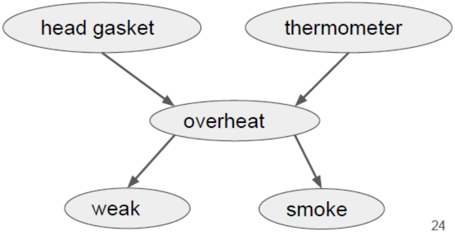
Oct 14 ~~Multivariate Probability~~

Oct21 Markov Models



Suppose we have the relevant conditional tables. How do we compute P(head gasket busted) *given* smoke?

Announcements and To-Dos

Announcements:

1. Practicum posted! Due November 7 ~~2~~, but I could push it back: but the next HW will be right on Nov 9 either way!

Last time we learned:

1. Bayes nets enumeration and calculations

Bayes Nets: Recap

Bayesian Networks wrapup:

- Bayes nets encode joint distributions as the product of local conditionals: $P(X_1 = x_1, X_2 = x_2, \ldots X_n = x_n) = \prod_{i=1}^{n} P(x_i|\text{parents}(X_i))$
- We compute specific probabilities and conditionals on a Bayesian network via *enumeration*: summing over all the possible values of the **unobserved** variables. For example, for our alarm network $(B \to A, E \to A \to (J, M))$, we had

  observed

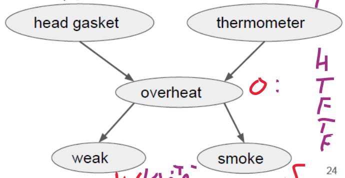
$$P(B|JM) = \alpha \sum_{E} \sum_{A} P(BJMAE)$$

- α is the inverse of the conditional probability denominator $P(JM) = P(JM\&B) + P(JM\&\neg B)$, so we can reuse our work to calculate it.

  numerator

Oct 14 Multivariate Probability

H : P F   T P F



O :

H	T	p(O)
T	T	p_1
F	T	p_2
T	F	p_3
F	F	p_4

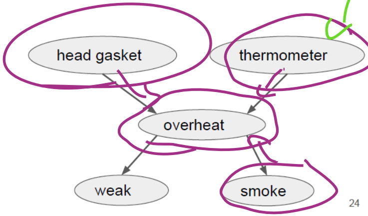
W \| O : T F   S

P(head gasket busted) *given* smoke?

$$P(H \mid S) = \frac{P(H, S)}{\boxed{P(S)}} = \alpha \cdot \underbrace{P(HS)}_{m,s,n \quad T,O,W}$$

P(mercury lung)

Oct 14 Multivariate Probability



hot
broken
cold

(joint)

marginal distribution

P(head gasket busted) *given* smoke?

$$P(H|S) = P(H\&S)/P(S) = \overset{P}{\alpha}(H\&S)$$

Then we enumerate over the other 3 outcomes, but we can skip weak since we actually only need all the parents of smoke!

Enumerate

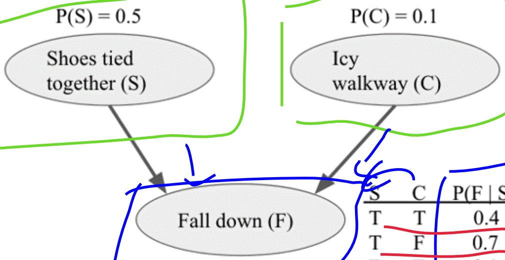
$$P(H\&S) = \alpha\sum_t\sum_o (H\&S\&T\&O) = P(H,T)\cdot P(O|H,T)\cdot P(S|O)$$

$$P(H,\, S) = \alpha P(H=h)\sum_t P(T=t)\sum_o P(O=o|H=h,\, T=t)P(S=s|O=o)$$

Sampling Example

simulate

Consider the network:



S	C	P(F | S, C)
T	T	0.4
T	F	0.7
F	T	0.1
F	F	0.01

→ np.random.choice

To *sample* on a Bayesian network, we move from the top down.

1. Sample from the priors. These are nodes without parents.
2. Sample from the conditionals. These are children given their parents.
3. Save a data frame with all the outcomes.

The data frame is holding *joint* outcomes, so whatever conditional probabilities we desire live in it's rows and just counting outcomes with Booleans!

Sampling Example

To *sample* on a Bayesian network, we move from the top down.

1. Sample from the priors. These are nodes without parents.
2. Sample from the conditionals. These are children given their parents.
3. Save a data frame with all the outcomes.

50-50 90-10

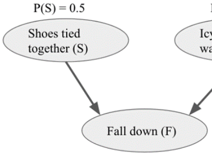
We may have a dfSamples:

Sample #	S	C	F
0	T	T	T
1	T	T	F
2	T	F	T
3	F	T	F
4	T	T	F
5	F	T	F
6	T	T	T
7	F	F	F
⋮	⋮	⋮	⋮

random function of S, C.

We can check $P(C = T|F = T) = \frac{P(C=T \textbf{ AND } F=T)}{P(F=T)}$ by counting the numbers of rows where each set of Booleans is true.

Enumeration Example



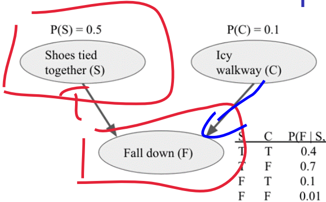
S	C	P(F | S, C)
T	T	0.4
T	F	0.7
F	T	0.1
F	F	0.01

p(shoes tied together | Not fell down)

$P(S=T \mid F=F)$

What is the fully enumerated probability $P(+S|-F)$?

Enumeration Example



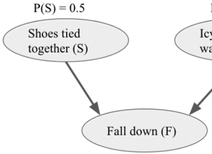
S	C	P(F | S, C)
T	T	0.4
T	F	0.7
F	T	0.1
F	F	0.01

What is the fully enumerated probability $P(+S|-F)$?

$$P(\overset{+}{S}|F) = P(\overset{+}{S} \textbf{ AND } \overset{-}{F})/P(F)$$ defn. cond prob

$$P(S|F) = \underset{50.50}{\alpha} P(S)P(F|S)$$

Enumeration Example



S	C	P(F | S, C)
T	T	0.4
T	F	0.7
F	T	0.1
F	F	0.01

What is the fully enumerated probability $P(+S|-F)$?

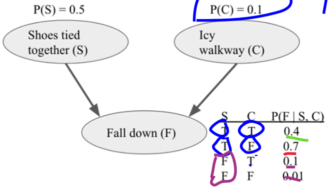
$$P(S|F) = P(S \textbf{ AND } F)/P(F)$$
$$P(S|F) = \alpha P(S)P(F|S)$$

P(F sum S)

Law of total prob

$$P(S|F) = \alpha P(S)[P(F|(S \textbf{ AND } +C) + P(F|(S \textbf{ AND } \neg C)]$$
$$P(S|F) = \alpha P(S)\sum_c P(F|SC)P(C)$$

Enumeration Example



T : .1
F : .9

What is the fully enumerated probability $P(+S|-F)$?

$$P(S|F) = P(S\ \textbf{AND}\ F)/P(F)$$
$$P(S|F) = \alpha P(S)P(F|S)$$
$$P(S|F) = \alpha P(S)\left[P(F|(S\ \textbf{AND}\ +C) + P(F|(S\ \textbf{AND}\ \neg C)\right]$$
$$P(S|F) = \alpha P(S)\textstyle\sum_I P(F|SC)P(C)$$
$$P(+S|\neg F) = \alpha P(+S)\textstyle\sum_C P(\neg F|+S,C)P(C)$$
$$P(+S|\neg F) = \alpha .5\,(.1\cdot .4 + .9\cdot .7)$$
$$P(-S|\neg F) = \alpha P(-S)\textstyle\sum_C P(\neg F|-S,C)P(C)$$
$$P(+S|\neg F) = \alpha .5\,(.1\cdot .1 + .9\cdot .01)$$

$\alpha\ .5(.1\cdot.4+.9\cdot.7)$

$=$

$+$

$\alpha(\quad) = 1$

Markov Models

That concludes our discussion of Bayesian networks - there's a notebook for Friday, though! Now, there's another week or two of similar models! A *Markov Model* provides reasoning about a **sequence of events**.

1. Robot localization
2. Speech recognition
3. Medical monitoring
4. Weather forecasting

In short: we need to introduce *time* into our Bayesian Network model.

Definition: The *Markov Property* of a system states that conditional on the **present** state of the system, its **future** and past states are *independent*. It's also know as "memoryless."

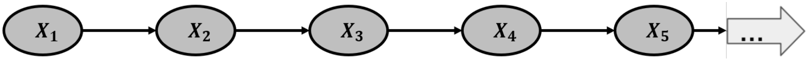
Markov Chains

A Markov model is a chain-structured Bayesian network.

The value of X at a time t is the state at time t

Stationary Markov model: All subsequent nodes have the same CPT (identically distributed)

Example: Is it raining? Let X_t denote the event that it is raining on day t



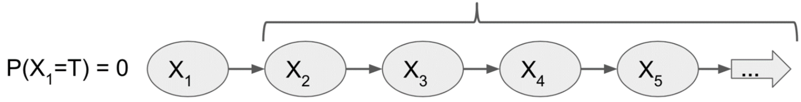
Markov Chains

Example: Is it raining? Let X_t denote the event that it is raining on day t

P(rain on day 3) =

Day 2 rain?	
T	.3
F	.1

X_t	$P(X_{t+1}=T \mid X_t)$
T	0.3
F	0.1



Just like a Bayesian network, the independence assumption of the Markov chain is a *causal chain*. So:

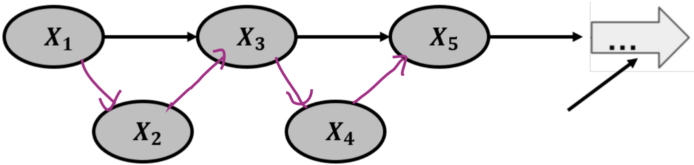
1. Past and future are independent of one another, given the present.
2. The state at time $t+1$ depends only on the state at time t (first order).
3. We again use *conditional probability tables* (CPT) to record the *transition probabilities*

Bigger Chains

Definition: The prior example was a *first-order* Markov chain, since the conditional dependence only used one prior state of the system.

1. Past and future are independent of one another, given the present.
2. The state at time $t+1$ depends only on the state at time t (first order).
3. We again use *conditional probability tables* (CPT) to record the *transition probabilities* from one state to another for this.

We could instead allow for *some* memory of the process. For example, a second-order process:



X_{t-1}	X_t	$P(X_{t+1}=T \mid X_t, X_{t-1})$
T	T	0.3
T	F	0.1
F	T	0.4
F	F	0.1

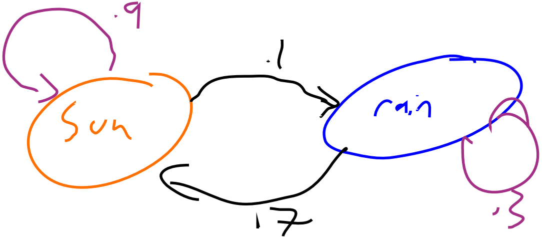
second order 2^2

third order 2^3

Weather Example

Example: Suppose we want to forecast the weather. From historical data, we know that in our town, if the current day was sunny, then the following day was also sunny $\boxed{90\%}$ of the time, and that if the current day was rainy, then the following day was also rainy 30% of the time.

Draw the state space graph and specify the CPT.

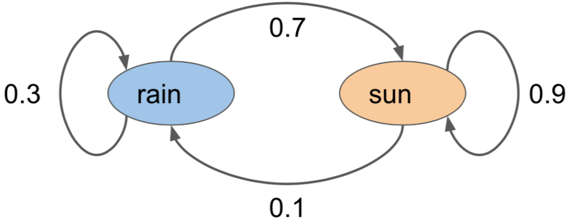


day 1 (X_i)		day 2 (X_{i+1})	
S	R	$P(X_2=S)$	$P(X_2=R)$
T	F	.9	.1
F	T	.7	.3

Weather Example

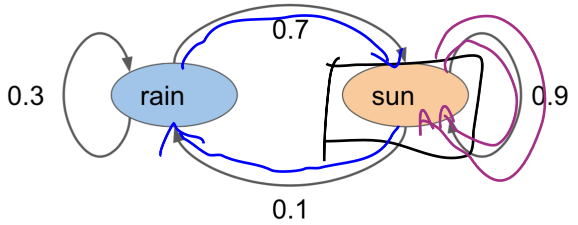
Example: Suppose we want to forecast the weather. From historical data, we know that in our town, if the current day was sunny, then the following day was also sunny 90% of the time, and that if the current day was rainy, then the following day was also rainy 30% of the time.

Draw the state space graph and specify the CPT.



Weather Example

Example: Suppose we want to forecast the weather. From historical data, we know that in our town, if the current day was sunny, then the following day was also sunny 90% of the time, and that if the current day was rainy, then the following day was also rainy 30% of the time.



1: $x_1 = \text{Sun}$, $x_2 = \text{Sun}$, $x_3 = \text{Sun}$ $\quad (.9)(.9)$

$+$

2: $x_1 = \text{Sun}$, $x_2 = \text{Rain}$, $x_3 = \text{Sun}$ $\quad (.1)(.7)$

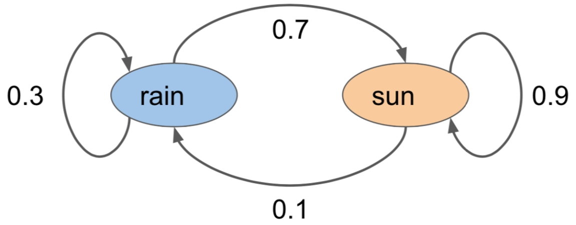
It is sunny today. What is the probability that it will be sunny 2 days from now?

$= .81 + .07$

$= .88$

Weather Example

Example: Suppose we want to forecast the weather. From historical data, we know that in our town, if the current day was sunny, then the following day was also sunny 90% of the time, and that if the current day was rainy, then the following day was also rainy 30% of the time.



It is sunny today. What is the probability that it will be sunny 2 days from now?

Solution: Two ways: Sun-Rain-Sun or Sun-Sun-Sun.

The Mini-Forward Algorithm

Example: It is sunny today. What is the probability that it will be sunny 2 days from now?

This is a general question about the state of X_3 *given* X_1. One way to track this is the *mini-forward algorithm.* For $t = 2, 3, 4, \dots$

$$P(x_t) = \sum_{x_{t-1}} P(x_t | x_{t-1}) P(x_{t-1})$$

$X = \begin{bmatrix} P(\text{sun}) \\ P(\text{rain}) \end{bmatrix}$: $X_1 \begin{bmatrix} P(\text{sun @ time} = 1) \\ P(\text{rain @ time} = 1) \end{bmatrix}$ $\left(= \begin{bmatrix} 1 \\ 0 \end{bmatrix} \right)$

$P(X_2) = P(X_2 | X_1) P(X_1) = P \begin{bmatrix} P(X_2 = \text{sun given } X_1) \cdot P(X_1) \\ P(X_2 = \text{rain given } X_1) \cdot P(X_1) \end{bmatrix}$

to get X_2 what X_1 we sum possibilities

	sun	rain
sun	.9	.1
rain	.7	.3

The Mini-Forward Algorithm

Example: It is sunny today. What is the probability that it will be sunny 2 days from now?

This is a general question about the state of X_3 *given* X_1. One way to track this is the *mini-forward algorithm.* For $t = 2, 3, 4, \ldots$

$$P(x_t) = \sum_{x_{t-1}} P(x_t|x_{t-1})P(x_{t-1})$$

$$\underset{X_1}{\begin{bmatrix} 1 \\ 0 \end{bmatrix}} \rightarrow \underset{X_2}{\begin{bmatrix} .9 \\ .1 \end{bmatrix}} \rightarrow \underset{X_3}{\begin{bmatrix} (.9)(.9) + .1(.7) \ldots \\ (.9)(.1) + .1(.3) \ldots \end{bmatrix}}$$

1. Given probabilities of the state at $t = 1$, we can take one forward-step of the chain to find the probabilities of the state at $t = 2$
2. ... and then get the probabilities of the states at $t = 3$.
3. At each time step, x_t is the *vector* holding of each state's probability. e.g. $x_2 = [P(sun\, at\, time\, t = 2), P(rain\, at\, time\, t = 2)]$

Matrix updating

It is sunny today. What is the probability that it will be sunny 2 days from now?

Definition: The *transition matrix* of a Markov process is the matrix with entry i, j of $T_{ij} :=$ the probability that the system in state i evolves to state j on the next time step.

$X_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$

$$\begin{array}{c} \\ Sun \\ Rain \end{array} \begin{array}{c} \begin{array}{cc} Sun & Rain \end{array} \\ \begin{bmatrix} .9 & .1 \\ .7 & .3 \end{bmatrix} \end{array}$$

$[1 \quad 0] = [.9 \quad .1]$

$X_2 = T X_1$

Notice what happens if we multiple this matrix by itself.

Matrix updating

It is sunny today. What is the probability that it will be sunny 2 days from now?

Definition: The *transition matrix* of a Markov process is the matrix with entry i, j of $T_{ij} :=$ the probability that the system in state i evolves to state j on the next time step.

$$\boldsymbol{T} = \begin{array}{c c} & \begin{array}{cc} X_1 & \qquad\qquad X_2 \end{array} \\ \begin{array}{c} X_1 \\ X_2 \end{array} & \begin{bmatrix} P(X_1 \to X_1) & P(X_1 \to X_2) \\ P(X_2 \to X_1) & P(X_2 \to X_2) \end{bmatrix} \end{array}$$

Notice what happens if we multiply this matrix by itself. $\boldsymbol{T}^2 =$

Matrix updating

It is sunny today. What is the probability that it will be sunny 2 days from now?

Definition: The *transition matrix* of a Markov process is the matrix with entry i, j of $T_{ij} :=$ the probability that the system in state i evolves to state j on the next time step.

$$\boldsymbol{T} = \begin{array}{c c} & \begin{array}{cc} X_1 & \qquad\qquad X_2 \end{array} \\ \begin{array}{c} X_1 \\ X_2 \end{array} & \begin{bmatrix} P(X_1 \to X_1) & P(X_1 \to X_2) \\ P(X_2 \to X_1) & P(X_2 \to X_2) \end{bmatrix} \end{array}$$

Notice what happens if we multiply this matrix by itself. $\boldsymbol{T}^2 =$

$$\begin{bmatrix} P(X_1 \to X_1)P(X_1 \to X_1) + P(X_1 \to X_2)P(X_2 \to X_1) & P(X_1 \to X_1)P(X_1 \to X_2) + P(X_1 \to X_2)P(X_2 \to X_2) \\ P(X_2 \to X_1)P(X_1 \to X_1) + P(X_2 \to X_2)P(X_2 \to X_1) & P(X_2 \to X_1)P(X_1 \to X_2) + P(X_2 \to X_2)P(X_2 \to X_2) \end{bmatrix}$$

Matrix updating

Result: The transition matrix gives us a shortcut for updating probabilities after multiple steps!

$$\boldsymbol{T}^k_{i,j} = \text{Probability that system in state } i \text{ is in state } j \text{ after exactly } k \text{ steps.}$$

```
transition_matrix = np.array([[0.9, 0.1],[0.7, 0.3]])

new = transition_matrix

for k in range(1,13):
    new = np.matmul(new, transition_matrix)
    print('T**{} = \n{}'.format(k+1, new))
```

What happens if we run this a very large number of times?

Moving Forward

- Coming up:
  1. Markov!