# A case study in dynamic belief networks: monitoring walking, fall prediction and detection

A. E. Nicholson
Department of Computer Science,
Monash University,
Clayton, VIC 3168, Australia,
tel: (03) 9905 5211. fax: (03) 9905 5146
annn@cs.monash.edu.au.

## Abstract

The task is to monitor walking patterns and give early warning of falls using foot switch and mercury trigger sensors. We describe a dynamic belief network model for fall diagnosis which, given evidence from sensor observations, outputs beliefs about the current walking status and makes predictions regarding future falls. The model represents possible sensor error and is parametrised to allow customisation to the individual being monitored.

# 1 Introduction

The task is to monitor the stepping patterns of elderly people, or recovering patients. Not only are actual falls to be detected causing an alarm to be raised, but irregular walking patterns, stumbles and near falls are to be monitored, and early warning of possible falls made in time for giving assistance. The monitoring is performed using two kinds of sensors: foot-switches which provide information about a foot step; and a mercury sensor which is triggered by a change in height such as going from standing upright to lying horizontal, and hence indicates a fall has occurred. Timing data for the observations is also given.

Previous work in this domain performed fall diagnosis with a simple state machine [6], however this does not allow representation of either degrees of belief as to the person's ambulatory status, or of the uncertainty in the sensor readings. *Dynamic belief networks* integrate a mechanism for inference under uncertainty with a secure Bayesian foundation, and are suitable for domains, such as the fall diagnosis problem, where the world changes and the focus is reasoning over time. In this paper we present a dynamic belief network model for the fall diagnosis problem, an interesting practical application of an AI approach to the real world problem of medical monitoring.

The organisation of this paper is as follows. The fall diagnosis problem is described in detail in Sect. 2. Sect. 3 gives an introduction to dynamic belief networks. In Sect. 4 we develop a complete belief network model for the fall diagnosis problem, with results given in Sect. 5. Extensions to the basic network are described in Sect. 6.

# 2 The Fall Diagnosis Problem

Davies [6] describes a project with Prof. Ian Brown at Monash University, Dept. of Electrical Engineering, for monitoring the stepping pattern of elderly people and patients. Step data is obtained using foot-switches and sent via a mobile data network to a remote monitoring station, which attempts to detect falls and near falls by using a state transition diagram, shown in Fig. 1. This model was developed by Davies in conjunction with expert medical practitioners.

The sensor observations are as follows: **L:** data from the left foot switch; **R:** data from the right foot switch; **M:** data from a mercury switch indicates a change in height. Each sensor observation is accompanied by a time, which is the time duration of the sensor observation. This timing information is crucial in performing fall diagnosis: $y$ is the threshold time below which a foot switch reading is considered a stumble; $x$ is the threshold time below which the mercury trigger is taken to indicate a fall (a slow change in height would be consistent with intended sitting or lying down). [1]

[1] Davies used the threshold times of $x = 2s$, $y = 0.8s$.

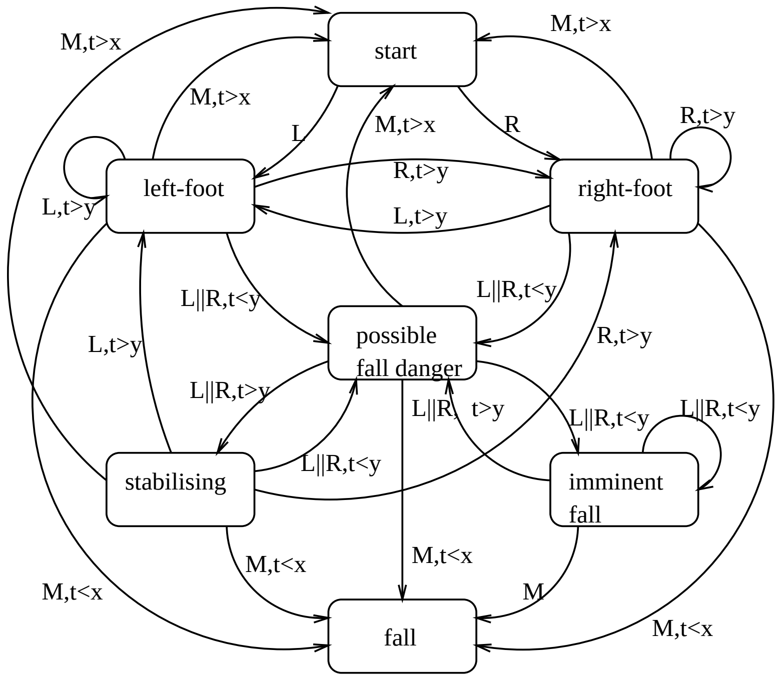


Figure 1: Davies's State Machine for the fall diagnosis problem [3]

The states of the state machine are as follows. **start** is a state of ignorance, entered when a slow mercury trigger is recorded, or when the machine is restarted after a fall alarm. **left-foot, right** are walking states, indicating which foot is currently forward in the process of walking. Normal stepping patterns, indicated by an observation time interval of $> y$, should see the state machine alternate between these two states. **possible-fall-danger** is an intermediate waiting state, entered after any abnormally fast step which may have been a stumble, as the next input will determine if the patient is really stumbling or if the reading was a lone occurrence. The **stabilising** state will be reached after `possible-fall-danger` if a slow-controlled step is observed. The **imminent-fall** state will be reached after `possible-fall-danger` if another quick step is observed. The system currently increments a counter storing the number of near falls detected in the day. The **fall** state will be reached from `imminent-fall` with any triggering of the mercury switch, in which case the system sounds a local alarm and places an emergency call to the base station. This fall state is also reached from states other than `imminent-fall`, however in these cases the time data for the mercury switch must be $< x$ seconds.

This state machine model has a number of limitations. First, there is no representation of degrees of belief in the current state of the person's ambulation. Second there is no distinction between actual states of the world and observations of that state, for example, the fall state is really a **fall-alarm** state. That is, there is no explicit representation of the uncertainty in the sensor observation

[14]. Possible sensor errors include:

- False positives: the sensor wrongly indicates that an action (left, right, lowering action) has occurred (also called clutter, noise or false alarms).
- False negatives: an action occurred but the sensor was not triggered and no observation was made (also called missed detection).
- Wrong time data: the sensor readings indicate the action which occurred, however the time interval reading is incorrect.

Also, in Davies's representation, there is no difference between a sequence of alternate foot steps, and a sequence of same foot steps (hopping); we would expect that the latter should probably increase the concern about an imminent fall.

# 3 Dynamic Belief Networks

## 3.1 Belief Networks

Belief networks are directed acyclic graphs, where nodes correspond to random variables, which we assume to take discrete values (although in general they need not be discrete). In this paper the variables pertain to the world state or the sensor observations. The relationship between any set of state variables can be specified by a joint probability distribution. The nodes in the network are connected by directed arcs, which may be thought of as causal or influence links. The connections also specify the independence assumptions between nodes. Each node has associated with it a *probability distribution*, which, for each combination of the variables of the parent nodes (called a *conditioning case*), gives a probability of each value of the node variable. The probability distribution for a node with no predecessors is the prior distribution. Evidence can be specified about the state of *any* of the nodes in the network — root nodes, leaf nodes or intermediate nodes. This evidence is propagated through the network affecting the overall joint distribution (as represented by the conditional probabilities). There are a number of exact and approximate inference algorithms available for performing belief updating [16]; in this paper we are not concerned with the particular algorithm.

## 3.2 Why *Dynamic* Belief Networks?

Belief networks have been been used in various applications, such as medical diagnosis [20] and model-based vision [12], which initially were more static, i.e. essentially the nodes and links do not change over time. Such approaches involve determining the structure of the network; supplying the prior probabilities for root nodes and conditional probabilities for other nodes; adding or retracting evidence about nodes; repeating the inference algorithm for each change in evidence.

Some work has been done on the dynamic *construction* of belief networks [2, 3], but the desired output is still a single static network. More recently researchers have used belief networks in *dynamic* domains such as the fall diagnosis problem, where the world changes and the focus is reasoning over time [7, 11, 15, 10] Such dynamic applications include robot navigation and map learning based on *temporal* belief networks [7], monitoring diabetes [1], monitoring robot vehicles [13], oil forecasting [5], [17], forecasting sleep apnea [4], automated vehicle control [8] and traffic plan recognition [18]. For such applications the network grows over time, as the state of each domain variable at different times is represented by a *series* of nodes. These dynamic networks are Markovian, which constrains the state space to some extent, however it is also crucial to limit the history being maintained in the network.

## 3.3 A Generic DBN Structure

A generic dynamic belief network structure for monitoring application is shown in Fig. 2 [15]. The types of nodes are: **World** nodes, which describe the central domain variables (for example, position, heading, velocity) variables; **Event** nodes, which represent a change in the state of a world node; **Observation** nodes, which represent direct observations of world nodes, or the observable effects of an event. Time is discretised at irregular intervals, usually divided by the occurrence of discrete events. Each time slice within the network represents the static environment during that time interval. The structure within time slices is often regular. These networks are typically highly connected, particularly between adjacent time slices. The conditional probability distributions (CPDs) are shown in rectangular boxes. The CPDs of nodes with parents in the previous time slice are usually a function of the time interval. After addition of sensor observations as evidence to the DBN (indicated by dark shading), belief updating is performed, providing prediction for the values of the world nodes at time slice $T + 1$.

Note the distinction between the two types of observation nodes in this model: $\mathtt{O}(T)$ for the direct observation of a world variable, and $\mathtt{O}(T_i,T_{i+1})$ for the observation of a change in the state of a world node, that is, the observation of an *event*. They provide the foundation for the dynamic construction of the network, identifying when the nodes for a new time slice should be added. An example of event observation is the robot vehicle monitoring in [14]: the environment (a laboratory in which a robot vehicle roams) is divided into regions by the light-beam sensors, which provide observations on events where an agent (person or robot) moves between regions.

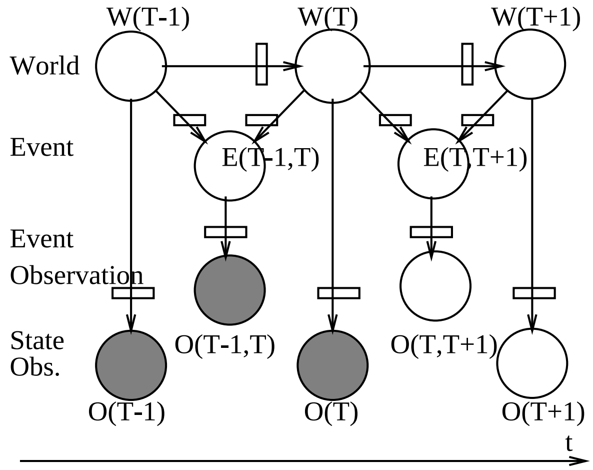


Figure 2: Generic structure for a dynamic belief network

### 3.4 Representing actions

For planning and monitoring applications, it is often useful to have the action as a separate node, where the action directly affects the value of the state variable at the next time instant.

A DBN with action nodes is common for planning application. The networks in both [7] and [9] have action nodes with no predecessors, and the prior is not important because planning is done by instantiating the action node for all combinations of goals, with the instantiation which maximises some utility function chosen as the plan. Pynadeth and Wellman, in work on plan recognition for a traffic monitoring domain [18], have a variation on this structure where there is a prior on the goal, which then is a predecessor of the action node; they instantiate the goal, and infer the action, rather observe the action. In [15], where the observations are on the light beam sensor crossing events, the action structure is added together with additional structure which maintains information about movement trends, and provides a conditional probability for the action.

## 4 Basic DBN Model for Fall Diagnosis

Davies's state machine essentially defines the fall diagnosis problem as a set of *if-then-else* rules. When developing the DBN model, a key difference is that we focus on the causal relationships between domain variables, and make a clear distinction between observations and actual states of the world. A DBN for the fall diagnosis problem is given in Fig. 3. In the rest of this section, we describe the various features of this network in such a way as provide an insight into the network development process.

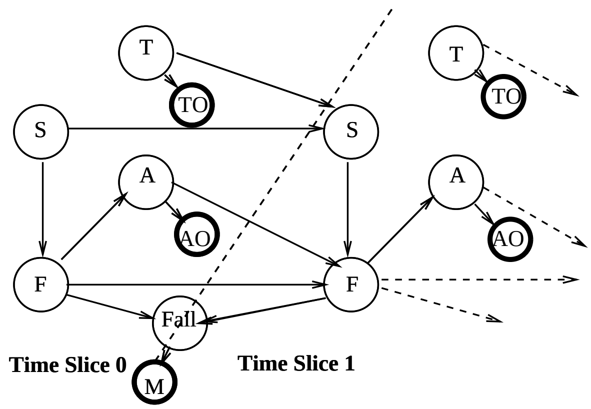


Figure 3: Dynamic Belief Network for fall diagnosis problem including interslice arcs. The smaller nodes with thicker outlines are the sensor observations.

## Nodes

When considering how to represent a person's walking situation, possibilities include whether the person is stationary on both feet, on a step with either the left or right foot forward, or has fallen and hence is no longer on their feet. We call the main world node representing this `F`, which may take 4 possible values: **both**, **left**, **right**, **off**. The event node `Fall`, which is a boolean, indicates whether a fall has taken place between time slices.

Fall warning and detection relies on an assessment of the person's walking pattern. The node `S` maintains the person's status, and may take the possible values **ok** and **stumbling**.

The action variable, `A`, may take the possible values **left**, **right**, or **none**. The last value is necessary for the situation where a time slice is added because the mercury sensor has triggered (i.e. the person has fallen) but no step was taken, or a foot switch false positive observation was registered. [2]

There is an observation node for each of the two sensors. The foot switch observations are essentially observations on the step actions, and are represented by the `AO` node which contains the same values as the action node. The mercury sensor trigger is represented by the node `M`, which represents a boolean variable.

While it is possible to not represent the time explicitly, but merely in the construction of the conditional probability distributions (say between $\mathtt{F}_i$ and $\mathtt{F}_{i+1}$), since time is one of our observations, we must represent it explicitly in order to be able to represent the sensor error uncertainty.

The time between sensor observations is given by the node `T`. Given the problems with combining continuous and discrete variables [19, p.465] and the limitations of the sensor, node `T` has discrete values representing tenths of seconds.

[2] Note that we can easily extend the model to handle a person jumping, by adding an additional possible value for `A`, say `jump`.

While the fact that there is no obvious upper limit on the time between readings may seem to make it difficult to define the state space of the T node, recall that a monitoring DBN is extended to the next time slice when a sensor observation is made, say $n$ tenths of a second. If we ignored error in time data, we could add a T with a single value **n**.

In order to represent the uncertainty in the sensor data, we say it can take values within an interval around the sensor time reading that generates the addition of a new time slice to the DBN. If there is some knowledge of the patients expected walking speed, values in this range can be added also. The time observation node, TO, has the same state space as T.

There is a new copy of each node added for each time slice; we will indicate the time slice by the subscript. The possibility of adding more time slices is shown by the dashed arcs to the right.

Note that there is no need to explicitly include **imminent-fall** or **fall** in the status node. The belief of a fall in the current time slice $i$ is given by the posterior obtaining after adding evidence and running the inference algorithm, that is, bel($\mathtt{Fall}_i$=T), [3], and a warning about an imminent fall can be based on the predictions for the next time slice, that is whether bel($\mathtt{Fall}_{i+1}$) is greater than some warning threshold.

## Structure and Conditional Probability Distributions

**World and Event Nodes.** The CPDs for the nodes A, F, Fall and S are given in Table 1. The model for walking is represented by the arcs from $\mathtt{F}_i$ to $\mathtt{A}_i$, and $\mathtt{F}_i$, $\mathtt{A}_i$ and $\mathtt{S}_i$ to $\mathtt{F}_{i+1}$.

We assume that normal walking involves alternating left and right steps. Where the left and right are symmetric, only one combination is included in the table. We have priors for starting on both feet ($r$) or already being off the ground ($s$). Because we have restricted the possible actions to moving either feet or neither, there is no way for this model to reflect a person getting to their feet; we are assuming use of the model will begin with the person upright, and if not, they stay off their feet. Looking at the CPD for $\mathtt{F}_{i+1}$, we can see that a left step can have the walker finish on one foot or both feet, depending on whether it is a half or full step. By definition, if a person finishes on a particular foot, it rules out some actions; for example, if $\mathtt{F}_{i+1} = \mathtt{left}$, the action could not have been right. These zero conditional probability are omitted from the table.

The CPD for $\mathtt{F}_{i+1}$ for the conditioning cases where $\mathtt{S}_{i+1} = \mathtt{stumbling}$ is exactly the same as for ok except the $p$ and $q$ probability parameters will have lower values, representing the higher expectation of a fall; that is, $p'_i < p_i$, $q'_i < q_i$, for all relevant $i$.

[3] Also given by bel($\mathtt{F}_i$=off); the redundancy is useful for describing the problem, but could be removed to improve computational efficiency.

Table 1: CPDs for step action node A, the foot node F, the Fall node and the walking status node S

| | | |
|---|---|---|
| P($F_0$=left‖right\|) | = (1-$r$-$s$)/2 | |
| P($F_0$=both\|) | = $r$ | |
| P($F_0$=off\|) | = $s$ | |
| | | |
| P(A=left\|F=right) | = $u$ | alternate feet |
| P(A=right\|F=right) | = $v$ | hopping |
| P(A=none\|F=right) | = 1-$u$-$v$ | stationary |
| P(A={left‖right}\|F=both) | = $w/2$ | start with left or right |
| P(A=none\|F=both) | = 1-$w$ | stationary |
| P(A=none\|F=off) | = 1 | can't walk when off feet |
| | | |
| P($F_{i+1}$=left\|$F_i$=right,$A_i$=left,$S_{i+1}$=ok) | = $p_1$ | succ. alternate step |
| P($F_{i+1}$=both\|$F_i$=right,$A_i$=left,$S_{i+1}$=ok) | = $q_1$ | half-step |
| P($F_{i+1}$=off\|$F_i$=right,$A_i$=left,$S_{i+1}$=ok) | = 1-$p_1$-$q_1$ | fall prob |
| P($F_{i+1}$=left\|$F_i$=left,$A_i$=left,$S_{i+1}$=ok) | = $p_2$ | succ. hop |
| P($F_{i+1}$=both\|$F_i$=left,$A_i$=left,$S_{i+1}$=ok) | = $q_2$ | half-hop |
| P($F_{i+1}$=off\|$F_i$=left,$A_i$=left,$S_{i+1}$=ok) | = 1-$p_2$-$q_2$ | fall prob |
| P($F_{i+1}$=left\|$F_i$=both,$A_i$=left,$S_{i+1}$=ok) | = $p_3$ | succ. first step |
| P($F_{i+1}$=both\|$F_i$=both,$A_i$=left,$S_{i+1}$=ok) | = $q_3$ | unsucc. first step |
| P($F_{i+1}$=off\|$F_i$=both,$A_i$=left,$S_{i+1}$=ok) | = 1-$p_3$-$q_3$ | fall prob |
| P($F_{i+1}$=left\|$F_i$=left,$A_i$=none,$S_{i+1}$=ok) | = $p_4$ | |
| P($F_{i+1}$=off\|$F_i$=left,$A_i$=none,$S_{i+1}$=ok) | = 1-$p_4$ | fall when on left foot |
| P($F_{i+1}$=right\|$F_i$=right,$A_i$=none,$S_{i+1}$=ok) | = $p_5$ | |
| P($F_{i+1}$=off\|$F_i$=right,$A_i$=none,$S_{i+1}$=ok) | = 1-$p_5$ | fall when on right foot |
| P($F_{i+1}$=both\|$F_i$=both,$A_i$=none,$S_{i+1}$=ok) | = $p_6$ | |
| P($F_{i+1}$=off\|$F_i$=both,$A_i$=none,$S_{i+1}$=ok) | = 1-$p_6$ | fall when on both feet |
| P($F_{i+1}$=off\|$F_i$=off,$A_i$=left,$S_{i+1}$=any) | = 1 | no "get up" action |
| | | |
| P(Fall=T \| $F_{i+1}$=off,$F_i$={left‖right‖both}) | = 1 | from upright to ground |
| P(Fall=F \| $F_{i+1}$=any,$F_i$=off) | = 1 | can't fall if on ground |
| | | |
| P($S_{i+1}$=ok\|$T_i$=t) | = 1 | if t $\geq$ y |
| P($S_{i+1}$=stumbling\|$T_i$=t) | = 1 | if t $<$ y |

If there are any variations on walking patterns for an individual patient, for example if one leg was injured, the DBN can be customised by varying the probability parameters, $s$, $r$, $p_i$, $q_i$, $u$, $v$ and $w$, and removing the assumption that left and right are completely symmetric. For example, we can relax the assumption that the person is equally likely to start on the left foot as the right. Note that having different $p$ parameters indicates different expectations of a fall when the person is walking compared to hopping. Also, a person can end up off their feet even if the status node `S` is indicating `ok`.

The fall event node `Fall` has $\mathtt{F}_i$ and $\mathtt{F}_{i+1}$ as predecessors; a fall only occurs when the subject was on his or her feet to start with ($\mathtt{F}_i \neq \mathtt{off}$), and finishes off their feet ($\mathtt{F}_{i+1} = \mathtt{off}$). [4]

The value of walking status node `S` is determined solely by the time between sensor readings (see next section for an extension which takes into account status history). In this DBN model, the `T` node has no predecessors. One possible model is to have uniform priors, or the prior can also be modified, based on sensor observations over time, to reflect an individual's ordinary walking speed.

**Observation Nodes.** When constructing the conditional probability distributions for the various observation nodes, the confidence in the observation is given by some value based on a model of the sensor's performance and is empirically obtainable; *pos* is the sensitivity of the positive sensor data, *neg* is the specificity of the negative sensor data (or, 1-*neg* is the probability of ghost data). We make the default assumption that missing or wrong data are equally likely — this need not be the case and can be replaced by any alternative plausible values.

Each observation node has a single predecessor: the mercury trigger observation node `M` has predecessor `F`; the foot-step action observation node `AO` has predecessor `A`; the time observation node `TO` has predecessor `T`. The conditional probability distributions for `M`, `AO` and `TO` are shown in Table 2. Note that the CPD for the case where the sensor is defective is uniform over the other time values; this could easily be changed to cluster around the true time interval. If the timing sensor fails and no data is obtained, fall diagnosis becomes impossible, so we do not handle the case of missing time data.

Note that when the monitoring begins, we do not need to have a known start state; we need only have a prior over the possible starting positions. Because the standard left foot, right foot, walking model, is represented by the conditional probability distribution between $\mathtt{F}_i$ and $\mathtt{F}_{i+1}$, if the first data received is a `left` $\mathtt{S}_i$, then after belief updating, the belief vector will include $\mathtt{bel(F}_i\mathtt{=off) = 0}$, $\mathtt{bel(F}_i\mathtt{=left)} < \mathtt{0.25}$, $\mathtt{bel(F}_i\mathtt{=left)} > \mathtt{0.25}$ and $\mathtt{bel(F}_i\mathtt{=left)} > \mathtt{0.25}$. The

[4] We do not model the situation Davies described where the mercury trigger data is ignored if the time is $\geq$ x; this would be more correctly modelled by: adding an additional value, `sitting` to the state `F`; adding an additional value, `sit`, to the action `A`; adding another alternative, `sat`, to the fall event `fall`; adding a connection from $\mathtt{T}_i$ to $\mathtt{A}_i$; changing the CPD for $\mathtt{A}_i$ to say that if the time is above the threshold, then the `sit` action is possible.

Table 2: CPDs for observation nodes M (mercury trigger), AO (foot switch), TO (time data)

| | | |
|---|---|---|
| P(M=T\|Fall=T) | = $pos_1$ | ok |
| P(M=F\|Fall=T) | = 1-$pos_1$ | missing |
| P(M=F\|Fall=F) | = $neg_1$ | ok |
| P(M=T\|Fall=F) | = 1-$neg_1$ | false alarm |
| | | |
| P(AO=left\|A=left) | = $pos_2$ | ok |
| P(AO=right\|A=right) | = $pos_2$ | ok |
| P(AO=right\|A=left) | = (1-$pos_2$)/2 | wrong |
| P(AO=left\|A=right) | = (1-$pos_2$)/2 | wrong |
| P(AO=none\|A=left) | = (1-$pos_2$)/2 | missing |
| P(AO=none\|A=right) | = (1-$pos_2$)/2 | missing |
| P(AO=none\|A=none) | = $neg_2$ | ok |
| P(AO=left\|A=none) | = (1-$neg_2$)/2 | false alarm |
| P(AO=right\|A=none) | = (1-$neg_2$)/2 | false alarm |
| | | |
| P(TO=x\|T=x) | = $pos_3$ | ok, y $\neq$ x, T and TO have $m$ values. |
| P(TO=y\|T=x) | = $1 - pos_3/m$-1, | ok, y $\neq$ x, T and TO have $m$ values. |

DBN presented is one possible model for the fall diagnosis problem; many other variations are possible. For example, the DBN does not handle the case where both foot switches provide data at the same time.

# 5 Results

The results described in this section were obtained using the Lisp-based IDEAL belief network development environment [21] on a GNU Common Lisp platform. We present results of a Fall Diagnosis network modelled for a given set of parameters: $s = 0.0$, $r = 0.9$, $u = 0.7$, $v = 0.2$, $w = 0.1$, $p_1 = 0.6$, $q_1 = 0.3$, $p'_1 = 0.5$, $q'_1 = 0.4$ $p_2 = 0.6$, $q_2 = 0.3$, $p'_2 = 0.5$, $q'_2 = 0.4$, $p_3 = 0.6$, $q_3 = 0.3$, $p'_3 = 0.5$, $q'_3 = 0.4$, $p_4 = 0.95$, $p'_4 = 0.85$, $p_5 = 0.95$, $p'_5 = 0.85$, $p_6 = 0.9$, $p'_6 = 0.8$, $pos_1 = 0.9$, $pos_2 = 0.9$, $pos_3 = 0.9$, $neg_1 = 0.95$, $neg_2 = 0.95$. The T and TO time nodes had 4 possible values, $t_1$, $t_2$, $t_3$, and $t_4$; the lowest, $t_1$ was below the threshold y and meant the subject was considered to be stumbling.

After constructing the DBN, we entered a sequence of evidence, that is simulated observations from the sensors, and performed belief updating after every new piece of evidence was added. Table 3 shows the posterior probabilities, or beliefs, of the values of nodes in the network across this sequence of data. For reasons of space, we left out the initial $S_0$ node and the $T_2$ and $TO_2$ nodes from the model, and do not give all the beliefs, especially if they are uniform or otherwise obvious. Probabilities have been rounded to 4 decimal places. Evidence added results in a 1.0 belief for that value, shown in bold in the table; also bolded are the beliefs described below in the text. The evidence sequence added, and the effect on the beliefs, was as follows.

**No evidence added:** All beliefs are based on the parameters. Belief in an immediate fall is small, bel($\texttt{Fall}_0$ = T)=0.1194, but chance of being off feet in 2 steps is higher, bel($\texttt{F}_0$=T)=0.2238.

$\texttt{TO}_0$ **set to** $\texttt{t}_1$**:** This increases the probability that the person is stumbling, that is, bel($\texttt{S}_1$ = **stumbling**)=0.9, which in turn slightly increases the belief in a fall, bel($\texttt{Fall}_0$ = T) = 0.1828.

$\texttt{AO}_0$ **set to left:** Foot switch information leads to a change in the belief in the initial starting state; bel($\texttt{F}_0$=**right**) has increased from 0.05 to 0.2550, reflecting the model of alternate foot steps.

$\texttt{M}_0$ **set to false:** The negative mercury trigger data makes it very unlikely that a fall occurred, bel($\texttt{Fall}_0$=T)=0.0203.

$\texttt{TO}_0$ **set to** $\texttt{t}_2$**:** "Resetting" of the original timing data makes it less likely the person was stumbling, reducing the belief in a fall, bel($\texttt{Fall}_0$=T) = 0.0098.

$\texttt{M}_0$ **set to true:** However, resetting the mercury trigger data makes a fall most probable, bel($\texttt{Fall}_0$=T)=0.6285, although there is still the chance that the sensor has given a wrong reading.

$\texttt{M}_1$ **set to false,** $\texttt{TO}_1$ **set to** $\texttt{t}_4$**,** $\texttt{AO}_1$ **set to none:** No action, and no mercury trigger data confirms the earlier fall, bel($\texttt{Fall}_0$=T)=0.7903, since if the person is already on the ground they won't take a left or right step.

# 6 Extensions to the Fall Diagnosis DBN

## 6.1 Maintaining a History

The states **imminent-fall**, possible-fall, and stabilising in the original state machine are an attempt to capture the idea that the history beyond the current time interval gives information about the likelihood of a fall soon. This is represented in a DBN by the use of a history node [15], which maintains a count of how long the agent has been exhibiting one type of behaviour. For our domain, this would be a status history node, $\texttt{H}_i$, for each time slice; its predecessors are the previous and current walking status nodes, $\texttt{S}_{i-1}$ and $\texttt{S}_i$. H then becomes a predecessor of $\texttt{F}_{i+1}$, and the CPD entries are changed so that the probability of falling is a function of the stumble count.

Table 3: Changing beliefs as evidence is added or changed.

| Node | Val | None | $TO_0$=$t_1$ | $AO_0$=left | $M_0$=F | $TO_0$=$t_2$ | $M_0$=T | SET |
|---|---|---|---|---|---|---|---|---|
| $T_0$ | $t_1$ | 0.25 | 0.9000 | 0.9000 | 0.8914 | 0.0305 | 0.0535 | 0.0616 |
| | $t_2$ | 0.25 | 0.0333 | 0.0333 | 0.0361 | 0.9026 | 0.8812 | 0.8736 |
| | $t_3$ | 0.25 | 0.0333 | 0.0333 | 0.0361 | 0.0334 | 0.0326 | 0.0323 |
| | $t_4$ | 0.25 | 0.0333 | 0.0333 | 0.0361 | 0.0334 | 0.0326 | 0.0323 |
| $TO_0$ | $t_1$ | 0.25 | **1.0** | **1.0** | **1.0** | 0.0 | 0.0 | 0.0 |
| | $t_2$ | 0.25 | 0.0 | 0.0 | 0.0 | **1.0** | **1.0** | **1.0** |
| $F_0$ | left | 0.05 | 0.05 | 0.0870 | 0.0860 | 0.0856 | 0.0964 | 0.0911 |
| | right | 0.05 | 0.05 | **0.2550** | 0.2717 | 0.2515 | 0.2792 | 0.2767 |
| | both | 0.90 | 0.90 | 0.6581 | 0.6422 | 0.6628 | 0.6244 | 0.6322 |
| | off | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| $A_0$ | left | 0.09 | 0.09 | 0.6403 | 0.6483 | 0.6453 | 0.6047 | 0.5427 |
| | right | 0.09 | 0.09 | 0.0356 | 0.0360 | 0.0359 | 0.0336 | 0.0302 |
| | none | 0.82 | 0.82 | 0.3241 | 0.3156 | 0.3188 | 0.3617 | 0.4271 |
| $AO_0$ | left | 0.1265 | 0.1265 | **1.0** | **1.0** | **1.0** | **1.0** | **1.0** |
| | right | 0.1265 | 0.1265 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| | none | 0.7470 | 0.7470 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| $Fall_0$ | True | **0.1194** | **0.1828** | 0.1645 | **0.0203** | **0.0098** | **0.6285** | **0.7903** |
| | False | 0.8806 | 0.8173 | 0.8355 | 0.9797 | 0.9902 | 0.3715 | 0.2096 |
| $M_0$ | True | 0.1515 | 0.2053 | 0.1898 | 0.0 | 0.0 | **1.0** | **1.0** |
| | False | 0.8485 | 0.7947 | 0.8102 | **1.0** | **1.0** | 0.0 | 0.0 |
| $S_1$ | ok | 0.75 | 0.1 | 0.1 | 0.1086 | 0.9695 | 0.9465 | 0.9383 |
| | stum'g | 0.25 | **0.9** | 0.9 | 0.8914 | 0.0305 | 0.0535 | 0.0617 |
| $F_1$ | left | 0.0638 | 0.0425 | 0.2737 | 0.3208 | 0.5120 | 0.1921 | 0.0340 |
| | right | 0.0638 | 0.0425 | 0.0168 | 0.0197 | 0.0303 | 0.0114 | 0.0020 |
| | both | 0.7530 | 0.7322 | 0.5451 | 0.6391 | 0.4478 | 0.1680 | 0.1736 |
| | off | 0.1194 | **0.1828** | 0.1645 | 0.0203 | 0.0098 | 0.6285 | 0.7903 |
| $T_1$ | $t_1$ | 0.25 | 0.25 | 0.25 | 0.25 | 0.25 | 0.25 | 0.0326 |
| | $t_4$ | 0.25 | 0.25 | 0.25 | 0.25 | 0.25 | 0.25 | 0.9006 |
| $TO_1$ | $t_4$ | 0.25 | 0.25 | 0.25 | 0.25 | 0.25 | 0.25 | **1.0** |
| $A_1$ | left | 0.0950 | 0.0749 | 0.0938 | 0.1099 | 0.1461 | 0.0548 | 0.0035 |
| | right | 0.0950 | 0.0749 | 0.2222 | 0.2605 | 0.3869 | 0.1451 | 0.0092 |
| | none | 0.8090 | 0.8502 | 0.6841 | 0.6296 | 0.4670 | 0.8001 | 0.9872 |
| $AO_1$ | left | 0.1308 | 0.1137 | 0.1297 | 0.1434 | 0.1741 | 0.0966 | 0.0 |
| | right | 0.1308 | 0.1137 | 0.2389 | 0.2714 | 0.3788 | 0.1734 | 0.0 |
| | none | 0.7383 | 0.7730 | 0.6315 | 0.5851 | 0.4671 | 0.7301 | **1.0** |
| $Fall_1$ | True | 0.1044 | 0.0975 | 0.0959 | 0.1124 | 0.1099 | 0.0412 | 0.0024 |
| | False | 0.8956 | 0.9025 | 0.9041 | 0.8876 | 0.8901 | 0.9588 | 0.9976 |
| $M_1$ | True | 0.1387 | 0.1329 | 0.1315 | 0.1455 | 0.1434 | 0.0850 | 0.0 |
| | False | 0.8612 | 0.8671 | 0.8685 | 0.8545 | 0.8566 | 0.9150 | **1.0** |
| $S_2$ | ok | 0.75 | 0.75 | 0.75 | 0.75 | 0.75 | 0.75 | 0.9673 |
| | stum'g | 0.25 | 0.25 | 0.25 | 0.25 | 0.25 | 0.25 | 0.0327 |
| $F_2$ | left | 0.0673 | 0.0531 | 0.0898 | 0.1053 | 0.1472 | 0.0552 | 0.0258 |
| | right | 0.0673 | 0.0531 | 0.1335 | 0.1565 | 0.2291 | 0.08594 | 0.0076 |
| | both | 0.6415 | 0.6136 | 0.5164 | 0.6055 | 0.5040 | 0.1891 | 0.1740 |
| | off | **0.2238** | 0.2802 | 0.2603 | 0.1327 | 0.1197 | 0.6698 | 0.7927 |

## 6.2 Representing a consistent walking pace

We can also improve the model of what a person's ordinary walking pace is by adding an arc from $\mathtt{T}_i$ to $\mathtt{T}_{i+1}$, which would allow a representation of the expectation that the walking pace should remain fairly constant.

## 6.3 Explaining incorrect data

The DBN described in the previous section provides a mechanism for handling (by implicitly rejecting) certain inconsistent data. It represents adequately the underlying assumptions about the data uncertainty, however it does not provide an explanation of *why* the observed sensor data might be incorrect. We can represent the most usual source of incorrect data, namely a defective sensor, by the addition of a sensor status node SS [14] for each sensor. Each sensor status node becomes a predecessor of the corresponding observation node, and there is a connection between sensor status nodes across time slices.

# 7 Conclusions

We have shown the development of a dynamic belief network model for fall diagnosis which overcomes the limitations of previous work. Given evidence from sensor observations, the model outputs beliefs about the current walking status and makes predictions regarding future falls. The model represents possible sensor error, and is parametrised to allow customisation to the individual being monitored.

**Acknowledgements** Thanks to Dr. Jon Oliver for suggesting this fall diagnosis problem as an application for dynamic belief networks.

# References

[1] S.A. Andreassen, J.J. Benn, R. Hovorks, K. G. Olesen, and R. E Carson. A probabilistic approach to glucose prediction and insulin dose adjustment: Description of metabolic model and pilot evaluation study. Unpublished draft, 1991.

[2] J.S. Breese. Construction of belief and decision networks. Technical Memorandum 30, Rockwell Palo Alto Laboratory, 444 High Street, Palo Alto, California 94301, 1989.

[3] Eugene Charniak and Robert Goldman. Plan recognition in stories and in life. In *Proc. of the Fifth Workshop on Uncertainty in Artificial Intelligence*, pages 54–60, 1989.

[4] P. Dagum and A. Galper. Forecasting sleep apnea with dynamic network models. In *Proceedings of the Ninth Conference on Uncertainty in AI*, pages 64–71, 1993.

[5] P. Dagum, A. Galper, and E. Horvitz. Dynamic network models for forecasting. In *Proceedings of the 8th Conference on Uncertainty in Artificial Intelligence*, pages 41–48, 1992.

[6] James Davies. Fall diagnosis with a mobile data network. unpublished bcse honours report, dept. of electrical engineering, monash university, 1995.

[7] Thomas Dean and Michael P. Wellman. *Planning and control.* Morgan Kaufman Publishers, San Mateo, Ca., 1991.

[8] Jeff Forbes, Tim Huang, Keiji Kanazawa, and Stuart Russell. The batmobile: Towards a bayesian automated taxi. In *Proceedings of the 14th Int. Joint Conf. on Artificial Intelligence (IJCAI'95)*, pages 1878–1885, 1995.

[9] Robert P. Goldman and Eugene Charniak. Dynamic construction of belief networks. In *Proc. of 1990 Workshop on Uncertainty in Artificial AIntelligence*. 1990.

[10] Keiji Kanazawa. *Probability, Time, and Action.* PhD thesis, Brown University, Providence, RI, 1992.

[11] U. Kjærulff. A computational scheme for reasoning in dynamic probabilistic networks. In *Proceedings of the 8th Conference on Uncertainty in Artificial Intelligence*, pages 121–129, 1992.

[12] T.S. Levitt, J. M. Agosta, and T.O. Binford. Model-based influence diagrams for machine vision. In *Proc. of the Fifth Workshop on Uncertainty in Artificial Intelligence*, pages 233–244, 1989.

[13] A. E. Nicholson and J. M. Brady. The data association problem when monitoring robot vehicles using dynamic belief networks. In *Proc. of the 10th European Conf. on Artificial Intelligence (ECAI-92)*, pages 689–693, 1992.

[14] A. E. Nicholson and J. M. Brady. Sensor validation using dynamic belief networks. In *Proceedings of the 8th Conference on Uncertainty in Artificial Intelligence*, pages 207–214, 1992.

[15] A. E. Nicholson and J. M. Brady. Dynamic belief networks for discrete monitoring. *IEEE Systems, Man and Cybernetics*, 24(11), 1994.

[16] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems.* Morgan Kaufmann, San Mateo, Ca., 1988.

[17] Gregory M. Provan. Tradeoffs in constructing and evaluating temporal influence diagrams. In *Proceedings of the 9th Conference on Uncertainty in Artificial Intelligence*, pages 40–47, 1993.

[18] David Pynadeth and Michael P. Wellman. Accounting for context in plan recognition, with application too traffic monitoring. In *Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence*, pages 472–481, 1995.

[19] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach.* Prentice-Hall, 1994.

[20] D. Spiegelhalter, R. Franklin, and K. Bull. Assessment criticism and improvement of imprecise subject probabilities for a medical expert system. In *Proc. of the Fifth Workshop on Uncertainty in Artificial Intelligence*, pages 335–342, 1989.

[21] Sampath Srinivas and Jack Breese. Ideal: Influence diagram evaluation and analysis in lisp. Technical Report Technical Memorandum No. 23, Rockwell International Science Center, 1989.