Intelligent Systems (AI-2)

Computer Science cpsc422, Lecture 31

Nov, 25, 2016

Slide source: from Pedro Domingos UW & Markov Logic: An Interface Layer for Artificial Intelligence Pedro Domingos and Daniel Lowd University of Washington, Seattle

Lecture Overview

- Finish Inference in MLN
  - Probability of a formula, Conditional Probability
- Markov Logic: applications
  - Entity resolution
  - Statistical Parsing! (not required, just for fun ;-)

Markov Logic: Definition

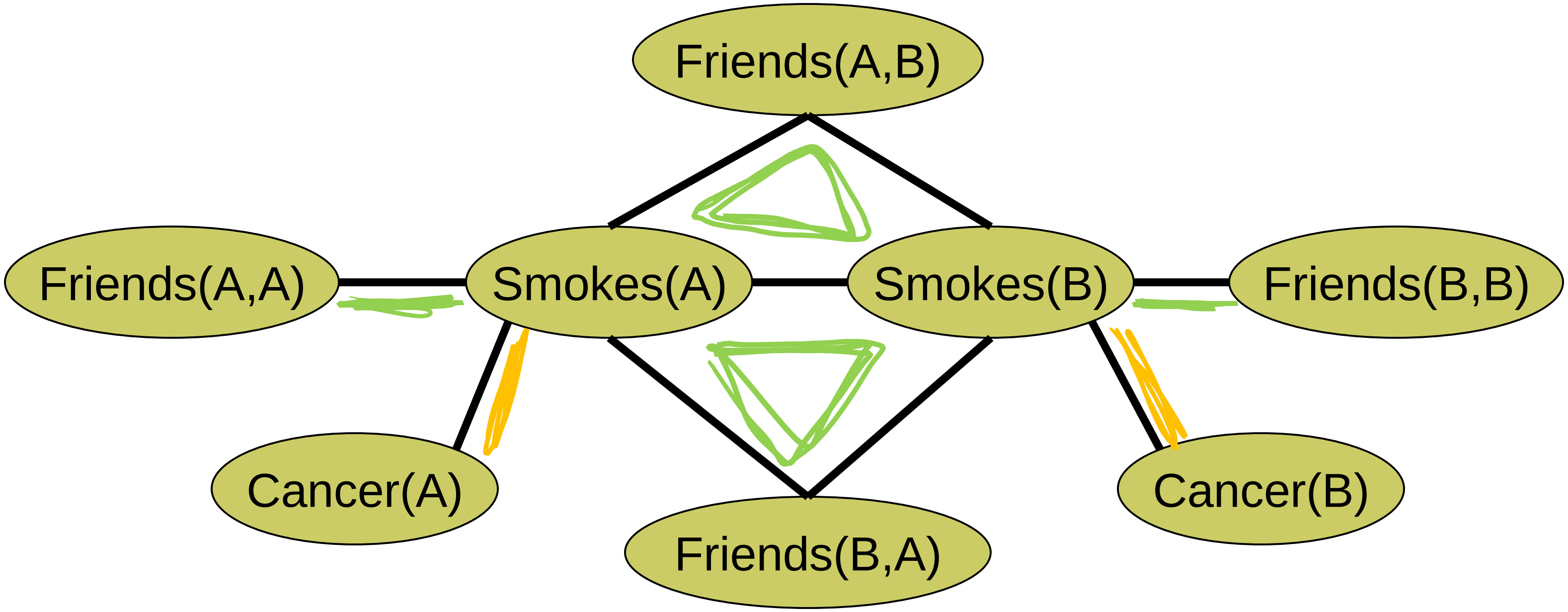
- A Markov Logic Network (MLN) is
  - a set of pairs (F, w) where
    - F is a **formula** in first-order logic
    - w is a **real number**
  - Together with a set C of **constants**,
- It defines a **Markov network** with
  - One *binary node* for each **grounding** of each **predicate** in the MLN
  - One *feature/factor* for each **grounding** of each **formula** F in the MLN, with the corresponding weight w

Grounding: substituting vars with constants

MLN features

1.5	$\forall x\ Smokes(x) \Rightarrow Cancer(x)$
1.1	$\forall x, y\ Friends(x, y) \Rightarrow \left(Smokes(x) \Leftrightarrow Smokes(y)\right)$

Two constants: **Anna** (A) and **Bob** (B)



Computing Probabilities

P(Formula,$M_{L,C}$) = ?

- **Brute force:** Sum probs. of possible worlds where formula holds

$M_{L,C}$ Markov Logic Network

PW_F possible worlds in which F is true

$$P(F, M_{L,C}) = \sum_{pw \in PW_F} P(pw, M_{L,C})$$

- **MCMC:** Sample worlds, check formula holds

S all samples

S_F samples (i.e. possible worlds) in which F is true

$$P(F, M_{L,C}) = \frac{|S_F|}{|S|}$$

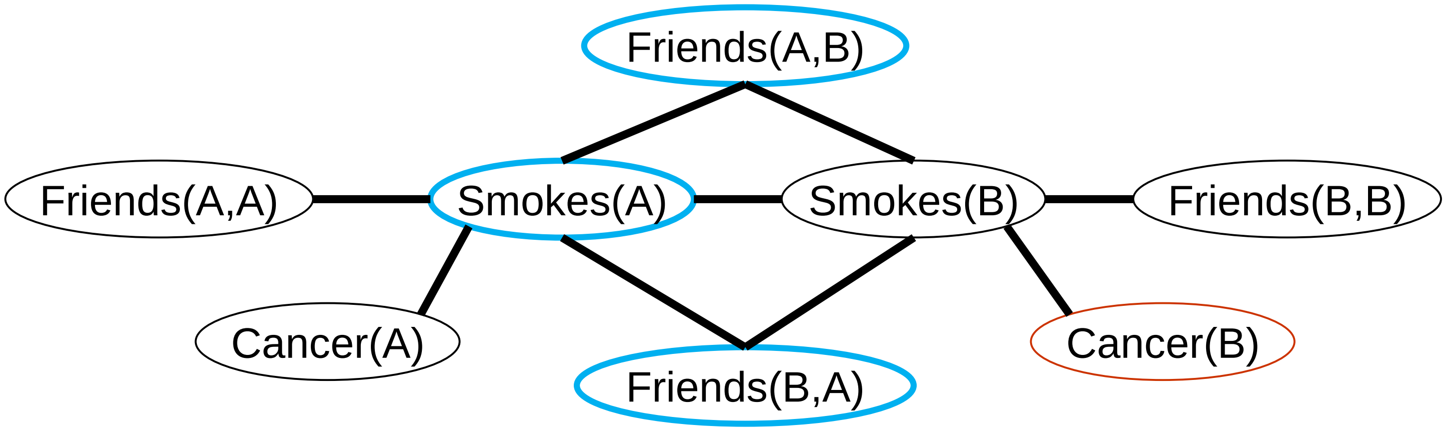
Computing Cond. Probabilities

1.5	$\forall x\ Smokes(x) \Rightarrow Cancer(x)$
1.1	$\forall x, y\ Friends(x,y) \Rightarrow \left(Smokes(x) \Leftrightarrow Smokes(y)\right)$

Let's look at the simplest case

P(ground literal | conjuction of ground literals, $M_{L,C}$)

P(Cancer(B)| Smokes(A), Friends(A, B), Friends(B, A))



To answer this query do you need to create (ground) the whole network?

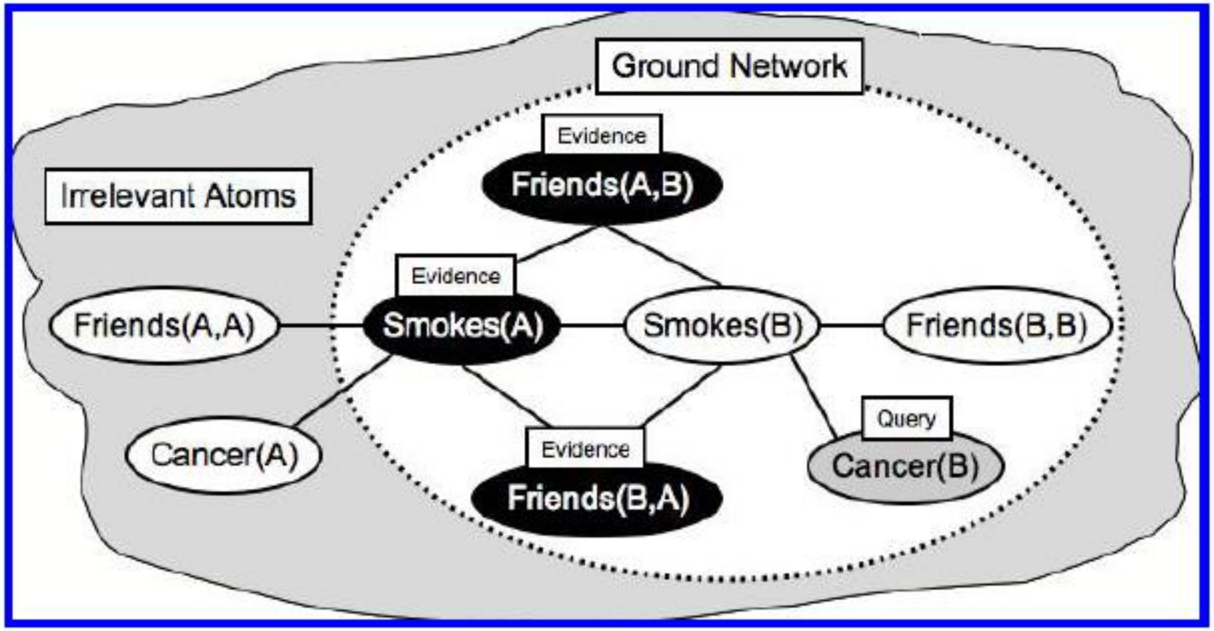
A. Yes B. No C. It depends ...

Computing Cond. Probabilities

Let's look at the simplest case

P(ground literal | conjuction of ground literals, $M_{L,C}$)

P(Cancer(B)| Smokes(A), Friends(A, B), Friends(B, A))



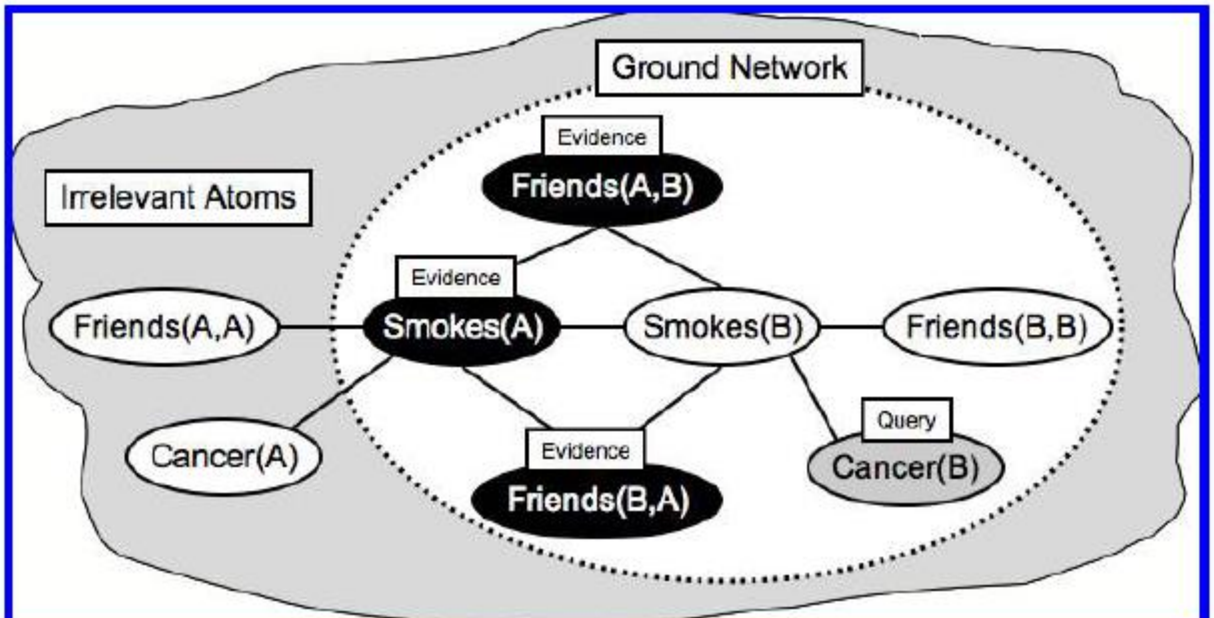
You do not need to create (ground) the part of the Markov Network from which the query is independent given the evidence

Computing Cond. Probabilities

P(Cancer(B)| Smokes(A), Friends(A, B), Friends(B, A))

The sub network is determined by the formulas (the logical structure of the problem)

1.5	$\forall x\ Smokes(x) \Rightarrow Cancer(x)$
1.1	$\forall x, y\ Friends(x, y) \Rightarrow (Smokes(x) \Leftrightarrow Smokes(y))$



You can then perform Gibbs Sampling in this Sub Network

Lecture Overview

- Finish Inference in MLN
  - Probability of a formula, Conditional Probability
- Markov Logic: applications
  - Entity resolution
  - Statistical Parsing!

Entity Resolution

- Determining which observations correspond to the same real-world objects

- (e.g., database records, noun phrases, video regions, etc)

- Crucial importance in many areas

(e.g., data cleaning, NLP, Vision)

Entity Resolution: Example

AUTHOR: *H. POON & P. DOMINGOS*
TITLE: *UNSUPERVISED SEMANTIC PARSING*
VENUE: *EMNLP-09*

SAME?

AUTHOR: *Hoifung Poon and Pedro Domings*
TITLE: *Unsupervised semantic parsing*
VENUE: *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*

AUTHOR: *Poon, Hoifung and Domings, Pedro*
TITLE: *Unsupervised ontology induction from text*
VENUE: *Proceedings of the Forty-Eighth Annual Meeting of the Association for Computational Linguistics*

SAME?

AUTHOR: *H. Poon, P. Domings*
TITLE: *Unsupervised ontology induction*
VENUE: *ACL-10*

Entity Resolution (relations)

Problem: Given citation database, find duplicate records
Each citation has `author`, `title`, and `venue` fields
We have 10 relations

provided as evidence

`Author(bib,author)`
`Title(bib,title)`
`Venue(bib,venue)`
relate citations to their fields

`HasWord(author, word)`
`HasWord(title, word)`
`HasWord(venue, word)`
indicate which words are present in each field;

To be inferred

`SameAuthor (author, author)` represent field equality;
`SameTitle(title, title)`
`SameVenue(venue, venue)`

`SameBib(bib, bib)` represents citation equality;

Entity Resolution (formulas)

Predict citation equality based on words in the fields

```
Title(b1, t1) ∧ Title(b2, t2) ∧
HasWord(t1,+word) ∧ HasWord(t2,+word) ⇒
SameBib(b1, b2)
```

1000s of rules one for each word

```
(NOTE: +word is a shortcut notation, you
actually have a rule for each word e.g.,
Title(b1, t1) ∧ Title(b2, t2) ∧
HasWord(t1,”bayesian”) ∧
HasWord(t2,”bayesian” ) ⇒ SameBib(b1, b2) )
```

Same 1000s of rules for author

Same 1000s of rules for venue

Entity Resolution (formulas)

Transitive closure

SameBib(b1,b2) ∧ SameBib(b2,b3) ⇒ SameBib(b1,b3)

SameAuthor(a1,a2) ∧ SameAuthor(a2,a3) ⇒ SameAuthor(a1,a3)

Same rule for **title**

Same rule for **venue**

Link fields equivalence to citation equivalence – *e.g., if two citations are the same, their authors should be the same*

Author(b1, a1) ∧ Author(b2, a2) ∧ SameBib(b1, b2) ⇒ SameAuthor(a1, a2)

…and that citations with the same author are more likely to be the same

Author(b1, a1) ∧ Author(b2, a2) ∧ SameAuthor(a1, a2) ⇒ SameBib(b1, b2)

Same rules for **title**

Same rules for **venue**

Benefits of MLN model

Standard non-MLN approach: build a classifier that given two citations tells you if they are the same or not, and then apply transitive closure

New MLN approach:

- performs *collective* entity resolution, where resolving one pair of entities helps to resolve pairs of related entities

e.g., inferring that a pair of citations are equivalent can provide evidence that the names *AAAI-06* and *21st Natl. Conf. on AI* refer to the same venue, even though they are superficially very different. This equivalence can then aid in resolving other entities.

Other MLN applications

- **Information Extraction**
- **Co-reference Resolution Robot Mapping** (infer the map of an indoor environment from laser range data)
- **Link-based Clustering** (uses relationships among the objects in determining similarity)
- **Ontologies extraction from Text**
- **…..**

Lecture Overview

- Finish Inference in MLN
  - Probability of a formula, Conditional Probability
- Markov Logic: applications
  - Entity resolution
  - Statistical Parsing!

Not required

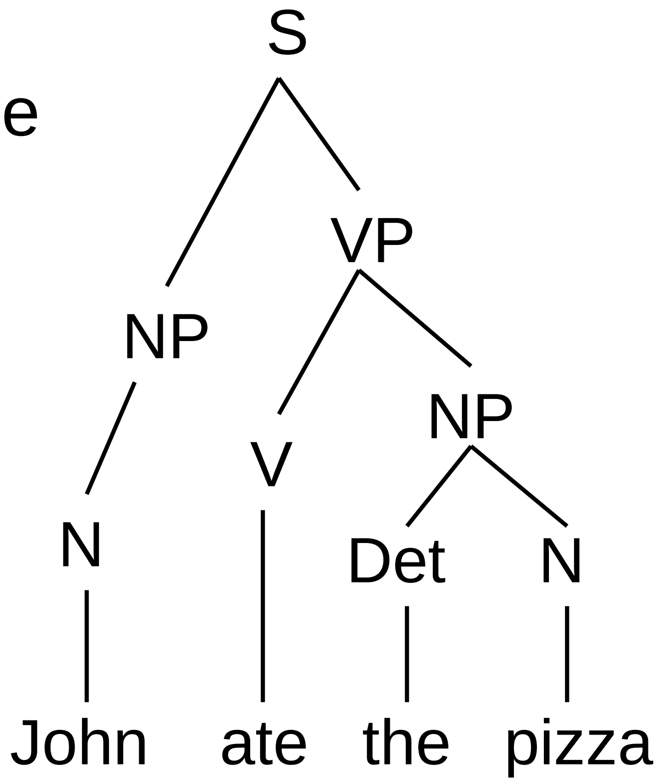
Statistical Parsing

- **Input:** Sentence
- **Output:** Most probable parse
- **PCFG:** Production rules with probabilities

  E.g.: 0.7 NP → N

  0.3 NP → Det N
- **WCFG:** Production rules with weights (equivalent)
- Chomsky normal form:

  A → B C or A → a



Logical Representation of CFG

→ rewrites as ⇒ logical implication

$$S \rightarrow NP\ VP$$

(A) NP ^ VP => S

(B) NP(i,j) ^ VP(j,k) => S(i,k)

(C) S(i,k) => NP(i,j) ^ VP(j,k)

i>clicker

Which one would be a reasonable representation in logics?

0 the 1 dog 2 chases 3 the 4 cat 5

Logical Representation of CFG

$$S \rightarrow NP\ VP$$
$$NP \rightarrow Adj\ N$$
$$NP \rightarrow Det\ N$$
$$VP \rightarrow V\ NP$$

NP(i,j) ^ VP(j,k) => S(i,k)
Adj(i,j) ^ N(j,k) => NP(i,k)
Det(i,j) ^ N(j,k) => NP(i,k)
V(i,j) ^ NP(j,k) => VP(i,k)

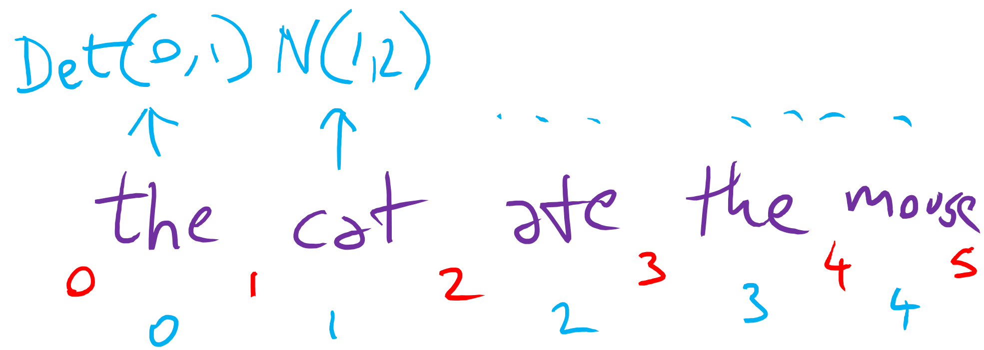
Lexicon….

// Determiners
Token("a",i) => Det(i,i+1)
Token("the",i) => Det(i,i+1)

// Adjectives
Token("big",i) => Adj(i,i+1)
Token("small",i) => Adj(i,i+1)

// Nouns
Token("dogs",i) => N(i,i+1)
Token("dog",i) => N(i,i+1)
Token("cats",i) => N(i,i+1)
Token("cat",i) => N(i,i+1)
Token("fly",i) => N(i,i+1)
Token("flies",i) => N(i,i+1)

// Verbs
Token("chase",i) => V(i,i+1)
Token("chases",i) => V(i,i+1)
Token("eat",i) => V(i,i+1)
Token("eats",i) => V(i,i+1)
Token("fly",i) => V(i,i+1)
Token("flies",i) => V(i,i+1)



Avoid two problems (1)

- If there are two or more rules with the same left side (such as **NP -> Adj N** and **NP -> Det N** need to enforce the constraint that only one of them fires :

NP(i,k) ^ Det(i,j) => ¬Adj(i,j)

``If a noun phrase results in a determiner and a noun, it cannot result in and adjective and a noun''.

Avoid two problems (2)

- **Ambiguities in the lexicon.**

homonyms belonging to different parts of speech,
e.g., Fly (noun or verb),
only one of these parts of speech should be assigned.

We can enforce this constraint in a general manner by making mutual exclusion rules for each part of speech pair, i.e.:

¬ Det(i,j) v ¬ Adj(i,j)
¬ Det(i,j) v ¬ N(i,j)
¬ Det(i,j) v ¬ V(i,j)
¬ Adj(i,j) v ¬ N(i,j)
¬ Adj(i,j) v ¬ V(i,j)
¬ N(i,j) v ¬ V(i,j)

Statistical Parsing Representation: Summary

- For each rule of the form A → B C:
  Formula of the form **`B(i,j) ^ C(j,k) => A(i,k)`**

  E.g.: `NP(i,j) ^ VP(j,k) => S(i,k)`
- For each rule of the form A → a:
  Formula of the form **`Token(a,i) => A(i,i+1)`**

  E.g.: `Token("pizza", i) => N(i,i+1)`
- For each nonterminal: state that exactly one production holds (solve problem 1)
- Mutual exclusion rules for each part of speech pair (solve problem 2)

Statistical Parsing : Inference

- **Evidence predicate:** **`Token(token,position)`**

  E.g.: **`Token("pizza", 3) etc.`**
- **Query predicates:**
  `Constituent` **`(position,position)`**

  E.g.: **`S(0,7} "is this sequence of seven words a sentence?" but also NP(2,4)`**

- What inference yields the most probable parse?

MAP!

Semantic Processing

Example: John ate pizza.

Grammar: S → NP VP   VP → V NP   V → ate
NP → John   NP → pizza

```
Token("John",0) => Participant(John,E,0,1)
Token("ate",1) => Event(Eating,E,1,2)
Token("pizza",2) => Participant(pizza,E,2,3)
```

```
Event(Eating,e,i,j) ^ Participant(p,e,j,k)
  ^ VP(i,k) ^ V(i,j) ^ NP(j,k) => Eaten(p,e)
```

```
Event(Eating,e,j,k) ^ Participant(p,e,i,j)
  ^ S(i,k) ^ NP(i,j) ^ VP(j,k) => Eater(p,e)
```

```
Event(t,e,i,k) => Isa(e,t)
```

Result: `Isa(E,Eating), Eater(John,E), Eaten(pizza,E)`

422 big picture: Where are we?

StarAI (statistical relational AI)

Hybrid: Det +Sto

Prob CFG
Prob Relational Models
Markov Logics

	Deterministic	Stochastic
Query	*Logics* *First Order Logics* *Ontologies* • Full Resolution • SAT	*Belief Nets* Approx. : Gibbs *Markov Chains and HMMs* Forward, Viterbi…. Approx. : Particle Filtering *Undirected Graphical Models* *Markov Networks* *Conditional Random Fields*
Planning		*Markov Decision Processes and Partially Observable MDP* • Value Iteration • Approx. Inference *Reinforcement Learning*

Applications of AI

Representation

Reasoning Technique

MAP MaxWalkSAT

Templates

Diff dot for MarkovBl

rewrite as ops on factors

Learning Goals for today's class

You can:

- Compute Probability of a formula, Conditional Probability
- Describe two applications of ML and explain the corresponding representations

Next Class on Mon

- **Start Probabilistic Relational Models**

Keep working on **Assignment-4**

Due **Dec 2**