Belief Propagation in Fuzzy Bayesian Networks

Christopher Fogelberg[1] and **Vasile Palade** and **Phil Assheton**[2]

Abstract. Fuzzy Bayesian networks are a generalisation of classic Bayesian networks to networks with fuzzy variable state. This paper describes our formalisation and outlines how belief propagation can be conducted. Fuzzy techniques can lead to more robust inference. A key advantage of our formalisation is that it can take advantage of all existing network inference and Bayesian network algorithms. Another key advantage is that we have developed several techniques to control the algorithmic complexity. When these techniques can be applied it means that fuzzy Bayesian networks are only a small linear factor less efficient than classic Bayesian networks. With appropriate pre-processing they may be substantially more efficient.

1 Introduction

Modern machine learning research frequently uses *Bayesian networks* (BNs)[6; 7; 15; 16]. However, BN inference is NP-complete due to cycles in the undirected graph[4], and belief propagation is exponential in the *tree-width* of the network. This makes them difficult to use for large problems.

Fuzzy[3] and hybrid fuzzy systems[11; 13] are also frequently used. In a fuzzy system, a variable's *state* is represented by a set of *fuzzy values* (FVs). Because fuzzy systems do not force a model to artificially discretise a continuous underlying state they are often more robust in the face of noise.

To date there has been very little research into BNs with fuzzy variable states. What there is has centred around the use of fuzzy approximations to perform inference and belief propagation in a *hybrid BN*[1; 12]. A hybrid BN is one where the parameters are a mix of continuous and multinomial distributions.

This paper's key contribution is a formal generalisation of classic BNs to *fuzzy Bayesian networks* (FBNs). In a FBN the variables can have *fuzzy states*. The paper also describes tractable belief propagation over FBNs. An important advantage of the presented formalisation is that existing inference algorithms (e.g. MCMC, simulated annealing) can be used without modification. Furthermore, FBNs may be only a small linear constant less efficient than classic BNs of the same size, and appropriate pre-processing may make some problems which were intractable for classic BNs tractable for FBNs.

The paper is structured as follows. Section 2 presents a fuzzy Bayesian network which will be used as an example. Section 3 introduces some notation (subsection 3.1), and presents belief propagation for variables with one parent (subsection 3.3) and for variables with multiple parents (subsection 3.4). Section 4 analyses the algorithmic efficiency of FBNs and how it can be controlled. Section 5 outlines an important bioinformatic domain where FBNs may be especially useful, and section 6 concludes the paper.

2 A Fuzzy Bayesian Network

The structure of the FBN that is used as an example in this paper is shown in figure 1. Call this FBN $G = \langle \eta, \theta \rangle$, where η denotes the structure of G and θ its parameters.

For clarity of presentation, G is a multinomial (discrete) BN. However, the formalisation generalises easily and transparently to continuously-valued FBNs and hybrid FBNs.

The relevant conditional distributions of G are shown in figure 2. D's distribution is not shown and we will later assume a state for D with no loss of generality.

Because we have restricted the differences between BNs and FBNs to belief propagation, the specification of a FBN and a BN are identical.

3 Belief Propagation

Belief propagation in a Bayesian network involves calculating the updated probability distributions of variables in the network, given θ and the observed states of other variables.

3.1 Some Notation

The terminology and notation is as follows. A variable has a state, either a fuzzy state (FS) or a *discrete state* (DS).

A fuzzy state is made up of one or more *components*, and each component is annotated with the variable's degree of membership (μ) in that component. For example, equation 1 is an example of a variable (S) with two components. It has membership 0.7 in the component *hi* and membership 0.3 in the component *mid*. *hi* and *mid* are examples of *values* that the variable can take. When annotated with μ they are referred to as fuzzy values (FV). The set of all possible values (fuzzy values) that a variable can take is the *range* of that variable, e.g. *hi*, *mid*, *lo*.

$$S = [hi_{0.7}, mid_{0.3}] \tag{1}$$

In general, the components of a variable's state are enclosed in square brackets. A discrete state is just a special case of a fuzzy state. Discrete states have just one component with $\mu = 1$, and the square brackets and μ subscript can be omitted in this situation. We assume that $\sum_{c \in C} \mu_c = 1$ for a FS with C components and have not considered other situations.
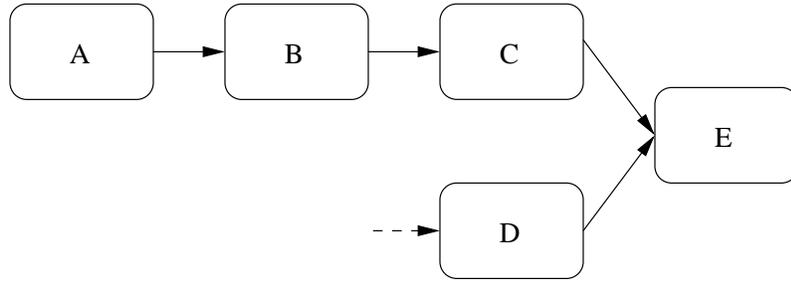
Figure 1. The fuzzy Bayesian network G, used as an example in this paper.

$\rightarrow A$	$A = lo$	$A = mid$	$A = hi$
	0.7	0.1	0.2

(a) θ_A, A's prior distribution

$A \rightarrow B$	$B = lo$	$B = mid$	$B = hi$
$A = lo$	0.6	0.2	0.2
$A = mid$	0.1	0.1	0.8
$A = hi$	0.1	0.2	0.7

(b) θ_B, B's conditional distribution

$B \rightarrow C$	$C = lo$	$C = mid$	$C = hi$
$B = lo$	0.1	0.1	0.8
$B = mid$	0.1	0.8	0.1
$B = hi$	0.7	0.2	0.1

(c) θ_C, C's conditional distribution

$C, D \rightarrow E$		$E = lo$	$E = mid$	$E = hi$
$C = lo$	$D = lo$	0.6	0.2	0.2
$C = lo$	$D = mid$	0.1	0.1	0.8
$C = lo$	$D = hi$	0.1	0.1	0.8
$C = mid$	$D = lo$	0.6	0.2	0.2
$C = mid$	$D = mid$	0.1	0.6	0.3
$C = mid$	$D = hi$	0.1	0.6	0.3
$C = hi$	$D = lo$	0.1	0.2	0.7
$C = hi$	$D = mid$	0.1	0.2	0.7
$C = hi$	$D = hi$	0.8	0.1	0.1

(d) θ_E, E's conditional distribution

Figure 2. θ for G. The conditional distributions of A, B, C and E.

Just as a component can be a value from the range of a variable, e.g. hi, a component can also be a probability distribution (PD). PD are denoted with curly brackets, e.g. $\{hi_{0.3}, mid_{0.2}, lo_{0.5}\}$. Because the value associated with each subscripted probability is implicit in the tuple order the value names can be omitted: $\{0.3, 0.2, 0.5\}$.

An example of a fuzzy state which mixes values and probability distributions is shown in equation 2.

$$T = [\{0.2, 0.1, 0.7\}_{0.2}, \{0.1, 0.8, 0.1\}_{0.6}, mid_{0.2}] \qquad (2)$$

A PD which is annotated (subscripted) with μ is called a

fuzzy probability distribution (FPD). Samples are drawn from a FPD in the same way that they are drawn from a PD. However, a variable with membership μ in a FPD can only have μ proportion of its state determined by that FPD; a sample from a FPD will have the same μ as the FPD does. For example, a sample from $\{0.2, 0.1, 0.7\}_{0.2}$ will be one of $lo_{0.2}$, $mid_{0.2}$ or $hi_{0.2}$, and each of these components will be drawn with probability 0.2, 0.1 and 0.7 respectively.

This means that the state of a sample from the uncertain variable T (equation 2 will be some member from the set $[lo_{[0..0.8]}, mid_{[0.2..1]}, hi_{[0..0.8]}]$ and the distribution over members in this set is determined by the two FPD and one FV which make up the fuzzy state of T.

3.2 Assumptions

The full and general analysis of FBNs would also consider unrestricted interactions amongst components in a fuzzy state, allowing $\sum \mu \neq 1$ and so forth. In this article we make a number of linearising assumptions which make FBN belief propagation cheap, relative to the cost of full general propagation. They also greatly aid the clarity of the presentation in the space available. Furthermore, these assumptions are reasonable and do not restrict the general utility of FBNs. However it is important to make them explicit. The assumptions (and consequently the nature of full general propagation) will be briefly summarised in this section; a more general discussion is forthcoming.

3.2.1 Assumption: Total Membership

As noted above and in subsection 3.1, we assume that $\sum \mu = 1$. This is our first linearising assumption, and it can be conceptualised as follows. A variable's degree of membership in each of its $|C|$ components forms a $|C|$-dimensional *fuzzy state space*. If the variable has no uncertainty (no component is an FPD) then $|C|$ will be the same as the variable's range. Even if a FS has 0 membership in some of its range those values are still part of the state's FSS. By assuming that $\sum_{c \in C} \mu_c = 1$, we restrict our attention to a smaller $|C| - 1$ dimensional subspace. This subspace constrains the degrees of membership in each component in a cyclically conditional way on the degrees of membership in the other components. This assumption simplifies the combination of components of fuzzy states in subsection 3.4.

For example, if a FS has membership 0.5 in the value hi its membership in the values lo and mid in the FSS are constrained to be in the range $[0, 0.5]$. Furthermore, its membership in lo and mid mutually constrain (in this case define, as

there are no other values in the range) each other. Figure 3 illustrates the impact of the first assumption on the FSS for a fuzzy state with two components.
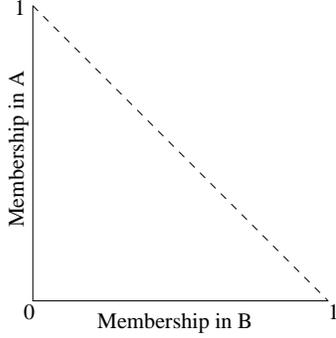

Figure 3. Imagine a fuzzy state with two components, A and B. Such a state would have a two-dimensional FSS, as in this figure. Without restriction, the state's degree of membership in each component could be specified by any point in the FSS. However, we assume that $\sum_{c \in C} \mu_c = 1$. Therefore its membership in components A and B must be specified by some point on the dashed line.

3.2.2 Assumption: Component Independence

We also make two further assumptions about FBN during belief propagation. The first is that components are independent. This means that when a variable has only one parent then its state will have one component for each component its parent has, and these components will have the same μ as the corresponding parent's component. For example, the children which have S (equation 1) as their only parent will have two components in their FS, one with $\mu = 0.7$ and one with $\mu = 0.3$.

When a variable has more than one parent then the components of the parents will be mixed and combined before propagation. This is described in subsection 3.4. Because we assume independence, the child's fuzzy state will have one component for each component in the mixed and combined parent set, and each of the child's component will have the same μ as the corresponding component in the parent set.

It will also be clear that we assume independence when we describe how the parents' components are mixed and combined in subsection 3.4.

3.2.3 Assumption: FPD Samples

The third assumption implicit in this model is the assumption that a sample from an FPD with $\mu = x$ will be a single fuzzy component with $\mu = x$. Although natural and intuitive we do not believe that this is automatically entailed, thus we explicitly assume it.

Consider an uncertain variable with a range of r in a standard (discrete) Bayesian network. Its state will be a probability distribution which specified a single point in the r dimensional *probability space* (p-space).

Now consider this variable in a FBN. Any uncertainty in its state will be represented by an FPD component with some μ. As described in subsection 3.1, a FPD is just PD with a fixed (0 dimensional) μ associated with it.

This definition of a FPD could be generalised so that μ could vary independently but was fixed for each of the r possible samples that could be drawn from the FPD. Call this a *slightly general FPD* (SGFPD) and call the FPD defined in subsection 3.1 a *standard FPD*. An SGFPD would specify a single point in an $r + r$ dimensional space, where r of the dimensions are the probability of each value and the other r dimensions specify the μ of a sample of each value. Just as the first r dimensions specify a p-space, the second r dimensions specify a *μ-space*. An example of such a space is given diagrammatically in figure 4, and it is used to contrast a SGFPD with a standard FPD.

An example of an SGFPD might be "there is a 0.2 probability of drawing a sample of *hi*, and any sample of *hi* will have $\mu = 0.3$, and there is a 0.3 probability of drawing a sample of *mid*, and any sample of *mid* will have $\mu = 0.5$, and..." and so forth. The assumption that $\sum_{c \in C} \mu_c = 1$ for a state could be relaxed if SGFPD were used.

After considering figure 4 it will be clear that SGFPD could be further generalised so that the μ of any sample also varied probabilistically, conditional on the value (*hi*, *mid*, etc.) of the sample. Such a *general FPD* (GFPD) would be an $r + r$ dimensional probability distribution over the joint μ and range of the variable. We believe that this represents the most general kind of inference and belief propagation in a FBN. Such inference is intractable and we do not consider it in this paper.

In summary, the assumptions which we have made substantially reduce the dimensionality of belief propagation and are necessary for it to be tractable. However, more general FBNs with GFPD do not have these restrictions; their utility will be considered in a forthcoming publication.

3.3 Single-Parent Belief Propagation

Assume that observations indicate $A = [mid_{0.2}, hi_{0.8}]$ in G. With this information we can calculate the updated distributions on B and C.

Because A has an observed (certain) FS and is B's only parent the components of B's updated FS can be read from θ_B. This shows that:

$$B = [\{0.1, 0.1, 0.8\}_{0.2}, \{0.1, 0.2, 0.7\}_{0.8}] \tag{3}$$

The FS over C is calculated similarly. Just as each of the fuzzy values in A lead to a weighted FPD in the FS of B the same occurs for C, and $C = [\alpha_{0.2}, \beta_{0.8}]$. The weighted distributions α and β are calculated using standard BN belief propagation, based on the conditional distribution of B. This is shown in equations 4, 5 and 6.

$$
\begin{aligned}
p(C|B = lo) &= \{0.1, 0.1, 0.8\} \\
p(C|B = mid) &= \{0.1, 0.8, 0.1\} \\
p(C|B = hi) &= \{0.7, 0.2, 0.1\}
\end{aligned}
\tag{4}
$$

$$
\begin{aligned}
\alpha &= \{0.1, 0.1, 0.8\} \times 0.1 + \{0.1, 0.8, 0.1\} \times 0.1 + \\
&\quad \{0.7, 0.2, 0.1\} \times 0.8 \\
&= \{0.01, 0.01, 0.08\} + \{0.01, 0.08, 0.01\} + \\
&\quad \{0.56, 0.16, 0.08\} \\
&= \{0.58, 0.25, 0.17\}
\end{aligned}
\tag{5}
$$
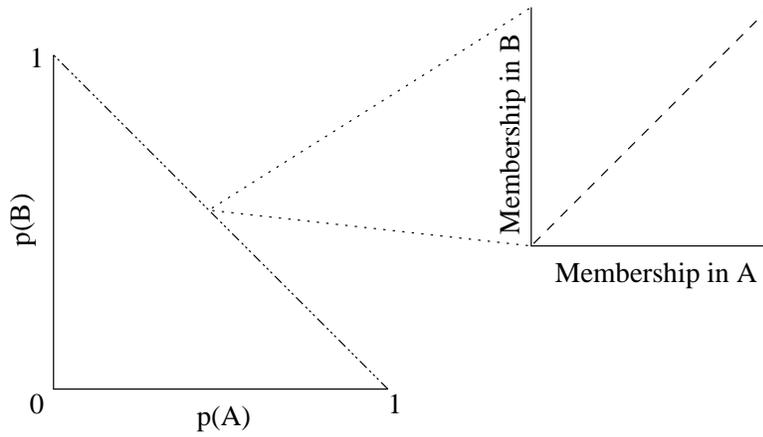
Figure 4. Assume a variable with a range (r) of 2 (A and B). Each point in the 2 dimensional p-space (left hand side) could be considered an index into a 2 dimensional μ-space (right hand side), as diagrammed. All proper probability distributions ($\sum p = 1$) fall on the dotted dashed line in the p-space. The μ-space that is indexed by some point in the p-space could be unique to that point. In a standard FPD, the μ-space is reduced to a single point on the dashed line and that point on the line in the μ-space is specified precisely by the μ of the FPD. Because any sample from a standard FPD, regardless of its value, will have the same μ, r of the dimensions are eliminated. In addition, we assume that an FPD has a proper probability distribution. In total, these reduces the dimensionality from $r + r$ to $r - 1$. In a SGFPD the μ-space would also be reduced to a single point, but any point in the μ-space would be a valid reduction, so an SGFPD with a proper distribution over the values still has $r + r - 1$ dimensions.

$$
\begin{aligned}
\beta &= \{0.1, 0.1, 0.8\} \times 0.1 + \{0.1, 0.8, 0.1\} \times 0.2 + \\
&\quad \{0.7, 0.2, 0.1\} \times 0.7 \\
&= \{0.01, 0.01, 0.08\} + \{0.02, 0.16, 0.02\} + \\
&= \{0.49, 0.14, 0.07\} \\
&= \{0.52, 0.31, 0.17\} \quad (6)
\end{aligned}
$$

The calculated FS for C is shown in equation 7.

$$
C = [\{0.58, 0.25, 0.17\}_{0.2}, \{0.52, 0.31, 0.17\}_{0.8}] \quad (7)
$$

3.4 Multi-Parent Belief Propagation

Subsection 3.3 illustrated belief propagation in a FBN when a variable has only one parent. This subsection shows *naive FBN belief propagation* in the case of a variable with multiple parents. Section 4 outlines several more nuanced approaches which address the problems with naive propagation.

Take the calculated value of C, and assume a fuzzy state for D (equation 8). What is the updated fuzzy state of E?

$$
\begin{aligned}
C &= [\{0.58, 0.25, 0.17\}_{0.2}, \{0.52, 0.31, 0.17\}_{0.8}] \\
D &= [\{0.45, 0.30, 0.25\}_{0.3}, \{0.1, 0.8, 0.1\}_{0.7}] \quad (8)
\end{aligned}
$$

Any *combination* of components, one from each parent, can be used to calculate an updated probability distribution for a variable. However, this raises the question of how to combine and weight each combination of component distributions in the parent FSs to calculate an updated FS for the child.

Because the parents are conditionally independent given the variable being updated[3], any particular combination of PD and observations can be summed over, as one was in each of equations 5 and 6. The summed over combinations became components of C's updated distribution.

---

[3] And also given their updated state and the acyclic nature of the graph

In the naive approach to belief propagation the Cartesian product of the parents' FS is used to find all possible combinations of components. μ for each one of these combinations is calculated using the *product t-norm*[2]. Any other fuzzy conjunction (normalising μ where necessary) could also be used. Because we assumed that $\sum \mu = 1$ holds for each of the parents though, using this fuzzy conjunction guarantees that $\sum \mu$ over the child's components will also equal 1 and no normalisation is necessary.

For example, if we use the first components of C and D ($\{0.58, 0.25, 0.17\}_{0.2}$ and $\{0.45, 0.30, 0.25\}_{0.3}$, respectively, equation 8) then standard Bayesian propagation and using the product t-norm to calculate μ shows that one member of E's updated FS is:

$$
\alpha = \{0.3165, 0.2189, 0.4647\}_{0.06} \quad (9)
$$

The full FPD for E will have four members, one for each member of $C \times D$ (equation 10, below). For clarity, the calculated α from equation 9 has not been substituted into this equation.

$$
E = [\alpha_{0.06},\ \beta_{0.14},\ \gamma_{0.24},\ \delta_{0.56}] \quad (10)
$$

In general a variable with k parents that each have an FS with m components will have an updated FS of size m^k. Assuming all variables have k parents, the grand-children will have updated FS of size m^{k^k}, and so forth. This is the *fuzzy state size explosion* (FSSE), and it makes naive belief propagation in a FBN intractable.

4 Dealing with Complexity

There are several ways that the explosion in the complexity can be controlled by approximating the FS. This section discusses four kinds of control. The bimodal fuzzy state $X = [\{0.9_\alpha, 0.1_\beta\}_{0.5}, \{0.1_\alpha, 0.9_\beta\}_{0.5}]$ is used as an example in several places in this section. Such a variable could represent

a committee of two in which the committee members (components) hold diametrically opposite beliefs about the outcome of some future event.

4.1 Linear Collapse

A first approximation that addresses the FSSE is to linearly collapse a FS that is made up only of FPDs, immediately after they are calculated. Each component can be weighted by its fuzzy membership and they can be summed to calculate a single, discrete, PD. For example, B (equation 3) can be collapsed as shown in equation 11. Collapsed FS are denoted with a prime.

$$
\begin{aligned}
B &= [\{0.1, 0.1, 0.8\}_{0.2}, \{0.1, 0.2, 0.7\}_{0.8}] \\
\therefore B' &= \{0.1, 0.18, 0.72\}
\end{aligned}
\tag{11}
$$

However, this approximation is unsatisfactory: it conflates probability with fuzziness and may change the *expected value* of the variable. Although it may be approximately correct in some circumstances, a simple thought experiment will show why it is insufficient.

Consider the bimodal FS X. The expected sample from X is $X' = [\alpha_{0.5}, \beta_{0.5}]$. Although this sample does not reflect any of the uncertainty in X it does reflect the bi-modality (indecision) of the variable (committee) as a whole. Subsection 4.4 returns to this approach.

If X is linearly collapsed though then $X' = \{0.5, 0.5\}$. No sample drawn from this PD can be half α and half β. Important information in X has been lost. Although further belief propagation will not be biased if this variable is summed over[4], there is no way to compare the linearly collapsed value X' with any observed value for X when trying to evaluate the quality of an inferred network.

Other approximations to the naive approach have been developed. They are discussed in the next three subsections.

4.2 Strict and Dynamic Top Fuzzy Combinations

Consider again the full (naive) FS of E, reproduced in equation 12.

$$
E = [\alpha_{0.06}, \ \beta_{0.14}, \ \gamma_{0.24}, \ \delta_{0.56}]
\tag{12}
$$

Some of the components barely contribute to the overall state and will not have a substantial influence on any children either. Such components could be ignored, and the remaining components could have their μ normalised. For example, if just the top three components of E were used then the updated FS would take the form:

$$
E = [\beta_{0.149}, \gamma_{0.255}, \delta_{0.596}]
\tag{13}
$$

The number of components retained could be either *k-component strict selected* or *ϕ-dynamically selected*. In the former case, the top k components would be selected. In the latter, the $|C|$ components with greatest μ would be selected so that $\sum_{c \in C} \mu_c > \phi$. Strict selection would mean that FBNs were only a small linear factor less efficient than classic BNs of the same size. However, the top k components may not be

___
[4] Due to the use of the product t-norm.

an accurate reflection of the full FS, thus ϕ-dynamic selection may be more appropriate in some cases.

4.3 Clustering the Fuzziness

Another way of controlling the FSSE is to calculate the full FS of each variable during belief propagation. However, before using the full FS to update the state of its children, its components could be clustered so that FPD which specified similar distributions were combined together.

For example, the FS $[\ldots, \{0.7, 0.2, 0.1\}_{0.3}, \{0.6, 0.3, 0.1\}_{0.2}, \ldots]$ might cluster to $[\ldots, \{0.66, 0.24, 0.1\}_{0.5}, \ldots]$.

Because the clustering problem would only have as many dimensions as the range of each FPD, we speculate that a simple fixed-k clustering algorithm like k-means would work very well.

Although this approach is more complex than selection or linear collapse, the total increase in complexity in belief propagation would be related to and bound by the maximum indegree and range of a variable.

4.4 Expected Values

A fourth kind of control is inspired by particle filtering and the Condensation algorithm[9]. The general *sequential Monte Carlo* (SMC) method will be outlined first. Although this approach is not as efficient as others it is applicable in all cases and is strictly correct.

Consider X. An infinite sequence of independent samples drawn from this uncertain fuzzy state will take something like the form $[\alpha_{0.5}, \beta_{0.5}], [\alpha_{0.5}, \beta_{0.5}], \ldots, [\alpha_1], [\alpha_{0.5}, \beta_{0.5}] \ldots$ and so forth.

The properties of this sequence are identical to those of the fuzzy state, and a long-enough finite sequence will be a good approximation to it. For example, 100 samples could be drawn from X. Each of these samples could then be used to propagate the uncertain state of X to X's children. The relative efficiency of this technique compared to clustering depends on the range and k_{max} of the variables, but in certain situations it may also be better.

As noted in subsection 4.1, the expected value of a variable is easily calculated analytically. For example, the expected value of X is $[\alpha_{0.5 \times 0.9 + 0.5 \times 0.1}, \beta_{0.5 \times 0.1 + 0.5 \times 0.9}] = [\alpha_{0.5}, \beta_{0.5}]$.

Doing this expectation calculation is analogous to summing over or numerically integrating a probability distribution, and we call it *fuzzy integration*. Like clustering the impact on efficiency of fuzzy integration depends on the range and k_{max}.

This approach is very similar to linear collapse but it has a number of key advantages. Firstly, like linear collapse, it does not bias any further belief propagation. This is untrue of selection and clustering. Secondly, the expected value X' which is the result of this fuzzy integration can be meaningfully compared with observed values of X when performing network inference. In many cases, users are only interested in the expected (integrated) value. In these cases the expected value of a FS is ideal.

5 A Bioinformatic Domain

Inference of large *genetic regulatory networks* (GRN) is a central problem in modern bioinformatics. However, algorithmic complexity has limited detailed inference using BNs[6;

15; 16] to $N \lesssim 100$ genes[5]. Approaches which can be applied to larger numbers of genes include modern clustering methods[10] and the inference of *graphical Gaussian models* over clustered gene expression data[8; 14].

FBNs suggest a novel approach to detailed exploration of large GRN. Such a methodology generalises to the inference of other large causal networks as well.

If the data is pre-processed by using a *fuzzy cover* algorithm the dimensionality of the problem may be reduced by an order of magnitude or more. This could lead to an exponential reduction in the algorithmic complexity which would more than offset any increase caused by the fuzzy state size explosion and its collapse.

A fuzzy cover is a clustering algorithm which *covers* the data, rather than clusters it. In a fuzzy cover, a variable (gene) can have $\sum_{c \in C} \mu_c > 1$, where C is the set of covers that the algorithm finds.

Inference over the covers is performed using a standard algorithm to find a virtual GRN. Using the retained μ_c for each $n \in N$ and $c \in C$, most of the original fidelity can be recovered after the inference has been performed by linearly devolving and normalising the network of covers back down to a network of genes. The synergistic use of dimension-reduction and FBNs are what we believe will be most useful.

The authors are using this approach (fuzzy covering, FBN inference, FBN devolution) to infer and explore large *genetic regulatory networks*. With fuzzy clustering and FBNs we expect to be able to perform more detailed exploratory inference for $N \approx 1000$.

6 Contributions and Future Work

This paper has presented a new formalisation which combines fuzzy theory and Bayesian networks. Because of the way that it extends classic BNs, all existing algorithms, tools and machine learning techniques for classic BNs can be used immediately with FBNs.

Several techniques for tractably propagating fuzzy beliefs across a FBN are also described. Using these techniques, previously used BNs can be assigned fuzzy variable states and updated accordingly. This means that existing networks, often learnt only after substantial effort, can be easily reused.

Furthermore, the difference in BN and FBN efficiency with sensible fuzziness collapse may be as little as a small linear constant in some circumstances. This means that there are few disadvantages to using FBNs instead of BNs.

The possibility of integrating FBNs into a machine learning pipeline which involves dimension-reduction and network devolution also suggests that the inference of larger causal networks will be possible using FBNs.

Future research may uncover more efficient methods for integrating, clustering or otherwise collapsing a FS. In addition, the authors plan to present an even more generalised formalisation which relaxes the assumptions made in subsection 3.2.

REFERENCES

[1] Jim F. Baldwin and Enza Di Tomaso, 'Inference and learning in fuzzy Bayesian networks', in *FUZZ'03: The 12th IEEE International Conference on Fuzzy Systems*, volume 1, pp. 630–635, (May 2003).

[2] F. Bobillo and U. Straccia, 'A fuzzy description logic with product t-norm', in *Proceedings of the 16th IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2007)*, pp. 652–657, London (United Kingtom), (July 2007).

[3] Y. Cao, P. Wang, and A. Tokuta, 'Reverse engineering of NK boolean network and its extensions — fuzzy logic network (FLN)', *New Mathematics and Natural Computation*, **3**(1), 68–87, (2007).

[4] David M. Chickering, 'Learning Bayesian networks is NP-Complete', in *Learning from Data: Artificial Intelligence and Statistics V*, eds., D. Fisher and H. J. Lenz, 121–130, Springer-Verlag, (1996).

[5] Christopher Fogelberg and Vasile Palade, 'Machine learning and genetic regulatory networks: A review and a roadmap', Technical Report CS-RR-08-04, Computing Laboratory, Oxford University, Wolfson Building, Parks Road, Oxford, OX1-3QD, (April 2008).

[6] A. J. Hartemink, D. K. Gifford, T. S. Jaakkola, and R. A. Young, 'Combining location and expression data for principled discovery of genetic regulatory network models.', *Pacific Symposium on Biocomputing*, 437–449, (2002).

[7] David Heckerman, 'A tutorial on learning with Bayesian networks', Technical report, Microsoft Research, Redmond, Washington, (1995).

[8] Katsuhisa Horimoto and Hiroyuki Toh, 'Statistical estimation of cluster boundaries in gene expression profile data.', *Bioinformatics*, **17**(12), 1143–1151, (2001).

[9] Michael Isard and Andrew Blake. Condensation – conditional density propagation for visual tracking, 1998.

[10] Sara C. Madeira and Arlindo L. Oliveira, 'Biclustering algorithms for biological data analysis: a survey', *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, **1**(1), 24–45, (2004).

[11] Daniel Neagu and Vasile Palade, 'A neuro-fuzzy approach for functional genomics data interpretation and analysis', *Neural Computing and Applications*, **12**(3-4), 153–159, (2003).

[12] Heping Pan and Lin Liu, 'Fuzzy Bayesian networks - a general formalism for representation, inference and learning with hybrid Bayesian networks', *IJPRAI*, **14**(7), 941–962, (2000).

[13] Romesh Ranawana and Vasile Palade, 'Multi-classifier systems: Review and a roadmap for developers', *International Journal of Hybrid Intelligent Systems*, **3**(1), 35–61, (2006).

[14] Hiroyuki Toh and Katsuhisa Horimoto, 'Inference of a genetic network by a combined approach of cluster analysis and graphical Gaussian modeling.', *Bioinformatics*, **18**(2), 287–297, (2002).

[15] Jing Yu, V. Anne Smith, Paul P. Wang, Alexander J. Hartemink, and Erich D. Jarvis, 'Advances to Bayesian network inference for generating causal networks from observational biological data.', *Bioinformatics*, **20**(18), 3594–3603, (2004).

[16] Yu Zhang, Zhingdong Deng, Hongshan Jiang, and Peifa Jia, 'Dynamic Bayesian network (DBN) with structure expectation maximization (SEM) for modeling of gene network from time series gene expression data.', in *BIOCOMP*, eds., Hamid R. Arabnia and Homayoun Valafar, pp. 41–47. CSREA Press, (2006).