Probabilistic Reasoning

Unit # 8

SMILE API

- The API/wrapper is available for Java and .NET.
- The following commands are taken from GeNIe/SMILE official website.
- To create a network
  - Network net = new Network();
- Add a node named Success
  - net.AddNode(Network.NodeType.Cpt, “A");

SMILE API (Cont’d)

- Node A has two states: Success and Failure
  - net.SetOutcomeId(“A", 0, "Success");
  - net.SetOutcomeId(“A", 1, "Failure");
- Create an arc between Nodes A and B
  - net.AddArc(“A", “B");
- Assign probabilities
  - double[] aSuccessDef = {0.2, 0.8};
  - net.SetNodeDefinition(“A", aSuccessDef);
- Save the model
  - net.WriteFile("tutorial_a.xdsl");

SMILE API (Cont’d)

- Update probabilities
  - net.UpdateBeliefs();
- Set evidence
  - net.SetEvidence(“B", "Good");
- Get posterior probabilitiy
  - aValues = net.GetNodeValue("Success");
  - double P_SuccIsSuccGivenForeIsPoor = aValues[outcomeIndex];
  - Console.WriteLine("P(\"Success\" = Success | \"Forecast\" = Poor) = " + P_SuccIsSuccGivenForeIsPoor);

Stochastic Process

- A stochastic process is defined to be an indexed collection of random variables $\{X_t\}$, where the index t runs through a given set T.
- For example, X_t might represent the inventory level of a particular product at the end of week t.

Markovian Property

- A stochastic process {Xt} is said to have the Markovian property if $P(X_{i+1} = j \mid X_0 = k_0, X_1 = k_1, \ldots\ldots, X_{t-1} = k_{t-1}, X_i = i\} = P(X_{i+1} = j \mid X_i = i\}$, for t=0, 1, … and every sequence I, j, k_0, k_1, ….., k_{t-1}.
- This Markovian property says that the conditional probability of any future "event" given any past "events" and the present state $X_t = i$, is independent of the past events and depends only upon the present state.

Markov Chain

- A stochastic process $\{X_t\}$ (t = 0, 1, ….) is a Markov chain if it has the Markovian property.

- The conditional probabilities $P\{X_{t+1} = j \mid X_t = i\}$ for a Markov chain are called (one-step) transition probabilities.
- If for each i and j

$$P\{X_{t+1} = j \mid X_t = i\} = P(X_1 = j \mid X_0 = i\}$$

- Then the (one-step) transition probabilities are said to be stationary. Thus, having stationary transition probabilities implies that the transition probabilities do not change over time.

Weather Example

- Xt = 0 if day t is dry
- = 1 if day t has rain
- P{Xt+1 = 0 | Xt = 0} = 0.8
- P{Xt+1 = 0 | Xt = 1} = 0.6
- P = $\begin{pmatrix} p_{00} & p_{01} \\ p_{10} & p_{11} \end{pmatrix} = \begin{pmatrix} 0.8 & 0.2 \\ 0.6 & 0.4 \end{pmatrix}$

Stock Example

- At the end of a given day, the price of a stock is recorded. If the stock has gone up, the probability that it will go up tomorrow is 0.7. If the stock has gone down, the probability that it will go up tomorrow is 0.5.
- This is a Markov chain, where the possible states for each day are as follows:
  - State 0: the stock increased on this day
  - State 1: the stock decreased on this day
- P = $\begin{pmatrix} 0.7 & 0.3 \\ 0.5 & 0.5 \end{pmatrix}$

Gambling Example

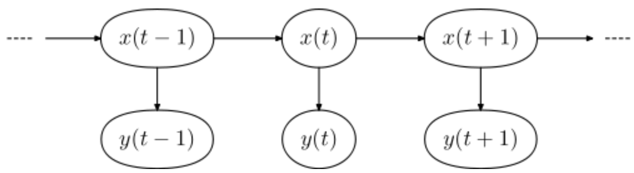
- Suppose that a player has \$1 and with each play of the game wins \$1 with probability p > 0 or loses \$1 with probability 1 – p. The game ends when the player either accumulates \$3 or goes broke.
- This game is a Markov chain with the states representing the player’s current holding of money, that is, 0, \$1, \$2, or \$3.
- P = ?

Chapman Kolmogorov Equation

- The n-step transition probability matrix $P^{(n)}$ can be obtained by computing the nth power of the one-step transition matrix P.
- For the weather example,
- $P^{(2)} = P \cdot P = \left(\begin{pmatrix} 0.8 & 0.2 \\ 0.6 & 0.4 \end{pmatrix}\right)\left(\begin{pmatrix} 0.8 & 0.2 \\ 0.6 & 0.4 \end{pmatrix}\right) = \left(\begin{pmatrix} 0.76 & 0.24 \\ 0.72 & 0.28 \end{pmatrix}\right)$
- If the weather is in state 0 (dry) on a particular day, the probability of being in state 0 two days later is 0.76. Similarly, if the weather is in state 1 now, the probability of being in state 0 two days later is 0.72.

Hidden Markov Model

- A hidden Markov model (HMM) is a Markov model in which the system being modeled is assumed to be a Markov process with unobserved (hidden) states.
- An HMM can be considered as the simplest dynamic Bayesian network.



Hidden Markov Model (Cont'd)

- In simpler Markov models (like a Markov chain), the state is directly visible to the observer, and therefore the state transition probabilities are the only parameters.
- In a hidden Markov model, the state is not directly visible, but output, dependent on the state, is visible.
- Each state has a probability distribution over the possible output tokens. Therefore the sequence of tokens generated by an HMM gives some information about the sequence of states.
- Note that the adjective 'hidden' refers to the state sequence through which the model passes, not to the parameters of the model; even if the model parameters are known exactly, the model is still 'hidden'.

Application of DBN in Anti-Money Laundering (Saleha and Sajjad)

- Suspicious activity reporting (SAR) has been a crucial part of anti-money laundering (AML) systems.
- Financial transactions are considered suspicious when they deviate from the regular behavior of their customers.
- Money launderers pay special attention to keep their transactions as normal as possible to disguise their illicit nature.

Application (Cont'd)

- This study presents an approach, called SARDBN (Suspicious Activity Reporting using Dynamic Bayesian Network), that employs a combination of clustering and dynamic Bayesian network (DBN) to identify anomalies in sequence of transactions.
- SARDBN applies DBN to capture patterns in a customer's monthly transactional sequences as well as to compute an anomaly index

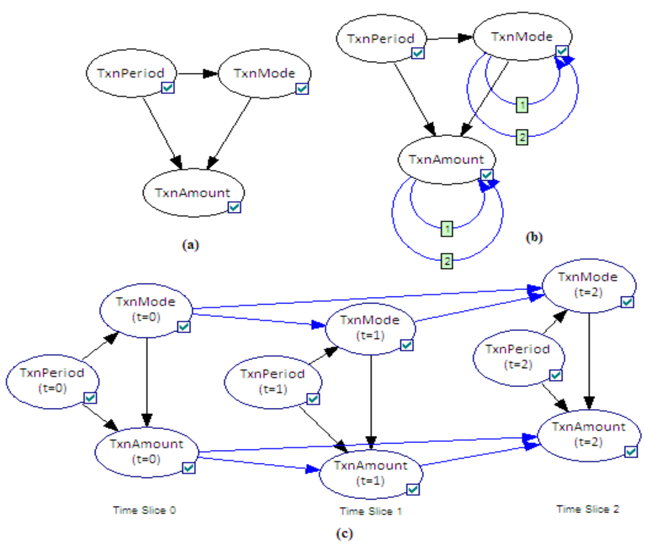
Application (Cont'd)

- The first step is to form clusters of customers that exhibit similar transactional pattern.
- The similarity in the transactional behavior is assessed by a customer's average monthly credit and debit amounts, frequency of credit and debit transactions, and delay in two consecutive transactions.

Application (Cont'd)

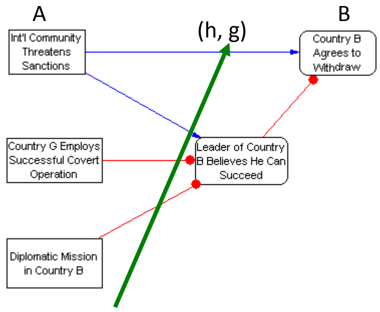
- Once clustering is performed, the next step is to learn the parameters of a DBN for each cluster.
- The structure of these DBNs can vary in different scenarios and is devised with the aid of subject matter experts.
- For this study, the three variables under consideration are transaction amount (TxnAmount), mode of transaction (TxnMode) (e.g. cheque withdrawal/deposit, ATM withdrawal, salary transfer, POS payment, bank draft), and period of transaction (TxnPeriod) (i.e. start/middle/end of month).

Application (Cont'd)



Influence Nets

- A set of random variables that makes up the nodes of an IN. All the variables in the IN have binary states.
- A set of directed links that connect pairs of nodes.
- Each link has associated with it a pair of parameters that shows the causal strength of the link (usually denoted as h and g values).



h is Influence of A on B: Analogous to P(B | A)

g is Influence of ¬A on B: Analogous to P(B | ¬A)

Positive Impact

Negative Impact

Root Nodes

Non-root Nodes

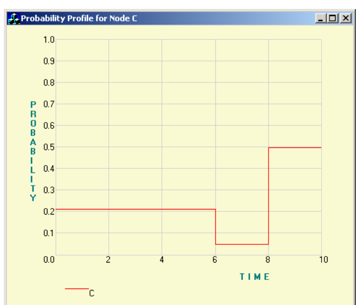
Timed Influence Net

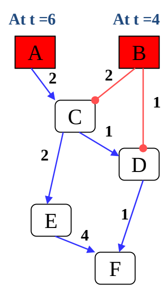
The specification of a TIN require the following additional parameters besides the one required for by an ordinary IN:

A time delay is associated with each arc.

A time delay is associated with each node.

Each actionable event is assigned time stamp(s) at which the decision(s) regarding the state of that action is(are) made



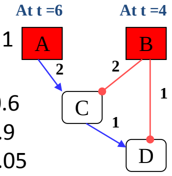


TIN Exercise

- At what time stamps, the probabilities of C and D is changed?
  - C @ {6, 8}
  - D @ {5, 7, 9}
- Compute probabilities of C & D at each of the time stamp mentioned above.

P(A) = 0.1

P(B) = 0.3

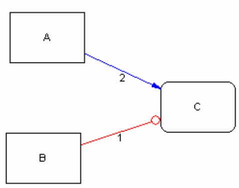


P(C|A,B) =0.6
P(C|A,~B) =0.9
P(C|~A,B) =0.05
P(C|~A,~B)=0.1

P(D|B, C) =0.7
P(D|B,~C) =0.9
P(D|~B,C) =0.05
P(D|~B,~C)=0.1

Time-Varying Influence (Persistence of Influence)

- Current implementation of TINs models time-invariant influences
- A scheme is proposed for modeling time-varying influences
- A list of influences along with their time of effect is specified for each arc in a TIN
- The proposed scheme can be used to model time-dependent structural changes in a TIN



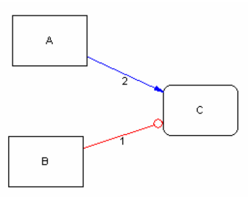
Influence of A on C when information at A is t time units old
Strong: $2 \leq t < 4$
Moderate: $4 \leq t \leq 6$
Low: $t > 6$

Influence of B on C when information at B is t time units old
Strong: $1 \leq t < 3$
Low: $t > 3$

Time-Varying Influence (Cont'd)

P(A) = 0.05 @ 0
= 1.0 @ 4

P(B) = 0.1 @ 0
= 0.6 @ 7
= 1.0 @ 10



Influence of A on C when information at A is t time units old
Strong: 2 ≤ t < 4
Moderate: 4 ≤ t ≤ 6
Low: t > 6

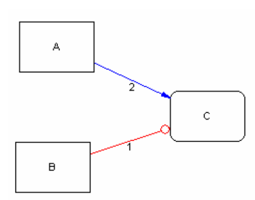
Influence of B on C when information at B is t time units old
Strong: 1 ≤ t < 3
Low: t > 3

- C is updated at time: 6, 8, 11
- At time 6: P(A) @ 4 is used, while P(B) @ 0 is used
  - Information coming from A is 2 time units old
  - Information coming from B is 6 time units old
    - C has strong influence of A and low influence of B at time 6
- At time 8: P(A) @ 4 is used, while P(B) @ 7 is used
  - Information coming from A is 4 time units old
  - Information coming from B is 1 time units old
    - C has moderate influence of A and strong influence of B at time 8

Non-Stationary Conditional Probabilities

P(A) = 0.05 @ 0
= 1.0 @ 4

P(B) = 0.1 @ 0
= 0.6 @ 7
= 1.0 @ 10



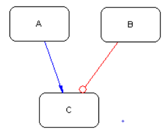
Influence of A on C when information at A is t time units old
Strong: 2 ≤ t < 4
Moderate: 4 ≤ t ≤ 6
Low: t > 6

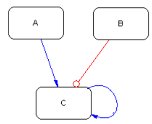
Influence of B on C when information at B is t time units old
Strong: 1 ≤ t < 3
Low: t > 3

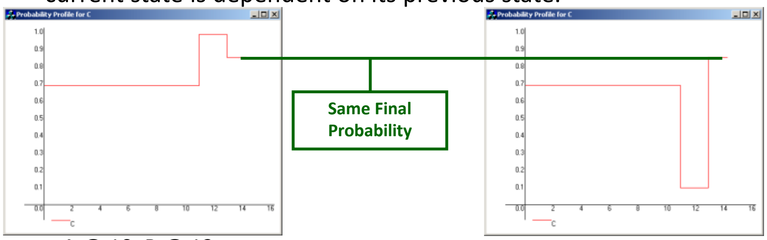
		Time	
Parents Combination	**6**	**8**	**11**
P(C|¬A,¬B)	**0.07**	**0.85**	**0.93**
P(C|¬A, B)	**0.03**	**0.02**	**0.03**
P(C| A,¬B)	**0.97**	**0.98**	**0.97**
P(C| A, B)	**0.93**	**0.15**	**0.07**

Adding Memory to the Nodes in a TIN

- Current implementation of TINs assume that the nodes are memoryless
  - The impact of different sequences of actions on the final probability is not captured.
- An approach is proposed that adds memory to the nodes in a TIN
  - A self-loop is added to each node whose current state is dependent on its previous state.



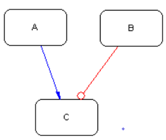


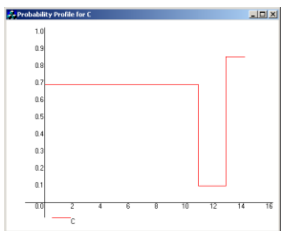


A @ 10, B @ 12

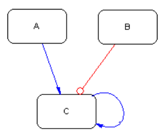
B @ 10, A @ 12

Adding Memory to the Nodes in a TIN (Cont'd)

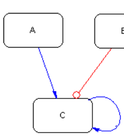




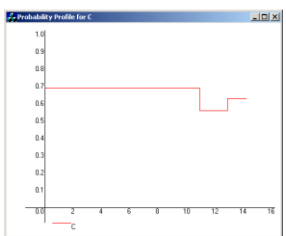
B @ 10, A @ 12

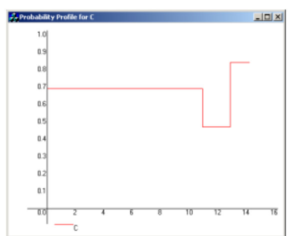


(0.33, -0.33)
Weak Memory



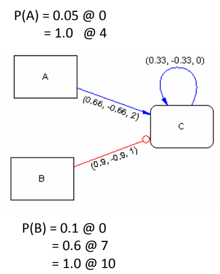
(0.90, -0.90)
Strong Memory





Dynamic Influence Nets

- Timed Influence Nets with
  - Time-varying influences
  - Memory represented by a self-loop

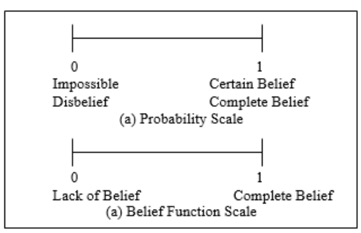


Influence of A on C when information at A is t time units old
Strong: $2 \leq t < 4$
Moderate: $4 \leq t \leq 6$
Low: $t > 6$

Influence of B on C when information at B is t time units old
Strong: $1 \leq t < 3$
Low: $t > 3$

Dempster-Shafter Theory of Belief Functions

- Like probability theory, Dempster-Shafer theory of belief functions is a calculus to represent and reason uncertainty present in a problem domain.
- The main difference between the two approaches lies in the way they handle ignorance.



Basic Constructs

- Support: The quantitative weight assigned to each subset is referred to as support or basic probability assignment (bpa).
- Belief: For any non-empty subset X; Bel{X} indicates the total support assigned to X and to any of its non-empty subsets. For singleton elements, support{X} = Believe{X}
- Doubt: For any set X, Dou{X} is related to the extent X is disbelieved or X^c is believed. Formally Dou{X}=Bel{X^c}.
- Plausibility: Also referred to as the upper probability, this term indicates the degree to which a modeler fails to doubt X.

Examples

	Y0	Y1	Y2	Y0,Y1	Y0,Y2	Y1,Y2	Y0,Y1,Y2	{}
Support	0.2	0.4	0.1	0.1	0.2	0.0	0.0	0.0
Belief	0.2	0.4	0.1	0.7	0.5	0.5	1.0	0.0
Doubt	0.5	0.5	0.7	0.1	0.4	0.2	0.0	1.0
Plausibility	0.5	0.5	0.3	0.9	0.6	0.8	1.0	0.0

Hypothesis	Mass	Belief	Plausibility
Null (neither alive nor dead)	0	0	0
Alive	0.2	0.2	0.5
Dead	0.5	0.5	0.8
Either (alive or dead)	0.3	1.0	1.0

Combining Evidence

- Suppose there are 100 cars in a parking lot consisting of type A and B.
- Two policemen check the type of cars in the lot.
- First policeman says that there are 30 A cars and 20 B cars among the 50 cars he counted.
- Second policeman says that there are 20 A cars and 20 cars that could A or B.