CAP6938-02
Plan, Activity, and Intent Recognition

Review of Material

Instructor: Dr. Gita Sukthankar
Email: gitars@eecs.ucf.edu
Schedule: T & Th 1:30-2:45pm
Location: CL1 212
Office Hours (HEC 232):
T 3-4:30pm, Th 10-11:30am

Exam Format

- Exam Oct 4th: closed-book, can bring 1 page of notes
- Oct 11th: 2 page writeup of your project results (informal in-class presentation)
- Oct 18th: Project Phase 2
  - Chance to start a new project or refine your old one
  - 1 page writeup and informal class presentation describing changes you want to make in your project

Definitely on Exam

- Specific questions on:
  - Bayes networks
  - Hidden Markov Models
    - Representation
    - Forward algorithm
- General research questions on the 5 papers (Kautz, Tambe, Pynadath, Kaminka, Starner)

Not on Exam

- Logic proofs or e-graphs
- SOAR
- Inference using stochastic grammars
- Variable elimination for loopy graphs
- Details of Baum-Welch algorithm
- Vision based tracking

What makes PAIR hard?

- High computational cost
- Plan library requirements:
  - Libraries can be incomplete or inaccurate
  - Difficult to author (making learning attractive)
  - Individual differences
  - Mistakes/irrational behavior
- Domain-specific characteristics make generalization across domains difficult
- Specific to activity recognition:
  - Identifying transitions between behavior
  - Data association
  - Obtaining reliable tracking data (vision)

Application Areas

- Robocup (not on the exam)
- Quality of Life (not on the exam)
- Adversarial reasoning for games and battlefield analysis (Tambe)
- Gesture recognition (Starner)

Symbolic (Consistency-based)

- Based on the idea that plan recognition is a consistency-checking process.
- A model matches the set of observations if the observed actions don't violate any of the constraints specified in the plan library.
- Example techniques (first 2 weeks of reading)
  - Event hierarchy circumscription (Kautz)
  - Event tracking/model tracing (Tambe)
  - Fast/complete symbolic plan recognition (Kaminka)
- Output: return complete set of models that pass consistency checking

Probabilistic (Likelihood-based)

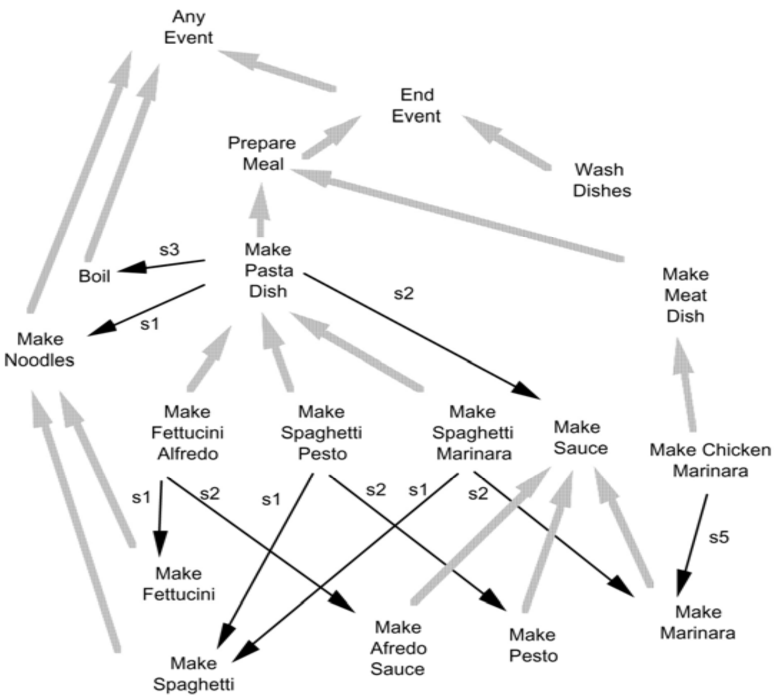
- Based on the idea of selecting the plan that has a high probability based on the observed evidence
- Belief is usually calculated using some variant on Bayesian belief update (but Dempster-Shafer evidential reasoning has also been used)
- Includes both directed/undirected graphical model based procedures
  - Examples: dynamic Bayes networks (DBNs), hidden Markov/semi-Markov models (HMMs),
- Output: model with the maximum likelihood at the current time step given the set of previous observations

Decision-theoretic (Utility-based)

- Based on the idea that the agent is rational and acts to maximize a known utility function.
- Plan recognition process occurs by calculating utility of all plans in current situation.
- Game-theory is applicable for adversarial reasoning when the agent is simultaneously trying to maximize their utility while minimizing their opponents.
- Output: a rank-ordering of models by utility
- Note: this method is well-suited for prioritizing or pruning the search process and is often used in combination with one of the previous methods

Event Hierarchy Circumscription

Event hierarchy



General axioms

$\forall x$. *MakePastaDish(x)* $\supset$

	MakeNoodles(step1(x)) $\wedge$
Components	*MakeSauce(step2(x))* $\wedge$
	Boil(step3(x)) $\wedge$
Equality	*agent(step1(x)) = agent(x)* $\wedge$
Constraints	*result(step1(x)) = input(step3(x))* $\wedge$
Temporal	*During(time(step1(x)), time(x))* $\wedge$
Constraints	*BeforeMeets(time(step1(x)), time(step3(x)))* $\wedge$
	Overlaps(time(x), postTime(x)) $\wedge$
Preconditions	*InKitchen(agent(x), time(x))* $\wedge$
	Dexterous(agent(x)) $\wedge$
Effects	*ReadyToEat(result(x), postTime(x))* $\wedge$
	PastaDish(result(x))

H. Kautz, A Formal Theory of Plan Recognition and its Implementation, in Reasoning about Plans

Kautz's Model

- First order predicate calculus
- Event hierarchy (logical encoding of a semantic network)
  - Event predicates
  - Abstaction axioms
  - Decomposition axioms
- General axioms: hardest to use for inference
  - Includes temporal constraints between the steps
  - Equality constraints between the agents executing steps or objects involved in steps
  - Preconditions
- Special event predicates*: End, AnyEvent (*top-level abstraction*)*

Kautz's Assumptions

- Exhaustiveness: Known ways of specializing an event type are the only ways of specializing it
- Disjointedness: Types are disjoint, unless one abstracts the other, or they abstract a common type
- Component/Use: Seeing an event implies the disjunction of the plans which include it as a component
- Minimum Cardinality Assumption: Assume parsimony: the minimum number of plans to explain the observations

RESC Algorithm (Tambe)

- Simple insight: model what you would do if you were in the opponent's position
- What are problems with this?
  - High overhead: must program an agent capable of solving the problem
  - Modeling the opponent's world state can be difficult (what is the opponent's sensor model?)
  - Maintaining multiple hypotheses is even more expensive
- What are the strengths?
  - Allows designer to leverage extra domain knowledge
  - Does not require enumerating chains of possible events

Ambiguity in Event Tracking

- Ambiguity: the bane of plan recognition!
- Potential solutions:
  - Maintain multiple operator hierarchies (continue considering all valid hypotheses)
  - Delay until more evidence presents itself
- Tambe solution: attempt to resolve ambiguity and commit to a single interpretation
  - Passive ambiguity resolution (game-theoretic)
  - Active resolution: modify agent’s actions to resolve ambiguity
  - Detect incorrect interpretation through match failure
  - Recovery mechanisms (assumption injection, backtracking)

Stochastic Grammars

- Refer to the shorter version of the Pynadath paper
- Understand how to represent plan recognition as a grammar parsing problem
- Difference between plan recognition using context-free and context-sensitive grammars
- Understand Pynadath's representation of the driving domain

Speedups for Plan Recognition

- Smart data structures (Kaminka)
- Use of dynamic programming (forwards-backwards algorithm, variable elimination)
- Be able to suggest new speedups
- Understand the purpose of the ones proposed in the Kaminka paper
  - Speeding observation matching (tagged feature tree)
  - Improving efficiency of current state query
  - Hypotheses graph data structure

Rules of Probability

- Product Rule

$$P(X,Y) = P(X \mid Y)P(Y) = P(Y \mid X)P(X)$$

- Marginalization

$$P(Y) = \sum_{i=1}^{n} P(Y, x_i)$$

X binary: $$P(Y) = P(Y, x) + P(Y, \bar{x})$$

Bayes Rule

$$P(H,E) = P(H \mid E)P(E) = P(E \mid H)P(H)$$

$$\boxed{P(H \mid E) = \frac{P(E \mid H)P(H)}{P(E)}}$$

$$P(h \mid e) = \frac{P(e \mid h)P(h)}{P(e,h) + P(e,\bar{h})}$$

$$= \frac{P(e \mid h)P(h)}{P(e \mid h)P(h) + P(e \mid \bar{h})P(\bar{h})}$$

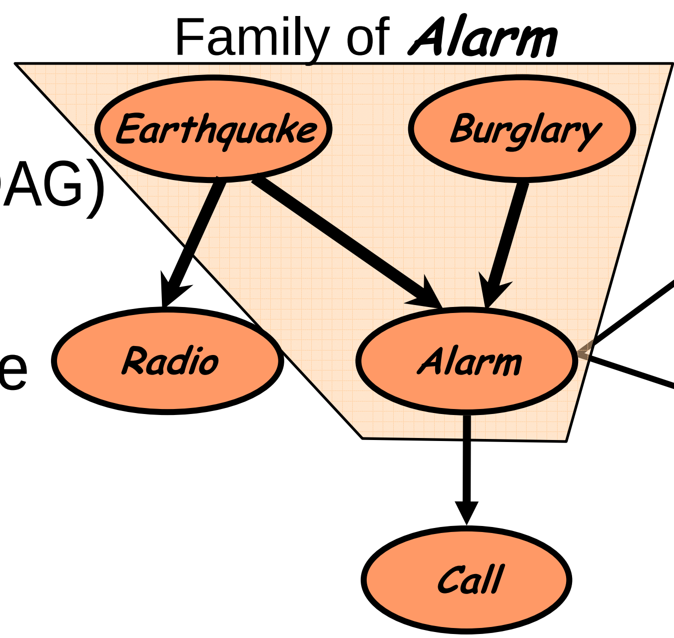
What is a Bayes (belief) net?

Compact representation of joint probability distributions via conditional independence

Qualitative part:

Directed acyclic graph (DAG)

- Nodes - random vars.
- Edges - direct influence



E	B	$P(A \mid E,B)$	
e	b	0.9	0.1
e	$\bar{b}$	0.2	0.8
$\bar{e}$	b	0.9	0.1
$\bar{e}$	$\bar{b}$	0.01	0.99

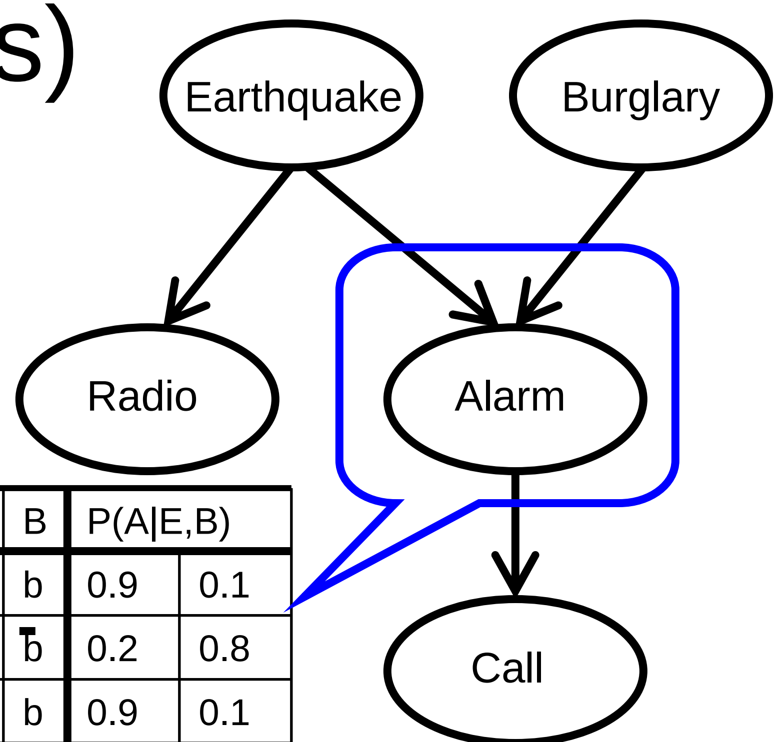
Together:
Define a unique distribution in a factored form

Quantitative part:
Set of conditional probability distributions

$$P(B,E,A,C,R) = P(B)P(E)P(A \mid B,E)P(R \mid E)P(C \mid A)$$

Figure from N. Friedman

Conditional probability distributions (CPDs)



- Each node specifies a distribution over its values given its parents values $P(X_i \mid X_{Pa_i})$
- Full table needs 2^5-1=31 parameters, BN needs 10

E	B	P(A|E,B)	
e	b	0.9	0.1
e	$\bar{b}$	0.2	0.8
$\bar{e}$	b	0.9	0.1
$\bar{e}$	$\bar{b}$	0.01	0.99

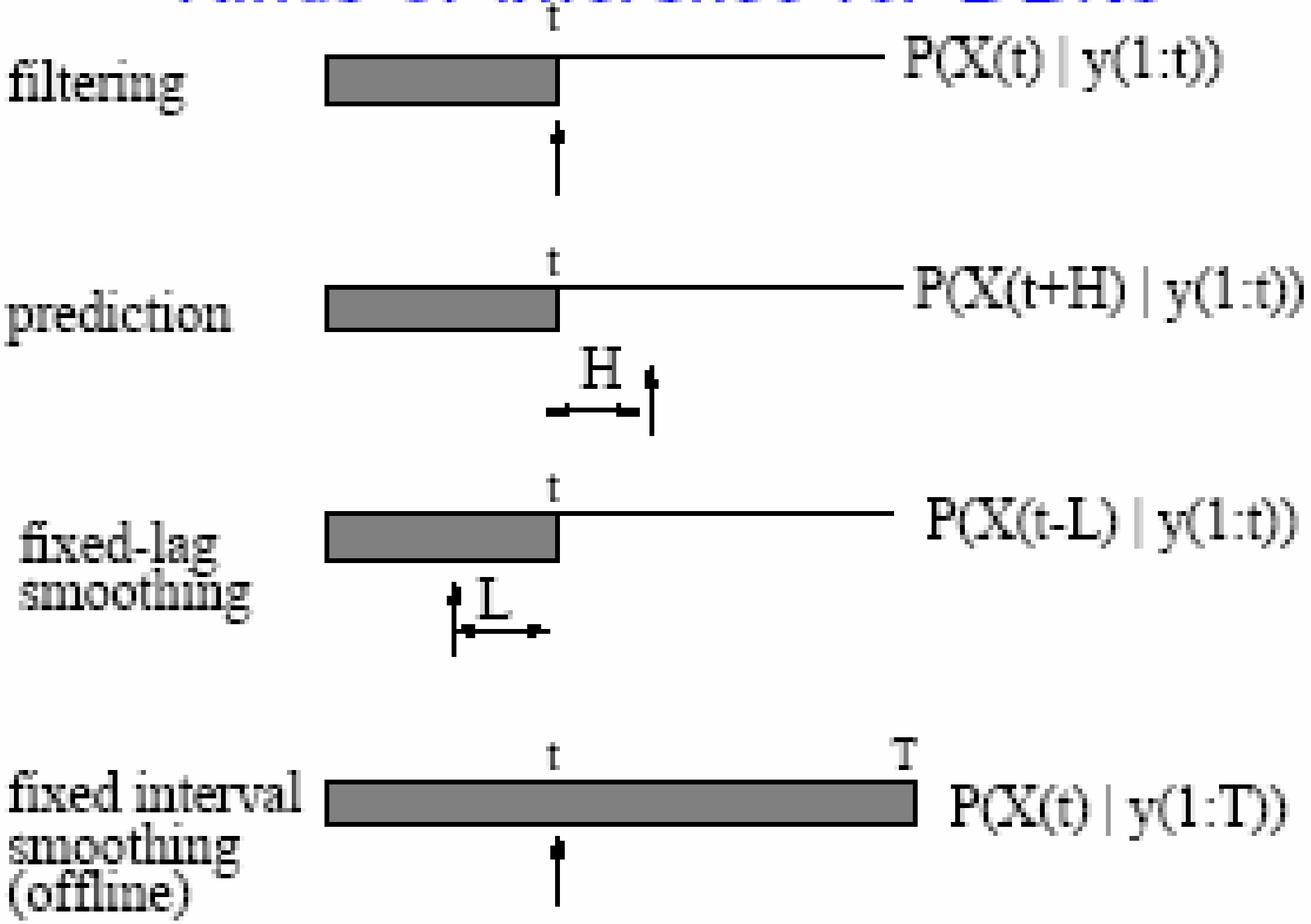
$$\underbrace{P(B,E,A,R,C)}_{31}$$

$$= \underbrace{P(B)}_{1}\underbrace{P(E|B)}_{2}\underbrace{P(A|B,E)}_{4}\underbrace{P(R|A,B,E)}_{8}\underbrace{P(C|R,A,B,E)}_{16}$$

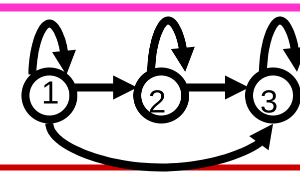
$$= \underbrace{P(B)}_{1}\underbrace{P(E)}_{1}\underbrace{P(A|B,E)}_{4}\underbrace{P(R|E)}_{2}\underbrace{P(C|A)}_{2}$$

Pearl, 1988

Kinds of inference for DBNs

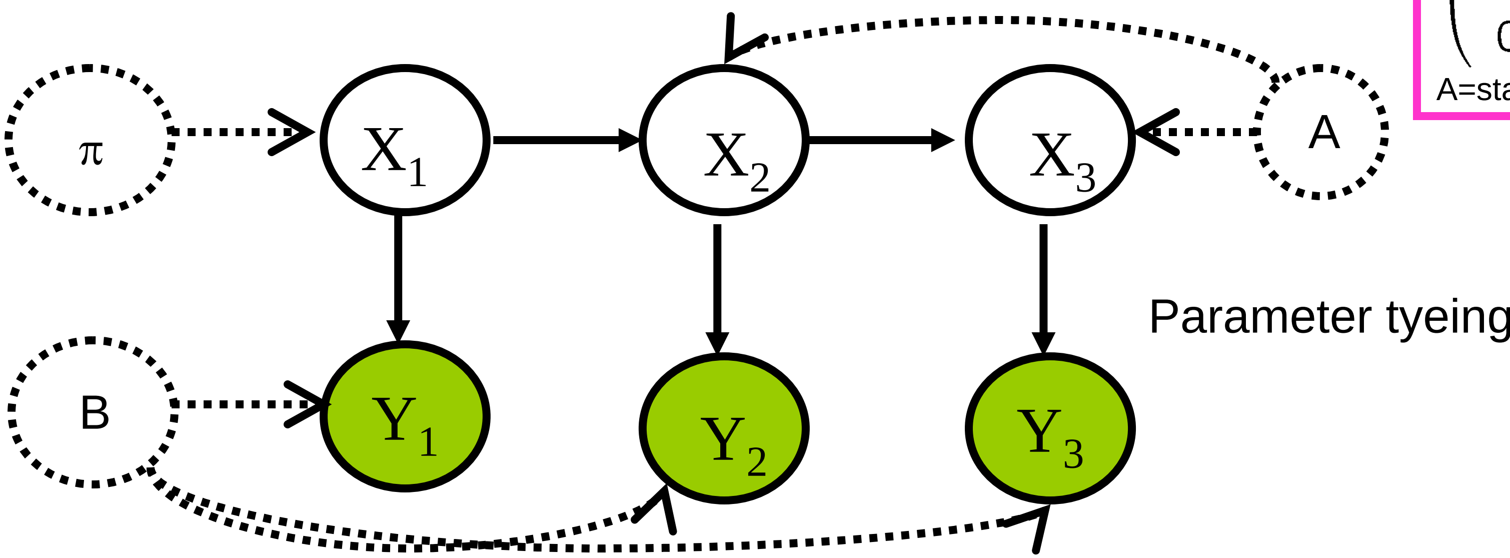


CPDs for HMMs



$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a_{22} & a_{23} \\ 0 & 0 & a_{33} \end{pmatrix}$$

A=state transition matrix



$$P(X_{1:T}, Y_{1:T}) = P(X_1)P(Y_1|X_1) \prod_{t=2}^{T} P(X_t|X_{t-1})P(Y_t|X_t)$$

Transition matrix $$P(X_t = j | X_{t-1} = i) = A(i,j)$$

Observation matrix $$P(Y_t = j | X_t = i) = B(i,j)$$

Initial state distribution $$P(X_1 = i) = \pi(i)$$

Nuisance variable=hidden node that we don't care about but that we don't know the value for

Inference tasks

- Posterior probabilities of Query given Evidence
  - Marginalize out Nuisance variables
  - Sum-product

$$P(X_Q|X_E = x_e) = \frac{\sum_{x_n} P(X_Q, x_n, x_e)}{\sum_{x_q} \sum_{x_n} P(x_q, x_n, x_e)}$$

- Most Probable Explanation (MPE)/ Viterbi
  - max-product

$$x_q^* = \arg\max_{x_q} P(x_q|x_e) = \arg\max_{x_q} P(x_q, x_e)$$

- "Marginal Maximum A Posteriori (MAP)"
  - max-sum-product

$$x_q^* = \arg\max_{x_q} P(x_q|x_e) = \arg\max_{x_q} \sum_{x_n} P(x_q, x_n, x_e)$$

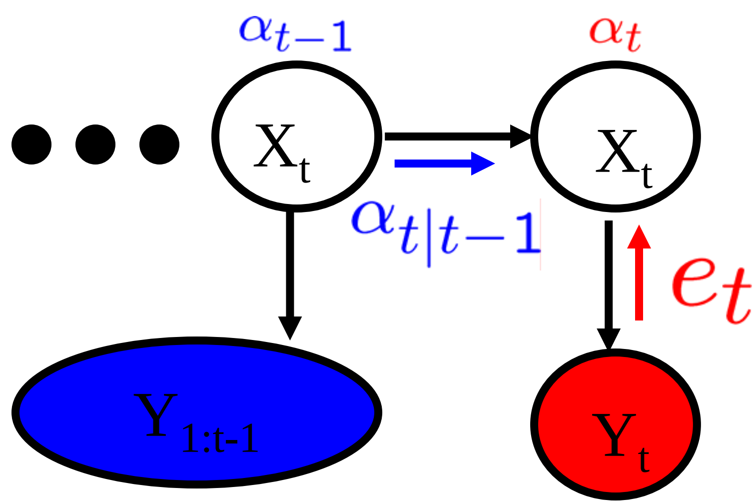
Variable/bucket elimination

Kschischang01, Dechter96

- Push sums inside products (generalized distributive law)
- Carry out summations right to left, storing intermediate results (factors) to avoid recomputation (dynamic programming)

$$
\begin{aligned}
P(b|j,m) &= \alpha \sum_e \sum_a P(b)P(e)P(a|b,e)P(j|a)P(m|a) \\
&= \alpha P(b) \sum_e P(e) \sum_a P(a|b,e)P(j|a)P(m|a)
\end{aligned}
$$

Forwards algorithm (filtering)



Use the Markov assumptions

$$
\begin{aligned}
\alpha_t(j) &\stackrel{\text{def}}{=} P(X_t = j | y_{1:t}) \\
&\propto P(y_t | X_t = j, \cancel{y_{1:t-1}}) P(X_t = j | y_{1:t-1}) \\
&= P(y_t | X_t = j) \sum_i P(X_t | X_{t-1}, \cancel{y_{1:t-1}}) P(X_{t-1} | y_{1:t-1}) \\
&= e_t(j) \sum_i A(i,j) \alpha_{t-1}(i)
\end{aligned}
$$

$$\alpha_t \propto e_t . * A^T \alpha_{t-1}$$

Gesture Recognition (Starner)

- Be able to describe how the recognition aspect of the system works
- Don’t have to understand the visual tracking
- Don’t have to understand the use of Gaussian probability densities