, A. K. and Levis, A. H., A Heuristic Approach for Best Set of Actions Determination in Influence Nets, In Proceedings of the IEEE International Conference on Information Reuse and Integration, Las Vegas, NV, pp. 600-605, 2004.
[23] Jensen,F., Aldenryd, S., and Jensen, K., “Sensitivity analysis in Bayesian networks,” Symbolic and Quantitative Approaches to Reasoning and Uncertainty, 1995, pp. 243-250.
[24] Kjaerulff, U, “Reduction of computational complexity in Bayesian networks through removal of weak dependencies”, In Proceedings of the 10th Conference on Uncertainty in Artificial Intelligence, pp. 374 – 382, 1994.
[25] Mittal, A. and Kassim, A., “Bayesian Network Technologies: Applications and Graphical Models”, IGI Publishing, 2007.
[26] Pearl, J., “Fusion, propagation, and structuring in belief networks”. Artificial Intelligence, 29, pp. 241–288, 1986.
[27] Pearl, J., “Probabilistic Reasoning in Intelligent Systems: Network of Plausible Inference”, Morgan Kaufmann, 1987.
[28] Pourret, O., Naim, Patrick, and Marcot, B., “Bayesian Networks: A Practical Guide to Applications”, Wiley, 2008.
[29] Rosen, J. A., and Smith, W. L., “Influence Net Modeling with Causal Strengths: An Evolutionary Approach, In Proceedings of the Command and Control Research and Technology Symposium, Monterey CA, 1996.
[30] Takikawa, M., d’Ambrosio, B. and Wright, E., “Real-Time Inference with Large-Scale Temporal Bayes Nets”, In Proceedings of the 18th Conference on Uncertainty in Artificial Intelligence, 2002.
[31] Wagenhals, L. W., Janssen, R. A., and DeGregorio, E. A., “Model Interoperation for Effects Based Planning”, In Proceedings of the 9th International Command and Control Research and Technology Symposium, San Diego, June 2004.
[32] Wagenhals, L. W., Levis, A. H., “Course of Action Analysis in a Cultural Landscape Using Influence Nets”, In Proceedings of the IEEE Symposium on Computational Intelligence for Security and Defense Applications, Honolulu, 2007.
[33] Wagenhals, L. W., Shin, I., and Levis, A. H., “Creating Executable Models of Influence Nets with Coloured Petri Nets”, International Journal On Software Tools for Technology Transfer, Spring-Verlag, 2(2), 168-181, 1998.
[34] Wagenhals, L. W., Reid, T. J., Smillie, R. J., and Levis, A. H., “Course of Action Analysis for Coalition Operations”, In Proceedings of the 6th International Command and Control Research and Technology Symposium, Annapolis, Maryland, 2001.
[35] Wagenhals, L. W. and Wentz L. K., “New Effects-Based Operations Models in War Games”, In Proceedings of the 8th International Command and Control Research and Technology Symposium, Washington DC, 2003.
[36] Wentz, L. K. and Wagenhals, L. W., “Effects Based Operations for Transnational Terrorist Organizations: Assessing Alternative Courses of Action to Mitigate Terrorist Threats”, In Proceedings of the 9th International Command and Control Research and Technology Symposium, San Diego, CA, 2004.
[37] Zhi-fu, S. and Hai-yan, L., “Intelligent Situation Fusion Assessment Using Bayesian Networks,” Information and Computing Science, International Conference on, Los Alamitos, CA, USA: IEEE Computer Society, 2009, pp. 212-215.