Technical Report: Bayesian Belief Nets

Sarah Constantin

May 23, 2013

1 Introduction

We introduce here a formalism for drawing probabilistic inference from medical data using Bayesian belief nets. Medical questions naturally lend themselves to a Bayesian formalism. Certain variables (demographics, lifestyle choices, genes, comorbid diseases, medical treatments or drugs) affect the risk of diseases or other physical conditions, which in turn affect symptoms. The problem of diagnosis is, in essence, a problem of estimating the probability of unknown variables (the diseases) given available information about the patient's symptoms and risk factors. The problem of treatment also naturally falls into this formalism. A medical treatment has a causal relationship with other factors (it has some chance of curing disease, some chance of alleviating symptoms, and some chance of causing side effects). The decision whether or not to perform a given treatment should be informed by predictions of its expected effects, given other available data about the patient, compared with the expected effects of no treatment; probabilistically, this is an inference problem, the comparison of $P(disease|treatment)$ with $P(disease|\overline{treatment})$.

2 Bayesian Networks in Medicine

Bayesian networks were introduced in the 1980's, and their applicability to medicine was obvious. Several studies have shown their efficacy at diagnosis – for example, in 1999, [1] a Bayesian network model of pneumonia diagnosis with only 10 nodes was better correlated with gold standard expert diagnosis than with clinical practice. Clinical decision support systems were defined by Wyatte and Spiegelhalter in 1991 as "active knowledge systems which use two or more items of patient data to generate case-specific advice." CDSSs have been shown to significantly improve physicians' diagnostic performance and improve quality of care. The oldest CDSSs, such as MYCIN, were based on propositional logic rather than Bayesian inference. Bayesian inference systems fell out of favor in the late 1970's due to the difficulty of obtaining conditional independence and the burdensome necessity

of recomputing probability estimates with each new piece of data. The development of Bayesian belief networks in the 1980s, which limit the number of conditional dependencies, made Bayesian methods in expert systems computationally feasible.

CDSSs have been used and studied at a limited number of institutions, such as the HELP system at LDS hospital, Regenstrief Medical Institute, and Vanderbilt University. Unfortunately, the proportion of correct diagnoses in a study of four general diagnostic systems ranged from 0.52-0.71. [3] Many decision support systems are nothing more than simple rule-based reminders for doctors to order tests or take safety precautions; such reminder systems, while straightforward, have been shown to improve health outcomes in the hospitals that use them.

MYCIN, one of the most famous clinical decision support systems, used a "certainty factor" ranging from -1 (false) to +1 (true). The certainty factor of a combined statement is the minimum of the certainty factors of all the statements. This is imprecise. For example, the probability of two independent claims is the product of their probabilities, which is lower than the minimum of their two probabilities; the statement is only precise if one claim implies the other. However, even with this simplification, MYCIN still outperformed clinicians at diagnosis in an experimental setting.

The HELP decision support system used at the LDS Hospital in Salt Lake City diagnosed nosocomial infections with a sensitivity of 90 percent and a false positive rate of 23 percent, while practitioners had a sensitivity of 76 percent and a false positive rate of 19 percent. This was a simple Boolean logic tool. A similar Boolean rule for diagnosing adverse drug events decreased the incidence by 30 percent.[4]

PATHFINDER, developed in 1992,[5] is an expert system for the diagnosis of lymph node diseases; MUNIN, from 1987, is used for interpreting electromyographic findings; and BayPAD, developed in 2003, is used for diagnosing pulmonary embolisms. DxPlain is a decision support system, developed in 1986 at the Massachusetts General Hospital.

The source of the probabilities in a Bayesian network is important. In the DXPlain knowledge base, each disease and finding pair (a finding is a symptom, sign, or piece of epidemiologic, radiologic, laboratory, or endoscopic data) is associated with two numbers, one representing the frequency with which the finding occurs in the disease and the other the degree to which the presence of the finding suggests consideration of the disease. These numbers are extracted from standard medical reference texts, counting the frequency with which phrases related to the finding occur in conjunction with phrases related to the disease.[6] Greater weight is given to words that represent high frequency (such as "always") than low frequency (such as "occasionally"). In other words, the probabilities in the system do not actually represent the experimentally observed probability of having the disease given the finding; they are extrapolated from subjective statements in medical textbooks. This means that the output of models such as DxPlain can be no better than the guidelines in

standard references and do not adapt to changes in the scientific state of the art.

Moreover, the fact that "term-evoking power" (how strongly the finding supports the diagnosis of a disease) is a separate variable from "term frequency" (how often a finding occurs in the presence of the disease) means that Bayes' rule is not enforced by the system, as "term frequency" is a subjective analogue to conditional probability and "term-evoking power" is a subjective analogue to the Bayes factor. If medical textbooks fail to be Bayesian, (for example by neglecting base rates so that a high "term frequency" finding is judged to also have a high "term-evoking power" even when the disease is extremely rare) then decision support systems based on the assumptions in those textbooks will also fail to be Bayesian. Since base rate neglect, among other errors, is a common cognitive bias, and part of the goal of decision-support systems is to prevent errors caused by cognitive biases, a system that fails to build in such corrections may not improve diagnostic accuracy as much as would be desired.

The history of clinical support systems in medicine, and in particular Bayesian networks, is a mixture of success and failure. On the one hand, even very simple automated decision tools and automated diagnoses can improve diagnostic accuracy and health options compared to human judgment alone. On the other hand, adoption of clinical decision support systems is slow, and general-purpose decision support systems still have a high rate of misdiagnoses. Even some Bayesian network-based CDSSs are not truly based on what one might call "Bayesian principles" in that the probability figures they incorporate do not actually represent conditional probabilities and base rates in an internally consistent manner.

The reasons for the slowness of adoption of clinical decision support systems tend to be associated with the fact that they are designed for a hospital setting. Integration with electronic health records, convenient user interfaces, and trust from doctors are all difficult to achieve. In a study of medical residents given access to a CDSS, very few even used the software.[3] Despite their potential to improve the practice of medicine, CDSSs never gained wide appeal. It's possible that CDSSs could be better put to use in a consulting setting, which would avoid the challenges of integrating into a hospital. It's also possible that they could be improved by the use of probabilities that actually represent experimentally observed frequencies and relative risks.

3 Defining the model

A Bayesian belief net is a directed graph, together with an associated set of probability tables. The nodes represent random variables (in our cases, risk factors, symptoms, signs, or diseases), and arrows denote causality (for example, "sleep apnea" might be a node with an arrow pointing to "insomnia.") For each node, we also have a probability table,

capturing the conditional probabilities given its parents. For example, we would have Prob(insomnia = 1, apnea = 1, anxiety = 0) as one of the elements in the node probability table. We make the standard assumption in probabilistic graphical models that the nodes are independent given their parents.

We also assume that nodes are discrete random variables.

For our purposes, the node probability tables are drawn from the research literature. We assign a single odds ratio to every pair of variables with a causal relationship (for example a weighted average of the odds ratios in the most reliable studies) and thus compute the conditional probabilities.

If OR denotes the odds ratio between a variable B and a correlated variable A,

$$OR = \frac{P(B|A)/P(\bar{B}|A)}{P(B|\bar{A})/P(\bar{B}|\bar{A})},$$

then we can derive the conditional probability by an application of Bayes' Rule:

$$P(B|A) = \frac{P(B) - P(B)OR - 1 - \sqrt{(-P(B) + P(B)OR + 1)^2 - 4(1 - OR)P(B)OR}}{2(1 - OR)}$$

The belief net ceases to be a directed acyclic graph if we allow bidirectional edges; that is, if we permit both A to cause B and B to cause A. Probabilistic inference is impossible in graphs with cycles – if A and B mutually cause each other

Propagation refers to computing the probability of a node having a given value (or a set of nodes having a set of given values) given the conditional probabilities and given the values of the known nodes.

In general, if e_x^- refers to the evidence introduced through the arrows between x and its children, and e_X^+ is the evidence introduced through the arrows between x and its parents,

$$p(x|e_x^-, e_x^+) = \frac{p(e_x^-|x, e_x^+)p(x|e_x^+)}{p(e_x^-|e_x^+)} = \frac{p(e_x^-|x)p(x|e_x^+)}{p(e_x^-)}$$

where the second equality is due to the fact that e_x^+ and e_x^- are independent since nodes in a probabilistic graphical model are independent given their parents. Thus we can write the evidence for x as

$$p(x|e_x^-, e_x^+) = \alpha p(x|e_x^+)p(e_x^-|x)$$

where

$$\alpha = 1/p(e_x^-)$$

Or, if we denote $\pi(x) = p(x|e_x^+)$ and $\lambda(x) = p(e_X^-|x)$ this is

$$\alpha\pi(x)\lambda(x)$$

The product of diagnostic or retrospective support, which is λ, and causal or predictive support, which is π. α is the inverse prevalence of the diagnostic data (e.g. the symptoms), $\pi(x)$ is the likelihood of the condition given the predictive data (e.g. the risk factors), and $\lambda(x)$ is the likelihood of diagnostic data given the condition. So, for example, the likelihood that you have condition x given your symptoms and risk factors is higher if your symptoms are rarer (larger α), higher if the disease is more common given your risk factors (larger π), and higher if the disease usually causes the symptoms you have (larger λ).

In principle, one could simply calculate the probabilities of all value assignments on the entire network which are consistent with the known values, and sum them, but the time to compute this grows exponentially in the number of nodes, and so is impractical.

4 Propagation in Trees

Suppose we have a chain network $W \to X \to Y$, where X's only parent is W and Y's only parent is X. Then

$$\lambda(x) = p(e_X^-|x) = \sum_y p(y|x)p(e_Y^-|y)$$

$$= \sum_y \lambda(y)p(y|x)$$

That is, the diagnostic evidence for x is the sum, over all possible values of y, of the diagnostic evidence for y times the probability of y given x. If each node only has two states, this is

$$\lambda(x=1) = p(e_X^-|x=1) = p(y=1|x=1)p(e_Y^-|y=1) + p(y=0|x=1)p(e_Y^-|y=0)$$

In other words, the λs propagate backwards, against the flow of the causality arrows. You can compute the diagnostic evidence λ for x by summing the diagnostic evidence for different possible values of X's child Y, weighted by the conditional probabilities $p(y|x)$.

By contrast, the π messages propagate in the opposite direction:

$$\pi(x) = p(x|e_X^+) = \sum_w p(x|w)p(w|e_W^+) = \sum_w \pi(w)P(x|w)$$

or, if there are only two possible values,

$$\pi(x=1) = p(x=1|e_X^+) = p(x=1|w=1)p(w=1|e_W^+) + p(x=1|w=0)p(w=0|e_W^+)$$

Or, in other words, the causal evidence for a value of X can be derived by summing the causal evidence for all possible values of X's parent W, weighted by the conditional probabilities $p(x|w)$.

Now if parent nodes can have multiple child nodes, the λs must be combined as they're propagated up, and the πs separated as they are propagated down the tree.

If X has child nodes Y_1, Y_2, then

$$\lambda(x) = \lambda_{Y_1}(x)\lambda_{Y_2}(x)$$

where these denote "the diagnostic evidence from Y_1 for x" and "the diagnostic evidence from Y_2 for x," respectively, if the diagnostic evidence for X is partitioned into two sets $e^-_{XY_1}$ and $e^-_{XY_2}$, since children are independent given their parents in the assumption of a graphical model. More generally, if there are more children,

$$\lambda(x) = \prod_{j=1}^{K} \lambda_{Y_j}(x)$$

If we know the $\lambda(y_1) = p(y_1|e^-_{Y_1})$ and $\lambda(y_2) = p(y_2|e^-_{Y_2})$, the diagnostic evidence for the children, then it's possible from this to compute λ for X.

$$\lambda(x) = p(x|e^-_X) = (\sum_{y_1} p(y_1|e^-_{Y_1})p(x|y_1))(\sum_{y_2} p(y_2|e^-_{Y_2})p(x|y_2))$$

$$\lambda(x) = (\sum_{y_1} \lambda(y_1)p(x|y_1))(\sum_{y_2} \lambda(y_2)p(x|y_2))$$

So the diagnostic evidence λ for a node with multiple children is the product of the λ's for all its children, summed over all possible values for the children and weighted by the conditional probabilities on the edges between the parent and its children. This is how we can get λs to "propagate upwards" – if we know them for the children, we know them for the parent.

How do we get π's to "propagate downwards" to the children if we know them for the parent? The predictive support for $X = x$ is $\pi(x) = p(x|e^+_X)$.

$$\pi_{Y_1}(x) = p(x|e^+_X, e^+_{Y_2})$$

Using Bayes' Rule,

$$\pi_{Y_1}(x) = \alpha p(e^-_{Y_2}|x)p(x|e^+_X) = \alpha\lambda_{Y_2}(x)\pi(x)$$

More generally, for the kth child of K children of X,

$$\pi_{Y_k}(x) = \alpha \prod_{j=1-\{k\}}^{K} \lambda_{Y_j}(x)\pi(x)$$

$$= \alpha \frac{Bel(x)}{\lambda_{Y_k}(x)}$$

if $Bel(x) = \prod_j^K \lambda_{Y_j}(x)\pi(x)$, the posterior probability for $X = x$. In other words, π messages propagate downwards by "splitting" into a different value for all children; the original π gets multiplied by the λ messages from all children except the one in question, and by the normalizing constant α.

$$\alpha = \frac{1}{p(e_X^-)} = (\sum_x \pi(x)\lambda(x))^{-1}$$

If we want to find $\pi(y_1) = p(y_1|e_{Y_1}^+)$, the causal evidence for $Y_1 = y_1$, and we know all the $\pi(x)$'s for values of x, we simply note that

$$\pi(y_1) = \sum_x p(y_1|x)p(x|e_X^+)$$

$$= \sum_x p(y_1|x)\pi(x)$$

We can derive the causal evidence for children from the causal evidence for the parents, and the diagnostic evidence for parents from the diagnostic evidence for children.

So, for example, in a tree with parents, children, and grandchildren where only grandchildren are known, we would derive posterior probabilities for the children by computing λ's (observe $p(x|y) = p(y|x)p(x)/p(y)$ and multiply the $p(x|y_i)$ for all children) and π's ($\sum_w p(w)p(x|w)$) and multiplying this by the prior probabilities $p(x_i)$ of the children. Now to compute parents, there is no causal evidence (because they're the roots) and the diagnostic evidence is of the form

$$\prod_i (\sum_{x_i} p(w|x_i)p(x_i|e_{X_i}^-)$$

$$= \prod_i (\sum_{x_i} p(x_i|w)p(w)/p(x_i)p(x_i|y_1, y_2, \ldots y_k))$$

$$= \prod_i \sum_{x_i} p(x_i|w)p(w) \prod_j p(x_i|y_j)/p(y_j)$$

On a tree, belief propagation (also known as the sum-product algorithm) is a two-step process. First, starting at the leaves, each node passes a λ message "up" to its parent. For a leaf L, either it is known or it isn't. If it's unknown we use the prior probabilities $P(l)$. If it's known we observe $P(L = l) = 1$. Then $\lambda(M)$ for M the parent of L is

$$p(e_M^-|m) = p(l_1, l_2, \ldots l_k|m)$$

for all known-valued children k.

$$= \prod_i p(l_i|m). = \prod_i \lambda_{L_i}(m)$$

And so on, at each step the λ "message" is the product of the λ messages at each child node. Finally at the root, there is a combined λ function. Then, propagating downward

5 Propagation in Polytrees

If each node can have multiple parents, but there are no loops, we call this a polytree rather than a tree. In this case, we also have to combine π messages from multiple parent nodes and split λ messages between multiple parent nodes. If W_1 and W_2 are both parents of X, and Y_1, Y_2 are both children of X, the λ messages $\lambda(X) = \lambda_{Y_1}(X)\lambda_{Y_2}(X)$ need to be split (since they propagate backwards):

$$\lambda_X(w_1) = p(e_{W_1X}^-|W_1) = \sum_{w_1} p(e_X^-, e_{W_2X}^+|w_1)$$

$$\lambda_X(w_1) = \beta \sum_x p(e_X^-|x) \sum_{w_2} p(w_2|e_{W_2X}^+) p(x|w_1, w_2)$$

where β is a normalizing constant. and in general

$$\lambda_X(w_i) = \beta \sum_x p(e_X^-|x) \sum_{w_k, k\neq i} p(x|w_1 \ldots w_n) \prod_{k\neq i} \pi_X(w_k)$$

In other words, the diagnostic evidence for $W_1 = w_1$ is the probability of seeing the available evidence downstream of W_1 given $W_1 = w_1$, which can be divided into the probability of seeing the evidence from the subtree with X and the evidence from all other parents $W_2, W_3, \ldots$ of X. This in turn is the sum over all possible values of x, of the diagnostic evidence for x, times the sum over all possibilities for the parents $W_i, i \neq 1$, of the joint probability of x given the parents, times the product of the causal evidence for the parents.

Likewise, we have to alter the predictive support π to get $\pi(x)$ from the predictive supports for the parents $\pi_X(w_k) = p(w_k|e^+_{W_kX})$. The causal support for each parent of X depends on the other parents of X.

$$\begin{aligned}\pi(x) &= p(x|e^+_X) \\ &= \sum_{w_1}\sum_{w_2} p(x|w_1, w_2)\pi_X(w_1)\pi_X(w-2)\end{aligned}$$

so in general, for n parents, this may be written

$$\pi(x) = \sum_{w_1,w_2\ldots w_n} p(x|w_1, w_2, \ldots w_n)\prod_{i=1}^{n}\pi_X(w_i)$$

In other words, to get the causal evidence for children, we have to sum over all possible values for the parents, compute the joint probabilities of x given the parents, and then multiply by the product of the causal evidence for the parents.

6 Noisy-OR and missing joint distributions.

In general, in medical applications, we do not have full joint probability distributions available. For example, if there are several possible diseases which may cause a single symptom, or several risk factors which influence the chance of getting a disease, we may have data from the research literature about the pairwise relationship between parents and children ("Disease X causes symptom Y with probability P ") but not full joint distributions.

To accommodate this limitation, we must make simplifying assumptions on the joint distributions. Heckerman's probabilistic similarity network formulation makes the assumption that diseases are mutually exclusive and exhaustive. [7] We consider that an overly burdensome restriction, especially in the case where parent nodes refer to risk factors and it is clearly possible to have more than one risk factor.

Instead, we make the Noisy-OR assumption, due to Pearl, which assumes causal independence between the parent nodes:

$$p(B|A_1, A_2, \ldots A_n) = 1 - \prod(1 - p(B|A_i))$$

This follows the principle that for B not to occur, independently all the A_i must fail to cause B.

One can generalize this to a formula called Recursive Noisy-OR [8], which incorporates those joint probability distributions which happen to be available. This is, for all subsets C of A,

$$P^R(B|C) = P(B|C)$$

if the complete joint distribution $P(B|C)$ is provided; and

$$1 - \prod_{i=0}^{n-1} \frac{1 - P^R(C - \{A_i\})}{1 - P^R(C - \{A_i, A_{i+1} mod(n)\})}$$

This permits us to incorporate any incomplete joint information. In the case where no additional information is available beyond pairwise $P(B|A_i)$, this is equivalent to the original Noisy-Or algorithm.

Noisy-OR is meant to update the children based on known parents. However, in practice we often have parents which are unknown, and we wish to update the posterior probability of the children based on updated posterior probabilities of the parents. To do this, we simply look at all configurations of values of the parents consistent with the known parents, and compute a weighted sum

$$\sum_i P(B|\bar{A}^i)P(A^i)$$

where each A^i is a configuration, like "$A_1 = 1, A_2 = 0, A_3 = 1$". The joint probability of the configuration is, in general, unknown. For our purposes we treat all the parents as though they're independent. This is a false assumption, but a conservative one in a certain sense; it underestimates the risk of a symptom (or disease) in the case of someone with many correlated causes (or risk factors).

Using this modified version of Noisy-OR assumptions, we can estimate updates in a polytree with some known nodes. First, every child updates all its parents' posterior probabilities, and each updated child updates all its parents in turn, and so on until all the roots of the tree are updated.

$$\begin{aligned} P(B|C_1 \dots C_n) &= P(C_1 \dots C_n|B)P(B)/P(C_1 \dots C_n) \\ &= \frac{\prod_i P(C_i|B)P(B)}{P(C_1 \dots C_n|B)P(B) + P(C_1 \dots C_n|\bar{B})P(\bar{B})} \\ &= \frac{\prod_i P(C_i|B)P(B)}{P(C_1 \dots C_n|B)P(B) + (P(B) - P(C_1 \dots C_n|B))(1 - P(B))} \end{aligned}$$

Then, every parent updates its children, and each child updates its children, and so on, using modified noisy-OR assumptions. [This is equivalent to passing the λ and π messages described in the sum-product algorithm; except that here we don't assume knowledge of joint distributions, and we get posterior probabilities on all elements in the Bayes net at once. In fact, this algorithm can be parallelized.

7 Cluster trees

In general, Bayesian networks need not be trees; they only must be directed acyclic graphs (no cycles with arrows pointing the same way, though loops with arrows pointing different ways are permitted). So we must extend the algorithms used for trees so they can be used in more general cases.

Grouping variables into compound variables so that each cluster is a node in a tree allows us to simplify networks which have loops (i.e. which are multiply connected). We can do belief propagation on the joint distribution of nodes in clusters. "Join trees" are a particular form of clustering in which clusters are allowed to grow until a tree structure is formed.

7.1 Shenoy-Shafer Algorithm

1.) "Moralize" the network by drawing an edge between all parents of a common child (bigamy is legal) and removing the arrows from the edges.

2.) Triangulate the graph.

3.) Identify cliques (complete subgraphs) as compound variables and connect them to form a join tree.

4.) Add evidence nodes as dummy variables to the tree. The λ messages are sent to the evidence nodes.

5.) Generate joint probability distribution matrices for the links between compound nodes.

Then run updates as described before, passing π messages down and λ messages up. For each cluster, we need to pass n_i messages from its neighbors, computing

$$\lambda_X(w_i) = \beta \sum_x p(e_X^-|x) \sum_{w_k, k \neq i} p(x|w_1 \dots w_n) \prod_{k \neq i} \pi_X(w_k)$$

and then multiply all the λs together,

$$\pi_{Y_k}(x) = \alpha \prod_{j \neq k} \lambda_{Y_j}(x) \pi(x)$$

so in total this takes

$$\sum_i O(n_i(n_i - 1)exp(|C_i|) + n_i exp(|C_i)|$$

time, where n_i is the number of neighbors in each cluster and C_i is the number of elements in each cluster.

This is quite a bit better than exponential time in the size of the entire network!

7.2 Alternative algorithm for belief propagation

The algorithm by which parents are updated from children all the way up to a "root" and then children are updated from parents, using noisy-OR assumptions, can be extended to every acyclic graph. Each path following the directions of arrows must terminate to execute this algorithm, but loops are allowed.

For instance, suppose we have the four-node network, A, B, C, D, with $A \to B, A \to C, B \to D, C \to D$ in the shape of a diamond. If A is true,

$$P(B) = P(B|A)$$

$$P(C) = P(C|A)$$

$$\begin{aligned} P(D) = P(B|A)P(C|A)(1-(1-P(D|B))(1-P(D|B)) \\ +P(B|A)(1-P(C|A))(1-(1-P(D|B)(1-P(D|\bar{B})) \\ +(1-P(B|A)P(C|A)(1-(1-P(D|\bar{C}))(1-P(D|B)) \\ +(1-P(B|A))(1-P(C|A))(1-(1-P(D|\bar{C}))(1-P(D|\bar{B})) \end{aligned}$$

If only B is known, first we update A, then update C, then use both B and the new posterior over C to update D.

If only D is known, first we update B and C separately (making the Noisy-Or assumption, not attempting to compute a joint distribution) and then update A, using the Noisy-Or assumption and the assumption of independence of B and C.

Further investigation should determine the properties of this algorithm, but it should permit approximate belief propagation under certain conditions (where the Noisy-Or and independence assumptions are not too far wrong, where "too far wrong" still needs to be made rigorous.) This is a potential alternative algorithm for belief propagation in loopy directed acyclic graphs which requires no knowledge of joint distributions and runs in time exponential only in the number of children per node, not in exponential time in the size of the entire network.

8 Stochastic sampling

In large networks, sometimes exact inference is infeasible and probabilistic methods must be used.

Given known nodes, to determine the most likely propositions at certain hypothesis nodes. the general form of a stochastic sampling algorithm goes as follows:

1.) Calculate the posterior probability distribution of the variable given the assumed states of the other variables in the network.

2.) Draw a random sample from this distribution and set the variable equal to that value.

Forward sampling, for instance, processes each variable in order from the top of the causal network. One problem is that it does not guarantee that known variables will reach their instantiated values. If an instantiated variable fails to draw its instantiated value, the simulation run must be discarded. One way of viewing Gibbs Sampling is that it starts with a configuration constrained to have the known values at the known nodes, and then randomly changes the values of the unknown nodes in causal order, updating probabilities as it goes down the arrows. Once through the network, this instantiation is used as the starting configuration for the next pass.

9 Conclusion

We have here given a brief overview of inference in belief nets and their historical use in medical decision making. We have also explained our methods for inference in a belief net where nodes refer to risk factors, diseases, treatments, or symptoms, and the conditional probabilities are drawn from meta-analyses of the research literature. We have proposed an algorithm for belief propagation in situations where joint probability distributions are unknown, which makes some assumptions on the probabilities (e.g. independent causality) but, we believe, less extreme assumptions than have been made by previous medical Bayes net systems (e.g. Heckerman's assumption of mutually exclusive causes for each effect.) This structure for inference on Bayes nets could produce a knowledge base usable for general diagnosis or risk prediction. It could form a way to systematize the information found in the medical research literature into a probabilistically consistent model of causes and risks that took account of all available knowledge to update estimated probabilities of a variety of conditions and symptoms.

References

[1] Lucas, Peter. "Bayesian networks in medicine: a modern approach to medical decision making." Proceedings of the EUNITE workshop on Intelligent Systems in patient Care, Vienna, Oct. 2001, pp. 73-97

[2] Lucas, Peter. "Bayesian networks in biomedicine and health care." Artificial Intelligence in Medicine 30 (2004).

[3] Berner, Eta et al. *Clinical decision support systems: theory and practice*. New York: Springer, 2007.

[4] Haug, Peter J. et al. "Clinical decision support at Intermountain Healthcare." *Clinical decision support systems: theory and practice*. New York: Springer, 2007.

[5] Heckerman, David, Horvitz, Eric, and Nathwani, Bharat. "Toward normative expert systems: the Pathfinder project." Methods of Information in Medicine, 1992.

[6] Barnett, Octo et al. "DXplain: an evolving diagnostic decision-support system. " JAMA, July 3, 1987.

[7] Heckerman, David. *Probabilistic Similarity Networks*. Cambridge, Massachussetts: The MIT Press, 1991.

[8] Lemmer, John, and Gossink, Don. "Recursive Noisy-Or – a rule for estimating complex probabilistic interactions." IEEE Transactions on Systems, Man, and Cybernetics: Vol. 34, No. 6, 2004.