CS 188: Artificial Intelligence

Review of Probability, Bayes' nets

DISCLAIMER: It is insufficient to simply study these slides, they are merely meant as a quick refresher of the high-level ideas covered. You need to study all materials covered in lecture, section, assignments and projects !

Pieter Abbeel – UC Berkeley

Many slides adapted from Dan Klein

Probability recap

- Conditional probability $P(x|y) = \frac{P(x,y)}{P(y)}$
- Product rule $P(x,y) = P(x|y)P(y)$
- Chain rule $P(X_1, X_2, \ldots X_n) = P(X_1)P(X_2|X_1)P(X_3|X_1, X_2)\ldots$
- X, Y independent iff: $\forall x, y : P(x,y) = P(x)P(y)$
  equivalently, iff: $\forall x, y \ : \ P(x|y) = P(x)$
  equivalently, iff: $\forall x, y \ : \ P(y|x) = P(y)$
- X and Y are conditionally independent given Z iff:
  $\forall x, y, z : P(x, y|z) = P(x|z)P(y|z)$

equivalently, iff: $\forall x, y, z \ : \ P(x|y,z) = P(x|z)$
equivalently, iff: $\forall x, y, z \ : \ P(y|x,z) = P(y|z)$

2

Inference by Enumeration

- P(sun)?
- P(sun | winter)?
- P(sun | winter, hot)?

S	T	W	P
summer	hot	sun	0.30
summer	hot	rain	0.05
summer	cold	sun	0.10
summer	cold	rain	0.05
winter	hot	sun	0.10
winter	hot	rain	0.05
winter	cold	sun	0.15
winter	cold	rain	0.20

3

Bayes' Nets Recap

- Representation
  - Chain rule -> Bayes' net = DAG + CPTs
- Conditional Independences
  - D-separation
- Probabilistic Inference
  - Enumeration (exact, exponential complexity)
  - Variable elimination (exact, worst-case exponential complexity, often better)
  - Probabilistic inference is NP-complete
  - Sampling (approximate)

4

Chain Rule → Bayes net

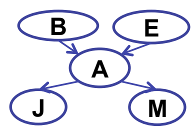
- Chain rule: can ***always*** write ***any*** joint distribution as an incremental product of conditional distributions

$$P(x_1, x_2, x_3) = P(x_1)P(x_2|x_1)P(x_3|x_1, x_2)$$

$$P(x_1, x_2, \ldots x_n) = \prod_i P(x_i|x_1 \ldots x_{i-1})$$

- Bayes nets: make conditional independence assumptions of the form:

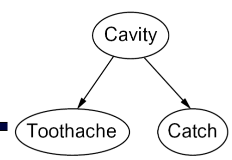
$$P(x_i|x_1 \cdots x_{i-1}) = P(x_i|parents(X_i))$$



giving us:

$$P(x_1, x_2, \ldots x_n) = \prod_{i=1}^{n} P(x_i|\mathit{parents}(X_i))$$

Probabilities in BNs



- Bayes' nets implicitly encode joint distributions
  - As a product of local conditional distributions
  - To see what probability a BN gives to a full assignment, multiply all the relevant conditionals together:

$$P(x_1, x_2, \ldots x_n) = \prod_{i=1}^{n} P(x_i|\mathit{parents}(X_i))$$

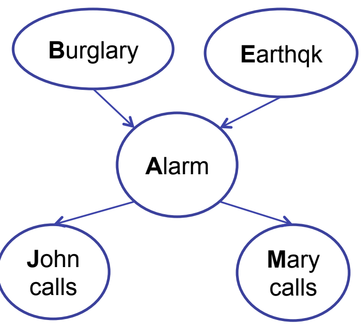
  - Example:

$$P(+\mathit{cavity},\ +\mathit{catch},\ \neg\mathit{toothache})$$

- This lets us reconstruct any entry of the full joint
- Not every BN can represent every joint distribution
  - The topology enforces certain conditional independencies

Example: Alarm Network

B	P(B)
+b	0.001
¬b	0.999



E	P(E)
+e	0.002
¬e	0.998

B	E	A	P(A|B,E)
+b	+e	+a	0.95
+b	+e	¬a	0.05
+b	¬e	+a	0.94
+b	¬e	¬a	0.06
¬b	+e	+a	0.29
¬b	+e	¬a	0.71
¬b	¬e	+a	0.001
¬b	¬e	¬a	0.999

A	J	P(J|A)
+a	+j	0.9
+a	¬j	0.1
¬a	+j	0.05
¬a	¬j	0.95

A	M	P(M|A)
+a	+m	0.7
+a	¬m	0.3
¬a	+m	0.01
¬a	¬m	0.99

Size of a Bayes' Net for $P(X_1, X_2, \ldots X_n)$

- How big is a joint distribution over N Boolean variables?

  2^N
- Size of representation if we use the chain rule

  2^N
- How big is an N-node net if nodes have up to k parents?

  $O(N * 2^{k+1})$

- Both give you the power to calculate
- BNs:
  - Huge space savings!
  - Easier to elicit local CPTs
  - Faster to answer queries

8

Bayes Nets: Assumptions

- Assumptions made by specifying the graph:

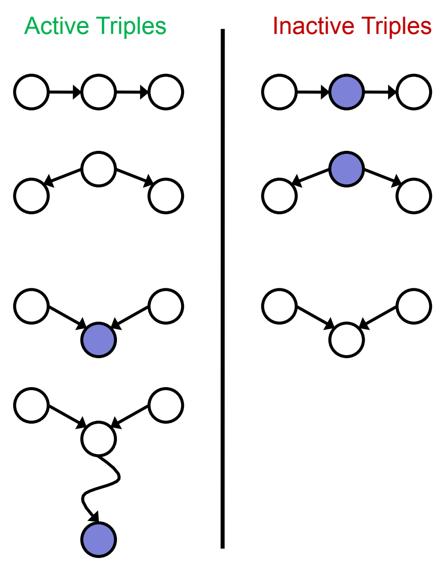
$$P(x_i|x_1 \cdots x_{i-1}) = P(x_i|parents(X_i))$$

- Given a Bayes net graph additional conditional independences can be read off directly from the graph
- Question: Are two nodes *guaranteed to be independent* given certain evidence?
- If no, can prove with a counter example
  - I.e., pick a set of CPT's, and show that the independence assumption is violated by the resulting distribution
- If yes, can prove with
  - Algebra (tedious)
  - D-separation (analyzes graph)

9

D-Separation

- Question: Are X and Y conditionally independent given evidence vars {Z}?
  - Yes, if X and Y "separated" by Z
  - Consider all (undirected) paths from X to Y
  - No active paths = independence!
- A path is active if each triple is active:
  - Causal chain A → B → C where B is unobserved (either direction)
  - Common cause A ← B → C where B is unobserved
  - Common effect (aka v-structure) A → B ← C where B *or one of its descendents* is observed
- All it takes to block a path is a single inactive segment



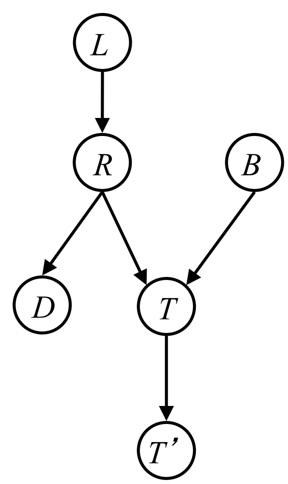
D-Separation

- Given query $X_i \overset{?}{\perp\!\!\!\perp} X_j | \{X_{k_1}, ..., X_{k_n}\}$
- Shade all evidence nodes
- For all (undirected!) paths between and
  - Check whether path is active
    - If active return $X_i \not\perp\!\!\!\perp X_j | \{X_{k_1}, ..., X_{k_n}\}$
- (If reaching this point all paths have been checked and shown inactive)
  - Return $X_i \perp\!\!\!\perp X_j | \{X_{k_1}, ..., X_{k_n}\}$

11

Example

$L \perp\!\!\!\perp T' \| T$	*Yes*
$L \perp\!\!\!\perp B$	*Yes*
$L \perp\!\!\!\perp B \| T$	
$L \perp\!\!\!\perp B \| T'$	
$L \perp\!\!\!\perp B \| T, R$	*Yes*



12

All Conditional Independences

- Given a Bayes net structure, can run d-separation to build a complete list of conditional independences that are necessarily true of the form

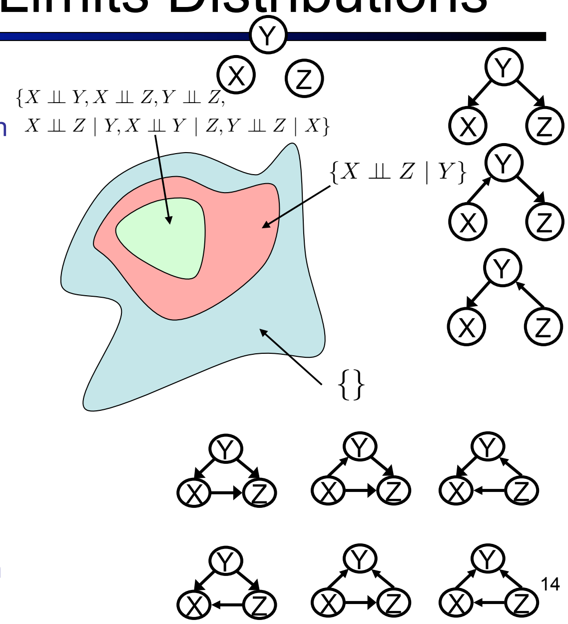
$$X_i \perp\!\!\!\perp X_j | \{X_{k_1}, ..., X_{k_n}\}$$

- This list determines the set of probability distributions that can be represented by Bayes' nets with this graph structure

13

Topology Limits Distributions

- Given some graph topology G, only certain joint distributions can be encoded
- The graph structure guarantees certain (conditional) independences
- (There might be more independence)
- Adding arcs increases the set of distributions, but has several costs
- Full conditioning can encode any distribution

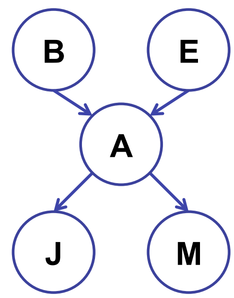


14

Inference by Enumeration

- Given unlimited time, inference in BNs is easy
- Recipe:
  - State the marginal probabilities you need
  - Figure out ALL the atomic probabilities you need
  - Calculate and combine them
- Example:

$$P(+b|+j,+m) = \frac{P(+b,+j,+m)}{P(+j,+m)}$$



15

Example: Enumeration

- In this simple method, we only need the BN to synthesize the joint entries

$$\begin{aligned}P(+b,+j,+m) = \\ &P(+b)P(+e)P(+a|+b,+e)P(+j|+a)P(+m|+a)+ \\ &P(+b)P(+e)P(-a|+b,+e)P(+j|-a)P(+m|-a)+ \\ &P(+b)P(-e)P(+a|+b,-e)P(+j|+a)P(+m|+a)+ \\ &P(+b)P(-e)P(-a|+b,-e)P(+j|-a)P(+m|-a)\end{aligned}$$

16

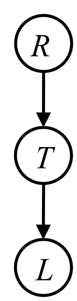
Variable Elimination

- Why is inference by enumeration so slow?
  - You join up the whole joint distribution before you sum out the hidden variables
  - You end up repeating a lot of work!
- Idea: interleave joining and marginalizing!
  - Called "Variable Elimination"
  - Still NP-hard, but usually much faster than inference by enumeration

17

Variable Elimination Outline

- Track objects called factors
- Initial factors are local CPTs (one per node)



$P(R)$

+r	0.1
-r	0.9

$P(T|R)$

+r	+t	0.8
+r	-t	0.2
-r	+t	0.1
-r	-t	0.9

$P(L|T)$

+t	+l	0.3
+t	-l	0.7
-t	+l	0.1
-t	-l	0.9

- Any known values are selected
  - E.g. if we know $L = +\ell$, the initial factors are

$P(R)$

+r	0.1
-r	0.9

$P(T|R)$

+r	+t	0.8
+r	-t	0.2
-r	+t	0.1
-r	-t	0.9

$P(+\ell|T)$

+t	+l	0.3
-t	+l	0.1

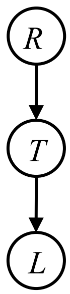
- VE: Alternately join factors and eliminate variables

18

Variable Elimination Example

$P(R)$

+r	0.1
-r	0.9



$P(T|R)$

+r	+t	0.8
+r	-t	0.2
-r	+t	0.1
-r	-t	0.9

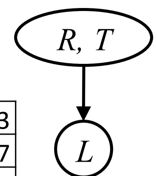
$P(L|T)$

+t	+l	0.3
+t	-l	0.7
-t	+l	0.1
-t	-l	0.9

Join R

$P(R,T)$

+r	+t	0.08
+r	-t	0.02
-r	+t	0.09
-r	-t	0.81



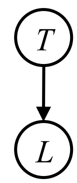
$P(L|T)$

+t	+l	0.3
+t	-l	0.7
-t	+l	0.1
-t	-l	0.9

Sum out R

$P(T)$

+t	0.17
-t	0.83

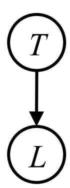


$P(L|T)$

+t	+l	0.3
+t	-l	0.7
-t	+l	0.1
-t	-l	0.9

19

Variable Elimination Example



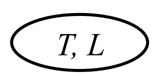
$P(T)$

+t	0.17
-t	0.83

$P(L|T)$

+t	+l	0.3
+t	-l	0.7
-t	+l	0.1
-t	-l	0.9

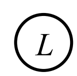
Join T



$P(T,L)$

+t	+l	0.051
+t	-l	0.119
-t	+l	0.083
-t	-l	0.747

Sum out T



$P(L)$

+l	0.134
-l	0.886

* VE is variable elimination

Example

$P(B|j,m) \propto P(B,j,m)$

$P(B)$	$P(E)$	$P(A|B,E)$	$P(j|A)$	$P(m|A)$

Choose A

$P(A|B,E)$
$P(j|A)$
$P(m|A)$

$\xrightarrow{\times}$ $P(j,m,A|B,E)$ $\xrightarrow{\sum}$ $P(j,m|B,E)$

$P(B)$	$P(E)$	$P(j,m|B,E)$

21

Example

$P(B)$	$P(E)$	$P(j,m|B,E)$

Choose E

$P(E)$
$P(j,m|B,E)$

$\xrightarrow{\times}$ $P(j,m,E|B)$ $\xrightarrow{\sum}$ $P(j,m|B)$

$P(B)$	$P(j,m|B)$

Finish with B

$P(B)$
$P(j,m|B)$

$\xrightarrow{\times}$ $P(j,m,B)$ $\xrightarrow{\text{Normalize}}$ $P(B|j,m)$

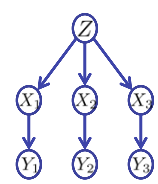
22

General Variable Elimination

- Query: $P(Q|E_1 = e_1, \ldots E_k = e_k)$
- Start with initial factors:
  - Local CPTs (but instantiated by evidence)
- While there are still hidden variables (not Q or evidence):
  - Pick a hidden variable H
  - Join all factors mentioning H
  - Eliminate (sum out) H
- Join all remaining factors and normalize

23

Another (bit more abstractly worked out) Variable Elimination Example



Query: $P(X_3|Y_1 = y_1, Y_2 = y_2, Y_3 = y_3)$

Start by inserting evidence, which gives the following initial factors:

$$p(Z)p(X_1|Z)p(X_2|Z)p(X_3|Z)p(y_1|X_1)p(y_2|X_2)p(y_3|X_3)$$

Eliminate X_1, this introduces the factor $f_1(Z, y_1) = \sum_{x_1} p(x_1|Z)p(y_1|x_1)$, and:

$$p(Z)f_1(Z, y_1)p(X_2|Z)p(X_3|Z)p(y_2|X_2)p(y_3|X_3)$$

Eliminate X_2, this introduces the factor $f_2(Z, y_2) = \sum_{x_2} p(x_2|Z)p(y_2|x_2)$, and:

$$p(Z)f_1(Z, y_1)f_2(Z, y_2)p(X_3|Z)p(y_3|X_3)$$

Eliminate Z, this introduces the factor
$f_3(y_1, y_2, y_3, X_3) = \sum_z p(z)f_1(z, y_1)f_2(z, y_2)p(X_3|z)p(y_3|X_3)$, and:

$$f_3(y_1, y_2, y_3, X_3)$$

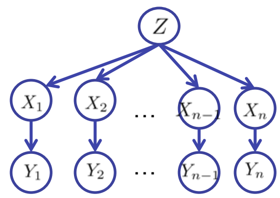
Normalizing over X_3 gives $P(X_3|y_1, y_2, y_3)$.

Computational complexity critically depends on the largest factor being generated in this process. Size of factor = number of entries in table. In example above (assuming binary) all factors generated are of size 2 --- as they all only have one variable (Z, Z, and X3 respectively).

24

Variable Elimination Ordering

- For the query $P(X_n|y_1,\ldots,y_n)$ work through the following two different orderings as done in previous slide: $Z, X_1, \ldots, X_{n-1}$ and $X_1, \ldots, X_{n-1}, Z$. What is the size of the maximum factor generated for each of the orderings?



- Answer: 2^n versus 2 (assuming binary)
- In general: the ordering can greatly affect efficiency.

25

Computational and Space Complexity of Variable Elimination

- The computational and space complexity of variable elimination is determined by the largest factor
- The elimination ordering can greatly affect the size of the largest factor.
  - E.g., previous slide's example 2^n vs. 2
- Does there always exist an ordering that only results in small factors?
  - No!

26

Worst Case Complexity?

- Consider the 3-SAT clause:

$$(x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_3 \vee \neg x_4) \wedge (x_2 \vee \neg x_2 \vee x_4) \wedge (\neg x_3 \vee \neg x_4 \vee \neg x_5) \wedge (x_2 \vee x_5 \vee x_7) \wedge (x_4 \vee x_5 \vee x_6) \wedge (\neg x_5 \vee x_6 \vee \neg x_7) \wedge (\neg x_5 \vee \neg x_6 \vee x_7)$$

which can be encoded by the following Bayes' net:

$P(X_i = 0) = P(X_i = 1) = 0.5$

$Y_1 = X_1 \vee X_2 \vee \neg X_3$

...

$Y_8 = \neg X_5 \vee X_6 \vee X_7$

$Y_{1,2} = Y_1 \wedge Y_2$

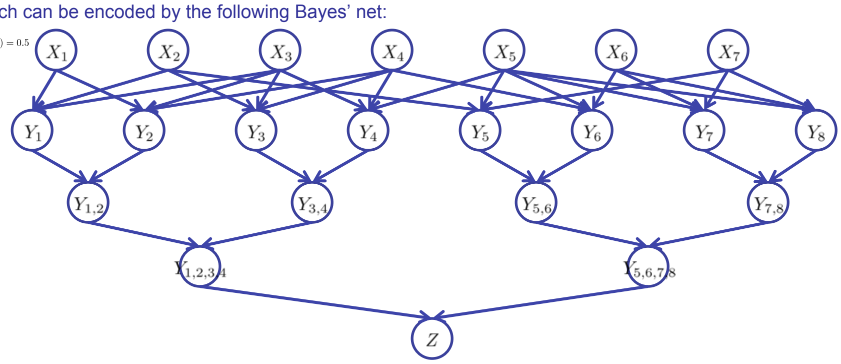
...

$Y_{7,8} = Y_7 \wedge Y_8$

$Y_{1,2,3,4} = Y_{1,2} \wedge Y_{3,4}$

$Y_{5,6,7,8} = Y_{5,6} \wedge Y_{7,8}$

$Z = Y_{1,2,3,4} \wedge Y_{5,6,7,8}$



- If we can answer P(z) equal to zero or not, we answered whether the 3-SAT problem has a solution.
- Subtlety: why the cascaded version of the AND rather than feeding all OR clauses into a single AND? Answer: a single AND would have an exponentially large CPT, whereas with representation above the Bayes' net has small CPTs only.
- Hence inference in Bayes' nets is NP-hard. No known efficient probabilistic inference in general.

27

Polytrees

- A polytree is a directed graph with no undirected cycles
- For poly-trees you can always find an ordering that is efficient
  - Try it!!
- Cut-set conditioning for Bayes' net inference
  - Choose set of variables such that if removed only a polytree remains
  - Think about how the specifics would work out!

28

Approximate Inference: Sampling

- Basic idea:
  - Draw N samples from a sampling distribution S
  - Compute an approximate posterior probability
  - Show this converges to the true probability P
- Why? Faster than computing the exact answer
- Prior sampling:
  - Sample ALL variables in topological order as this can be done quickly
- Rejection sampling for query $P(Q|E_1 = e_1, \ldots E_k = e_k)$
  - = like prior sampling, but reject when a variable is sampled inconsistent with the query, in this case when a variable E_i is sampled differently from e_i
- Likelihood weighting for query $P(Q|E_1 = e_1, \ldots E_k = e_k)$
  - = like prior sampling but variables E_i are not sampled, when it's their turn, they get set to e_i, and the sample gets weighted by P(e_i | value of parents(e_i) in current sample)
- Gibbs sampling: repeatedly samples each non-evidence variable conditioned on all other variables → can incorporate downstream evidence

29

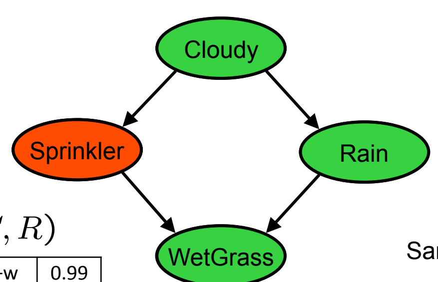
Prior Sampling

$P(C)$

+c	0.5
-c	0.5

$P(S|C)$

+c	+s	0.1
	-s	0.9
-c	+s	0.5
	-s	0.5



$P(R|C)$

+c	+r	0.8
	-r	0.2
-c	+r	0.2
	-r	0.8

$P(W|S,R)$

+s	+r	+w	0.99
		-w	0.01
	-r	+w	0.90
		-w	0.10
-s	+r	+w	0.90
		-w	0.10
	-r	+w	0.01
		-w	0.99

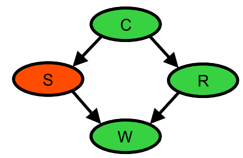
Samples:

+c, -s, +r, +w
-c, +s, -r, +w
…

30

Example

- We'll get a bunch of samples from the BN:
  - +c, -s, +r, +w
  - +c, +s, +r, +w
  - -c, +s, +r, -w
  - +c, -s, +r, +w
  - -c, -s, -r, +w



- If we want to know P(W)
  - We have counts <+w:4, -w:1>
  - Normalize to get P(W) = <+w:0.8, -w:0.2>
  - This will get closer to the true distribution with more samples
  - Can estimate anything else, too
  - What about P(C| +w)? P(C| +r, +w)? P(C| -r, -w)?
  - Fast: can use fewer samples if less time

31

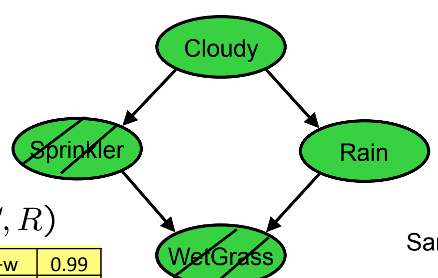
Likelihood Weighting

$P(C)$

+c	0.5
-c	0.5

$P(S|C)$

+c	+s	0.1
	-s	0.9
-c	+s	0.5
	-s	0.5



$P(R|C)$

+c	+r	0.8
	-r	0.2
-c	+r	0.2
	-r	0.8

$P(W|S,R)$

+s	+r	+w	0.99
		-w	0.01
	-r	+w	0.90
		-w	0.10
-s	+r	+w	0.90
		-w	0.10
	-r	+w	0.01
		-w	0.99

Samples:

+c, +s, +r, +w

…

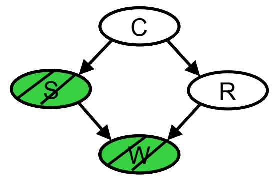
$$w = 1.0 \times 0.1 \times 0.99$$

32

Likelihood Weighting

- Sampling distribution if z sampled and e fixed evidence

$$S_{WS}(\mathbf{z},\mathbf{e}) = \prod_{i=1}^{l} P(z_i|\text{Parents}(Z_i))$$



- Now, samples have weights

$$w(\mathbf{z},\mathbf{e}) = \prod_{i=1}^{m} P(e_i|\text{Parents}(E_i))$$

- Together, weighted sampling distribution is consistent

$$S_{\text{WS}}(z,e) \cdot w(z,e) = \prod_{i=1}^{l} P(z_i|\text{Parents}(z_i)) \prod_{i=1}^{m} P(e_i|\text{Parents}(e_i))$$
$$= P(\mathbf{z},\mathbf{e})$$

33

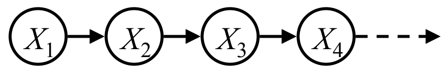
Gibbs Sampling

- *Idea*: instead of sampling from scratch, create samples that are each like the last one.
- *Procedure*: resample one variable at a time, conditioned on all the rest, but keep evidence fixed.
- *Properties*: Now samples are not independent (in fact they're nearly identical), but sample averages are still consistent estimators!
- *What's the point*: both upstream and downstream variables condition on evidence.

34

Markov Models

- A Markov model is a chain-structured BN
  - Each node is identically distributed (stationarity)
  - Value of X at a given time is called the state
  - As a BN:

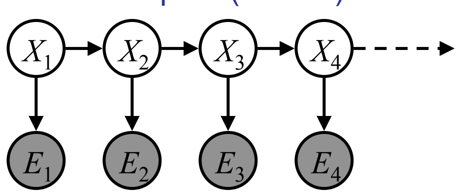


$P(X_1)$ $P(X|X_{-1})$

- The chain is just a (growing) BN
  - We can always use generic BN reasoning on it if we truncate the chain at a fixed length
- Stationary distributions
  - For most chains, the distribution we end up in is independent of the initial distribution
  - Called the stationary distribution of the chain
- Example applications: Web link analysis (Page Rank) and Gibbs Sampling

Hidden Markov Models

- Underlying Markov chain over states S
- You observe outputs (effects) at each time step

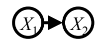


- Speech recognition HMMs:
  - X_i: specific positions in specific words; E_i: acoustic signals
- Machine translation HMMs:
  - X_i: translation options; E_i: Observations are words
- Robot tracking:
  - X_i: positions on a map; E_i: range readings

Online Belief Updates

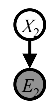
- Every time step, we start with current P(X | evidence)
- We update for time:

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



- We update for evidence:

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

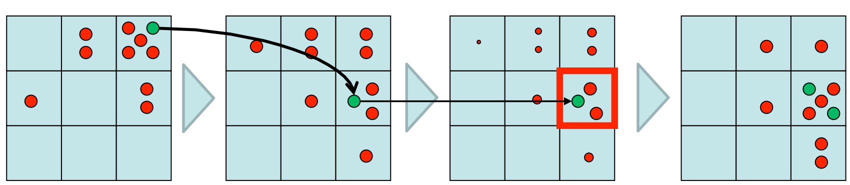


- The forward algorithm does both at once (and doesn't normalize)

Recap: Particle Filtering

- Particles: track samples of states rather than an explicit distribution

Elapse Weight Resample

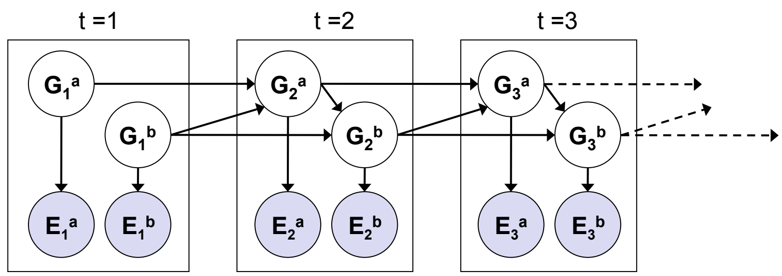


Particles:	Particles:	Particles:	(New) Particles:
(3,3)	(3,2)	(3,2) w=.9	(3,2)
(2,3)	(2,3)	(2,3) w=.2	(2,2)
(3,3)	(3,2)	(3,2) w=.9	(3,2)
(3,2)	(3,1)	(3,1) w=.4	(2,3)
(3,3)	(3,3)	(3,3) w=.4	(3,3)
(3,2)	(3,2)	(3,2) w=.9	(3,2)
(1,2)	(1,3)	(1,3) w=.1	(1,3)
(3,3)	(2,3)	(2,3) w=.2	(2,3)
(3,3)	(3,2)	(3,2) w=.9	(3,2)
(2,3)	(2,2)	(2,2) w=.4	(3,2)

38

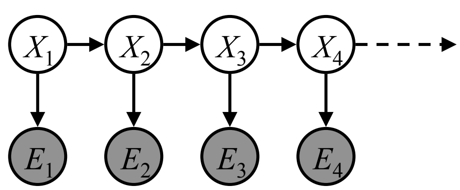
Dynamic Bayes Nets (DBNs)

- We want to track multiple variables over time, using multiple sources of evidence
- Idea: Repeat a fixed Bayes net structure at each time
- Variables from time *t* can condition on those from *t-1*



- Discrete valued dynamic Bayes nets are also HMMs

Best Explanation Queries



- Query: most likely seq:

$$\arg\max_{x_{1:t}} P(x_{1:t}|e_{1:t})$$

40

Best Explanation Query Solution Method 1: Search

$$\arg\max_{x_{1:t}} P(x_{1:t}|e_{1:t}) = \arg\max_{x_{1:t}} \prod_{i=1}^{t} P(x_t|x_{t-1})P(e_t|x_t)$$
$$= \arg\max_{x_{1:t}} \sum_{i=1}^{t} \log(P(x_t|x_{t-1})P(e_t|x_t))$$

slight abuse of notation, assuming $P(x_1|x_0) = P(x_1)$

- States: $\{(), +x_1, -x_1, +x_2, -x_2, \ldots, +x_t, -x_t\}$
- Start state: ()
- Actions: in state x_k, choose any assignment for state x_{k+1}
- Cost: $\log\left(P(x_{k+1}|x_k)P(e_{k+1}|x_{k+1})\right)$
- Goal test: goal(x_k) = true iff k == t

→ Can run uniform cost graph search to find solution

→ Uniform cost graph search will take O(t d^2). Think about this!

Best Explanation Query Solution Method 2: Viterbi Algorithm (= max-product version of forward algorithm)

$$x^*_{1:T} = \arg\max_{x_{1:T}} P(x_{1:T}|e_{1:T}) = \arg\max_{x_{1:T}} P(x_{1:T}, e_{1:T})$$

$$m_t[x_t] = \max_{x_{1:t-1}} P(x_{1:t-1}, x_t, e_{1:t})$$
$$= \max_{x_{1:t-1}} P(x_{1:t-1}, e_{1:t-1})P(x_t|x_{t-1})P(e_t|x_t)$$
$$= P(e_t|x_t) \max_{x_{t-1}} P(x_t|x_{t-1}) \max_{x_{1:t-2}} P(x_{1:t-1}, e_{1:t-1})$$
$$= P(e_t|x_t) \max_{x_{t-1}} P(x_t|x_{t-1}) m_{t-1}[x_{t-1}]$$

Viterbi computational complexity: O(t d^2)

Compare to forward algorithm:

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

42

Parameter Estimation

- Estimating distribution of random variables like X or X | Y
- *Empirically:* use training data
  - For each outcome x, look at the *empirical rate* of that value:

$$P_{\text{ML}}(x) = \frac{\text{count}(x)}{\text{total samples}}$$

r g g

$$P_{\text{ML}}(\text{r}) = 1/3$$

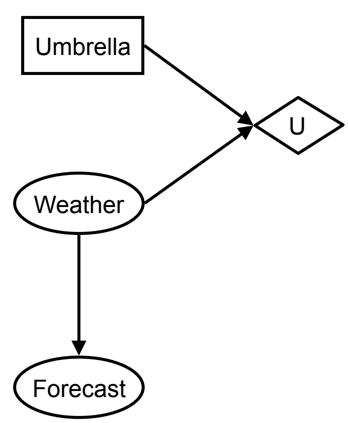
  - This is the estimate that maximizes the *likelihood of the data*

$$L(x, \theta) = \prod_i P_\theta(x_i)$$

- *Laplace smoothing*
  - Pretend saw every outcome k extra times $P_{LAP,k}(x) = \frac{c(x) + k}{N + k|X|}$
  - Smooth each condition independently: $P_{LAP,k}(x|y) = \frac{c(x,y) + k}{c(y) + k|X|}$

Decision Networks

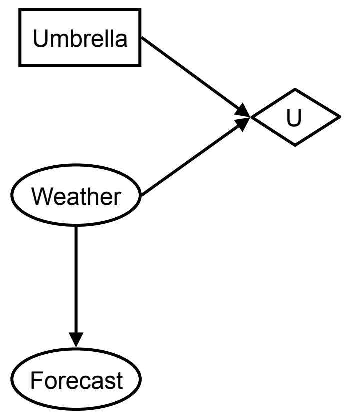
- MEU: choose the action which maximizes the expected utility given the evidence
- Can directly operationalize this with decision networks
  - Bayes nets with nodes for utility and actions
  - Lets us calculate the expected utility for each action
- New node types:
  - Chance nodes (just like BNs)
  - Actions (rectangles, cannot have parents, act as observed evidence)
  - Utility node (diamond, depends on action and chance nodes)



44

Decision Networks

- Action selection:
  - Instantiate all evidence
  - Set action node(s) each possible way
  - Calculate posterior for all parents of utility node, given the evidence
  - Calculate expected utility for each action
  - Choose maximizing action



45

Example: Decision Networks

Umbrella = leave

$$\text{EU(leave)} = \sum_w P(w)U(\text{leave}, w)$$

$$= 0.7 \cdot 100 + 0.3 \cdot 0 = 70$$

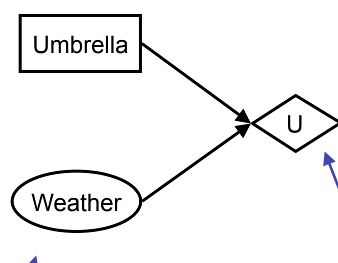
Umbrella = take

$$\text{EU(take)} = \sum_w P(w)U(\text{take}, w)$$

$$= 0.7 \cdot 20 + 0.3 \cdot 70 = 35$$

Optimal decision = leave

$$\text{MEU}(\phi) = \max_a \text{EU}(a) = 70$$



W	P(W)
sun	0.7
rain	0.3

A	W	U(A,W)
leave	sun	100
leave	rain	0
take	sun	20
take	rain	70

Example: Decision Networks

Umbrella = leave

$$\text{EU}(\text{leave}|\text{bad}) = \sum_w P(w|\text{bad})U(\text{leave}, w)$$

$$= 0.34 \cdot 100 + 0.66 \cdot 0 = 34$$

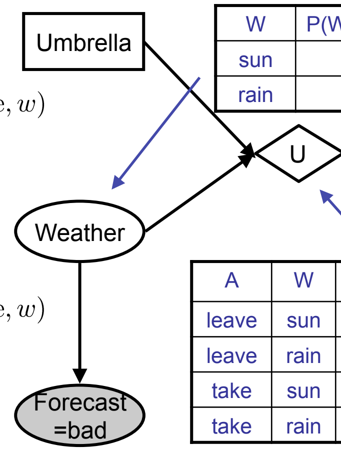
Umbrella = take

$$\text{EU}(\text{take}|\text{bad}) = \sum_w P(w|\text{bad})U(\text{take}, w)$$

$$= 0.34 \cdot 20 + 0.66 \cdot 70 = 53$$

Optimal decision = take

$$\text{MEU}(F = \text{bad}) = \max_a \text{EU}(a|\text{bad}) = 53$$

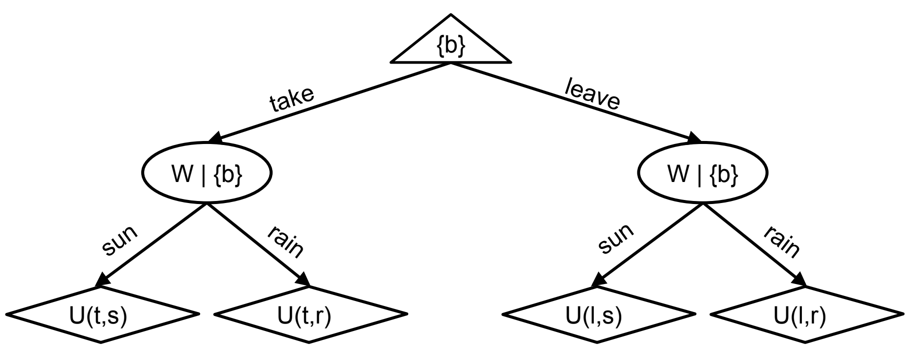


W	P(W|F=bad)
sun	0.34
rain	0.66

A	W	U(A,W)
leave	sun	100
leave	rain	0
take	sun	20
take	rain	70

47

Decisions as Outcome Trees



48

VPI Example: Weather

MEU with no evidence

$$\text{MEU}(\emptyset) = \max_a \text{EU}(a) = 70$$

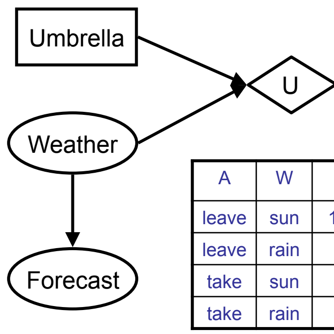
MEU if forecast is bad

$$\text{MEU}(F = \text{bad}) = \max_a \text{EU}(a|\text{bad}) = 53$$

MEU if forecast is good

$$\text{MEU}(F = \text{good}) = \max_a \text{EU}(a|\text{good}) = 95$$

Forecast distribution



A	W	U
leave	sun	100
leave	rain	0
take	sun	20
take	rain	70

F	P(F)
good	0.59
bad	0.41

$$0.59 \cdot (95) + 0.41 \cdot (53) - 70$$
$$77.8 - 70 = 7.8$$

$$\text{VPI}(E|e') = \left(\sum_{e'} P(e'|e)\text{MEU}(e, e')\right) - \text{MEU}(e)$$

49

Value of Information

- Assume we have evidence E=e. Value if we act now:

  $$\text{MEU}(e) = \max_a \sum_s P(s|e)\, U(s,a)$$

- Assume we see that E' = e'. Value if we act then:

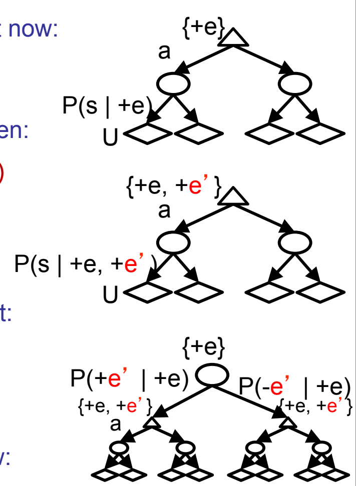
  $$\text{MEU}(e, e') = \max_a \sum_s P(s|e, e')\, U(s,a)$$

- BUT E' is a random variable whose value is unknown, so we don't know what e' will be
- Expected value if E' is revealed and then we act:

  $$\text{MEU}(e, E') = \sum_{e'} P(e'|e)\text{MEU}(e, e')$$

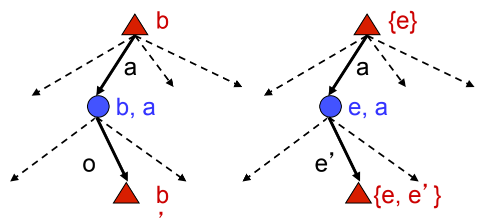
- Value of information: how much MEU goes up by revealing E' first then acting, over acting now:

  $$\text{VPI}(E'|e) = \text{MEU}(e, E') - \text{MEU}(e)$$



Example: Ghostbusters

- In (static) Ghostbusters:
  - Belief state determined by evidence to date {e}
  - Tree really over evidence sets
  - Probabilistic reasoning needed to predict new evidence given past evidence



- Solving POMDPs
  - One way: use truncated expectimax to compute approximate value of actions
  - What if you only considered busting or one sense followed by a bust?
  - You get a VPI-based agent!