Lecture Overview

- Finish Inference in MLN
  - Probability of a formula, Conditional Probability
- Markov Logic: applications
  - Entity resolution
  - Statistical Parsing!
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 \( \forall x \text{ Smokes}(x) \Rightarrow \text{Cancer}(x) \)

1.1 \( \forall x, y \text{ Friends}(x, y) \Rightarrow (\text{Smokes}(x) \Leftrightarrow \text{Smokes}(y)) \)

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

\[ P(\text{Formula}, M_{L,C}) = ? \]

- **Brute force:** Sum probs. of possible worlds where formula holds
  \[ M_{L,C} \text{ Markov Logic Network} \]
  \[ PW_F \text{ possible worlds in which F is true} \]
  \[ P(F, M_{L,C}) = \sum_{pw \in PW_F} P(pw, M_{L,C}) \]

- **MCMC:** Sample worlds, check formula holds
  \[ S \text{ all samples} \]
  \[ S_F \text{ samples (i.e. possible worlds) in which F is true} \]
  \[ P(F, M_{L,C}) = \frac{|S_F|}{|S|} \]
Let’s look at the simplest case

\[ P(\text{ground literal} \mid \text{conjunction of ground literals}, M_{L,C}) \]

\[ P(\text{Cancer}(B) \mid \text{Smokes}(A), \text{Friends}(A, B), \text{Friends}(B, A)) \]

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

\[
P(\text{ground literal} \mid \text{conjunction of ground literals, } M_{L,C})
\]

\[
P(\text{Cancer(B)} \mid \text{Smokes(A), Friends(A, B), Friends(B, A)})
\]

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

\[
P(\text{Cancer}(B) | \text{Smokes}(A), \text{Friends}(A, B), \text{Friends}(B, A))
\]

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

\[
\forall x \text{ Smokes}(x) \Rightarrow \text{Cancer}(x)
\]

\[
\forall x, y \text{ Friends}(x, y) \Rightarrow (\text{Smokes}(x) \Leftrightarrow \text{Smokes}(y))
\]

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

AUTHOR: Hoifung Poon and Pedro Domings
TITLE: Unsupervised semantic parsing

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

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

Author(bib,author) provided as evidence
Title(bib,title) relate citations to their fields
Venue(bib,venue)

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

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)

(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
Transitive closure
\[ \text{SameBib}(b_1, b_2) \land \text{SameBib}(b_2, b_3) \Rightarrow \text{SameBib}(b_1, b_3) \]

\[ \text{SameAuthor}(a_1, a_2) \land \text{SameAuthor}(a_2, a_3) \Rightarrow \text{SameAuthor}(a_1, a_3) \]

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
\[ \text{Author}(b_1, a_1) \land \text{Author}(b_2, a_2) \land \text{SameBib}(b_1, b_2) \Rightarrow \text{SameAuthor}(a_1, a_2) \]

…and that citations with the same author are more likely to be the same
\[ \text{Author}(b_1, a_1) \land \text{Author}(b_2, a_2) \land \text{SameAuthor}(a_1, a_2) \Rightarrow \text{SameBib}(b_1, b_2) \]

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!
Statistical Parsing

- **Input:** Sentence
- **Output:** Most probable parse
- **PCFG:** Production rules with probabilities
  - E.g.: 0.7 \( NP \rightarrow N \)
  - 0.3 \( NP \rightarrow \text{Det} N \)
- **WCFG:** Production rules with weights (equivalent)
- **Chomsky normal form:**
  - \( A \rightarrow B C \) or \( A \rightarrow a \)
Logical Representation of CFG

S → NP VP

rewrites as

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

NP ^ VP => S

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

Which one would be a reasonable representation in logics?

the dog chases the cat
Logical Representation of CFG

\[
\begin{align*}
S & \rightarrow NP \ VP \\
NP & \rightarrow Adj \ N \\
NP & \rightarrow Det \ N \\
VP & \rightarrow V \ NP
\end{align*}
\]

\[
\begin{align*}
NP(i,j) \land VP(j,k) & \implies S(i,k) \\
Adj(i,j) \land N(j,k) & \implies NP(i,k) \\
Det(i,j) \land N(j,k) & \implies NP(i,k) \\
V(i,j) \land NP(j,k) & \implies VP(i,k)
\end{align*}
\]
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 $\text{NP} \rightarrow \text{Adj} \text{ N}$ and $\text{NP} \rightarrow \text{Det} \text{ N}$) need to enforce the constraint that only one of them fires:

\[
\text{NP}(i,k) \land \text{Det}(i,j) \Rightarrow \neg \text{Adj}(i,j)
\]

``If a noun phrase results in a determiner and a noun, it cannot result in an 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.:

\[ \neg (A \land B) \]

\[ \neg \text{Det}(i,j) \lor \neg \text{Adj}(i,j) \]
\[ \neg \text{Det}(i,j) \lor \neg \text{N}(i,j) \]
\[ \neg \text{Det}(i,j) \lor \neg \text{V}(i,j) \]
\[ \neg \text{Adj}(i,j) \lor \neg \text{N}(i,j) \]
\[ \neg \text{Adj}(i,j) \lor \neg \text{V}(i,j) \]
\[ \neg \text{N}(i,j) \lor \neg \text{V}(i,j) \]
Statistical Parsing
Representation: Summary

- For each rule of the form $A \rightarrow B \ C$:
  Formula of the form $B(i,j) \ ^{\land} \ C(j,k) \Rightarrow A(i,k)$
  E.g.: $NP(i,j) \ ^{\land} \ VP(j,k) \Rightarrow S(i,k)$

- For each rule of the form $A \rightarrow a$:
  Formula of the form $Token(a,i) \Rightarrow A(i,i+1)$
  E.g.: $Token(\text{"pizza"}, i) \Rightarrow N(i,i+1)$

- 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:**

\[ S \rightarrow NP \ VP \quad VP \rightarrow V \ NP \quad V \rightarrow \text{ate} \]
\[ NP \rightarrow \text{John} \quad NP \rightarrow \text{pizza} \]

Token(“John”, 0) \Rightarrow Participant(John, E, 0, 1)
Token(“ate”, 1) \Rightarrow Event(Eating, E, 1, 2)
Token(“pizza”, 2) \Rightarrow Participant(pizza, E, 2, 3)

Event(Eating,e,i,j) ^ Participant(p,e,j,k)
^ VP(i,k) ^ V(i,j) ^ NP(j,k) \Rightarrow Eaten(p,e)

Event(Eating,e,j,k) ^ Participant(p,e,i,j)
^ S(i,k) ^ NP(i,j) ^ VP(j,k) \Rightarrow Eater(p,e)

Event(t,e,i,k) \Rightarrow Isa(e,t)

**Result:** Isa(E, Eating), Eater(John, E), Eaten(pizza, E)
422 big picture: Where are we?

Deterministic

Stochastic

Logics

First Order Logics

Ontologies

Query

Planning

Applications of AI

Reinforcement Learning

Value Iteration

Approx. Inference

Markov Networks

Conditional Random Fields

Markov Decision Processes and Partially Observable MDP

Forward, Viterbi...

Approx. : Particle Filtering

Belief Nets

Approx. : Gibbs

Markov Chains and HMMs

Undirected Graphical Models

Markov Logics

StarAI (statistical relational AI)

Hybrid: Det + Sto

Prob CFG

Prob Relational Models

Markov Logics

Diff dot for Markov81

rewrite as ops on factors

Approx. : Particle Filtering

Representation

Reasoning Technique
Learning Goals for today’s class

You can:

• Compute Probability of a formula, Conditional Probability
• Describe the entity resolution application of ML and explain the corresponding representation
Next Class on Mon

- Start Probabilistic Relational Models

Keep working on Assignment-4
Due Dec 1

In the past, a similar hw took students between 8 - 15 hours to complete. Please start working on it as soon as possible!