Tuning a Bayesian Knowledge Base

Eugene Santos Jr. and **Qi Gu**
Thayer School of Engineering
Dartmouth College, Hanover, NH 03755

Eunice E. Santos
Department of Computer Science
University of Texas at El Paso, El Paso, TX 79968

Abstract

For a knowledge-based system that fails to provide the correct answer, it is important to be able to tune the system while minimizing overall change in the knowledge-base. There are a variety of reasons why the answer is incorrect ranging from incorrect knowledge to information vagueness to incompleteness. Still, in all these situations, it is typically the case that most of the knowledge in the system is likely to be correct as specified by the expert(s) and/or knowledge engineer(s). In this paper, we propose a method to identify the possible changes by understanding the contribution of parameters on the outputs of concern. Our approach is based on Bayesian Knowledge Bases for modeling uncertainties. We start with single parameter changes and then extend to multiple parameters. In order to identify the optimal solution that can minimize the change to the model as specified by the domain experts, we define and evaluate the sensitivity values of the results with respect to the parameters. We discuss the computational complexities of determining the solution and show that the problem of multiple parameters changes can be transformed into Linear Programming problems, and thus, efficiently solvable. Our work can also be applied towards validating the knowledge base such that the updated model can satisfy all test-cases collected from the domain experts.

Introduction

How do you tune your probabilistic knowledge base? Let us say that your favorite probabilistic diagnostic system is not generating the answer that you or your expert friends expected. First, we would need to ask "why" are we getting that probabilistic answer (i.e., "why is this the most probable answer?"). The "why" could come in many forms ranging from determining which events have contributed most to the probabilistic mass to determining which rules or probabilistic relationships (e.g., direct conditional dependencies) were used and/or how frequently. In essence, this provides an explanation of the answer which can help in determining the cause of the discrepancy. Now secondly, what would we need to change in the probabilistic knowledge-base in order to correct the problem? If we believe that most of our knowledge is correct, should we modify as "little" as possible? Of course, what is "little"? Depending on your probabilistic representation, even a small change can have semantic rippling effects throughout the knowledge-base. Related to this is the effect of indirect relationships. Even though we may determine the primary contributors in the "why" above, the correction may involve a change to a single indirect relationship potentially far-removed from the contributor.

In related work, various efforts have been applied to Bayesian Networks (BNs) (Laskey 1995; Chan & Darwiche 2004). However, the quantification of the BN which often requires specification of a huge number of conditional probabilities is always a daunting task. In reality though, the causal mechanism is likely to vary across the population, which happens when the causal link between two variables becomes weak given some evidence. Under these situations, the complete conditional probability specification turns out to be unnecessary, as the redundant information only increases the reasoning complexity. Moreover, the underlying causal relationship may even be cyclic, which is not permitted in a BN.

To avoid these limitations, we represent our knowledge using Bayesian Knowledge-Bases (BKBs) (Santos & Santos 1999; Rosen et al 2004). BKBs are a rule-based probabilistic model that subsumes BNs by specifying dependence at the instantiation level (versus BNs that specify only at the random variable level); by allowing for cycles between variables; and, by loosening the requirement in specifying probability distributions (especially allowing for incompleteness).

To better understand the system and explain why the system is generating a conflict with regards to users' expectations, we look into the contributing factors leading to the conflict and show how changing them can assist in correcting the system. On the other hand, identifying just the primary contributing factors may not be enough. Thus, our second goal is to tune our knowledge system while minimizing the possible changes. This is crucial to the modelers who want to maintain system credibility by protecting certain knowledge from being changed. Previous work (Santos & Dinh 2008) proposed an algorithm for modifying a BKB such that the test-cases entered by the user can be satisfied. However, to validate a query, the given BKB often ends up with a large portion of rules/parameters being changed. In contrast, we start by changing only one rule (probability). Single-rule change, of course, can never be achieved in BNs because once a rule is changed, all the other rules in the same conditional probability table (CPT) column must be changed accordingly, such that the probability sum continues to be

1. We will show that the properties of BKBs can allow for such singular changes in different situations. Furthermore, we present a procedure to identify the candidate changes by using the fact that any joint probability can be expressed as a linear function of a rule, where the coefficients can be efficiently calculated using importance sampling method, instead of reasoning across the entire BKB. Single rule changes are good for maintaining the original probability distribution specified by the experts, but may still not guarantee a solution in every situation. We discuss such situations in terms of rule contribution and provide an intuitive explanation on when and why the single rule change is (in-)sufficient.

In the second half of this paper, we extend our work to allow multiple rules (probabilities) to change, in particular, a set of rules that are analogous to a conditional probability table (CPT) in a BN, but still allows for a cyclic relationship between random variables. Multiple parameter changes can be more intuitive and meaningful since the rules involved in the same conditional statement $X \rightarrow Y$ are always correlated in practice. Another contribution of this work is that we show how to formulate the problem of multiple rule changes while still maintaining the same computation cost needed for single rule changes. In fact, it can be solved efficiently through Linear Programming.

Though the rules with direct contributions are more likely to be critical to the results, the optimal solution may contain some rules whose influence is underestimated due to their low weights, but may cause a "butterfly effect" as the weights change. We incorporate the ideas of sensitivity analyses and evaluate how the changes to a set of rules can help correct the system. Our underlying philosophy is that we want to change the rules that are most significant to the query, such that the amount of change can be minimized or controlled to some extent (as specified by the users if so desired). Sensitivity analysis has been used in several fields, including selective parameter updating in BNs (Wang et al 2002) and large-scale network quantification (Coupe et al 1999). However, sensitivity analysis in BNs (Chan & Darwiche 2004) cannot be applied to tuning a BKB since BKBs do not form CPTs providing more flexibility and avoid the need for *Proportional Scaling* when the variable is multi-valued.

In the next sections, we start with a detailed description of BKBs, followed by our tuning process from single to multiple parameter changes. Another application of our results to knowledge validation will be introduced in the last part.

Bayesian Knowledge-base

BKBs subsume BNs. Instead of specifying the causal structure using conditional probability tables (CPT) as in BNs, it collects the conditional probability rules (CPR) in an "if-then" style. Fig. 1 shows a graph structure of a BKB fragment, in which A is a random variable with possible instantiations $\{a_1, a_2\}$. Each instantiation of a random variable is represented by an I-node, or "instantiation node", e.g. $A = a_1$, and the rule specifying the conditional probability of an I-node is encoded in an S-node, or "support node" with a certain weight. For example, q_7 corresponds to a CPR which can be interpreted as: if $A = a_2$ and $C = c_1$, then $B = b_2$ with a probability 0.5.

One benefit of this approach is that it allows for independence to be specified at the instantiation level instead of the random variable level. Also, it does not require the full table representation of the CPTs and allows for reasoning even with incompleteness. As the BKB shows in Fig.1, the dependency relationship at the variable level implies that variable B depends on both A and C. However, given the evidence of $A = a_1$, B becomes independent of C. This could happen in the real world when the role of a critical variable can dominate some local dependency relationships between variables. In a BN, in order to represent the probability distribution of B dependent on A and C, all CPT entries of $P(B|A,C)$ are required to fill in, which could grow exponentially when the number of the parent nodes is large. BKBs in contrast, only capture the knowledge that is available and does not require a complete distribution specification.

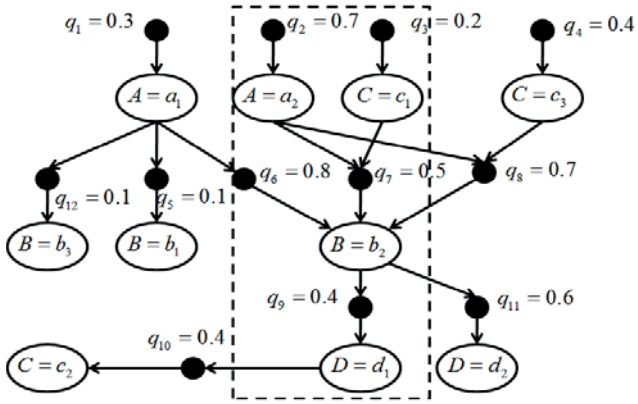

Figure 1: Example of a BKB fragment

The other feature of BKBs is that they also allow cyclic relationships among random variables. Imagine if the direction of some causal mechanism also depends on specific states of the variables. (Santos et al. 2009) gives an example of BKB modeling of a political election, in which the type of "race" may flip the causal direction between the belief of a piece of evidence and the voting action.

The formal definition of the graphical representation of a BKB from (Santos & Dinh 2008) is given below:

Definition 1. 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.

For each S-node q in a correlation graph G, we denote $Pred_G(q)$ as the set of I-nodes pointing to q, i.e. $Pred_G(q) = \{\alpha \in I | \alpha \rightarrow q \in E\}$ and $Desc_G(q)$ as the I-node supported by q, i.e. the α such that $q \rightarrow \alpha \in E$.

Two I-nodes, α_1 and α_2 are said to be *mutually exclusive* if they are the different instantiations of the same

random variables. Similarly, two sets of I-nodes I_1 and I_2 are mutually exclusive if there exists two I-nodes $\alpha_1 \in I_1$ and $\alpha_2 \in I_2$, such that α_1 and α_2 are mutually exclusive. For example, the sets of I-nodes $\{A = a_1, B = b_2\}$ and $\{A = a_2, B = b_2, C = c_1\}$ are mutually exclusive. A *state* is a set of I-nodes such that it contains no more than one instantiation of each random variable.

Definition 2. A set of S-nodes R is said to be *complementary* if for all $q_1, q_2 \in R$, $Desc_G(q_1)$ and $Desc_G(q_2)$ are mutually exclusive, but $Pred_G(q_1)$ and $Pred_G(q_2)$ are not mutually exclusive. Variable v is said to be the *consequent variable* of R, if for any S-node $q \in R$, $Desc_G(q)$ is an instantiation of v.

We denote ρ_v as the set that contains all possible complementary sets of S-nodes w.r.t. variable v, such that for any complementary set $r \in \rho_v$, v is the consequent variable of r. We also introduce Ψ_v to denote the subset of ρ_v that removes all complementary sets in ρ_v that are a subset of some other complementary set, i.e. $\Psi_v = \{r | r \in \rho_v \wedge \nexists r' \in \rho_v, r \subseteq r'\}$ For example in Fig. 1, $\{q_5, q_6\}$ is a complementary set w.r.t. variable B and $\Psi_B = \{\{q_5, q_6, q_{12}\}, \{q_7\}, \{q_8\}\}$. Note that for a certain variable v in Bayesian Network, the size of Ψ_v goes exponentially with the number of its parent variables. However, since BKBs only capture context-specific dependence rules, the size of Ψ_v can be considerably smaller in real-world cases.

Definition 3. A set of S-nodes B is said to be a *causal rule set (CRS)* for the random variable v if B contains all S-nodes pointing to the instantiations of v. As a note, each S-node only belongs to one CRS.

For each complementary set $r' \in \Psi_v$, r' is a subset of B. For example, the CRS for variable B in Fig.1 is $\{q_5, q_6, q_7, q_8, q_{12}\}$.

Definition 4. 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. $\forall q \in S, Pred_G(q)$ 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, $Pred_G(q_1)$ and $Pred_G(q_2)$ are mutually exclusive.
3. for any complementary set of S-nodes $R \subseteq S$, R is normalized: $\sum_{q \in R} w(q) \leq 1$ where $w(q)$ is a weight function that represents the conditional probability of $P(Desc_G(q) | Pred_G(q))$.

The intuition behind these three conditions is that each S-node can only support one I-node; two rules supporting the same I-node cannot be satisfied at the same time; and, to ensure normalization of the probability distribution, every complementary set of S-nodes should be normalized.

Definition 5. The CRS B w.r.t. the variable v is called normalized if for any complementary set $r \in \rho_v$, r is normalized.

Theorem 1. Let B be the CRS of variable v, if for any complementary set $r' \in \Psi_v$, r' is normalized, then B is also normalized.

Theorem 1 can be easily proved from the definition of Ψ_v, since any complementary set in ρ_v is a subset of some complementary set in Ψ_v.

As in BNs, reasoning with BKB is also based on the calculation of joint probabilities over the possible worlds. Here, a *world* is a subgraph of a BKB including at most one I-node of each random variable and the associated S-nodes (a world is referred to as an *inference* in (Santos & Santos 1999). For example in Fig.1, the dotted rectangle circles a world. As proved in (Santos & Santos 1999), the joint probability of a world τ is just the product of the weights of all S-nodes in τ, i.e. $P(\tau) = \prod_{q \in \tau} w(q)$. The idea of world plays an important role in two forms of reasoning with BKBs, belief revision (also called maximum a posteriori or MAP) (Santos & Santos 1999; Pearl 1988) and belief updating. In belief updating, the goal is to calculate the probability of a state of a random variable given some evidence, e.g. $P(Y = y | X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n)$. As shown in (Rosen et al 2004), the joint probability of a state $\theta = \{X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n\}$ is a summation of the probabilities of all possible worlds where θ is true, i.e. $P(\theta) = \sum_{\tau \in I_\theta} P(\tau)$, where I_θ represents the set of worlds containing θ. The worst-case complexity of reasoning with BKBs is NP hard. Some approximation algorithms, e.g. stochastic sampling methods, have been introduced to make the reasoning process more efficient (Rosen et al. 2004). However, given that BKBs only captures the knowledge available, for real-world problems (Santos et al. 2009), exact methods have been shown to be computationally feasible at least of moderate size.

As we discussed earlier, calculating the contributing factors is important to explaining the phenomenon. An algorithm used to compute the *contribution* of node q to the probability of state θ, represented by $c_\theta(q)$, was presented in (Santos et al. 2009), in which the contribution is the sum of the probabilities of the worlds including both θ and q, i.e. $c_\theta(q) = \sum_{\tau \in I_\theta \wedge q \in \tau} P(\tau)$. Here, q is not restricted to just S-nodes. If it is an I-node, then the contribution measures how much influence an event can have on a state.

Tuning a BKB

The main goal of this work is to tune the BKB by understanding the contributing factors leading to the inconsistent answer. More specifically, three problems will be addressed in this section. First, before we actually tune a set of rules (weights of S-nodes), we want to know how they contribute to the original answer. Second, given a BKB, $K = (G, w)$ and some evidence e entered by the user, we need to determine how we can identify necessary changes to a set of S-nodes such that the tuned BKB can enable $P'(\boldsymbol{v} = v_1 | e) > P'(\boldsymbol{v} = v_2 | e)$, where P' is the distribution after we apply changes on the rules (as compared to the original distribution P). Third, we want to select a set of S-nodes that are most sensitive to correcting the answer and focus our efforts on refining their weights. In other words, we want to keep the change as small as possible.

We begin by focusing on the goal of changing just a single S-node. Unlike BNs, which requires the sum of the

parameters in the same CPT column to remain 1.0, BKBs in contrast, allows for incompleteness, which makes it possible to minimize the change to a single S-node.

Let θ_1 and θ_2 be two states representing $\{\boldsymbol{v} = v_1, e\}$ and $\{\boldsymbol{v} = v_2, e\}$, following Bayes rule, the original result: $P(\boldsymbol{v} = v_1|e) < P(\boldsymbol{v} = v_2|e)$ can be transformed into: $P(\boldsymbol{v} = v_1, e) < P(\boldsymbol{v} = v_2, e)$ or $P(\theta_1) < P(\theta_2)$. As defined in the previous section, given an S-node q, we can express $P(\theta_1)$ in terms of $c_{\theta_1}(q)$ as:

$$P(\theta_1) = \sum_{\tau \in I_{\theta_1}} P(\tau) = \sum_{\tau \in I_{\theta_1} \wedge q \in \tau} P(\tau) + \sum_{\upsilon \in I_{\theta_1} \wedge q \notin \upsilon} P(\upsilon) = c_{\theta_1}(q) + a \quad (1)$$

Similarly,

$$P(\theta_2) = \sum_{\tau \in I_{\theta_2}} P(\tau) = c_{\theta_2}(q) + b \quad (2)$$

where a and b are two constants independent of $w(q)$.

An S-node with high contribution value can be considered as a critical parameter to the probability of a state. In Fig.1 currently, $P(\theta_1) = P(A = a_1, D = d_1) = 0.096$ and $P(\theta_2) = P(A = a_2, D = d_1) = 0.106$, which indicates that the most probable state for variable A is a_2 given evidence $D = d_1$. After calculating the contributions of each S-node, we determine that q_9 participates in all possible worlds including θ_2, and thus has the largest contribution to $P(\theta_2)$, i.e. $c_{\theta_2}(q_9) = 0.106$. Therefore, it is reasonable to decrease the probability of $P(\theta_2)$ by lowering the weight of q_9. However, q_9 is also the largest contributor to $P(\theta_1)$, i.e. $c_{\theta_1}(q_9) = 0.096$, which suggests that lowering the weight of q_9 will proportionally decrease $P(\theta_1)$ and $P(\theta_2)$ at the same time. As a result, q_9 is not an appropriate choice to correct the system.

So, which kind of S-node can help to correct the system? Considering that we only adjust the weights of S-nodes without changing the structure of BKB, the corresponding worlds of a state will stay the same. Let $\boldsymbol{q}$ be the S-node that we want to change and $\Delta\boldsymbol{w}$ be the amount of change that we apply to $\boldsymbol{q}$. In order to achieve $P'(\theta_1) > P'(\theta_2)$, from equations (1) and (2), the following condition must hold:

$$c'_{\theta_1}(\boldsymbol{q}) + a > c'_{\theta_2}(\boldsymbol{q}) + b \quad (3)$$

where,

$$c'_{\theta_i}(\boldsymbol{q}) = c_{\theta_i}(\boldsymbol{q})\left(1 + \frac{\Delta\boldsymbol{w}}{w(\boldsymbol{q})}\right) \qquad i = 1,2$$

In addition, to ensure the validity of the tuned system, all the complementary sets of S-nodes that involves $\boldsymbol{q}$ should remain normalized. Let $\boldsymbol{B}$ be the CRS for variable $\boldsymbol{v}$, such that $\boldsymbol{q} \in \boldsymbol{B}$, then $Desc_G(\boldsymbol{q})$ is an instantiation of $\boldsymbol{v}$. From the definition of CRSs, every complementary R, such that $\boldsymbol{q} \in R$, is a subset of $\boldsymbol{B}$. Therefore, as long as B is normalized, the updated BKB is still valid. From Theorem 1, we can simplify the conditions of normalization as follows:

$$\forall R \in \Psi_v(\boldsymbol{q}), \sum_{q \in R} w(q) \le 1$$

where $\Psi_{\mathbf{v}}(\boldsymbol{q})$ only collects the complementary sets in $\Psi_{\mathbf{v}}$ that contain $\boldsymbol{q}$. Combining with inequality (3), the conditions to correct the system for S-node $\boldsymbol{q}$ can be integrated as follows:

$$\begin{cases} \forall R \in \Psi_v(\boldsymbol{q}), \sum_{\boldsymbol{q} \in R} w(\boldsymbol{q}) + \Delta\boldsymbol{w} \le 1 \\ w(\boldsymbol{q}) + \Delta\boldsymbol{w} > 0 \\ (\alpha_{\boldsymbol{q}} - \beta_{\boldsymbol{q}})\Delta\boldsymbol{w} > b - a + c_{\theta_2}(\boldsymbol{q}) - c_{\theta_1}(\boldsymbol{q}) \end{cases} \quad (4)$$

where

$$\alpha_{\boldsymbol{q}} = \frac{c_{\theta_i}(\boldsymbol{q})}{w(\boldsymbol{q})} \qquad \text{and} \qquad \beta_{\boldsymbol{q}} = \frac{c_{\theta_2}(\boldsymbol{q})}{w(\boldsymbol{q})}$$

Since $\Delta\boldsymbol{w}$ is the only unknown variable, the solution space can be easily found by solving for the equality condition. The size of $\Psi_{\mathbf{v}}(\boldsymbol{q})$ can be larger than 1 due to the incompleteness, but still remains small in practice. The time complexity of computing all contribution $c_\theta(\boldsymbol{q})$ in terms of $\boldsymbol{q}$ is worst-case NP-hard, which is the same as belief updating. However, with the aid of importance sampling method introduced in (Rosen et al. 2004), we can approximate all coefficients α and β.

Considering the fact that $P(\theta_1)$ and $P(\theta_2)$ can be written as a function of $w(\boldsymbol{q})$

$$\begin{cases} P(\theta_1) = w(\boldsymbol{q})\alpha_{\boldsymbol{q}} + a \\ P(\theta_2) = w(\boldsymbol{q})\beta_{\boldsymbol{q}} + b \end{cases} \quad (5)$$

Let x_1, x_2 be two different weights assigned to $\boldsymbol{q}$, Then the values of $\alpha_{\boldsymbol{q}}$ can be easily determined as the following:

$$\begin{cases} p^1_{\theta_1} = x_1\alpha_{\boldsymbol{q}} + a \\ p^2_{\theta_1} = x_2\alpha_{\boldsymbol{q}} + a \end{cases}$$

where $p^1_{\theta_1}$ and $p^2_{\theta_1}$ denote the corresponding probabilities of $P(\theta_1)$ respectively. The time complexity of computing the probability $P(\theta_1)$ using sampling method is $O(X^2)$, where X is the number of random variables. The way of computing β is similar to above.

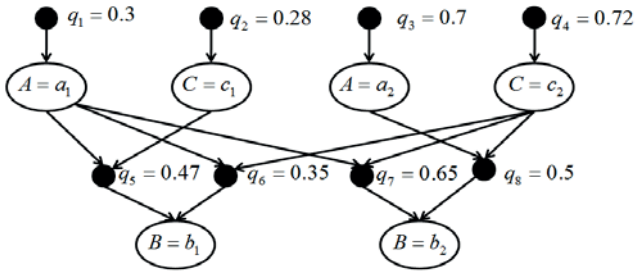

Figure 2: Example of a BKB with no single S-node solution to make $B = b_1$ be the most probable instantiation of variable B given evidence $C = c_2$.

Unfortunately, the single S-node solution does not always exist. We now describe the situations when we are unable to find a solution for inequalities (4). Intuitively, the single S-node solution does not exist if there is no room to increase $P(V = v_1, e)$ due to the normalization constraint (consider BNs). Instead, a second method may be to decrease the probability of $P(\boldsymbol{v} = v_2, e)$, such that $\{\boldsymbol{v} = v_1\}$ becomes the most probable state given evidence e. As the example shown in Fig. 2, we want to select the S-node that contributes to $P(B = b_2, C = c_2)$ and reduce its weight, e.g. q_1, q_3, q_4, q_7 and q_8. Tab.1 shows the probability of $P'(B = b_1, C = c_2)$ and $P'(B = b_2, C = c_2)$

after reducing the weight of these five S-nodes to 0 respectively. Obviously, no matter which S-node we pick to change, the joint probability of $P'(B = b_2, C = c_2)$ is still larger or equal to $P'(B = b_1, C = c_2)$. In other words, no S-node is significant enough to $P(\boldsymbol{v} = v_2, e)$ over $P(\boldsymbol{v} = v_1, e)$.

$w(q_i)$ =0	q_1	q_3	q_4	q_7	q_8
$P'(B = b_2, C = c_2)$	0.252	0.140	0	0.252	0.140
$P'(B = b_1, C = c_2)$	0	0.076	0	0.076	0.076

Table 1: Updated Probabilities of states $\{B = b_1, C = c_2\}$ and $\{B = b_2, C = c_2\}$ with one weight of S-node assigned to 0.

Sensitivity of a Single S-node

Now we address the third problem of selecting the optimal S-node that can minimize the possible change. Though we may determine the primary contributors as we discussed above, an S-node with small direct contribution to the results may actually assist in correcting the system indirectly. For example, changing its weight may impact another high-contributing S-node, and thus flip the final answer. Therefore, the contribution to the original results alone is insufficient for correcting the system in terms of S-node change. To overcome this, we incorporate sensitivity analysis. Sensitivity analysis has been used to evaluate the change in the posterior probability of the target query caused by parameter variation. (Laskey, 1995) proposed to measure sensitivity by using partial derivatives of the probability in terms of the parameter. In our work, in order to achieve $P'(\boldsymbol{v} = v_1|e) - P'(\boldsymbol{v} = v_2|e) > 0$ while minimizing change, we measure the sensitivity with respect to the ratio $r(\boldsymbol{v}|e) = P(\boldsymbol{v} = v_1|e)/P(\boldsymbol{v} = v_2|e)$ as a function of the S-node to be changed. The idea is to make $r(\boldsymbol{v}|e)$ larger than 1.0 with a small $|\Delta \boldsymbol{w}|$.

Definition 6. Given a BKB $K = (G, w)$, let x be the weight of S-node q, the sensitivity of x on $r(\boldsymbol{v}|e)$ is a partial derivative:

$$S(x \mid \boldsymbol{v}, e) = \frac{\partial r(\boldsymbol{v} \mid e)}{\partial x}$$

The ratio $r(\boldsymbol{v}|e)$ can be expressed in terms of x as:

$$r(\boldsymbol{v} \mid e) = \frac{P(\boldsymbol{v} = v_1 \mid e)}{P(\boldsymbol{v} = v_2 \mid e)} = \frac{P(\theta_1)}{P(\theta_2)} = \frac{\alpha x + a}{\beta x + b}$$

which is a fraction of two linear functions on x. Then the partial derivative of $r(\boldsymbol{v}|e)$ on x turns to be:

$$\frac{\partial r(\boldsymbol{v} \mid e)}{\partial x} = \frac{\alpha b - a\beta}{(\beta x + b)^2} \qquad (6)$$

The S-nodes with higher $|S(x|\boldsymbol{v}, e)|$ are more sensitive to the results, and thus more likely to correct the system with small change. Note that the S-nodes with $S(x|\boldsymbol{v}, e) < 0$ implies a negative $\Delta \boldsymbol{w}$, which means that there is no need to check the normalization condition in inequalities (4).

Tune a Variable CRS

Tuning a single S-node is simple to apply. However, as the example shown in Fig. 2 above demonstrates, there may not exist a single S-node solution. In order to overcome this limitation, we instead consider changing the S-nodes in the CRS of a random variable.

There are several advantages to applying single CRS tuning instead of single S-node tuning. First, allowing CRS tuning enlarges the range of its possible changes since the S-nodes in a same complementary set can be changed simultaneously. Therefore it becomes more likely to find a solution to correct the system. Back to the example in Fig.2, if we apply a positive change of 0.3 to q_6 and a negative change of -0.3 to q_7, which still satisfies normalization, then the updated BKB will have the probability of $P'(\boldsymbol{v} = v_1|e) > P'(\boldsymbol{v} = v_2|e)$. Second, in contrast to single s-node tuning, single CRS tuning may be more intuitive and meaningful. For example, if the chance of "Rain = True" gets higher, it is reasonable to believe that the probability of "Rain = False" should become lower. Furthermore, if we want to test how a sensor influences the outcome in a mechanical system, we may want to change both the false-positive and false-negative rates at the same time.

Let $\boldsymbol{B}$ be the CRS set of S-nodes w.r.t. variable $\boldsymbol{v}$, $\Delta \boldsymbol{w}(q)$ be the change applied on S-node $q \in \boldsymbol{B}$ respectively. $\Delta \boldsymbol{w}(q) = 0$ if the weight of q stays the same. Considering the fact that two S-nodes q_1 and q_2 can not coexist in the same world since either $Pred_G(q_1)$ and $Pred_G(q_2)$ are mutually exclusive or $Desc_G(q_1)$ and $Desc_G(q_2)$ are mutually exclusive, equalities (1) and (2) in this case will be replaced with:

$$P(\theta_1) = \sum_{\tau \in I_{\theta_1}} P(\tau) = \sum_{q \in \boldsymbol{B}} c_{\theta_1}(q) + a$$

$$P(\theta_2) = \sum_{\tau \in I_{\theta_2}} P(\tau) = \sum_{q \in \boldsymbol{B}} c_{\theta_2}(q) + b$$

Combining with normalization constraints of $\Psi_{\boldsymbol{v}}$ to find the solution of parameter changes over the CRS that satisfies $P'(\boldsymbol{v} = v_1|e) - P'(\boldsymbol{v} = v_2|e) > 0$, the following conditions must hold:

$$\begin{cases} \forall R \in \Psi_{\boldsymbol{v}}, \sum_{q \in R} w(q) + \sum_{q \in R} \Delta \boldsymbol{w}(q) \le 1 \\ \forall q \in \boldsymbol{B}, w(q) + \Delta \boldsymbol{w}(q) > 0 \\ \sum_{i=1}^{n} (\alpha_i - \beta_i) \Delta \boldsymbol{w}(q_i) > b - a + \sum_{i=1}^{n} [c_{\theta_1}(q_i) - c_{\theta_2}(q_i)] = C_0 \end{cases} \qquad (7)$$

where n is the number of S-nodes in $\boldsymbol{B}$,

$$\alpha_i = \frac{c_{\theta_1}(q_i)}{w(q_i)} \quad \text{and} \quad \beta_i = \frac{c_{\theta_2}(q_i)}{w(q_i)}$$

Solving inequalities (7) can be treated as a Linear Programming (LP) problem using simplex algorithms, i.e.$Min \sum_i^n |\Delta \boldsymbol{w}(q_i)|$, such that the upper conditions hold. The objective function with absolute value can be transformed into a linear objective with additional linear constraints and variables (Shanno & Weil, 1971). The simplex method can be very efficient in practice with a linear to polynomial complexity on average. Note that the time complexity of computing all coefficients α_i, β_i and searching for $\Psi_{\boldsymbol{v}}$ is the same as for the single S-node solution. The only concern is that, when n is big, the size of the coefficient matrix A in linear programming may become too large to solve. Considering that A in the linear programming problem is always a sparse matrix, various efficient revised simplex methods can be used.

Sensitivity of a Variable CRS

In this section, we analyze the sensitivity of a CRS B on the ratio $r(\boldsymbol{v}|e) = P(\boldsymbol{v} = v_1|e)/P(\boldsymbol{v} = v_2|e)$.

Definition 7. Given a BKB $K = (G, w)$, let $\vec{x} = [x_1, x_2 \ldots x_n]$ be the vector that denotes the weights of S-node in CRS $B = \{q_1, q_2 \ldots q_n\}$, the sensitivity of $\vec{x}$ on $r(\boldsymbol{v}|e)$ is a partial derivative:

$$S(\vec{x} \mid \boldsymbol{v}, e) = \frac{\partial r(\boldsymbol{v} \mid e)}{\partial \vec{x}}$$

Similar to the way we derive $r(\boldsymbol{v}|e)$ as a function of the weight of a single S-node, $r(\boldsymbol{v}|e)$ on $\vec{x}$ can be given by:

$$r(\boldsymbol{v} \mid e) = \frac{P(\theta_1)}{P(\theta_2)} = \frac{\alpha_1 x_1 + \alpha_2 x_2 + \ldots + \alpha_n x_n + a}{\beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n + b}$$

Then the sensitivity of $\vec{x}$ on $r(\boldsymbol{v}|e)$ is a partial derivative vector:

$$\frac{\partial r(\boldsymbol{v} \mid e)}{\partial \vec{x}} = [\frac{\partial r(\boldsymbol{v} \mid e)}{\partial x_1}, \frac{\partial r(\boldsymbol{v} \mid e)}{\partial x_2}, \ldots, \frac{\partial r(\boldsymbol{v} \mid e)}{\partial x_n}] \qquad (8)$$

A Euclidean norm of the partial derivative vector can be used to compute the sensitivity value. The CRS with higher sensitivity value will cause larger changes to the results.

As a reminder, every BN can be transformed into a BKB by representing the entries in the CPT with "if-then" rules in the BKB. Therefore, tuning a variable CRS can be also applied to a BN based system. We can prove that if two S-nodes in the same CPT column change together, the tuned system is still a valid BN, i.e. the probability sum of a CPT column remains 1.0. Let x and y denote the amount of change to q_1 and q_2, respectively. Given $\alpha x - \beta y = C_0$, where α, β and C_0 have the same definition as in inequalities (7). If we plot $|x| + |y|$ in terms of $x + y$, we will find out that the optima always shows up at $x + y = 0$ regardless if α is larger than β or not.

Now we provide a concrete example on how to correct the system by tuning a CRS. We again refer to the correlation-graph of the BKB in Fig. 1, and assume the goal is to flip the most probable answer for variable A given evidence $D = d_1$. Currently, $P(A = a_1, D = d_1) = 0.096$ and $P(A = a_2, D = d_1) = 0.106$. We start by evaluating the sensitivity of CRSs in terms of variables A, B, C and D. The sensitivity value of these four CRSs computed from equation (8) is: $S([q_3, q_4, q_{10}]) = 0.121 > S([q_5, q_6, q_7, q_8, q_{12}]) = 0.113 > S([q_1, q_2]) = 0.107 > S([q_9, q_{11}]) = 0$. We use the simplex method to minimize $|\Delta \boldsymbol{w}(q_3)| + |\Delta \boldsymbol{w}(q_4)| + |\Delta \boldsymbol{w}(q_{10})|$ while maintaining constraints in inequalities (7). The minimum is attained when $\Delta \boldsymbol{w}(q_4) = -0.053$ and $\Delta \boldsymbol{w}(q_{10}) = 0.053$. As we can see, this solution only requires a small change in the original BKB while keeping the internal conditional probabilities collected from experts the same.

Conclusion

This paper proposed a method to tune a BKB. Specifically, we described how to explain and fix the conflict between the users' expectation and system answer by making a small change to the parameters. The optimal set of parameters is determined by evaluating their sensitivity, which minimizes the amount of change. We started by tuning a single parameter, and then extended to multiple parameters. Another contribution of this work is that we demonstrated that tuning a system via multiple parameter changes has the same computational complexity as single parameter changes. The problem of multiple parameter changes can be transformed into a Linear Programming (LP) problem and efficiently solved.

Another application of system tuning with multiple parameters in BKB is on automatic knowledge validation, in which the goal is to tune the given BKB with the minimum necessary changes such that the updated BKB will pass all test cases. The problem complexity of solving multiple queries altogether is the same as for a single query, since it is still an LP problem with only an additional inequality needed for each query. One way to measure the sensitivity value w.r.t. multiple queries is to just simply take the average over all the queries.

Acknowledgments. This work was supported in part by grants from AFOSR, DHS, and DTRA.

References

Santos, Jr. E., and Santos, S. 1999. *A framework for building knowledge bases under uncertainty*. Journal of Experimental and Theoretical Artificial Intelligence. 11(2):265-286

Pearl, J. 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann Publishers Inc., San Mateo, CA

Rosen, T., Shimony S. E., and Santos, Jr, E. 2004. *Reasoning with bkbs – algorithms and complexity*, Annals of Mathematics and Artificial Intelligence.40(3-4),403-425

Santos, Jr., E., and Dinh, H. T. 2008. *On automatic knowledge validation for Bayesian knowledge bases*. Data Knowl. Eng. 64(1); 218-241

Laskey K. B. 1995. *Sensitivity analysis for probability assessments in Bayesian networks*. IEEE Trans. on Systems, Man and Cybernetics 25(6):901-909

Shanno D. F. and Weil R. I. 1971, '*Linear Programming with absolute-value functional*, Technical Notes.

Santos, E. E., Santos, Jr. E., Wilkinson, J. T., and Xia, H. 2009. *On a Framework for the Prediction and Explanation of Changing Opinions*. Proceedings of the IEEE International Conference on SMC, 1146-1452

Santos, Jr. E., Wilkinson, J. T., and Santos, E. E.,2009 *Bayesian Knowledge Fusion*. Proceedings of the 22nd International FLAIRS Conference. AAAI Press. 559-564

Chan. H. and Darwiche, A. 2004. *A. Sensitivity Analysis in Bayesian Networks: From Single to Multiple Parameters,* In Proceedings of UAI, 67-75

Wang. H., Rish. I. and Ma. S. 2002. *Using Sensitivity Analysis for Selective Parameter Update in Bayesian Network*. AAAI 02 Spring Symp., TR# SS-02-03, 29–36

Coupe, V. M.; Peek, N.; Ottenkamp, J. and Habbema, J. D. F. 1999. *Using sensitivity analysis for efficient quantification of a belief network.* Artificial Intelligence in Medicine 17:223-247