Uncertainty

S. Sheerazuddin

May 5, 2013

1 Bayesian Networks

Bayesian networks (BN), also known as **belief networks** (or **Bayes nets** for short), belong to the family of probabilistic graphical models (GMs). These graphical structures are used to represent knowledge about an uncertain domain. In particular, each node in the graph represents a random variable, while the edges between the nodes represent probabilistic dependencies among the corresponding random variables. These conditional dependencies in the graph are often estimated by using known statistical and computational methods. Hence, Bayesian Networks combine principles from graph theory, probability theory, computer science, and statistics.

Graphical Models with undirected edges are generally called **Markov random fields** or **Markov networks**. These networks provide a simple definition of independence between any two distinct nodes based on the concept of a Markov blanket. Markov networks are popular in fields such as statistical physics and computer vision.

Bayesian Networks correspond to another GM structure known as a directed acyclic graph (DAG) that is popular in the statistics, the machine learning, and the artificial intelligence societies. Bayesian Networks are both mathematically rigorous and intuitively understandable. They enable an effective representation and computation of the joint probability distribution (JPD) over a set of random variables.

The structure of a DAG is defined by two sets: the set of nodes (vertices) and the set of directed edges. The nodes represent random variables and are drawn as circles labeled by the variable names. The edges represent direct dependence among the variables and are drawn by arrows between nodes. In particular, an edge from node X_i to node X_j represents a statistical dependence between the corresponding variables. Thus, the arrow indicates that a value taken by variable X_j depends on the value taken by variable

X_i , or roughly speaking that variable X_i "influences" X_j . Node X_i is then referred to as a **parent** of X_j and, similarly, X_j is referred to as the child of X_i . An extension of these genealogical terms is often used to define the sets of "descendants"– the set of nodes that can be reached on a direct path from the node, or "ancestor" nodes – the set of nodes from which the node can be reached on a direct path. The structure of the acyclic graph guarantees that there is no node that can be its own ancestor or its own descendant. Such a condition is of vital importance to the factorization of the joint probability of a collection of nodes as seen below. Note that although the arrows represent direct causal connection between the variables, the reasoning process can operate on Bayesian Networks by propagating information in any direction.

A Bayesian Network reflects a simple conditional independence statement. Namely that each variable is independent of its nondescendants in the graph given the state of its parents. This property is used to reduce, sometimes significantly, the number of parameters that are required to characterize the JPD of the variables. This reduction provides an efficient way to compute the posterior probabilities given the evidence.

In addition to the DAG structure, which is often considered as the "qualitative" part of the model, one needs to specify the "quantitative" parameters of the model. The parameters are described in a manner which is consistent with a Markovian property, where the conditional probability distribution (CPD) at each node depends only on its parents. For discrete random variables, this conditional probability is often represented by a table, listing the local probability that a child node takes on each of the feasible values – for each combination of values of its parents. The joint distribution of a collection of variables can be determined uniquely by these local conditional probability tables (CPTs).

Following the above discussion, a more formal definition of a Bayesian Network can be given. A Bayesian network B is an annotated acyclic graph that represents a JPD over a set of random variables V. The network is defined by a pair $B = (G, \Theta)$, where G is the DAG whose nodes $X_1, X_2, \cdots, X_n$ represents random variables, and whose edges represent the direct dependencies between these variables. The graph G encodes independence assumptions, by which each variable X_i is independent of its nondescendants given its parents in G. The second component Θ denotes the set of parameters of the network. This set contains the parameter $\theta_{x_i|\pi_i} = P_B(x_i|\pi_i)$ for each realization x_i of X_i conditioned on π_i , the set of parents of X_i in G. Accordingly, B defines a unique JPD over V, namely:

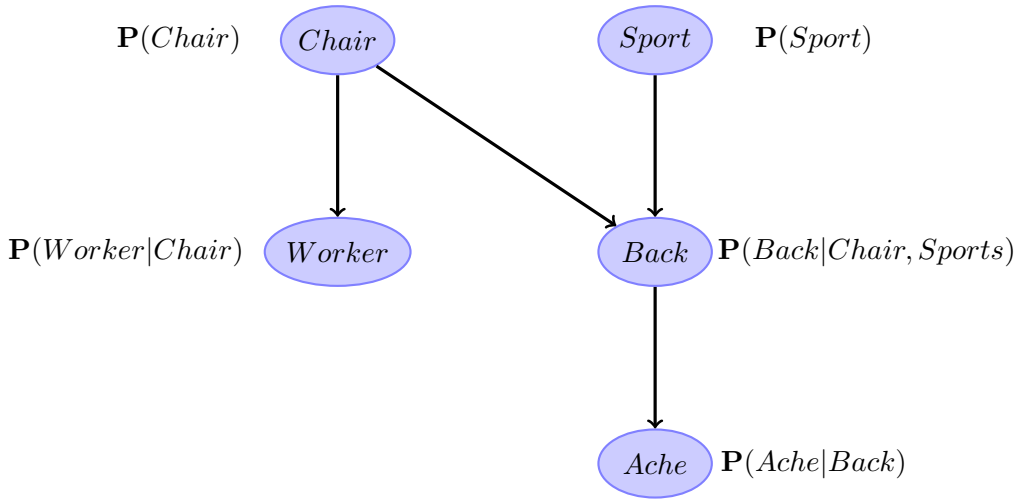


Figure 1: Backache Bayesian Network

$$P_B(X_1, X_2, \cdots, X_n) = \prod_{i=1}^{n} P(X_i|parents(X_i)) = \prod_{i=1}^{n} P(X_i|\pi_i)$$

For simplicity of representation we omit the subscript B henceforth. If X_i has no parents, its local probability distribution is said to be unconditional, otherwise it is conditional. If the variable represented by a node is observed, then the node is said to be an **evidence** node, otherwise the node is said to be **hidden** or **latent**.

Consider the following example that illustrates some of the characteristics of Bayesian Networks. The example shown in Figure 1 has a similar structure to the classical "earthquake" example given by Judea Pearl as shown in the Figure 2. The backache example considers a person who might suffer from a back injury, an event represented by the variable *Back* (denoted by B). Such an injury can cause a backache, an event represented by the variable *Ache* (denoted by A). The back injury might result from a wrong sport activity, represented by the variable *Sport* (denoted by S) or from new uncomfortable chairs installed at the persons office, represented

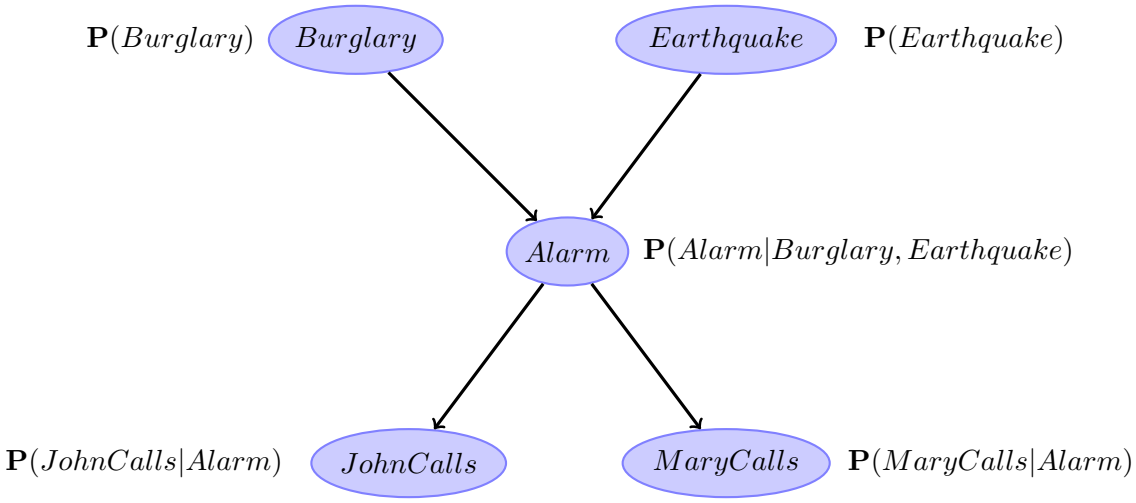


Figure 2: Earthquake Alarm Bayesian Network

by the variable $Chair$ (denoted by C). In the latter case, it is reasonable to assume that a coworker will suffer and report a similar backache syndrome, an event represented by the variable $Worker$ (denoted by W). All variables are binary; thus, they are either true (denoted by T) or false (denoted by F).

In this example the parents of the variable $Back$ are the nodes $Chair$ and $Sport$. The child of $Back$ is $Ache$, and the parent of $Worker$ is $Chair$. Following the Bayesian Network independence assumption, several independence statements can be observed in this case. For example, the variables Chair and Sport are marginally independent, but when $Back$ is given they are conditionally dependent. This relation is often called explaining away. When $Chair$ is given, $Worker$ and $Back$ are conditionally independent. When $Back$ is given, $Ache$ is conditionally independent of its ancestors $Chair$ and $Sport$.

Each node X in the Bayesian network is labelled by a conditional probability distribution $\mathbf{P}(X|parents(X))$ which can be depicted as a table called **conditional probability table** (CPT). For example $\mathbf{P}(Back|Chair, Sports)$ as a table is given as follows:

C	S	$P(B=T\|C,S)$	$P(B=F\|C,S)$
T	T	0.9	0.1
T	F	0.2	0.8
F	T	0.9	0.1
F	F	0.001	0.999

The other CPTs for the nodes in the Backache example are

$\mathbf{P}(A|B)$ as a table is

B	$P(A=T\|B)$	$P(A=F\|B)$
T	0.7	0.3
F	0.1	0.9

$\mathbf{P}(W|C)$ as a table is

B	$P(W=T\|C)$	$P(W=F\|C)$
T	0.9	0.1
F	0.01	0.09

$\mathbf{P}(C)$ as a table is

$P(C=T)$	$P(C=F)$
0.8	0.2

$\mathbf{P}(S)$ as a table is

$P(S=T)$	$P(S=F)$
0.02	0.98

The conditional independence statement of the Bayesian Network provides a compact factorization of the JPDs. Instead of factorizing the joint distribution of all the variables by the chain rule, i.e.,

$$P(C,S,W,B,A) = P(C)P(S|C)P(W|S,C)P(B|W,S,C)P(A|B,W,S,C),$$

the Bayesian Network defines a unique JPD in a factored form, i.e.

$$P(C,S,W,B,A) = P(C)P(S)P(W|C)P(B|S,C)P(A|B).$$

Note that the factored form (which is unique) depends on the topology of the Bayesian network. For a different topology, with the same set of random variables, the factored form would be different.

Also, note that the Bayesian Network form reduces the number of the model parameters, which belong to a multinomial distribution in this case, from $2^5 - 1 = 31$ to 10 parameters. Such a reduction provides great benefits from inference, learning (parameter estimation), and computational perspective. Thus, Bayesian network provides a substantially more efficient representation of a belief state compared to joint probability distribution.

2 Inference via Bayesian Network

Given a Bayesian Network that specified the JPD in a factored form, one can evaluate all possible inference queries by marginalization, *i.e.*, summing out over 'irrelevant' variables. Two types of inference support are often considered: predictive support for node X_i , based on evidence nodes connected to X_i through its parent nodes (also called top-down reasoning), and diagnostic support for node X_i , based on evidence nodes connected to X_i through its children nodes (also called bottom-up reasoning).

Note that even for the binary case, the JPD has size $O(2^n)$, where n is the number of nodes. Hence, summing over the JPD takes exponential time. In general, the full summation (or integration) over discrete (continuous) variables is called exact inference and known to be an NP-hard problem. Some efficient algorithms exist to solve the exact inference problem in restricted classes of networks. One of the most popular algorithms is the message passing algorithm that solves the problem in $O(n)$steps (linear in the number of nodes) for polytrees (also called singly connected networks), where there is at most one path between any two nodes. The algorithm was extended to general networks by Lauritzen and Spiegelhalter. Other exact inference methods include the cycle-cutset conditioning and variable elimination.

2.1 Approximate Inference in Bayesian Networks

Approximate inference methods were also proposed in the literature, such as Monte Carlo sampling that gives gradually improving estimates as sampling proceeds. A variety of standard Markov chain Monte Carlo (MCMC) methods, including the Gibbs sampling and the Metropolis-Hastings algorithm, were used for approximate inference. Other methods include the loopy belief propagation and variational methods that exploit the law of large numbers to approximate large sums of random variables by their means.

3 Other Approaches to Uncertain Reasoning

The earliest expert systems of the 1970s ignored uncertainty and used strict logical reasoning. Soon, it became clear that such an approach is unsuitable for most real-world domains. The next generation of expert systems, especially in medical domains, used probabilistic techniques. Unfortunately, they do not scale up because exponential number of probabilities are required in a full joint distribution. As a result, a variety of alternatives to probability were tried by AI theorists. We briefly discuss a few of them.

3.1 Rule-based Approaches for Uncertain Reasoning

Logical systems in general, and logic rule-based systems in particular, have three desirable properties:

Locality: In logical systems, whenever we have a rule of the form $A \rightarrow B$, we can conclude B given evidence A, *without worrying about other rules.*

Detachment: Once a logical proof is found for a proposition B, the proposition can be used regardless of how it was derived. That is, the proposition B can be *detached* from its justification.

Truth-functionality: In logic, the truth of complex sentences can be computed from the truth of its components.

Note that, these properties may not be true for probabilistic (uncertain) systems. In rule-based systems, degrees of belief are attached to propositions and rules. Thereafter, purely local schemes are described for combining and propagating those degrees of belief. These schemes are truth functional in the sense that degree of belief in $A \vee B$ is a function of the belief in A and belief in B.

Unfortunately, rule-based systems with properties of locality, detachment and truth-functionality are not suitable for uncertain reasoning.

3.2 Dempster-Schafer Theory

The **Dempster-Schafer** theory is designed to deal with the distinction between **uncertainty** and **ignorance**. Rather than computing the probability of a proposition, it computes the probability that *the evidence supports the proposition.* This measure of belief is called a **belief function**, denoted by $Bel(X)$ (probability that the evidence supports X).

Let us consider an example to illustrate belief functions. Suppose that a shady character comes up to you and offers to bet you \$10 that his coin will come up heads on the next flip. Given that the coin may or may not be fair, what belief should you ascribe to the event that it comes up heads? According to Dempster-Schafer theory, as we have no evidence either way, $Bel(Heads) = 0$ and $Bel(\neg Heads) = 0$. The Dempster-Schafer reasoning systems are by default skeptical of a proposition.

Now suppose that you have an expert at your disposal who testifies with 90% certainty that the coin is fair. Then, Dempster-Schafer theory gives $Bel(Head) = 0.9 \times 0.5 = 0.45$ and $Bel(\neg Head) = 0.9 \times 0.5 = 0.45$. There is still 10 percentage point gap that is not accounted by the evidence.

Dempster rule shows how to combine evidence to give new values for Bel, and Schafer's work extends the system into a complete computation model.

One interpretation of Dempster-Schafer theory is that it defines a probability interval: the interval for $Heads$ is $[0, 1]$ before the expert testimony (evidence) and $[0.45, 0.55]$ after. The width of the interval may help in deciding when and whether we need to acquire more evidence.

In Dempster-Schafer theory there is a problem in connecting beliefs to actions. It allows no definite action in many cases where probabilistic inference does yield a specific choice. In fact, the notion of utility in Dempster-Schafer theory is not yet well understood.

3.3 Fuzzy Sets and Fuzzy Logic

Fuzzy set theory is a means of specifying how well an object satisfies a vague description. For example, consider the proposition "Nate is tall". Is this true, if Nate is $5'\ 10''$? For Indians Nate is tall but for Americans Nate is not so tall. We are sure of Nate's height, what we are not sure about is the interpretation of the linguistic term "tall" as it does not refer to a sharp demarcation of objects into two classes– there are *degrees* of tallness. Fuzzy set theory treats $Tall$ as a fuzzy predicate and says that the truth value of $Tall(Nate)$ is a number between 0 and 1. The name "fuzzy set" derives from the interpretation of the predicate as implicitly defining a set of its members.

Fuzzy logic is a method of reasoning with logical expressions describing membership in fuzzy sets. For example, the complex sentence $Tall(Nate) \wedge Heavy(Nate)$ has a fuzzy truth value that is a function of the truth values of its components. The standard rules for evaluating the fuzzy truth T of a complex sentence are:

$T(A \wedge B) = min\{T(A), T(B)\}$
$T(A \vee B) = max\{T(A), T(B)\}$
$T(\neg A) = 1 - T(A)$

Fuzzy control is a methodology for constructing control systems in which the mapping between real-valued input and output parameters is represented by fuzzy rules. Fuzzy control has been very successful in commercial products such as automatic transmissions, video cameras and electric shavers.

Fuzzy predicates can be given probabilistic interpretation in terms if **random sets**–that is, random variables whose possible values are sets of objects. For example $Tall$ is a random set whose possible values are sets of

people. The probability $P(Tall = S_1)$, where S_1 is some particular set of people, is the probability that exactly that set would be identified as "tall" by an observer. Then, the probability that "Nate is tall" is the sum of probabilities of all the sets of which Nate is a member.

4 Temporal Models

Bayesian networks, which are just an efficient representation of joint probability distributions, are used to model knowledge about an uncertain world *which does not change with time*. In dynamic situations, where the state of the world changes with time we need to have richer models.

Diabetes Monitoring Example

Consider the problem of treating a diabetic patient. Here, we have measurable evidences such as recent insulin doses, food intake, blood sugar levels, and other physical ones like pulse rate etc. The task is to assess the current state of the patient (which is not directly observable), including the actual blood sugar level and insulin level. From the state information, the doctor has to make a decision about the patient's food intake and insulin dose.

In this case, the blood sugar levels and the measurements thereof can change rapidly over time, depending on many factors viz., recent food intake and insulin doses, metabolic activity, time of the day, and so on. Clearly, to assess the current state from the history of evidences and to predict the outcomes of treatment actions (future states), we must model these changes.

The basic approach in such models is as follows: the process of change can be viewed as a series of **snapshots**, each of which describes the state of the world at a particular time. Each snapshot, or **time slice** contains a set of random variables, some of which are observable and others not. For simplicity, we assume that same subset of variables is observable in each slice. We denote unobservable variables at time slice t by $\mathbf{X}_t$ and observable (evidence) variables by $\mathbf{E}_t$. The observation at time t is $\mathbf{E}_t = e_t$, for some set of values e_t.

For the diabetes example the set $\mathbf{X}_t$ contains state variables such as $BloodSugar_t$ and $StomachContents_t$ whereas the set $\mathbf{E}_t$ contains evidence variables $MeasuredBloodSugar_t$ and $PulseRate_t$. Note that actual blood sugar contents are not observable and may be different from the measured blood sugar contents. Similarly, the contents of stomach are also unseen but we can always observe the pulse rate.

Umbrella Example

Suppose there is a security guard at a secret underground installation who wants to know whether it is raining today. The only access of the security guard to the outer world occurs once each morning when he sees the director of the facility coming with or without an umbrella. For each day t, the set $\mathbf{E}_t$ (of observable variables) contains a single evidence variable U_t (whether the umbrella appears), and the set $\mathbf{X}_t$ (of unobservable variables) contains a single variable R_t (whether it is raining).

The interval between time slices depends on the problem in hand: for diabetes monitoring example t would be hours whereas for umbrella example t would most suitably be days. In any case, we assume that t is discrete.

We also assume that state sequence starts at $t = 0$ whereas the evidences start arriving at time $t = 1$. Hence, the umbrella world is represented by a **random (stochastic) process** containing state variables $R_0, R_1, \ldots$ and evidence variables $U_1, U_2, \ldots$.

Once we have decided the set of state and evidence variables for a given problem, we need to specify the dependencies among these variables. Clearly, we can divide these dependencies into two:

- **local** dependencies among the random variables of each time slice t, among $\mathbf{X}_t$ and $\mathbf{E}_t$ and
- **temporal (non-local)** dependencies among the random variables across the time slices t and $t-1$ or t and $t+1$, and so on.

As t is unbounded it creates two problems:

1. We may have to specify an unbounded number of conditional probability tables, one for each variable in each time slice t and
2. Each variable may involve an unbounded number of parents.

The first problem is solved by assuming that changes in the world state are caused by a **stationary process**—that is, a process of change that is governed by laws that themselves do not change over time. In umbrella world it means the conditional probability that umbrella appears, $\mathbf{P}(U_t|parents(U_t))$, is the same for all t. Therefore, we need to specify conditional distributions only for the variables within a "representative" slice. Clearly, this assumption is instrumental in specifying a finite description of the CPTs in the problem.

The second problem is solved by making a **Markovian assumption**—that is, the current state depends only on a *finite* history of previous states.

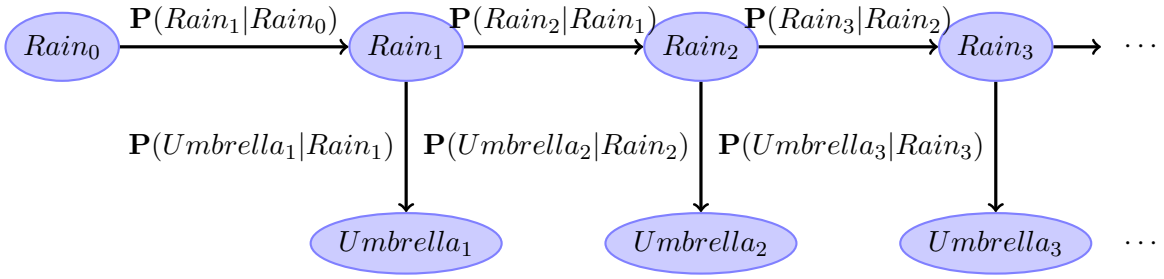


Figure 3: Bayesian Network for Umbrella World

Processes that satisfy this assumptions are called **Markov processes** or **Markov chains**. The simplest Markov chains are **first order Markov chains**, in which the current state depends only on the previous state and not on any earlier states. That is, for all t,

$$\mathbf{P}(\mathbf{X}_t|\mathbf{X}_{t-1}\dots\mathbf{X}_0) = \mathbf{P}(\mathbf{X}_t|\mathbf{X}_{t-1}).$$

In addition to restricting the parents of the state variables $\mathbf{X}_t$, we restrict the parents of the evidence variables $\mathbf{E}_t$ too. Typically, we assume that the evidence at time t depends only on the current state

$$\mathbf{P}(\mathbf{E}_t|\mathbf{X}_t\dots\mathbf{X}_0, \mathbf{E}_{t-1}\dots\mathbf{E}_0) = \mathbf{P}(\mathbf{E}_t|\mathbf{X}_t).$$

The conditional distribution $\mathbf{P}(\mathbf{X}_t|\mathbf{X}_{t-1})$ is called the **transition model** whereas the $\mathbf{P}(\mathbf{E}_t|\mathbf{X}_t)$ is called the **sensor model**. The transition model describes how the state evolves over time, whereas the sensor model describes how the sensors (evidence variables) are affected by the actual world.

In addition to the transition model and sensor model, we need to specify a prior probability distribution $\mathbf{P}(\mathbf{X}_0)$ over the states at time $t = 0$. These three distributions, combined with the conditional independence assertions, give a specification of the complete joint probability distribution over all the variables of the random process. For any finite t, we have

$$\mathbf{P}(\mathbf{X}_0, \mathbf{X}_1, \dots, \mathbf{X}_t, \mathbf{E}_1, \dots, \mathbf{E}_t) = \mathbf{P}(\mathbf{X}_0)\prod_{i=1}^{t}\mathbf{P}(\mathbf{X}_i|\mathbf{X}_{i-1})\cdot\mathbf{P}(\mathbf{E}_i|\mathbf{X}_i)$$

For the umbrella example, modelled by Bayesian network in the Figure 3, due to Markovian and stationary process assumptions we have

$$\mathbf{P}(R_1|R_0) = \mathbf{P}(R_2|R_1) = \mathbf{P}(R_3|R_2)\cdots$$

$$\mathbf{P}(U_1|R_1) = \mathbf{P}(U_2|R_2) = \mathbf{P}(U_3|R_3) \cdots$$

Therefore, the transition model may be

R_{t-1}	$P(R_t = T\|R_{t-1})$	$P(R_t = F\|R_{t-1})$
T	0.7	0.3
F	0.3	0.7

and the sensor model may be

R_t	$P(U_t = T\|R_t)$	$P(U_t = F\|R_t)$
T	0.9	0.1
F	0.2	0.8

Assuming a prior distribution $\mathbf{P}(R_0) = \langle 0.5, 0.5 \rangle$, we can compute the probability distribution of a belief state corresponding to $t = 2$ as follows:

$$\mathbf{P}(R_0, R_1, R_2, U_1, U_2) = \mathbf{P}(R_0) \prod_{i=1}^{2} \mathbf{P}(R_i|R_{i-1}) \cdot \mathbf{P}(U_i|R_i)$$

$$\mathbf{P}(R_0, R_1, R_2, U_1, U_2) = \mathbf{P}(R_0) \cdot \mathbf{P}(R_1|R_0) \cdot \mathbf{P}(R_2|R_1) \cdot \mathbf{P}(U_1|R_1) \cdot \mathbf{P}(U_2|R_2)$$

The probability for an actual state is as follows:

$$P(R_0 = T, R_1 = T, R_2 = T, U_1 = T, U_2 = T) = P(R_0 = T) \cdot P(R_1|R_0 = T) \cdot$$

$$P(R_2 = T|R_1 = T) \cdot P(U_1 = T|R_1 = T) \cdot P(U_2 = T|R_2 = T)$$

$$P(R_0 = T, R_1 = T, R_2 = T, U_1 = T, U_2 = T) = 0.5 \times 0.7 \times 0.7 \times 0.9 \times 0.9 = 0.198$$

On the same lines, we can compute the probabilities of other states for $t = 2$ which cumulatively would give us the probability distribution for the belief state for $t = 2$.

5 Inference in Temporal Models

The basic inference tasks in temporal models are:

- **Filtering** or **Monitoring**: This is the task of computing the **belief state**—the posterior probability distribution over the current state given all the evidence till date. That is, the probability distribution $\mathbf{P}(\mathbf{X}_t|e_1 \cdots e_t)$. In the umbrella example, this would mean computing the probability of rain today given all the observations of umbrella carrier made so far.

  A similar calculation provides the **likelihood** of the evidence sequence $\mathbf{P}(e_1 \cdots e_t)$.

- **Prediction**: This is the task of computing the posterior distribution over the future state given all evidence till date, $\mathbf{P}(\mathbf{X}_{t+k}|e_1 \cdots e_t)$, for some $k > 0$.

  In the umbrella example, this would mean computing the probability of rain, say three days from today given all the observations of umbrella carrier made so far.

- **Smoothing** or **Hindsight**: This is the task of computing the posterior distribution over a past state given all evidence till date, $\mathbf{P}(\mathbf{X}_k|e_1 \cdots e_t)$, for some $0 \leq k < t$.

  In the umbrella example, this would mean computing the probability of rain, say last wednesday given all the observations of umbrella carrier made so far.

- **Most Likely Explanation**: Given a sequence of observations, say $e_1 \cdots e_t$, we may wish to find the sequence of states, say $\mathbf{x}_1 \cdots \mathbf{x}_t$, that is most likely to have generated those observations. That is, we wish to compute

  $$\underset{\mathbf{x}_1 \cdots \mathbf{x}_t}{\mathbf{argmax}}\, P(\mathbf{x}_1 \cdots \mathbf{x}_t|e_1 \cdots e_t)$$

  For example, if the umbrella appears for first three days and is absent on the fourth day then the most likely explanation may be that it rained on the first three days and did not on the fourth day.

  Algorithms of this kind are useful in many applications, including speech recognition—where evidences are series of sounds and aim is to find the most likely sequence of words that produced those sounds—and the reconstruction of bit strings transmitted over a noisy channel.

An important point note is as follows: the direction of modelled dependencies in temporal models is from states to evidences and states to states—$X_t \to E_t$ and $X_{t-1} \to X_t$, whereas the direction of inference is from evidences to states.

We use general Bayes' rule and conditioning to describe these inference rules.

Generalized Bayes' Rule

$$\mathbf{P}(\mathbf{Y}|\mathbf{X}, e) = \frac{\mathbf{P}(\mathbf{X}|\mathbf{Y}, e) \cdot \mathbf{P}(\mathbf{Y}|e)}{\mathbf{P}(\mathbf{X}|e)}$$

Conditioning

The rule of **conditioning** is a variant of **marginalization**. The rule of marginalization uses joint probabilities as follows:

$$\mathbf{P}(\mathbf{Y}) = \sum_{\mathbf{z}} \mathbf{P}(\mathbf{Y}, \mathbf{z})$$

The rule of conditioning uses conditional probabilities as follows:

$$\mathbf{P}(\mathbf{Y}) = \sum_{\mathbf{z}} \mathbf{P}(\mathbf{Y}|\mathbf{z})P(\mathbf{z})$$

The conditioning rule has a generalized form as follows:

$$\mathbf{P}(\mathbf{Y}|\mathbf{X}) = \sum_{\mathbf{z}} \mathbf{P}(\mathbf{Y}|\mathbf{X}, \mathbf{z})\mathbf{P}(\mathbf{z}|\mathbf{X})$$

It can be derived as follows:

$$\mathbf{P}(\mathbf{Y}|\mathbf{X}) = \frac{\mathbf{P}(\mathbf{Y}, \mathbf{X})}{\mathbf{P}(\mathbf{X})} \quad \text{using product rule}$$

$$\mathbf{P}(\mathbf{Y}|\mathbf{X}) = \frac{\sum_{\mathbf{z}} \mathbf{P}(\mathbf{Y}, \mathbf{X}, \mathbf{z})}{\mathbf{P}(\mathbf{X})} \quad \text{using marginalization}$$

$$\mathbf{P}(\mathbf{Y}|\mathbf{X}) = \frac{\sum_{\mathbf{z}} \mathbf{P}(\mathbf{Y}|\mathbf{X}, \mathbf{z})\mathbf{P}(\mathbf{X}, \mathbf{z})}{\mathbf{P}(\mathbf{X})} \quad \text{using product rule}$$

$$\mathbf{P}(\mathbf{Y}|\mathbf{X}) = \frac{\sum_{\mathbf{z}} \mathbf{P}(\mathbf{Y}|\mathbf{X}, \mathbf{z})\mathbf{P}(\mathbf{z}|\mathbf{X})\mathbf{P}(\mathbf{X})}{\mathbf{P}(\mathbf{X})} \quad \text{using product rule}$$

$$\mathbf{P}(\mathbf{Y}|\mathbf{X}) = \sum_{\mathbf{z}} \mathbf{P}(\mathbf{Y}|\mathbf{X}, \mathbf{z})\mathbf{P}(\mathbf{z}|\mathbf{X})$$

Apart from conditioning and general Bayes' rule, we shall heavily use the following forms of conditional independence both of which are equivalent. These rules hold when R and B are conditionally independent with respect to Y. This means, given the knowledge of occurrence of Y, knowledge of occurrence of R does not give any information on the occurrence of B and vice versa.

$$\mathbf{P}(R, B|Y) = \mathbf{P}(R|Y) \cdot \mathbf{P}(B|Y) \quad \& \quad \mathbf{P}(R|B, Y) = \mathbf{P}(R|Y)$$

Furthermore, for simplicity, we describe the inference rules with probabilities, $P(x)$, instead of probability distributions $\mathbf{P}(X)$.

5.1 Filtering

Given the result of filtering up to time t, that is the values $P(x_1|e_1)$, $P(x_2|e_{1:2})$ upto $P(x_t|e_{1:t})$, we can compute the result for $t+1$, that is $P(x_{t+1}|e_{1:t+1})$.

The estimation of $P(x_{t+1}|e_1, e_2 \cdots e_{t+1})$ is done recursively. The base case is $P(x_1|e_1)$ which is computed using product rule and marginal summation as follows:

$$P(x_1|e_1) = \frac{P(x_1, e_1)}{P(e_1)}$$

$$P(x_1|e_1) = \frac{\sum_{x_0} P(x_0, x_1, e_1)}{P(e_1)}$$

$$P(x_1|e_1) = \frac{\sum_{x_0} P(x_0)P(x_1|x_0)P(e_1|x_1)}{P(e_1)}$$

$$P(x_1|e_1) = \frac{P(e_1|x_1)\sum_{x_0} P(x_0)P(x_1|x_0)}{P(e_1)}$$

$$P(x_1|e_1) = \frac{P(e_1|x_1)\sum_{x_0} P(x_0)P(x_1|x_0)}{\sum_{x_0,x_1} P(x_0, x_1, e_1)}$$

$$P(x_1|e_1) = \frac{P(e_1|x_1)\sum_{x_0} P(x_0)P(x_1|x_0)}{\sum_{x_0,x_1} P(x_0)P(x_1|x_0)P(e_1|x_1)}$$

The inductive case is computed as follows:

$$P(x_{t+1}|e_{1:t+1}) = P(x_{t+1}|e_{1:t}, e_{t+1}) = \frac{P(e_{t+1}|x_{t+1}, e_{1:t})P(x_{t+1}|e_{1:t})}{P(e_{t+1}|e_{1:t})}$$

$$P(x_{t+1}|e_{1:t}, e_{t+1}) = \alpha \cdot P(e_{t+1}|x_{t+1}, e_{1:t})P(x_{t+1}|e_{1:t}) \text{ where } \alpha = \frac{1}{P(e_{t+1}|e_{1:t})}$$

Now, $P(e_{t+1}|x_{t+1}, e_{1:t}) = P(e_{t+1}|x_{t+1})$, due to the Markov property of evidence. Note that, $P(e_{t+1}|x_{t+1})$ can be obtained directly from the sensor model. Furthermore, $P(x_{t+1}|e_{1:t})$ represents **one-step prediction** which can be obtained as follows:

$$P(x_{t+1}|e_{1:t}) = \sum_{x_t} P(x_{t+1}|x_t, e_{1:t}) \cdot P(x_t|e_{1:t}) \quad \text{conditioning over } x_t$$

Now, $P(x_{t+1}|x_t, e_{1:t}) = P(x_{t+1}|x_t)$, due to Markov property on the states. Note that, $P(x_{t+1}|x_t)$ can be obtained directly from the transition model. $P(x_t|e_{1:t})$ has already been computed recursively. Putting all of these together we get:

$$P(x_{t+1}|e_{1:t+1}) = \alpha \cdot P(e_{t+1}|x_{t+1}) \sum_{x_t} P(x_{t+1}|x_t) \cdot P(x_t|e_{1:t})$$

We think of filtered estimate $P(x_t|e_{1:t})$ as a message $\mathbf{f}_{1:t}$ that is propagated forward along the sequence modified by each transition and updated by each new observation. This forward propagation is denoted as

$$\mathbf{f}_{1:t+1} = \alpha \cdot FORWARD(\mathbf{f}_{1:t}, e_{t+1})$$

where $FORWARD$ implements the update described in the above equation.

5.2 Prediction

The task of prediction can simply be seen as filtering without the addition of new evidence. It turns out that this computation involves only the transition model and not the sensor model.

Assuming that we have already computed $P(x_t|e_{1:t})$ as the base case, $P(x_{t+k+1}|e_{1:t})$, for any $k \geq 0$, can be computed recursively using multiple conditioning as follows:

$$P(x_{t+k+1}|e_{1:t}) = \sum_{x_{t+k}} P(x_{t+k+1}|x_{t+k}, e_{1:t}) \cdot P(x_{t+k}|e_{1:t})$$

$$P(x_{t+k+1}|e_{1:t}) = \sum_{x_{t+k}} P(x_{t+k+1}|x_{t+k}) \cdot P(x_{t+k}|e_{1:t})$$

$$P(x_{t+k+1}|e_{1:t}) = \sum_{x_{t+k}} P(x_{t+k+1}|x_{t+k}) \sum_{x_{t+k-1}} P(x_{t+k}|x_{t+k-1}) \cdot P(x_{t+k-1}|e_{1:t})$$

$$P(x_{t+k+1}|e_{1:t}) = \sum_{x_{t+k}} P(x_{t+k+1}|x_{t+k}) \sum_{x_{t+k-1}} P(x_{t+k}|x_{t+k-1}) \cdots \sum_{x_t} P(x_{t+1}|x_t) \cdot P(x_t|e_{1:t})$$

Clearly, $P(x_{t+k+1}|x_{t+k}), P(x_{t+k}|x_{t+k-1}), \cdots, P(x_{t+1}|x_t)$ can be directly found from the transition model and sensor model is never used if $P(x_t|e_{1:t})$ has already been computed.

5.3 Likelihood

We can use the same technique, of forward recursion, to compute the likelihood of the evidence sequence, i.e., $P(e_{1:t})$.

$$P(e_{1:t}) = \sum_{x_t} P(x_t, e_{1:t})$$

$$P(e_{1:t}) = \sum_{x_t} P(x_t, e_{1:t-1}, e_t)$$

$$P(e_{1:t}) = \sum_{x_t} P(e_t|x_t, e_{1:t-1}) \cdot P(x_t, e_{1:t-1})$$

$$P(e_{1:t}) = \sum_{x_t} P(e_t|x_t) \cdot P(x_t, e_{1:t-1})$$

$$P(e_{1:t}) = \sum_{x_t} P(e_t|x_t) \cdot P(x_t|e_{1:t-1}) \cdot P(e_{1:t-1})$$

$$P(e_{1:t}) = P(e_{1:t-1}) \sum_{x_t} P(e_t|x_t) \cdot P(x_t|e_{1:t-1})$$

$$P(e_{1:t}) = P(e_{1:t-1}) \sum_{x_t} P(e_t|x_t) \sum_{x_{t-1}} \cdot P(x_t|x_{t-1}, e_{1:t-1}) \cdot P(x_{t-1}|e_{1:t-1})$$

$$P(e_{1:t}) = P(e_{1:t-1}) \sum_{x_t} P(e_t|x_t) \sum_{x_{t-1}} \cdot P(x_t|x_{t-1}) \cdot P(x_{t-1}|e_{1:t-1})$$

$$P(e_{1:t}) = P(e_{1:t-1}) \sum_{x_t} P(e_t|x_t) \sum_{x_{t-1}} \cdot P(x_t|x_{t-1}) \cdot \mathbf{f}_{1:t-1}$$

For any t and for any x_t, $P(e_t|x_t)$ can be found from the sensor model, and for any x_t and x_{t-1}, $P(x_t|x_{t-1})$ can be found from the transition model. $P(e_{1:t-1})$ and $\mathbf{f}_{1:t-1}$ can be recursively computed as they are of a lower complexity.

5.4 Smoothing

Smoothing is the process of computing distribution over past states given evidence upto the present; that is, $P(x_k|e_{1:t})$ for $1 \leq i < t$. This is done most conveniently in two parts—the evidence upto k and the evidence from $k+1$ to t.

$$P(x_k|e_{1:t}) = P(x_k|e_{1:k}, e_{k+1:t})$$

$$P(x_k|e_{1:t}) = \alpha \cdot P(x_k|e_{1:k}) \cdot P(e_{k+1:t}|x_k, e_{1:k}) \text{ where } \alpha = \frac{1}{P(e_{k+1:t}|e_{1:k})}$$

$$P(x_k|e_{1:t}) = \alpha \cdot \mathbf{f}_{1:k} \cdot P(e_{k+1:t}|x_k, e_{1:k})$$

We observe that evidences $e_{k+1:t}$ and $e_{1:k}$ are conditionally independent with respect to the state x_k. Therefore, we have

$$P(x_k|e_{1:t}) = \alpha \cdot \mathbf{f}_{1:k} \cdot P(e_{k+1:t}|x_k) = \alpha \cdot \mathbf{f}_{1:k} \cdot \mathbf{b}_{k+1:t}$$

We denote $P(e_{k+1:t}|x_k)$ by $\mathbf{b}_{k+1:t}$, "backward message" analogous to the forward message $\mathbf{f}_{1:k}$. It turns out that the backward message is computed by a recursive message that runs backward from t:

$$P(e_{k+1:t}|x_k) = \sum_{x_{k+1}} P(e_{k+1:t}|x_{k+1}, x_k) \cdot P(x_{k+1}|x_k)$$

Observing that $e_{k+1:t}$ and x_k are conditionally independent, also with respect to x_{k+1},

$$P(e_{k+1:t}|x_k) = \sum_{x_{k+1}} P(e_{k+1:t}|x_{k+1}) \cdot P(x_{k+1}|x_k)$$

$$P(e_{k+1:t}|x_k) = \sum_{x_{k+1}} P(e_{k+1}, e_{k+2:t}|x_{k+1}) \cdot P(x_{k+1}|x_k)$$

The observation e_{k+1} and the observation sequence $e_{k+2:t}$ are conditionally independent (with respect to x_{k+1}),

$$P(e_{k+1:t}|x_k) = \sum_{x_{k+1}} P(e_{k+1}|x_{k+1}) \cdot P(e_{k+2:t}|x_{k+1}) \cdot P(x_{k+1}|x_k)$$

$$\mathbf{b}_{k+1:t} = \sum_{x_{k+1}} P(e_{k+1}|x_{k+1}) \cdot \mathbf{b}_{k+2:t} \cdot P(x_{k+1}|x_k)$$

Note that $e_{k+1}|x_{k+1})$ is obtained from the sensor model whereas $P(x_{k+1}|x_k)$ from the transition model. Using the message notation, we have the following equation where $BACKWARD$ implements the update described above.

$$\mathbf{b}_{k+1:t} = BACKWARD(\mathbf{b}_{k+2:t}, e_{k+1:t})$$

5.5 Finding the Most Likely Sequence

Suppose $\{true, true, false, true, true\}$ is the umbrella sequence for the security guard's first 5 days on the job. What is the weather sequence most likely to explain these observations?

In the case of umbrella example there are 2^5 possible state sequences. We need to find the probability $P(R_1 = x_1, R_2 = x_2, R_3 = x_3, R_4 = x_4, R_5 = x_5 | U_1 = true, U_2 = true, U_3 = false, U_4 = true, U_5 = true)$ for each $x_1 x_2 x_3 x_4 x_5 \in \{true, false\}^5$, and find that sequence which has the largest corresponding probability value.

In general, given an observation sequence $e_1 e_2 \dots e_t$ we need to find the state sequence $x_1 x_2 \dots x_t$ such that the probability $P(x_1 x_2 \dots x_t | e_1 e_2 \dots e_t)$ is maximized.

$$\underset{x_1 \cdots x_t}{\mathbf{argmax}} P(x_1 \cdots x_t | e_1 \cdots e_t)$$

We describe the Viterbi algorithm which solves the most likely sequence problem without explicitly computing all 2^n probabilities.

Given the observations $e_1 e_2 \dots e_t$ and the transition and sensor model find the message term $m_{1:t}$ using dynamic programming as follows:

$$m_{1:1} = \mathbf{P}(X_1 | e_1)$$

$$m_{1:2} = \max_{x_1} \mathbf{P}(x_1 X_2 | e_1 e_2)$$

$$m_{1:3} = \max_{x_1 x_2} \mathbf{P}(x_1 x_2 X_3 | e_1 e_2 e_3)$$

$$m_{1:4} = \max_{x_1 x_2 x_3} \mathbf{P}(x_1 x_2 x_3 X_4 | e_1 e_2 e_3 e_4)$$

$$m_{1:5} = \max_{x_1 x_2 x_3 x_4} \mathbf{P}(x_1 x_2 x_3 x_4 X_5 | e_1 e_2 e_3 e_4 e_5)$$

In general,

$$m_{1:t} = \max_{x_1 \dots x_{t-1}} \mathbf{P}(x_1 \dots x_{t-1} X_t | e_1 \dots e_t)$$

The message terms are recursively related as follows, where $\alpha = \frac{1}{P(e_{1:t+1})}$ is the normalization constant,

$$m_{1:t+1} = \alpha \cdot \mathbf{P}(e_{t+1} | X_{t+1}) \cdot \max_{x_t} \left(\mathbf{P}(X_{t+1} | x_t) \cdot m_{1:t} \right)$$

The base term $m_{1:1}$ is computed exactly as we compute the base term in filtering. $m_{1:1} = \mathbf{P}(X_1 | e_1) = \langle P(x_1 | e_1), P(\neg x_1 | e_1) \rangle$, where we abbreviate

$X_1 = true$ by x_1 and $X_1 = false$ by $\neg x_1$ assuming that the domain of X's contain only two elements $\{true, false\}$. $P(x_1|e_1)$ is computed as follows:

$$P(x_1|e_1) = \frac{P(x_1, e_1)}{P(e_1)}$$

$$P(x_1|e_1) = \frac{\sum_{x_0} P(x_0, x_1, e_1)}{P(e_1)}$$

$$P(x_1|e_1) = \frac{\sum_{x_0} P(x_0)P(x_1|x_0)P(e_1|x_1)}{P(e_1)}$$

$$P(x_1|e_1) = \frac{P(e_1|x_1)\sum_{x_0} P(x_0)P(x_1|x_0)}{P(e_1)}$$

$$P(x_1|e_1) = \frac{P(e_1|x_1)\sum_{x_0} P(x_0)P(x_1|x_0)}{\sum_{x_0,x_1} P(x_0, x_1, e_1)}$$

$$P(x_1|e_1) = \frac{P(e_1|x_1)\sum_{x_0} P(x_0)P(x_1|x_0)}{\sum_{x_0,x_1} P(x_0)P(x_1|x_0)P(e_1|x_1)}$$

$P(x_1|e_1)$ is computed exactly the same way as above. We just replace x_1 by $\neg x_1$ in the numerator in order to obtain the appropriate expression.

$$P(\neg x_1|e_1) = \frac{P(\neg x_1, e_1)}{P(e_1)} = \frac{P(e_1|\neg x_1)\sum_{x_0} P(x_0)P(\neg x_1|x_0)}{\sum_{x_0,x_1} P(x_0)P(x_1|x_0)P(e_1|x_1)}$$

For the umbrella example with five observations $\{true, true, false, true, true\}$ these message values are,

$$m_{1:1} = \langle .8182, .1818\rangle$$

$$m_{1:2} = \langle .5155, .0491\rangle$$

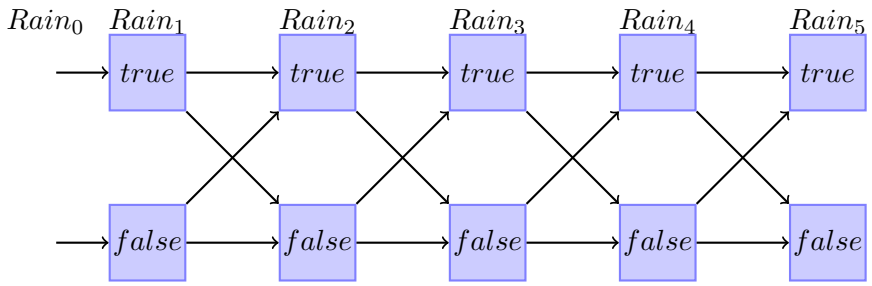
$$m_{1:3} = \langle .0361, .1237\rangle$$

$$m_{1:4} = \langle .0334, .0173\rangle$$

$$m_{1:5} = \langle .0210, .0024\rangle.$$

These values have been calculated using the following transition and sensor model. We also assume a prior distribution $\mathbf{P}(X_0) = \langle 0.5, 0.5\rangle$

All $2^5 = 32$ possible state sequences for $Rain_t$ viewed as paths through a graph.



Application of Viterbi Algorithm to umbrella observation sequence $\{T, T, F, T, T\}$.

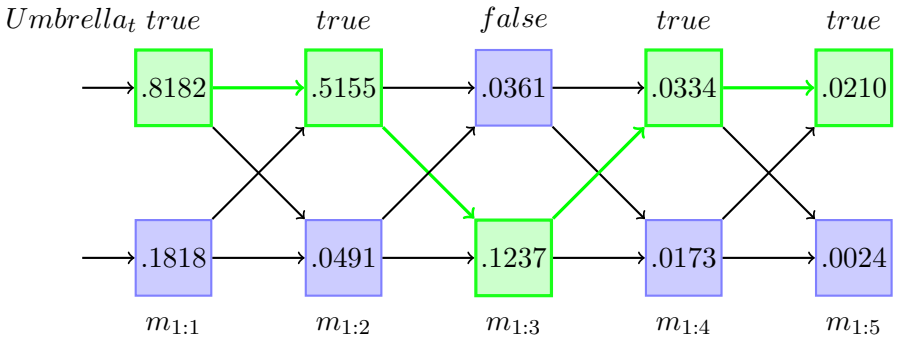


Figure 4: Finding most likely sequence using Viterbi algorithm

X_{t-1}	$P(X_t = T \mid X_{t-1})$	$P(X_t = F \mid X_{t-1})$
T	0.7	0.3
F	0.3	0.7

and the sensor model may be

X_t	$P(E_t = T \mid X_t)$	$P(E_t = F \mid X_t)$
T	0.9	0.1
F	0.2	0.8

These computations are pictorially described in the figure 4. At the end $m_{1:5}$ contains the probabilities for the most likely sequence reaching each of the final state x_5. We select that final state which gives the maximum value out of these probabilities and thus the most likely sequence overall is eventually found.

6 Hidden Markov Models

A **hidden Markov model** (HMM) is a **temporal probabilistic** model in which the state of the process is described by a **single discrete** random variable. Clearly, the possible values of the state variable are the possible states of the world. The umbrella example described previously is therefore an HMM with one state variable: R_t.

The restricted structure of HMMs allows for simple and elegant matrix implementation of all the basic (inference) algorithms. HMMs are used in speech recognition.

Consider a simple HMM with state variable X_t with domain $\{0, 1\}$ and evidence variable E_t with domain $\{0, 1\}$. The transition model for the HMM is a 2×2 matrix $\mathbf{T}$ where
$\mathbf{T}_{ij} = P(X_t = j \mid X_{t-1} = i) =$ probability of transition from state i to state j. The transition model can be written in matrix form as follows:

$$\mathbf{T} = \begin{pmatrix} P(\neg x_t \mid \neg x_{t-1}) & P(x_t \mid \neg x_{t-1}) \\ P(\neg x_t \mid x_{t-1}) & P(x_t \mid x_{t-1}) \end{pmatrix}$$

where, for any t, x_t means $X_t = 1$ and $\neg x_t$ means $X_t = 0$.

The sensor model can also be described as a matrix. This matrix is slightly different, because we fix the value of evidence variable E_t to be $e_t \in \{0, 1\}$ and enumerate the probabilities. For each time step t, we describe a diagonal matrix $\mathbf{O}_t$ where the diagonal entries are given by the values $P(e_t \mid X_t = i)$, $i \in \{0, 1\}$, and whose other entries are 0. In the matrix form,

$$\mathbf{O}_t = \begin{pmatrix} P(e_t \mid \neg x_t) & 0 \\ 0 & P(e_t \mid x_t) \end{pmatrix}$$

where, for any t, x_t means $X_t = 1$ and $\neg x_t$ means $X_t = 0$. Now, if we use column vectors to represent forward ($\mathbf{f}_{1:t}$) and backward ($\mathbf{b}_{k+1:t}$) messages, the computations become simple matrix-vector operations. The forward equation

$$\mathbf{P}(X_{t+1}|e_{1:t+1}) = \alpha \cdot \mathbf{P}(e_{t+1}|X_{t+1}) \sum_{x_t} \mathbf{P}(X_{t+1}|x_t) \cdot P(x_t|e_{1:t})$$

becomes

$$\mathbf{f}_{1:t+1} = \alpha \cdot \mathbf{O}_{t+1} \cdot \mathbf{T}^\top \mathbf{f}_{1:t}$$

and the backward equation

$$\mathbf{P}(e_{k+1:t}|X_k) = \sum_{x_{k+1}} P(e_{k+1}|x_{k+1}) \cdot P(e_{k+2:t}|x_{k+1}) \cdot \mathbf{P}(x_{k+1}|X_k)$$

becomes

$$\mathbf{b}_{k+1:t} = \mathbf{T} \cdot \mathbf{O}_{k+1} \cdot \mathbf{b}_{k+2:t}$$

Extra Reading

1. I have omitted discussion on basic probabilistic reasoning, in particular joint probability distribution and the relevant inference technique. You can read it from chapter 13 of the book.

2. The technique of exact inference in Bayesian network is in another set of notes which I am sending along with these two chapters.

3. The technique of approximate inference was not discussed in the class. You can find it in section 14.5, page 511, of the book.

4. Speech recognition using HMM was also not discussed in the class. You can find it in section 15.6, page 568, of the book.

5. Even though other approaches to uncertain reasoning like fuzzy logic etc. were not discussed in the class, you can find a section in the notes devoted to these topics. Do have a look at it.

6. I have seen questions on probabilistic reasoning in the Anna University examinations. It means nothing but joint probability distributions, Bayesian networks and the relevant inference techniques.