SIMON FRASER UNIVERSITY
ENGAGING THE WORLD

Bayes Nets (cont'd)
Hidden Markov Models

Dr. Angelica Lim
Assistant Professor
School of Computing Science
Simon Fraser University, Canada

Dec. 3, 2024

The Product Rule: Example

$$P(W \mid T)\, P(T) = P(W, T)$$

$P(W \mid T)$

	hot	cold
sun	0.90	0.30
rain	0.04	0.16
fog	0.06	0.54
meteor	0.00	0.00

$P(T)$

T	P
hot	0.5
cold	0.5

$\Longleftrightarrow$

$P(W, T)$

		Temperature	
		hot	cold
Weather	sun	0.45	0.15
	rain	0.02	0.08
	fog	0.03	0.27
	meteor	0.00	0.00

The Chain Rule

A joint distribution can be written as a **product** of **conditional distributions** by repeated application of the product rule:

$$P(x_1, x_2, x_3) = P(x_3 \mid x_1, x_2)\, P(x_1, x_2) = P(x_3 \mid x_1, x_2)\, P(x_2 \mid x_1)\, P(x_1)$$

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

Inference

- **Inference**: calculating some useful quantity from a joint probability distribution

e.g. what is the probability of having a disease given symptoms

- Examples:

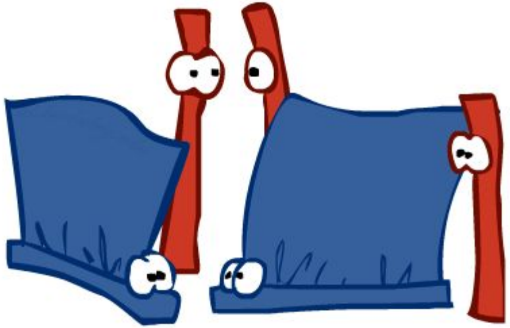
e.g. probability of spam/not spam given evidence

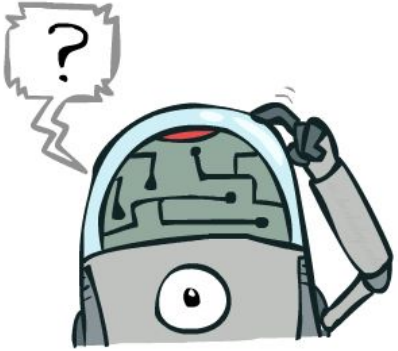
  - Posterior probability

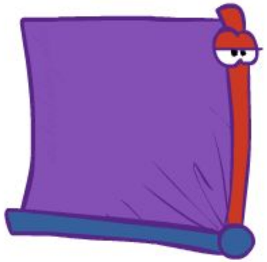
$$P(Q|E_1 = e_1, \dots E_k = e_k)$$

  - Most likely explanation:

$$\text{argmax}_q \; P(Q = q|E_1 = e_1 \dots)$$







Joint Distribution: Inference by Enumeration

Query variable

Key point: We can perform inference from a joint distribution… but the tables are huge.

Evidence variable

P(S | sun)?

- 1. Enumerate options with sun
- 2. Sum out irrelevant variable(s)
- 3. Normalize

P(S | sun) =
{summer: 0.45/(0.45+0.25), winter: 0.25/(0.45+0.25)}

Here, temperature was a hidden variable

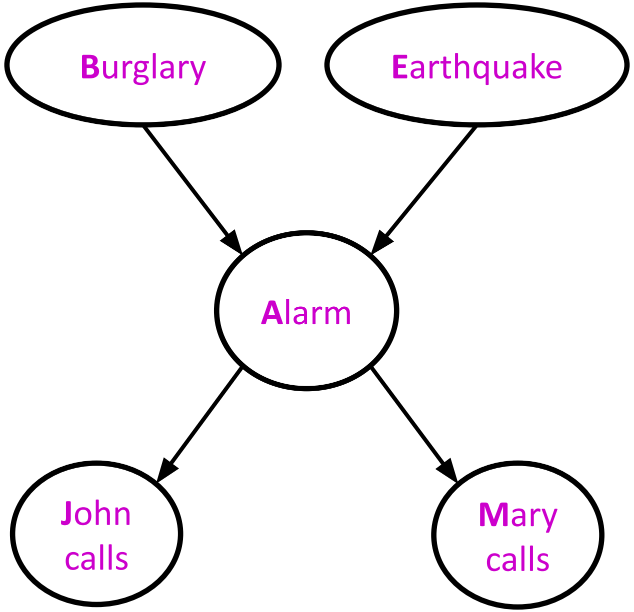
0.45

0.25

Season	Temp	Weather	P
summer	hot	sun	0.35
summer	hot	rain	0.01
summer	hot	fog	0.01
summer	hot	meteor	0.00
summer	cold	sun	0.10
summer	cold	rain	0.05
summer	cold	fog	0.09
summer	cold	meteor	0.00
winter	hot	sun	0.10
winter	hot	rain	0.01
winter	hot	fog	0.02
winter	hot	meteor	0.00
winter	cold	sun	0.15
winter	cold	rain	0.20
winter	cold	fog	0.18
winter	cold	meteor	0.00

Bayes Net Encodes Joint Distribution

P(B)	
true	false
0.001	0.999



P(E)	
true	false
0.002	0.998

$P(b, \neg e, a, \neg j, \neg m) =$

$P(b)\ P(\neg e)\ P(a|b,\neg e)\ P(\neg j|a)\ P(\neg m|a)$

$=.001 \times .998 \times .94 \times .1 \times .3 = .000028$

B	E	P(A|B,E)	
		true	false
true	true	0.95	0.05
true	false	0.94	0.06
false	true	0.29	0.71
false	false	0.001	0.999

A	P(J|A)	
	true	false
true	0.9	0.1
false	0.05	0.95

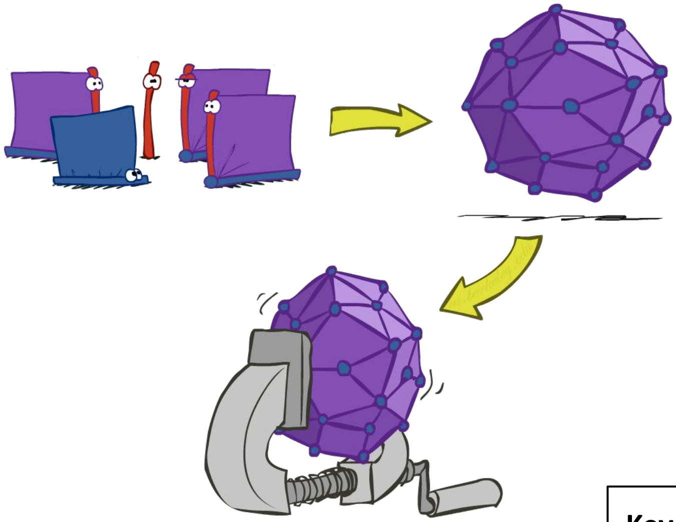
A	P(M|A)	
	true	false
true	0.7	0.3
false	0.01	0.99

Key point: With some assumptions on conditional independence, we can store the same data in a Bayes Net more succinctly.

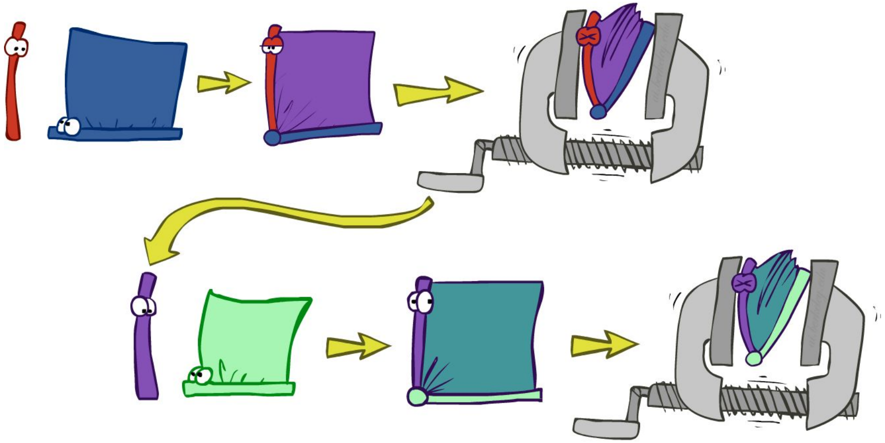
… but we'd still need to unpack it into huge tables.

Inference by Enumeration vs. Variable Elimination

- Why is inference by enumeration so slow?
  - You join up the whole joint distribution before you sum out the hidden variables

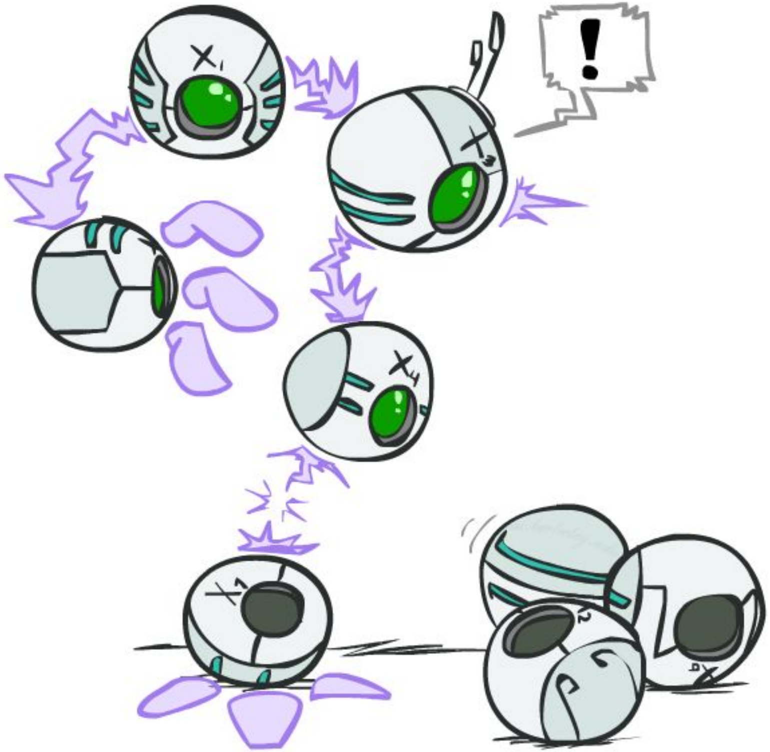


- Idea: interleave joining and marginalizing!
  - Called “Variable Elimination”
  - Still NP-hard, but usually much faster than inference by enumeration

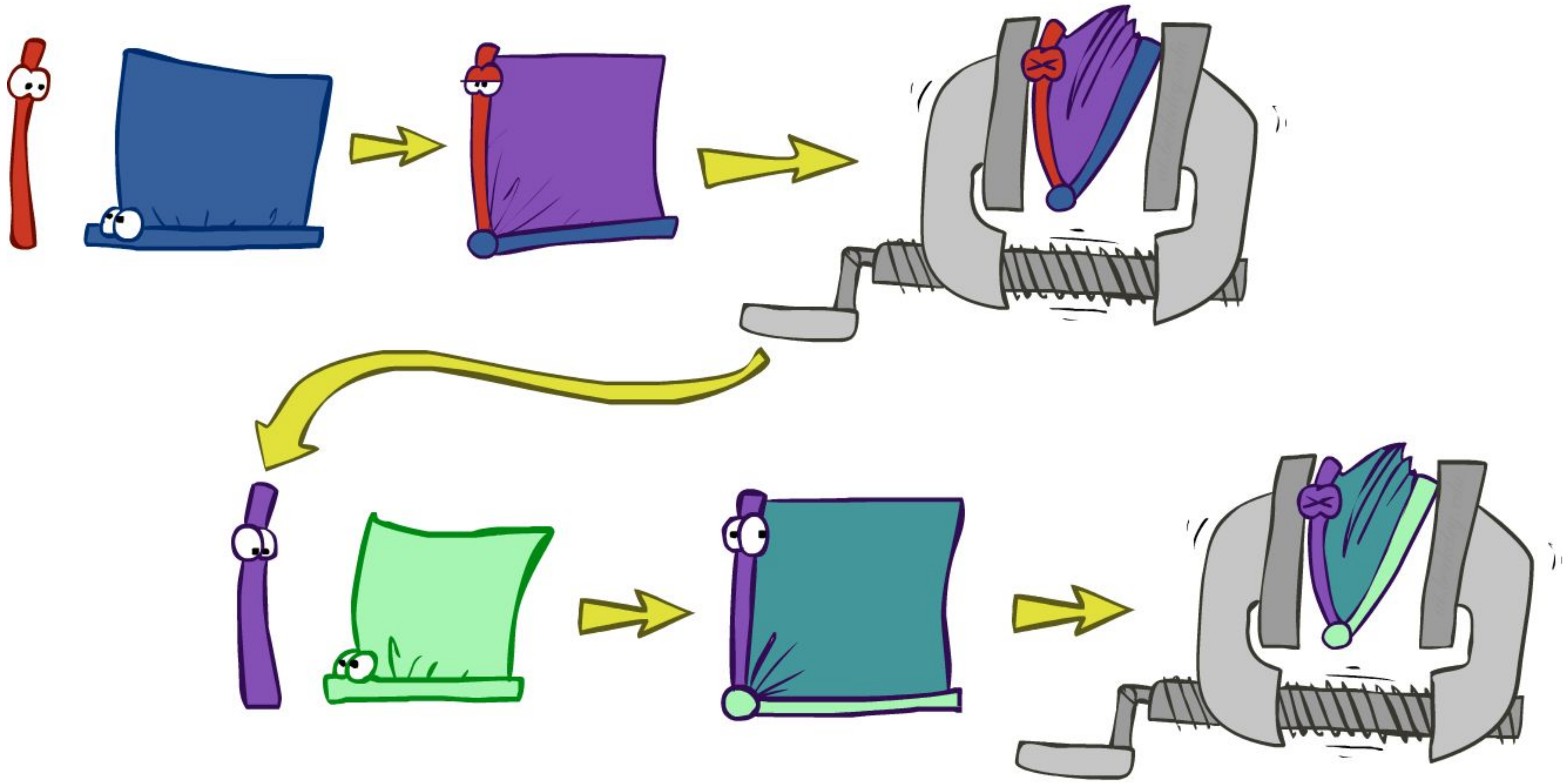


Key point: Instead of 2 phases (unpack and infer), we can try to limit the maximum size of tables by unpacking the Bayes Net gradually.

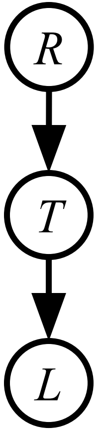
Variable Elimination (VE)



Variable Elimination => Marginalizing Early



Marginalizing Early == Variable Elimination



$P(R)$

+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

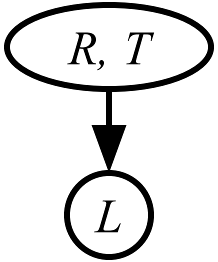
$P(L|T)$

+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



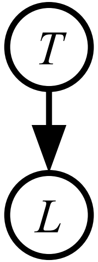
$P(L|T)$

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

Sum out R

$P(T)$

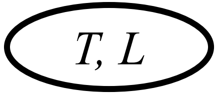
+t	0.17
-t	0.83



$P(L|T)$

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

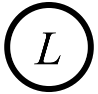
Join T



$P(T, L)$

+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	0.866

Key point: We interleave joining and marginalizing, and this is called variable elimination, where we eliminate hidden or irrelevant variables as we go.

Incorporating evidence

- If you have evidence, start with factors that select that evidence
  - No evidence uses these initial factors:

$P(R)$

+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

  - Computing $P(L|+r)$ the initial factors become:

$P(+r)$

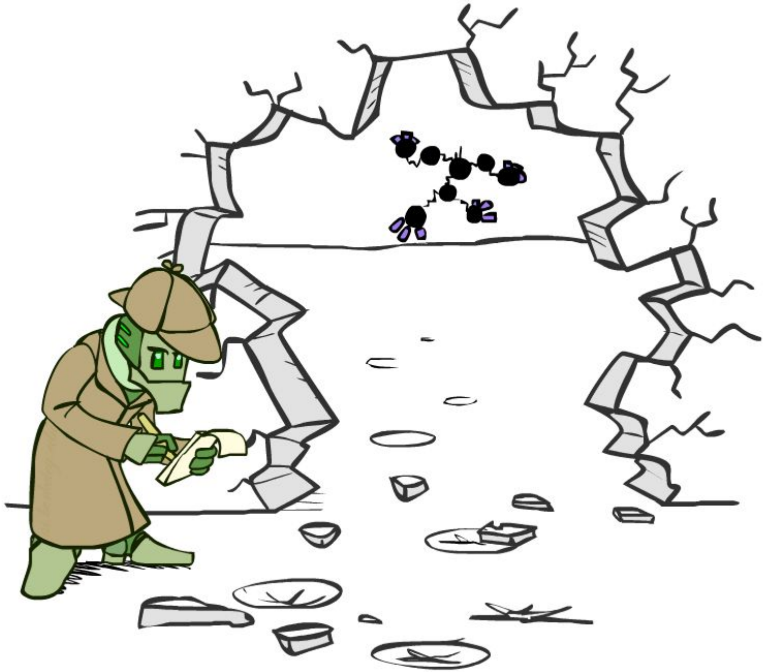
+r	0.1

$P(T|+r)$

+r	+t	0.8
+r	-t	0.2

$P(L|T)$

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



- We eliminate all vars other than query + evidence

Incorporating evidence

- Result will be a selected joint of query and evidence
  – E.g. for P(L | +r), we would end up with:

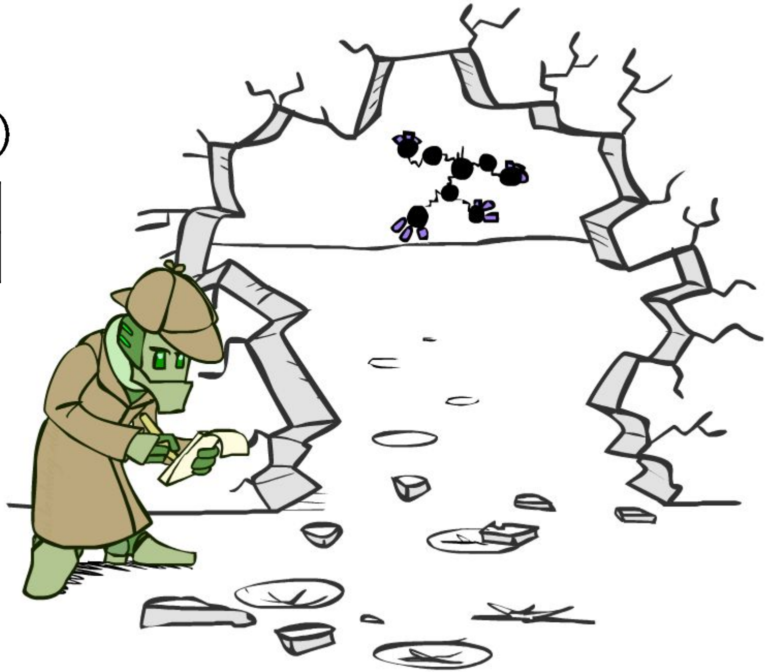
$P(+r, L)$

+r	+l	0.026
+r	-l	0.074

Normalize

$P(L|+r)$

+l	0.26
-l	0.74



- To get our answer, just normalize this!

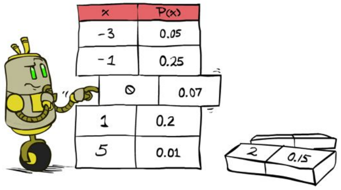
- That's it!

General Variable Elimination

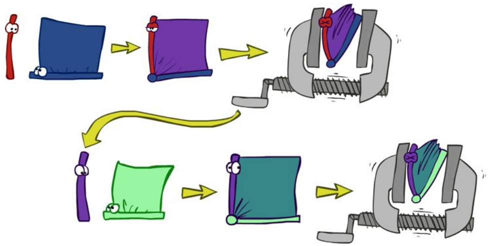
This slide is not examinable

- Query: $P(Q|E_1 = e_1, \ldots E_k = e_k)$

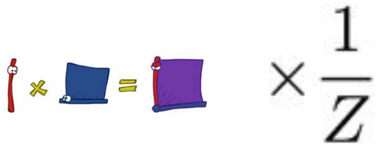
- Start with initial factors:
  - Local CPTs (but instantiated by evidence)



- 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



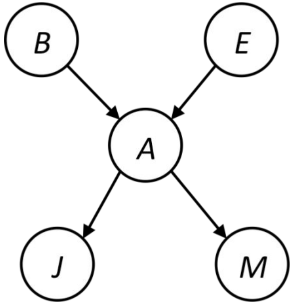
General Variable Elimination: Example

This slide is not examinable

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

Probability of a burglar if both John and Mary call

$$\boxed{P(B) \qquad P(E) \qquad P(A|B,E) \qquad P(j|A) \qquad P(m|A)}$$



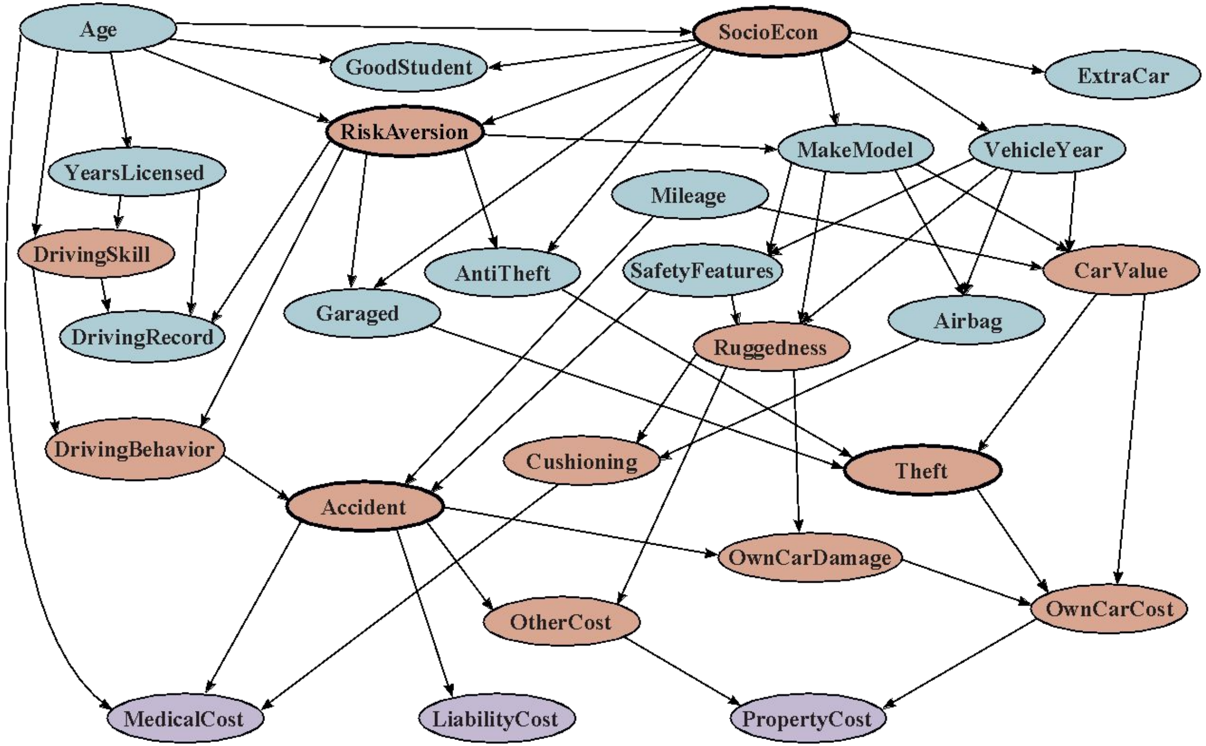
$$\begin{aligned} P(B|j,m) &\propto P(B,j,m) \\ &= \sum_{e,a} P(B)P(e)P(a|B,e)P(j|a)P(m|a) \\ &= \sum_{e} P(B)P(e)\sum_{a} P(a|B,e)P(j|a)P(m|a) \\ &= \sum_{e} P(B)P(e)f_1(B,e,j,m) \\ &= P(B)\sum_{e} P(e)f_1(B,e,j,m) \\ &= P(B)f_2(B,j,m) \end{aligned}$$

Join all factors involving A and marginalize out A

Join all factors involving E and marginalize out E

All we are doing is exploiting uwy + uwz + uxy + uxz + vwy + vwz + vxy +vxz = (u+v)(w+x)(y+z) to improve computational efficiency!

Example Bayes Net: Car Insurance



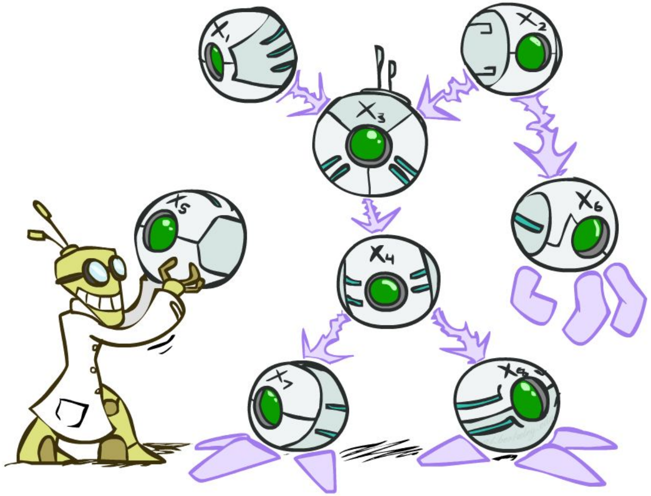
Enumeration: **227M** operations

Elimination: **221K** operations

Summary

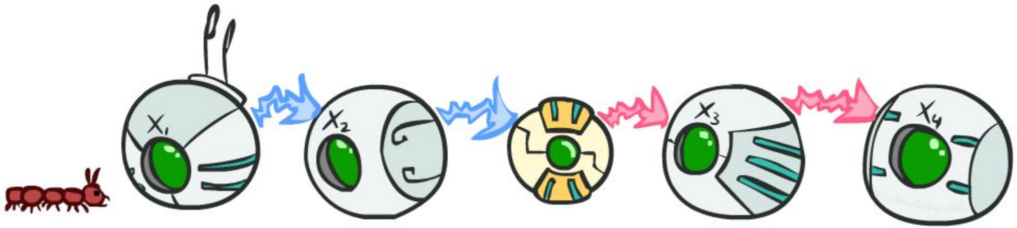
- Exact inference = sums of products of conditional probabilities from the network
- **Enumeration** is always **exponential**
- **Variable elimination** reduces this by avoiding the recomputation of repeated subexpressions
  - Massive speedups in practice
- Exact inference is #P-hard

  Approximation methods exist



CS 188: Artificial Intelligence

Markov Models



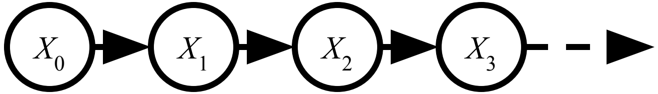
Uncertainty and Time

- Often, we want to reason about a ***sequence*** of observations where the state of the underlying system is ***changing***
  - Speech recognition
  - Robot localization
  - User attention
  - Medical monitoring
  - Global climate

- Need to introduce time into our models

Markov Models (inc. Markov Chains)

– Value of X at a given time is called the ***state*** (usually discrete, finite)



$P(X_0)$ $P(X_t \mid X_{t-1})$

– The ***transition model*** $P(X_t \mid X_{t-1})$ specifies how the state evolves over time
– ***Stationarity*** assumption: transition probabilities are the same at all times
– ***Markov*** assumption: “future is independent of the past given the present”

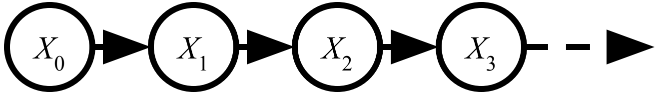
- X_{t+1} is independent of $X_0, \ldots, X_{t-1}$ given X_t
- This is a ***first-order*** Markov model (a kth-order model allows dependencies on k earlier steps)

– Joint distribution $P(X_0, \ldots, X_T) = P(X_0) \prod_t P(X_t \mid X_{t-1})$

SFU

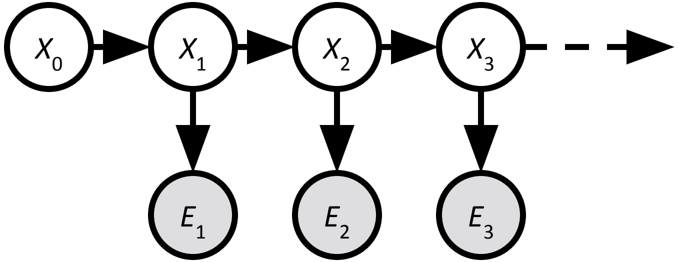
t starts at one?

"Markov" as in Markov Decision Processes?

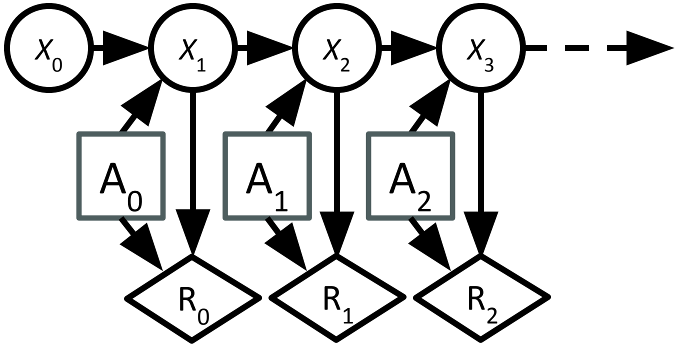


Markov Chain

Hidden models
have that inferences
are done indirectly
using other data.
Ex: a prison guard
infers that it is raining
based on someone entering the
prison with an umbrella



Hidden Markov Model



Markov Decision Process

???

Partially Observable
Markov Decision Process

Quiz: are Markov models a special case of Bayes nets?

- Yes and no!
- Yes:
  - Directed acyclic graph, joint = product of conditionals
- No:
  - Infinitely many variables (unless we truncate)
  - Repetition of transition model not part of standard Bayes net syntax

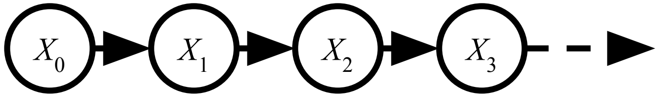
Example: n-gram models

We call ourselves *Homo sapiens*—man the wise—because our **intelligence** is so important to us. For thousands of years, we have tried to understand *how we think*; that is, how a mere handful of matter can perceive, understand, predict, and manipulate a world far larger and more complicated than itself. ….

- **State**: word at position t in text (can also build letter n-grams)
- **Transition model** (probabilities come from empirical frequencies):
  - Unigram (zero-order): $P(Word_t = i)$
    - "logical are as are confusion a may right tries agent goal the was . . ."
  - Bigram (first-order): $P(Word_t = i \mid Word_{t-1} = j)$
    - "systems are very similar computational approach would be represented . . ."
  - Trigram (second-order): $P(Word_t = i \mid Word_{t-1} = j, Word_{t-2} = k)$
    - "planning and scheduling are integrated the success of naive bayes model is . . ."
- Applications: text classification, spam detection, author identification, language classification, speech recognition

Auto complete in text messages is an example

Forward algorithm (simple form)



$P(X_0)$ $P(X_t \mid X_{t-1})$

- What is the state at time t?
  - $P(X_t) = \sum_{xt-1} P(X_t, X_{t-1}=x_{t-1})$
  - $\quad = \sum_{xt-1} P(X_{t-1}=x_{t-1})\, P(X_t \mid X_{t-1}=x_{t-1})$

Probability from previous iteration

Transition model

- Iterate this update starting at $t=0$
  - This is called a ***recursive*** update: $P_t = g(P_{t-1}) = g(g(g(g(\ldots P_0))))$

Hidden Markov Models

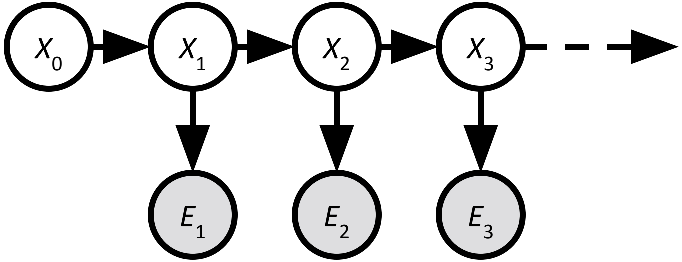
SFU

Hidden Markov Models

- Usually the true state is not observed directly

- Hidden Markov Models (HMMs)
  - Underlying Markov chain over states X
  - You observe evidence E at each time step

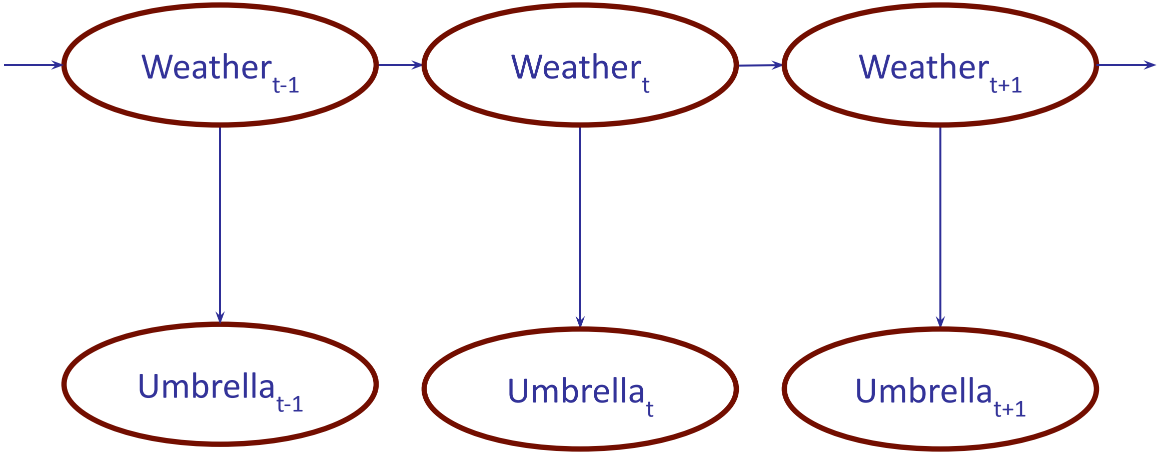
  - X_t is a single discrete variable; E_t may be continuous and may consist of several variables



Example: Weather HMM

W_{t-1}	$P(W_t \mid W_{t-1})$	
	sun	rain
sun	0.9	0.1
rain	0.3	0.7

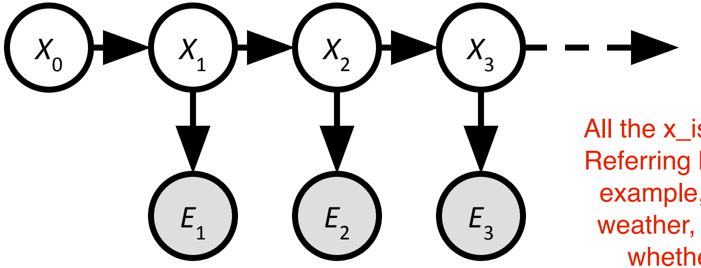
- An HMM is defined by:
  - Initial distribution: $P(X_0)$
  - Transition model: $P(X_t \mid X_{t-1})$
  - Sensor model: $P(E_t \mid X_t)$



W_t	$P(U_t \mid W_t)$	
	true	false
sun	0.2	0.8
rain	0.9	0.1

HMM as probability model

– Joint distribution for Markov model: $P(X_0,\ldots, X_T) = P(X_0) \prod_{t=1:T} P(X_t \mid X_{t-1})$
– Joint distribution for hidden Markov model:
  $P(X_0,E_0,X_1,E_1,\ldots, X_T,E_T) = P(X_0) \prod_{t=1:T} P(X_t \mid X_{t-1}) P(E_t \mid X_t)$
– Future states are independent of the past given the present
– Current evidence is independent of everything else given the current state
– Are evidence variables independent of each other?



All the x_is, where i > 0, is hidden.
Referring back to the prison guard example, the x_i's would be the weather, and the E_i's would be whether someone walks in with an umbrella

Useful notation:
$X_{a:b} = X_a, X_{a+1}, \ldots, X_b$

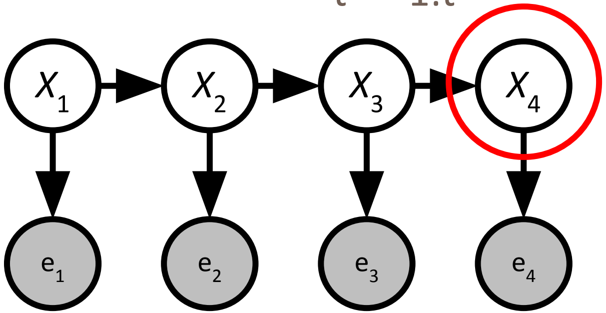
Real HMM Examples

- **Speech recognition HMMs:**
  - Observations are acoustic signals (continuous valued)
  - States are specific positions in specific words (so, tens of thousands)
- **Machine translation HMMs:**
  - Observations are words (tens of thousands)
  - States are translation options
- **Robot tracking:**
  - Observations are range readings (continuous)
  - States are positions on a map (continuous)
- **Molecular biology:**
  - Observations are nucleotides ACGT
  - States are coding/non-coding/start/stop/splice-site etc.

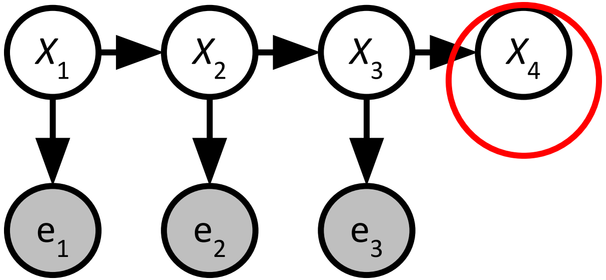
Inference tasks

Filtering: $P(X_t | e_{1:t})$

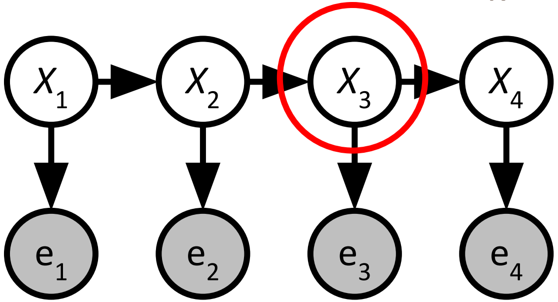
Ask Angelica about this slide.
Ask for examples.



Prediction: $P(X_{t+k} | e_{1:t})$

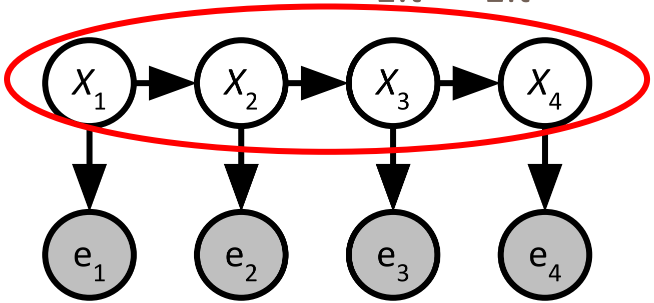


Smoothing: $P(X_k | e_{1:t})$, $k<t$



Explanation: $P(X_{1:t} | e_{1:t})$

Explanation tries to find the most probable option for what occurred.



Inference tasks

- ***Filtering***: $P(X_t | e_{1:t})$
  - ***belief state***—input to the decision process of a rational agent
- ***Prediction***: $P(X_{t+k} | e_{1:t})$ for $k > 0$
  - evaluation of possible action sequences; like filtering without the evidence
- ***Smoothing***: $P(X_k | e_{1:t})$ for $0 \leq k < t$
  - better estimate of past states, essential for learning
- ***Most likely explanation***: $\arg\max_{x1:t} P(x_{1:t} \mid e_{1:t})$
  - speech recognition, decoding with a noisy channel

Filtering / State Estimation

- Filtering, or monitoring, or state estimation, is the task of maintaining the distribution $\boldsymbol{f}_{1:t} = P(X_t | e_{1:t})$ over time

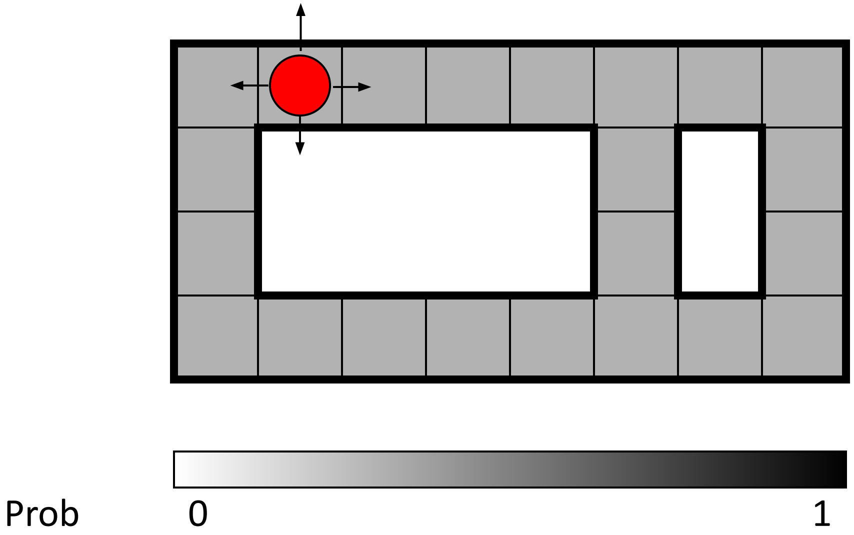
- We start with $\boldsymbol{f}_0$ in an initial setting, usually uniform

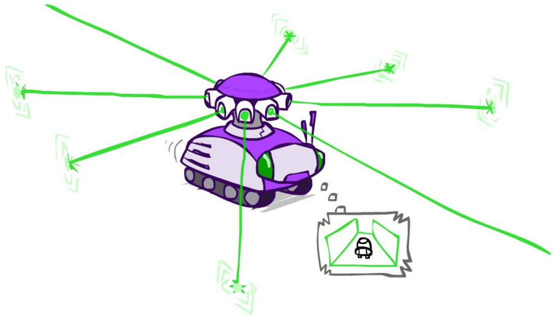
- Filtering is a fundamental task in engineering and science

- The Kalman filter (continuous variables, linear dynamics, Gaussian noise) was invented in 1960 and used for trajectory estimation in the Apollo program; core ideas used by Gauss for planetary observations; **>1,000,000** papers on Google Scholar

Example: Robot Localization

Example from
Michael Pfeiffer





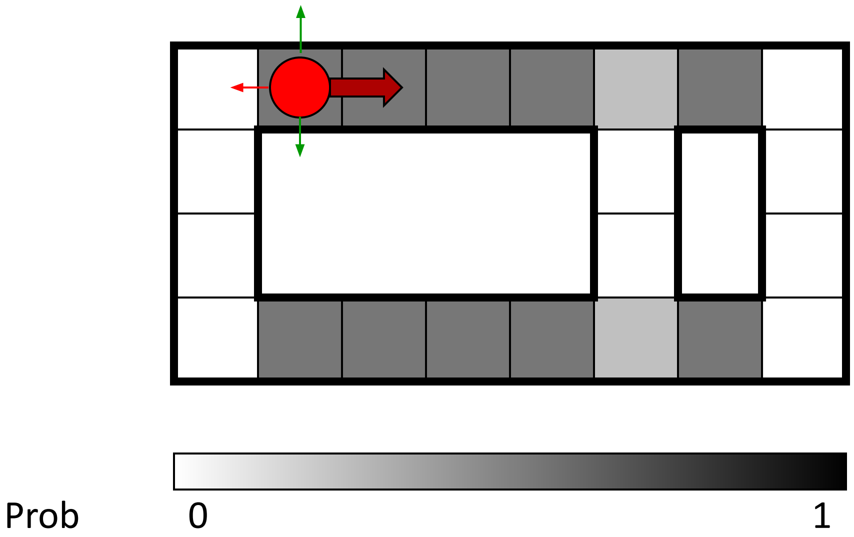
t=0

Sensor model: four bits for **wall/no-wall** in each direction,
never more than 1 mistake

At most once per turn

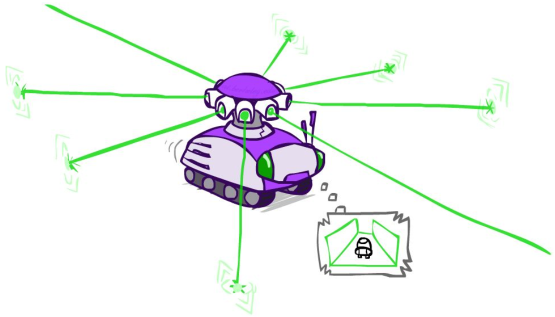
Transition model: **action may fail** with small prob.

Example: Robot Localization

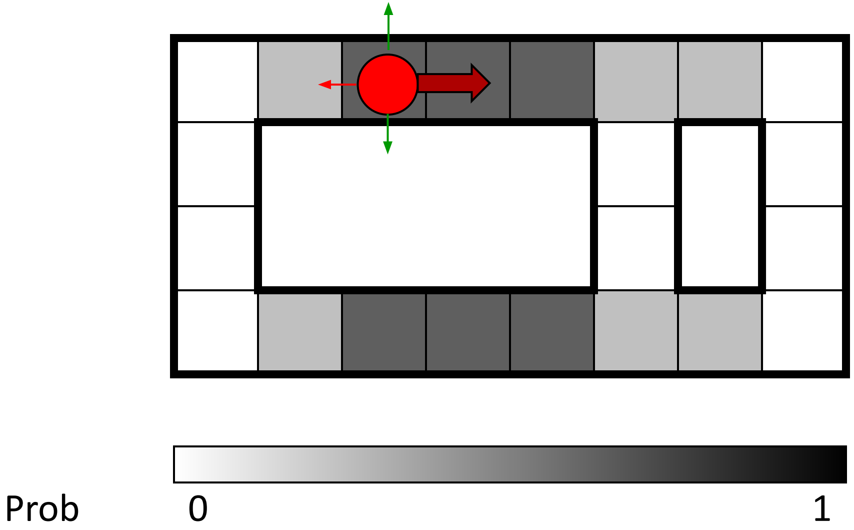


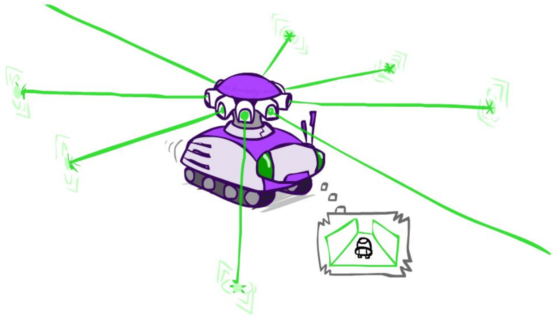
t=1

Lighter grey: was ***possible*** to get the reading, but ***less likely*** (required 1 mistake)



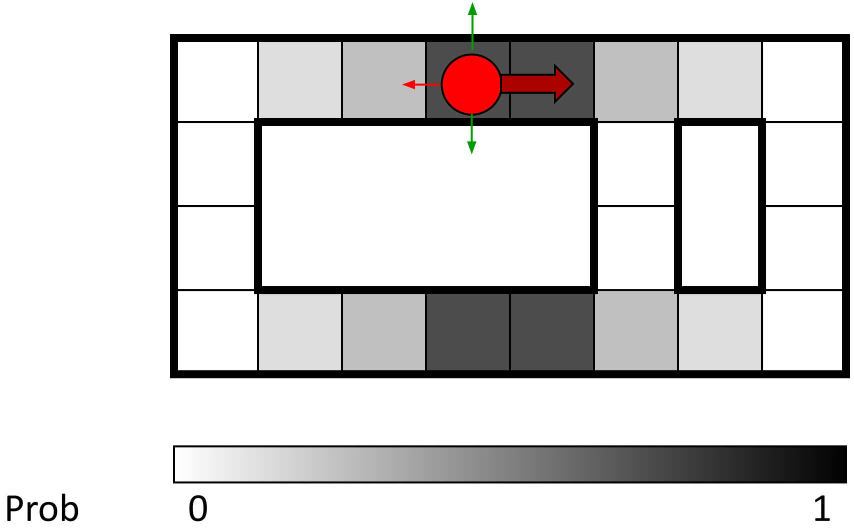
Example: Robot Localization

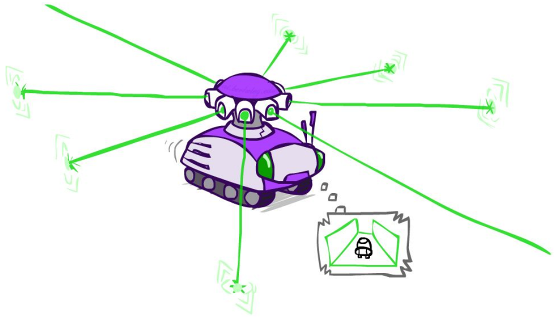




t=2

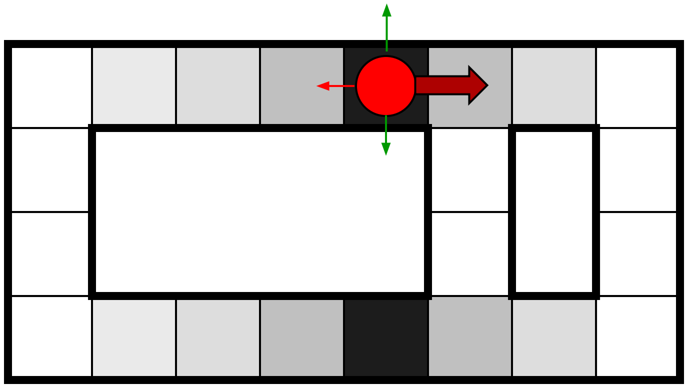
Example: Robot Localization

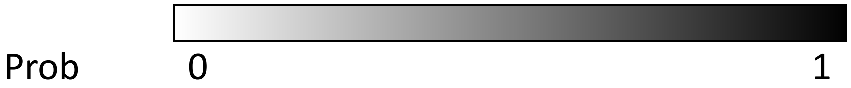




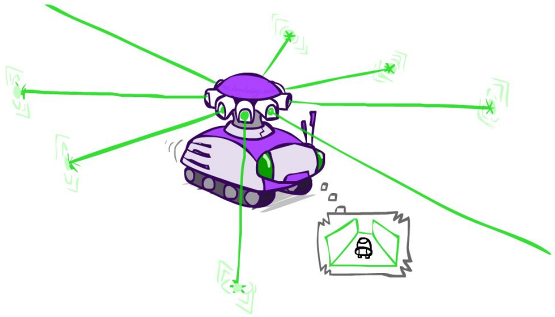
t=3

Example: Robot Localization

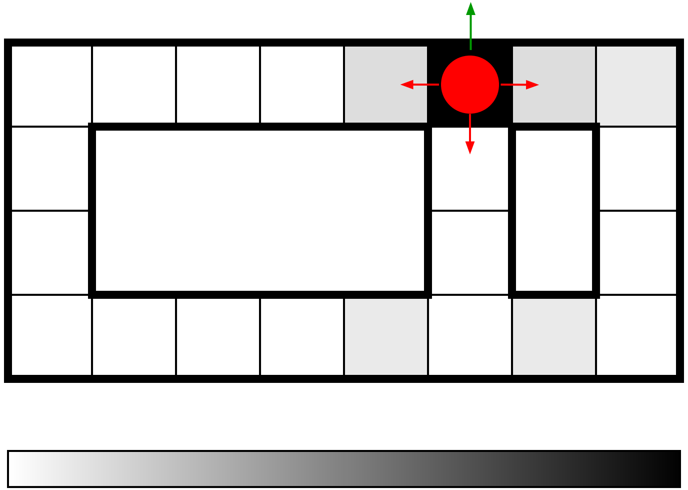




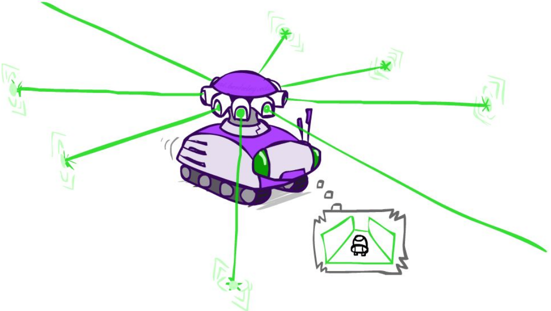
t=4



Example: Robot Localization



Prob 0 1



t=5

Example: Weather HMM

$P(R_t) = \sum_{r_{t-1}} P(R_t|r_{t-1})P(r_{t-1})$ = [0.7 0.3]*0.5 + [0.3 0.7]*0.5 = [0.5 0.5]

$P(R_t|u_t) = P(u_t|R_t)P(R_t)$ = [0.9 0.2][0.5 0.5] = [0.45 0.1] →(normalize)→0.818,0.182

In this example, let's say we observe an umbrella at time t and time t+1

f(rain) = 0.5
f(no rain) = 0.5

predict

0.5
0.5

update w/ observ.

f(rain) = 0.818
f(no rain) = 0.182

predict

0.627
0.373

update w/ observ.

f(rain) = 0.883
f(no rain) = 0.117

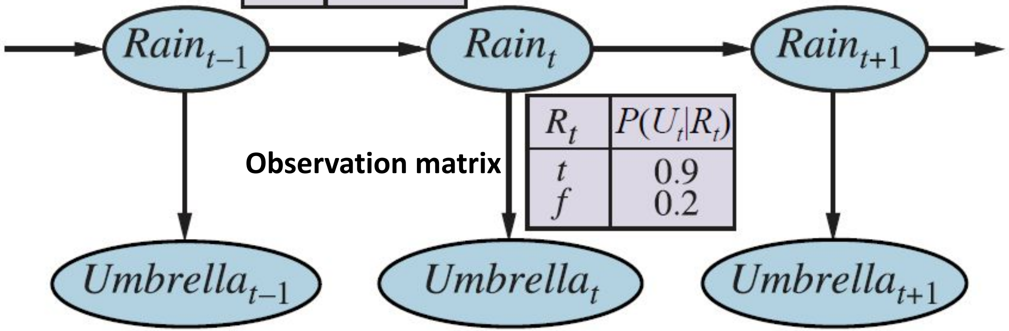
See pg. 467 in AIMA for more details

Transition matrix

R_{t-1}	$P(R_t\|R_{t-1})$
t	0.7
f	0.3

Observation matrix

R_t	$P(U_t\|R_t)$
t	0.9
f	0.2



This slide is not examinable

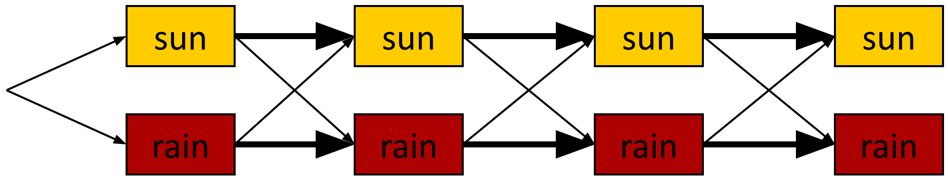
Most Likely Explanation

Inference tasks

- ***Filtering***: $P(X_t | e_{1:t})$
  - ***belief state***—input to the decision process of a rational agent
- ***Prediction***: $P(X_{t+k} | e_{1:t})$ for $k > 0$
  - evaluation of possible action sequences; like filtering without the evidence
- ***Smoothing***: $P(X_k | e_{1:t})$ for $0 \leq k < t$
  - better estimate of past states, essential for learning
- ***Most likely explanation***: $\arg\max_{x1:t} P(x_{1:t} \mid e_{1:t})$
  - speech recognition, decoding with a noisy channel

Most likely explanation = most probable path

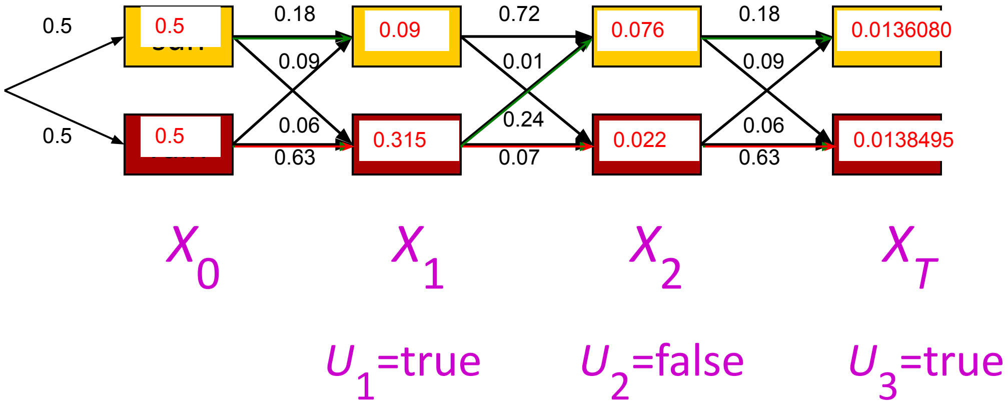
- ***State trellis***: graph of states and transitions over time



X_0 X_1 … X_T

- Each arc represents some transition $x_{t-1} \rightarrow x_t$
- Each arc has weight $P(x_t \mid x_{t-1})\, P(e_t \mid x_t)$ (arcs to initial states have weight $P(x_0)$)
- The ***product*** of weights on a path is proportional to that state sequence's probability
- Forward algorithm computes sums of paths, ***Viterbi algorithm*** computes best path

Viterbi algorithm example



Transition matrix

W_{t-1}	$P(W_t \mid W_{t-1})$	
	sun	rain
sun	0.9	0.1
rain	0.3	0.7

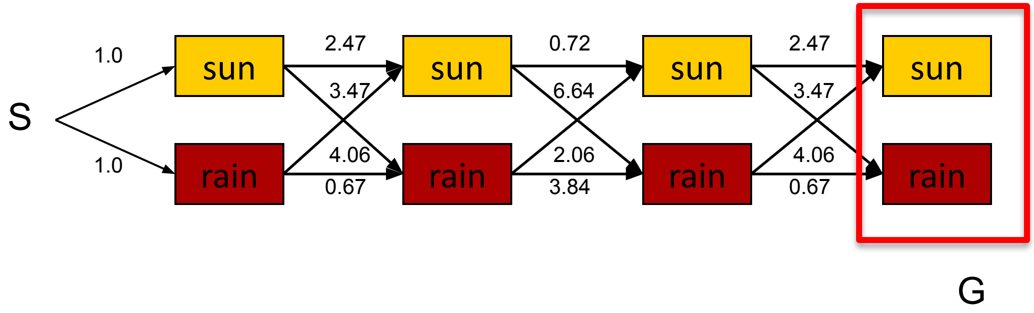
Observation matrix

W_t	$P(U_t \mid W_t)$	
	true	false
sun	0.2	0.8
rain	0.9	0.1

Viterbi chooses the step with the maximum probability at each step.
Hence, it will not look at all nodes in the graph.

This slide is not examinable

Viterbi is similar to BFS



Transition matrix

W_{t-1}	$P(W_t \mid W_{t-1})$	
	sun	rain
sun	0.9	0.1
rain	0.3	0.7

Observation matrix

W_t	$P(U_t \mid W_t)$	
	true	false
sun	0.2	0.8
rain	0.9	0.1

argmax of product of probabilities

= argmin of sum of negative log probabilities

= minimum-cost path

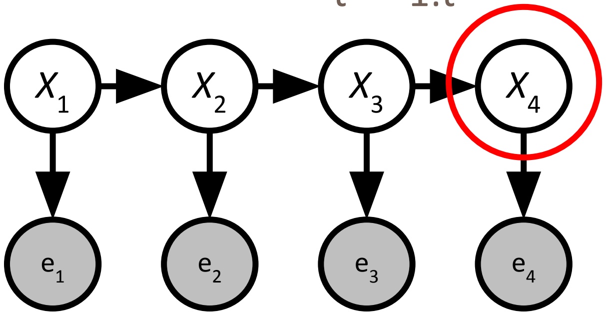
Viterbi is essentially breadth-first graph search

Recap on HMMS

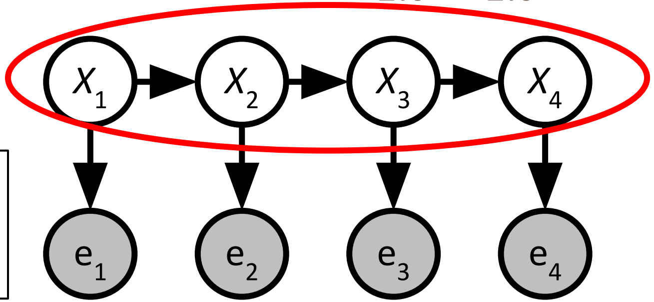
Hidden Markov Models: Inference tasks

- ***Filtering (e.g. Kalman)***: $P(X_t | e_{1:t})$
  - ***belief state***—input to the decision process of a rational agent
  - **predict** from previous step
  - **update** from current observation
- ***Most likely explanation***: $\arg\max_{x1:t} P(x_{1:t} \mid e_{1:t})$
  - speech recognition, decoding with a noisy channel
  - Incorporate observations and find most likely path

Filtering: $P(X_t | e_{1:t})$



Explanation: $P(X_{1:t} | e_{1:t})$



Key point: We might get observations (evidence) through time, like speech or the dynamics of a trajectory. We can model these as HMMs, e.g. what is my current state given transition dynamics and sensory evidence.

How does this course fit in?

CMPT 310 - Introduction to Artificial Intelligence
CMPT 353 - Computational Data Science

You are done!

CMPT 410 - Machine Learning
CMPT 420 - Deep Learning

CMPT 400 - 3D Computer Vision
CMPT 412 - Computer Vision
CMPT 413 - Computational Linguistics (NLP)
CMPT 417 - Intelligent Systems
CMPT 419 - Special topics in Artificial Intelligence

CMPT 720 - Robot Autonomy
CMPT 729 - Reinforcement Learning

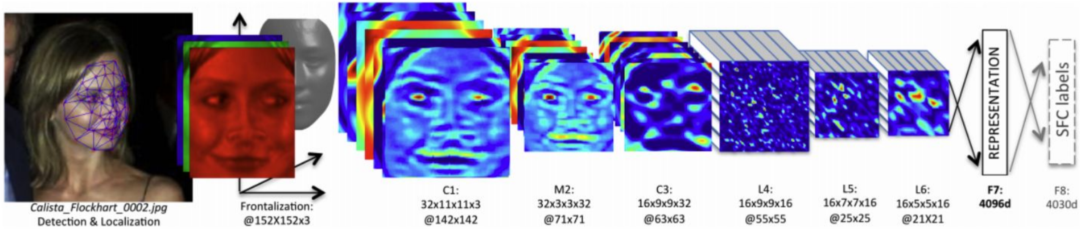
Course Review

Week 1 : Getting to know you
Week 2 : Introduction to Artificial Intelligence
Week 3: Machine Learning I: Basic Supervised Models (Classification)
Week 4: Machine Learning II: Supervised Regression, Classification and Gradient Descent, K-Means
Week 5: Machine Learning III: Neural Networks and Backpropagation
Week 6 : Search
Week 7 : Markov Decision Processes
Week 8 : Midterm
Week 9 : Reinforcement Learning
Week 10 : Games
Week 11 : Probability
Week 12 : Bayesian Networks
Week 13 : Markov Networks

Reflex-based models	State-based models	Variable-based models
Regression Classification Neural Networks	Search problems Markov decision processes Games	Probabilistic modeling Bayesian networks Markov networks

Reflex-based models

- A reflex-based model simply performs a fixed sequence of computations on a given input.
- Examples: linear classifiers, deep neural networks



- Most common models in machine learning
- Fully feed-forward (no backtracking, no considering alternative computations)
- Inference is fast

Reflex-based Models

Mathematical Foundations

- Multivariate calculus
- Partial derivatives, gradients
- Probability

Machine Learning

- Training sets, test sets
- Cross-validation
- Evaluation metrics

Unsupervised Learning

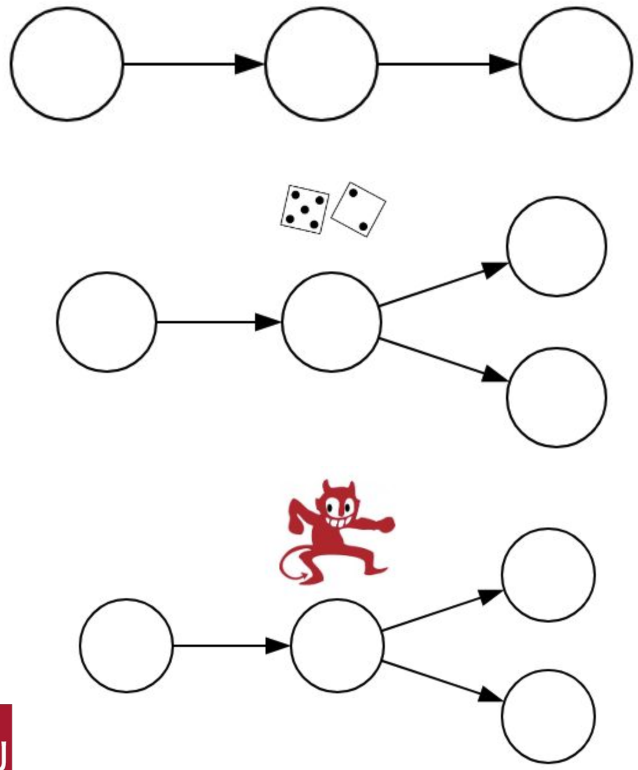
- K-Means Clustering

Supervised Learning

- Non-parametric methods:
  - K-Nearest Neighbors, Decision Trees
- Parametric methods*:
  - Hypothesis class, loss function, optimization algorithm
  - Linear regression
  - Linear classification
  - Gradient descent
  - Two-layer neural network
  - Backpropagation w/ computation graphs

**Credit: Stanford CS 221, used with permission*

State-based models



Search problems: when you have an environment with no uncertainty, ie. perfect information. But realistic settings are more complex

Markov decision processes (MDPs) handle situations with randomness, e.g. Blackjack

Game playing handles tasks where there is interaction with another agent. Adversarial games assume an opponent, e.g. Chess

State-based Models

Uninformed and Informed Search

- Search tree
- State space, state space graph
- DFS, BFS, UCS (review)
- A* search
- Heuristics, admissibility, optimality

Final will cover everything after the search lectures.

Markov Decision Processes

- Grid world
- Policies
- Discounting
- Search trees
- Valuation of states, Q-Values
- Value and Q-Value Iteration (Policy extraction)

Reinforcement Learning

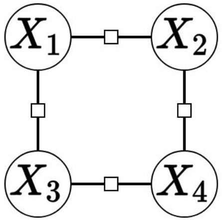
- Learning from Rewards
- Passive Reinforcement Learning
- Direct Evaluation
- Temporal Difference Learning
- Active Reinforcement Learning
- Temporal Difference Q-Learning

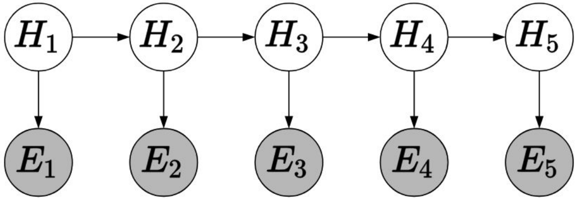
Games

- Types of games (adversarial, zero-sum etc)
- Minimax
- Limited depth search
- Evaluation functions
- Alpha-beta pruning
- Expectimax
- MCTS

Variable-based models

How is it different from a state-based model? In variable-based models, the order in which things are done is not important. Simply declare what you want, rather than micromanage how the solution is found.





Constraint satisfaction problems: hard constraints (e.g., Sudoku, scheduling)

Bayesian networks: soft dependencies (e.g., tracking cars from sensors). Variables are random variables dependent on each other, e.g. location of airplane H_3 depends on radar reading E_3 and H_2

Variable-based Models

Probability

- Marginal/conditional/joint distributions and probability
- Chain rule
- Product Rule
- Bayes Rule
- Conditional independence

Bayesian Networks

- Bayes nets
- Representation
- Independence
- Inference by enumeration using Bayes Nets
- Variable Elimination (concept / using tables)

Hidden Markov Models

- Filtering / state estimation
- Most likely explanation
- Real examples (NLP, robot localization)

Course Review

Week 1 : Getting to know you
Week 2 : Introduction to Artificial Intelligence
Week 3: Machine Learning I: Basic Supervised Models (Classification)
Week 4: Machine Learning II: Supervised Regression, Classification and Gradient Descent, K-Means
Week 5: Machine Learning III: Neural Networks and Backpropagation
Week 6 : Search
Week 7 : Markov Decision Processes
Week 8 : Midterm
Week 9 : Reinforcement Learning
Week 10 : Games
Week 11 : Probability
Week 12 : Bayesian Networks
Week 13 : Markov Networks

Reflex-based models	State-based models	Variable-based models
Regression Classification Neural Networks	Search problems Markov decision processes Games	Constraint satisfaction problems Bayesian networks Markov networks

CMPT 310 - Final Exam

- Thursday, Dec. 12 from 12-3pm in SSCB 9200. It will likely not take 3 full hours.
- Similar format to the midterm.
- Bring a pencil/eraser, SFU ID, and a <u>basic</u> calculator (no graphing or programmable calculators).
- It will focus mainly on the content after the midterm (Week 7 onward).

CMPT 310 - Course Evaluation

- Please fill out the course evaluation form. Thanks!