Constructing Bayesian Networks for Medical Diagnosis from Incomplete and Partially Correct Statistics

Daniel Nikovski*

The Robotics Institute
Carnegie Mellon University
5000 Forbes Ave
Pittsburgh, PA 15213
email: danieln@cs.cmu.edu

Keywords: Bayesian networks, medical diagnosis, combining risk factors, leaky noisy-OR nodes

Abstract

The paper discusses several knowledge engineering techniques for the construction of Bayesian networks for medical diagnostics when the available numerical probabilistic information is incomplete or partially correct. This situation occurs often when epidemiological studies publish only indirect statistics and when significant unmodeled conditional dependence exists in the problem domain. While nothing can replace precise and complete probabilistic information, still a useful diagnostic system can be built with imperfect data by introducing domain-dependent constraints. We propose a solution to the problem of determining the combined influences of several diseases on a single test result from specificity and sensitivity data for individual diseases. We also demonstrate two techniques for dealing with unmodeled conditional dependencies in a diagnostic network. These techniques are discussed in the context of an effort to design a portable device for cardiac diagnosis and monitoring from multi-modal signals.

*The reported research has been performed at Siemens Corporate Research, Inc., 755 College Road East, Princeton, NJ 08540.

1 Introduction

Bayesian networks (BN) have emerged as some of the most successful tools for medical diagnostics and many have been deployed in real medical environments or implemented in off-the-shelf diagnostic software [Spi87, Sho93, WSB97]. Constructing large BNs for such applications is in many ways similar to the knowledge-engineering process employed in the creation of expert systems (ES). However, building a BN is usually more difficult than building an ES, because a BN contains a lot of additional quantitative (numeric) information which is essential to its proper operation. The statistics for filling in this numeric information are not always readily available and often the designer of the system has to make use of indirect statistics instead. For example, it is not always clear how combinations of diseases determine the outcome of a particular diagnostic test – what is usually available are only the positive and negative predictive likelihoods of that test for each individual disease, regardless of what other diseases might be present.

This questions, estimating conditional probability tables for the joint effects of combinations of diseases, requires the use of partial statistics. The problem is of necessity ill-posed – more numbers have to be estimated than are available. In order to find a reasonable solution, some constraints have to be introduced, which usually depend on the problem domain. We propose a solution for the situation outlined above and discuss its advantages and limitations.

Another related problem in the area of constructing BNs for medical diagnostics is the incorporation of partial knowledge about the way the network should behave. In particular, overconfident diagnoses can usually be dealt with by introducing additional nodes, which represent hidden pathological states. The behavior of these nodes can sometimes be explained in terms of Boolean functions such as AND and OR, or compositions thereof. Expressing this knowledge in terms of CPTs for the nodes, however, is not straightforward. We propose one solution based on the use of generic logical nodes. We also propose an alternative technique that does not require the inclusion of new intermediate nodes.

Section 2 reviews briefly Bayesian networks and the process of their design. Section 3 discusses the problem of constructing networks from partial statistics and proposes solutions for the cases of finding the impact of several diseases on a single lab test. Section 4 proposes two methods for dealing with overconfident diagnosis, and section 5 summarizes the proposed solutions and concludes the paper.

2 Constructing Bayesian Networks

A Bayesian network is an efficient factorization of the joint probability distributions (JPD) over a set of variables $X_1, \ldots, X_n$, where each variable X_i has a domain of possible values. The JPD specifies a probability for each possible combination of values for all variables. If the JPD is known, inference can be performed by computing posterior probabilities. For example, if three variable p, q, and r exist in a domain of interest, we can compute the query

$$P(\bar{p}|r) = \frac{P(\bar{p}, r)}{P(r)} = \frac{P(\bar{p}, q, r) + P(\bar{p}, \bar{q}, r)}{P(p, q, r) + P(p, \bar{q}, r) + P(\bar{p}, q, r) + P(\bar{p}, \bar{q}, r)} \tag{1}$$

The summands in the numerator and denominator are all probabilities of atomic events and can be read off from the representation of the JPD. One possible way to represent the JPD is by means of a multidimensional joint probability table (JPT), the key into which is the combination of values for the variables. Reasoning with joint probability tables, however, is exponential in both space and time. The size of the JPT is exponential in the number of variables. The number of atomic probabilities in the numerator and denominator of equation (1) is exponential in the number of variables *not* mentioned in the query, and just performing the summation over them takes exponential time. If probabilistic reasoning is to be used in a practical system, an efficient form of representation and reasoning is necessary. Bayesian networks and the associated reasoning algorithms serve exactly this purpose.

The key to efficient representation of JPTs is in reducing the number of probabilities that are stored. This can be accomplished by introducing simplifying assumptions about the underlying JPD. One such assumption is that of conditional independence. Two variables A and B are said to be conditionally independent with respect to a third variable C if $P(A|C, B) = P(A|C)$, i.e., if we know the value of C, further knowledge of the value of B does not change our belief in A.

One convenient way of representing assumptions of conditional independence is by means of directed acyclic graphs (DAG), augmented by local conditional probability tables (LCPTs). Such graphs are called Bayesian or probabilistic networks. Three small networks are shown in Fig.1, each of them representing a different independence assumption.

After identifying the variables of interest in the problem domain, the next step in building a Bayesian network is to identify which of those variables are conditionally independent. It is possible to create the DAG of a Bayesian net from lists of statements of independence; in practice, however, a much simpler procedure is used. The DAG of a Bayesian net is closely related to the DAG of a set of propositional

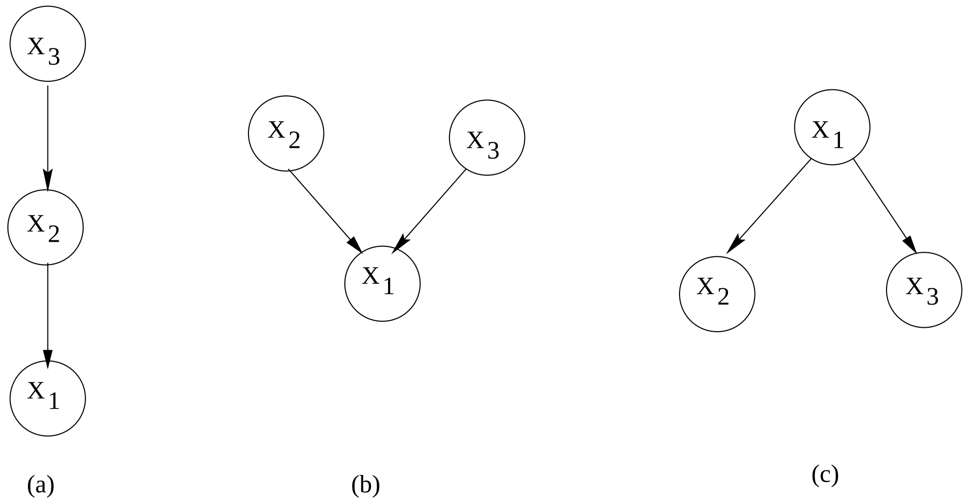

Figure 1: Three graphical representations of conditional independence. In (a), X_1 is independent of X_3 given X_2. In (b), X_2 and X_3 are marginally independent, but conditionally dependent given X_1. In (c), it is just the opposite — X_2 and X_3 are marginally dependent, but conditionally independent given X_1.

rules like those employed in rule-based expert systems.

The next stage in constructing a Bayesian net is to specify the numbers in the LCPTs of the graph. This task is much harder than constructing the DAG. Ideally, the designer of the net has access to statistical estimates of the probabilities in the LCPT. Some medical studies, for example [DF79], contain all the numbers required for the construction of the LCPT. Most often, however, such estimates are not available. The following sections discusses several ways to deal with this problem.

3 Constructing complete CPTs from partial statistics

A major issue in building successful Bayesian net applications is how to determine the entries in the local CPT tables for the nodes in the net. Ideally, those numbers can be estimated statistically or obtained via expert judgment. Most often, however, the numbers that are known are not in the form that is required for the LCPTs.

For example, two diseases D_1 and D_2 might cause the same finding F to appear. Usually, we can obtain the *sensitivity* $P(F|D)$ and *specificity* $P(\bar{F}|\bar{D})$ for the two diseases; for the LCPT, however, we need joint conditional probabilities — $P(F|D_1, D_2)$, etc.

Determining those joint conditional probabilities from the simple conditionals is an ill-posed problem. If n diseases are influencing a finding, the LCPT of the finding node has 2^n entries. At the same time, there are only $2n$ sensitivities and specificities for the n diseases. Inferring 2^n numbers from $2n$ givens is an ill-posed problem. Most commercial belief reasoning packages would require all of these 2^n point probabilities. The only way to obtain them is to introduce additional constraints — bias towards certain values for the entries, more assumptions of independence, or ad hoc techniques. These constraints depend on the particular type of interactions between the random variables at hand.

As noted above, the problem of LCPT compilation arises when a single finding is caused by two or more diseases. The LCPT of the finding node has to contain probabilities that the finding will be present, for all possible combinations of the parent nodes (diseases). For n parents of a finding node, 2^n numbers have to be specified. What we usually have instead is only sensitivities and specificities of the finding. The sensitivity of the finding $P(F|D)$ is a measure of the prevalence of the finding F among those people that have the disease D. The specificity $P(\bar{F}|\bar{D})$ shows how the absence of the finding indicates the absence of the disease. For two findings with equal sensitivity, their respective specificities will determine how unambiguously the presence of the finding indicates the presence of the disease. The finding with higher specificity is a better indicator for the disease than the one with lower specificity. Sensitivity and specificity data can be obtained from prevalence statistics.

There are several reasons why sensitivity and specificity numbers are the easiest to obtain. They are modular and represent objective relationship between a finding and a disease regardless of which other diseases are modeled. The entries in the LCPT, on the other hand, depend on all diseases that the designer of the network has decided to model. In addition, physicians relate easily to this type of information and can provide fairly good estimates. Pearl [Pea88] (pp. 15 and 33) has argued that those numbers are the ones most likely to be available from experiential knowledge and probably are represented explicitly in cognitive structures — for a particular disease, physicians seem to keep in memory a frame describing its characteristics, including the sensitivity and specificity estimates for various symptoms and findings.

For n parent nodes representing diseases, there are only $2n$ sensitivities and specificities available. Determining all 2^n entries in the LCPT is an ill-posed problem. The mapping from diseases to findings, however, has a specific structure, which can be exploited to turn the problem from under-determined to well-posed. It can be assumed that diseases act independently to produce a finding F, and the net effect of all diseases can be combined to produce the cumulative probability that the finding will be present. The finding can be caused by any disease, similarly to a logical OR gate. The relationship, though, is not deterministic — each of the diseases D_i acting alone can cause the finding to appear with probability p_i:

$$p_i = P(F|D_i only) = P(F|\bar{D_1}, \bar{D_2}, \ldots, D_i, \ldots, \bar{D_n})$$

This nondeterministic OR gate is known under the name noisy-OR and has been studied extensively [Sri93, HB94, RD98]. The probabilities p_i are also called *link* probabilities of the noisy-OR gate. Then, any entry in the LCPT can be obtained as ([SMH+91]):

$$P(F|H) = 1 - \prod_{D_i \in \mathcal{H}^+} [1 - p_i], \tag{2}$$

where H is a particular truth assignment to the parents of F, and $\mathcal{H}^+$ is the subset of those nodes in H that are set to *true*. If $\mathcal{H}^+$ is empty, the product term is assumed to be one and the resulting entry in the LCPT is zero. This result is rarely true of any real problem domain — even when no diseases are known to be present, there might be other, unmodeled diseases that can cause the finding. In addition, there is always measurement error and certain number of false positives in the detection of the finding is inevitable.

In order to represent the effect of unmodeled diseases and measurement errors, a *leak term* p_L is usually employed in a noisy-OR model. Conceptually, all unmodeled causes for the finding are represented by a disease node L that is always present. The leak probability p_L can be used directly in equation (2) just like any other link probability; L is always in $\mathcal{H}^+$.

In order to specify the LCPT, only $n+1$ parameters are needed (n link and one leak probabilities). Determining those $n+1$ numbers from the $2n$ sensitivities and specificities is no longer an ill-posed problem; instead, it changed to an over-determined one.

We propose a procedure for estimating link and leak probabilities. The first step is to estimate the link probabilities for each disease. A link probability is a characteristic of a particular disease and does not depend on sensitivities and specificities of other diseases. We can determine it from the sensitivity and specificity for this disease only. The second step is to estimate the leak probability for the whole model, based on all link probabilities determined in the first step. The leak probability is a characteristic of the whole gate — if we decide to add more diseases later, the leak probability should be updated.

In order to estimate the link probability p_i, we can employ again the leaky noisy-OR model, with two possible causes for F: the disease D_i, on one hand, and the combined action of *all* other factors that can lead to the appearance of F, such as other modeled or unmodeled diseases, measurement errors, etc. Conceptually, we can think of all of them as one leak factor L_{all} that is always present, and a corresponding leak probability p_{all}. Then, following equation (2) for two causes:

$$\begin{aligned} P(F|D_i) &= p_i + p_{all} - p_i p_{all} \\ P(F|\bar{D}_i) &= p_{all} \end{aligned}$$

The second equation expresses the fact that $P(F|\bar{D}_i) = 1 - specificity$ for disease D_i represents exactly the combined effect of all other diseases when D_i is not present. From those two equations:

$$p_i = \frac{P(F|D_i) - P(F|\bar{D}_i)}{1 - P(F|\bar{D}_i)}$$

For all cases of practical interest, this equation is well behaved. If a disease D_i causes F indeed, then $P(F|D_i) > P(F|\bar{D}_i)$, which also implies $P(F|\bar{D}_i) < 1$, since $P(F|D_i) \leq 1$. Furthermore, $P(F|D_i) - P(F|\bar{D}_i) \leq 1 - P(F|\bar{D}_i)$.

Once all link probabilities p_i are known, we can estimate the leak probability for the whole node. Let's consider $P(\bar{F}|\bar{D}_i)$, the specificity for a disease i. It expresses the probability that the finding will not be present when D_i is known *not* to be present. We can consider a world in which D_i is permanently absent and decompose the specificity of D_i across all possible such hypothesis:

$$P(\bar{F}|\bar{D}_i) = \frac{\sum_{H|D_i \in \mathcal{H}^-} P(\bar{F}|H)P(H)}{P(\bar{D}_i)}$$

Each of the LCPT entries $P(\bar{F}|H)$ can be expressed in terms of the already found link probabilities p_i and the unknown leak term p_L^i according to the noisy-OR formula:

$$P(\bar{F}|\bar{D}_i) = \frac{\sum_{H|D_i \in \mathcal{H}^-} (1 - p_L^i) \prod_{D_j \in \mathcal{H}^+} (1 - p_j) \prod_{D_j \in \mathcal{H}^+} P(D_j) \prod_{D_j \in \mathcal{H}^-} P(\bar{D}_j)}{P(\bar{D}_i)}$$

Solving for p_L^i, we obtain

$$p_L^i = 1 - \frac{P(\bar{F}|\bar{D}_i)P(\bar{D}_i)}{\sum_{H|D_i \in \mathcal{H}^-} \prod_{D_j \in \mathcal{H}^+} (1 - p_j) \prod_{D_j \in \mathcal{H}^+} P(D_j) \prod_{D_j \in \mathcal{H}^-} P(\bar{D}_j)}$$

We can produce an estimate p_L^i for each i, and combine them in an appropriate manner if they are not consistent, either by simple or weighted averaging. The weights can be, for example, the negated priors $P(\bar{D}_i) = 1 - P(D_i)$ of the diseases D_i:

$$p_L = \sum_i [1 - P(D_i)] p_L^i$$

The justification for this choice of weights is that the leak term p_L^i is needed when disease D_i is *not* present. At any rate, the separate estimates p_L^i should be consistent – if they are not, probably the sensitivity and specificity data are suspect, or a noisy-OR model is not appropriate for this particular set of diseases.

The following numerical example illustrates the operation of the algorithm. Suppose that we have two diseases: tuberculosis (D_1) and bronchitis (D_2). Both of them cause dyspnea (breathlessness) (F). We are given the incidence of these two diseases, as well as the sensitivity and specificity of the finding with respect to each of them:

$$P(D_1) = 0.005 \qquad P(F|D_1) = 0.85 \qquad P(\bar{F}|\bar{D}_1) = 0.983$$

$$P(D_2) = 0.01 \qquad P(F|D_2) = 0.7 \qquad P(\bar{F}|\bar{D}_2) = 0.986$$

The first step is to find the link probabilities of each disease:

$$p_1 = \frac{P(F|D_1) - P(F|\bar{D}_1)}{1 - P(F|\bar{D}_1)} = \frac{0.85 + 0.983 - 1}{0.983} = 0.8474059$$

$$p_2 = \frac{P(F|D_2) - P(F|\bar{D}_2)}{1 - P(F|\bar{D}_2)} = \frac{0.7 + 0.986 - 1}{0.986} = 0.695740365$$

The next step is to estimate the leak probability of the noisy-OR gate from each of the two specificities and the already found link probabilities:

$$p_L^1 = 1 - \frac{P(\bar{F}|\bar{D}_1)}{(1 - p_2)P(D_2) + 1 - P(D_2)} = 0.010112956$$

$$p_L^2 = 1 - \frac{P(\bar{F}|\bar{D}_2)}{(1 - p_1)P(D_1) + 1 - P(D_1)} = 0.009804513$$

We can see that the two estimates of the leak probability are fairly consistent and we can use either of them or their average in a noisy-OR gate.

4 Dealing with conditional dependence and over-confidence

Assumptions of conditional independence are the main factor for simplifying a JPD and representing it efficiently in a Bayesian network. Strict conditional independence, however, rarely exists in distributions coming from the real world. If two variables are nearly conditionally independent, the effect of making the assumption is usually not significant. There are, however, many situations when making an unwarranted assumption of conditional independence severely impairs the performance of the reasoning system.

One such case is observed when two findings are not completely independent given a disease. Over-confidence in the diagnosis is the end effect of failing to represent explicitly their dependency [YHL+88]. The reasons for over-confidence can be explained with the following example:

Two symptoms of the disease cardiac tamponade (D) are dyspnea (breathlessness, A) and rapid breathing (B) (over 20 inhalations per minute). Clearly, the two findings are related — if a patient has cardiac tamponade and has difficulty breathing, it would be very likely that the same patient is breathing rapidly and trying to compensate for the air shortage. If we fail to represent the dependency between the two findings, any time both of them are reported to be present, the posterior odds $O(D|A, B)$ of the disease will increase by the product of the likelihoods $L(A|D)$ and $L(B|D)$ of the two findings:

$$O(D|A, B) = L(A|D)L(B|D)O(D)$$

Multiplying the two likelihoods is clearly not right, because after we know that A is present, establishing B does not bring substantial new information. If we fail to recognize this effect, the posterior odds $O(D|A, B)$ become unreasonably high — the system is said to be overconfident in its diagnosis, and is not likely to be trusted by the users in the long run.

Reliable methods have been established in the medical community for the detection of over-confidence or under-confidence (diffidence) in diagnostic systems [HHB78a, HHB78b]. These methods require a dataset of diagnostic cases with known diagnoses.

There are several strategies for dealing with conditional dependency between findings. The simplest of all is to disregard all findings but the most diagnostic one. For example, the system Iliad uses this strategy in some cases [WSB97]. This is guaranteed to avoid over-confidence, indeed, but at the expense of ignoring useful diagnostic information.

Another strategy is to introduce intermediate nodes that represent pathophysiological states; they are also called clusters of findings [YHL+88]. This solution is shown in Fig. 2(a). In this scheme, the findings determine the truth value of the intermediate node, and only the latter influences the disease node directly. The problem with this approach is that the probabilities for the subnetwork have to be known, which is rarely the case — pathophysiological states are usually unobservable.

4.1 Expressing Boolean clusters

One practically important situation when the probabilities in the subnetwork are easy to find is that of Boolean clusters. For example, the intermediate node I representing the pathophysiological state "myocardial arrhythmias" should be on whenever one of the nodes corresponding to various arrhythmias is on. In other words, the node I is a simple OR function of other nodes, and the designer of the network uses only qualitative information about the behavior of the network, similarly to the principles of constructing Qualitative Bayesian Networks [Wel90, DH93].

Constructing the LCPT for an OR gate is trivial, but the direction of the edges of the graph is incompatible with that of the rest of the network (Fig. 2(b)). In Fig.2(b), the node I is a simple OR of the nodes F_1 and F_2. However, D is also a parent of I, and the LCPT of I includes mixed entries between the findings F_1 and F_2 and the disease node I. It is not at all obvious what those entries should be. Even if it is possible to construct them, the network would include both causal and anti-causal edges. In general, we would like to have consistent arrows in the DAG of the network — either in the causal or in the diagnostic direction, but not both.

One solution is to invert the direction of the Boolean rule and make it compatible with the rest of the network. The inversion is a mechanical procedure that manipulates the local structure and probability tables in a network, and is available as a feature in development environments such as Netica [Nor97a, Nor97b]. The resulting network represents the same JPD as the original one, but this factorization is different. It should be noted that when inverting Boolean rules with extremal entries in the LCPTs (0 and 1), degeneracies might occur. For example, one way to represent the rule $I = F_1 \vee F_2$ is by means of the following LCPT: $P(I|F_1, F_2) = P(I|\bar{F}_1, F_2) = P(I|F_1, \bar{F}_2) = 1$ and $P(I|\bar{F}_1, \bar{F}_2) = 0$. When the topology is inverted (Fig. 2(c)), the parents of F_1 are I and F_2 (note that the inversion introduces a new link from F_2 to F_1). The LCPT of F_1 then will include the entry $P(F_1|\bar{I}, F_2)$. Clearly, the truth assignment $\bar{I}$ and F_2 is impossible under the OR rule, and the entry $P(F_1|\bar{I}, F_2)$ cannot be filled in by a reversal algorithm. (Certainly, any value can be filled in, because it will never be used, but nevertheless reversal algorithms fail in this degenerate case.) A better choice for the LCPT of the original network would be, for example, $P(I|F_1, F_2) = P(I|\bar{F}_1, F_2) = P(I|F_1, \bar{F}_2) = 1 - \epsilon$ and

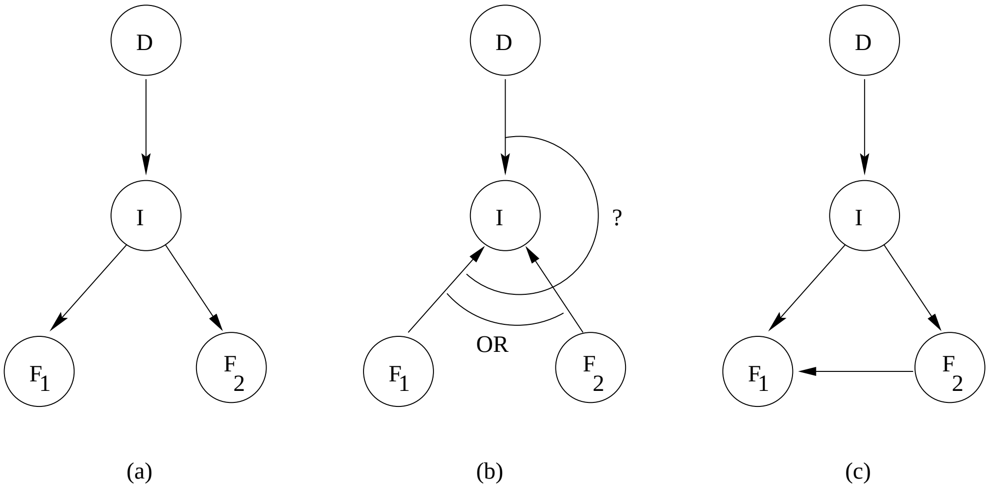

Figure 2: (a)Introduction of an intermediate node. (b)Mixing causal and anti-causal links is difficult. (c)Link reversal might introduce many unwanted links between findings.

$P(I|\bar{F}_1, \bar{F}_2) = \epsilon$, for some small number ϵ. If extremal values are avoided, the reversal is well behaved.

The technique of link reversal can be useful for relatively small networks — not more than five antecedents in a rule. For larger rules, the number of probabilities in the LCPTs becomes prohibitively large. During the reversal of a link, new links are added between the nodes in the Markov blanket of the nodes connected by that link. (The Markov blanket of a node is the set of nodes that are parents, children, or parents of the children of that node [Pea88].) For example, the inversion of an OR rule with 11 antecedents results in a network with 8192 numbers in its LCPTs. The computational effort is hardly worth the benefits of using the OR rule.

A question we can ask is whether all those numbers are really necessary for representing a simple logical rule. It would be very desirable if we could obtain the same or similar computational results with a smaller Bayesian net, for example the one shown in Fig. 2(a).

It turns out that this is possible. Fig. 2(a) shows a gate with edges pointing in the causal (generative) direction and no links between the leaves of the net. In theory, it is not possible to construct a net that implements a precise OR or AND gate in this direction. In practice, it is satisfactory to build a net that exhibits similar behavior within arbitrarily close error bounds.

Since AND and OR are commutative operators, it is reasonable to expect that the LCPTs of all child nodes in 2(a) should be identical and it is enough to specify one of them. An AND gate can be represented with the following LCPT: $P(F|I) = 1$,

$P(F|\bar{I}) = \epsilon$, for some small number ϵ. Analogously, the LCPT of an OR node can be $P(F|I) = 1 - \epsilon$, $P(F|\bar{I}) = 0$. The only difference between the two types of nodes is in how much the LCPT entries differ from the extremal values 0 and 1. The smaller the number ϵ, the closer the behavior of the gate is to the true logical function. It should be at least as small as the smallest prior probability of the consequent node D, and preferably smaller.

A NOT gate can be trivially expressed by the LCPT $P(F|I) = 0$, $P(F|\bar{I}) = 1$. By means of the three gates AND, OR, and NOT, arbitrary propositional rules can be expressed. Furthermore, rules represented by the proposed Bayesian subnetworks are able to propagate virtual (uncertain) evidence, while the original propositional rules cannot.

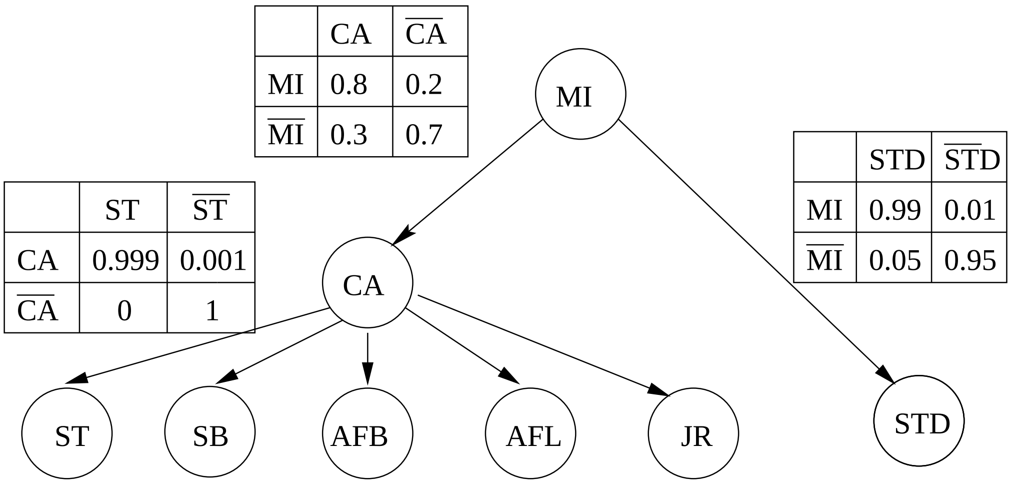

Figure 3: A segment of a network for diagnosing myocardial infarction (MI) based on detection of ST segment depression (STD) and five cardiac arrhythmias: sinus tachycardia (ST), sinus bradycardia (SB), atrial flutter (AFL), atrial fibrillation (AFB), and junctional rhythm (JR). Overconfident diagnosis is avoided by introducing the intermediate node CA, denoting any cardiac arrhythmia. The LCPTs of all arrhythmia nodes are identical to that of ST.

Figure 3 illustrates the use of an intermediate OR cluster to avoid overconfident diagnosis of myocardial infarction (MI) when several cardiac arrhythmias are detected. Arrhythmias are weak and inconclusive indications of MI and have supplemental role in diagnosis. The other finding, depression of the ST segment of the electrocardiogram (STD) is a much stronger finding, as reflected in its LCPT. However, if the intermediate node CA is not present, the effect of several cardiac arrythmias on diagnosis might be much stronger than that of STD. If the node CA is present, no matter how many arrythmias are detected, their combined effect is the same as if only one is detected. This prevents overconfidence and places more emphasis on the more predictive finding, STD.

4.2 Controlling the degree of conditional dependence

Inversion of logical rules helps in situations when beliefs on intermediate nodes are expressed as logical functions. When they are not, it is hard to determine the LCPTs of the intermediate nodes. Instead of introducing intermediate nodes, then, we can try to model explicitly the dependence between the findings. The result of such modification is shown in Fig. 4(a). At least one of the finding nodes has as parents both a disease node and another finding. The corresponding mixed entries in the LCPT of that node are hard to obtain from statistics. What is usually available is only the sensitivities and specificities of the two findings. In addition, the designer of the system usually has a qualitative idea of how dependent the two findings are.

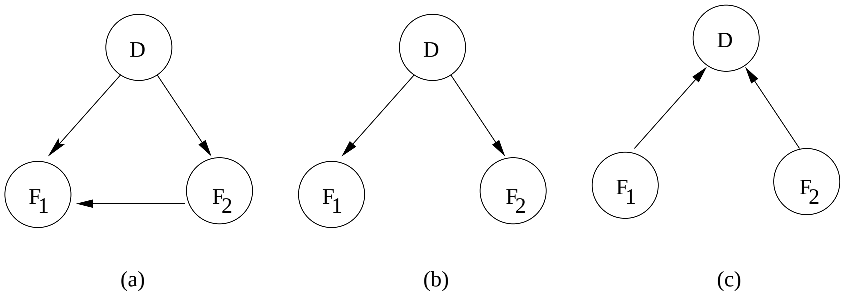

Figure 4: (a)Explicit control over dependency. (b) An overconfident net is created. (c) An inverse net is built by sampling the conditional distribution $P(D|F_1, F_2)$ represented by the previous net. The entries of the LCPT are corrected and the net is re-inverted.

This type of information is in fact sufficient to construct the subnetwork in Fig. 4(a). A special type of double network inversion can be employed to control explicitly the degree of dependency that two findings have. The idea of this technique is to obtain a network of marginally independent findings with edges in the diagnostic direction — opposite to the one in which sensitivity and specificity parameters are available. The LCPT of this network contains the posterior probabilities of the disease given all possible combinations of findings. All cases of over-confidence can be identified and corrected directly in that LCPT. After satisfactory diagnosis is achieved, the corrected network can be inverted again to its original generative direction (edges from diseases to findings). During the second inversion, additional links appear between findings, and the corresponding LCPTs are generated by the link reversal algorithm.

It should be noted that a standard link reversal algorithm can be employed only for the second inversion, but not for the first one. Such link reversal algorithms change the direction of the edges so that the resulting network represents the same *joint* probability distribution [Nor97a]. In doing so, they have to introduce links between findings, because findings have been marginally dependent in the original network.

In contrast to that, we need a link reversal algorithm that produces a network with marginally independent finding nodes and preserves the *conditional* probability distribution $P(D|F_1, F_2, \ldots, F_n)$, but not necessarily the *joint* probability distribution $P(D, F_1, F_2, \ldots, F_n)$. The former can be represented by a network with marginally independent finding nodes (no links between them), but the latter cannot.

Inverting the conditional probability distribution $P(D|F_1, F_2, \ldots, F_n)$ can be done by instantiating the nodes of the original overconfident network in Fig. 4(b) to a particular combination H of truth values for the finding nodes, doing belief updating, reading off the posterior probability $P(D|H)$ of node D, and entering that number in the LCPT of node D in the inverted net in Fig. 4(c). This process has to be repeated for all possible truth assignments of the finding nodes – each instantiation will produce one entry of the LCPT in the new network. Manual sampling of the original network is straightforward, but tedious and error-prone. In order to facilitate the inversion, a utility can be employed that samples an existing overconfident network and creates another one that represents the same conditional probability distribution. The utility we used was built by means of the Netica API library [Nor97b]. After the LCPT of the inverted net in Fig. 4(c) is corrected to avoid over-confidence, it is re-inverted back to the causal direction, which results in the appearance of the additional link in Fig. 4(a).

This procedure of double inversion will be illustrated with the following numerical example, involving the aforementioned problem of diagnosing cardiac tamponade D given two findings: breathlessness (F_1) and rapid breathing (F_2). The data we have available are the incidence of the disease and the sensitivity and specificity of the two findings:

$$P(D) = 0.001$$

$$P(F_1|D) = 0.5 \qquad P(F_1|\bar{D}) = 0.1$$

$$P(F_2|D) = 0.6 \qquad P(F_2|\bar{D}) = 0.2$$

In addition, we have the knowledge that the two findings are not conditionally independent and an approximate idea of their conditional dependency.

The first step is to build a network of the type shown in Fig. 4(b), using the disease prior and sensitivity and specificity information about the two findings. By means of belief updating in this network we can obtain the marginal probabilities of the findings:

$$P(F_1) = 0.1 \qquad P(F_2) = 0.2$$

These probabilities are approximate and are in this case very close to $P(F_2|\bar{D})$ and $P(F_2|\bar{D})$, respectively, because the disease is very rare.

This network can be used for diagnostic reasoning, by instantiating the finding nodes to all possible combinations of truth values:

$$P(D|F_1, F_2) = 0.0148 \qquad P(D|F_1, \bar{F}_2) = 0.0025$$

$$P(D|\bar{F}_1, F_2) = 0.0017 \qquad P(D|\bar{F}_1, \bar{F}_2) = 0.00028$$

$$P(D|F_1) = 0.005 \qquad P(D|F_2) = 0.003$$

We can clearly see that this network is over-confident. When only the first finding is present, the posterior probability of the disease is only five times the prior probability. Similarly, when only the second finding is present, the posterior probability is three times the prior one. However, when both findings are present, the posterior is 14.8 times the prior, even though the second finding does not add substantial new diagnostic information. This unreasonable increase in posterior probability might lead to imprecise diagnosis, and should be eliminated.

To this end, we build another network, this time of the type shown in Fig. 4(c). The priors of that network are the marginal probabilities of the findings, which were found in the first step. The LCPT of the disease node is filled with the posterior probabilities of the disease node from the first network, after appropriate corrections. In this particular case, the following corrections seem reasonable:

$$P(D|F_1, F_2) = 0.006 \qquad P(D|\bar{F}_1, \bar{F}_2) = 0.0004$$

The other two entries, $P(D|\bar{F}_1, F_2)$ and $P(D|F_1, \bar{F}_2)$, remain the same. After this network is built, its links are inverted by a general link reversal algorithm implemented in a package for belief reasoning [Nor97a]. The result is a network of the type shown in Fig. 4(a). An extra edge from F_2 to F_1 has been inserted in order to reflect the fact that the two findings are not conditionally independent. Moreover, the

LCPTs of the finding nodes contain the appropriate numbers, which are otherwise hard to come by:

$$P(F_2|D) = 0.4661 \qquad P(F_2|\bar{D}) = 0.1997$$

$$P(F_1|D, F_2) = 0.2817 \qquad P(F_1|\bar{D}, F_2) = 0.0996$$

$$P(F_1|D, \bar{F}_2) = 0.4098 \qquad P(F_1|\bar{D}, \bar{F}_2) = 0.0998$$

5 Conclusion

Building and using probabilistic networks for medical diagnosis requires reliable methods for knowledge engineering. Some of these methods can be borrowed from the general practice of constructing probabilistic systems; others have to be adapted to the problem domain. This paper described several such techniques. Methods were described for compiling local conditional probability tables from partial statistics for mappings from diseases to findings. The proposed solution to the problem of estimation of the parameters of leaky noisy-OR gates is significantly easier to use than other methods for assessment, and does not require statistics other than the specificity and sensitivity of the findings.

Over-confidence in diagnosis is a major problem in probabilistic diagnostic systems and handling it successfully is a crucial factor to the acceptance of diagnostic systems by medical decision makers. Two methods were proposed to deal with the major reason for over-confidence — unmodeled conditional dependence. These methods can be used to correct the numbers in the local conditional probability tables of a Bayesian network, if unmodeled conditional dependence is suspected. By means of a double link-reversal technique, the amount of conditional dependence between findings can be controlled precisely. Also, if findings are clustered by means of logical rules, the rules can be represented efficiently by the proposed Bayesian subnetworks that serve as AND, OR, and NOT logical gates.

References

[DF79] Diamond, G.A. and Forrester, J.S. (1979). Analysis of probability as an aid in the clinical diagnosis of coronary-artery disease. *The New England Journal of Medicine*, vol. 300. pp.1350–1359.

[DH93] Druzdzel, M.J. and Henrion. M. (1993). Efficient reasoning in qualitative probabilistic networks. In *Proceedings of the Eleventh National Conference on Artificial Intelligence (AAAI-93)*, pp.548–553, Washington, D.C.

[Hec90] Heckerman, D. (1990). A tractable inference algorithm for diagnosing multiple diseases. In *Proceedings of the 5th Conference Uncertainty in Artificial Intelligence 5*, Henrion, M. et al. (eds.), pp. 163–171. North-Holland: Elsevier Science Publishers.

[HB94] Heckerman, D. and Breese, J. (1994). A new look at causal independence. In *Proceedings of the 10th Conference on Uncertainty in Artificial Intelligence*, pp.286–292, San Mateo, CA: Morgan Kaufmann.

[HHB78a] Habbema, J.D.F., Hilden, J., and Bjerregaard, B. (1978a). The measurement of performance in probabilistic diagnosis: I. The problem, descriptive tools, and measures based on classification matrices. *Methods of Information in Medicine*, vol. 17, pp.217–226.

[HHB78b] Habbema, J.D.F., Hilden, J., and Bjerregaard, B. (1978b). The measurement of performance in probabilistic diagnosis: II. Trustworthiness of the exact values of the diagnostic probabilities. *Methods of Information in Medicine*, vol. 17, pp.227–237.

[Nor97a] *Netica application user's guide*. Norsys Software Corp., Vancouver, BC. [http://www.norsys.com]

[Nor97b] *Netica API programmer's library*. Norsys Software Corp., Vancouver, BC. [http://www.norsys.com]

[Pea88] Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems*. San Mateo: Morgan Kaufmann.

[RD98] Rish, I. and Dechter, R. (1998). On the impact of causal independence. 1998 Stanford Spring Symposium on Interactive and Mixed-Initiative Decision-Theoretic Systems, March 1998.

[Sho93] Shortliffe, E.H. (1993). The adolescence of AI in medicine: will the field come of age in the '90s? *Artificial Intelligence in Medicine 5*, pp. 93–106. North Holland: Elsevier Science Publishers.

[Spi87] Spiegelhalter, D.J. (1987). Probabilistic expert systems in medicine. *Statistical Science*, vol.2, no.1, pp. 3–44.

[Sri93] Srinivas, S. (1993). A generalization of the noisy-OR model, in *Proceedings of the 9th Conference on Uncertainty in Artificial Intelligence*, pp.208–218, San Mateo, CA: Morgan Kaufmann.

[SMH+91] Shwe, M.A., Middleton, B., Heckerman, D.E., Henrion, M., Horvitz, E.J., Lehmann, H.P., and Cooper, G.F. (1991). Probabilistic diagnosis using a reformulation of the INTERNIST-1/QMR knowledge base. *Methods of Information in Medicine*, vol.30, pp.241–255.

[WSB97] Warner, H.R., Sorenson, D.K., and Bouhaddou, O. (1997). *Knowledge Engineering in Health Informatics*. Berlin, New York: Springer Verlag.

[Wel90] Wellman, M.P. (1990). Fundamental concepts of qualitative probabilistic networks. *Artificial Intelligence*, vol. 44, no. 3, pp.257–303.

[YHL+88] Yu, H., Haug, P.J., Lincoln, M.J., Turner, C.W., Warner, H.R. (1988). Clustered knowledge representation: Increasing the reliability of computerized expert systems. In *Proceedings of the Twelfth Symposium for Computer Applications in Medical Care*, Washington, DC.

Daniel N. Nikovski received a Dipl.Eng. degree in computer systems from Technical University Sofia, Bulgaria in 1992, a MS degree in computer science from Southern Illinois University in 1995, and a MS degree in robotics from Carnegie Mellon University in 1998. In 1993, he was a postgraduate fellow in neural networks and artificial intelligence at the University of Limburg at Maastricht, Holland. He has worked for International Computers Limited PLC in Sofia, Bulgaria from 1991 to 1993, and Siemens Corporate Research in Princeton, USA, in 1996 and 1997. Daniel Nikovski is currently pursuing a doctoral degree in robotics from Carnegie Mellon University.