Natural Language Processing for Knowledge Representation and Reasoning

Michaël Thomazo

April 14th, 2014
Dresden

Today

- HMM approach for part of speech tagging;
- **introduction to Markov logic.**

What to think of the following knowledge base?

- smokes(*Alice*)
- smokes(*Bob*)
- cancer(*Alice*)
- $\neg$ cancer(*Bob*)
- dog(*Snoopy*)
- cat(*Garfiled*)
- $\forall x$ smokes(x) $\rightarrow$ Cancer(x)

A possible adaptation

- smokes($Alice$)
- smokes(Bob)
- cancer($Alice$)
- $\neg$ cancer(Bob)
- dog($Snoopy$)
- cat($Garfiled$)
- $\forall x$ smokes(x) $\rightarrow$ Cancer(x) with weight p.

Markov Random Field

Given an undirected graph $G = (V, E)$, a set of random variables $X = (X_v)_{v \in V}$ forms a Markov random field if it satisfies the following Markov property:

X_A is independent from X_B knowing X_S if S is an (A, B)-separator.

This is the **global Markov property**.

Other Markov Properties

Pairwise Markov property:

X_u is independent from X_v knowing all other variables if u and v are non-adjacent.

Local Markov property:

X_u is independent from all other variables knowing its neighbors.

Clique potentials

The joint probability probability of

$$P(X = x) = \frac{1}{Z} \Pi_k \phi_k(x_{\{k\}})$$

where ϕ_k are potential functions (non-negative real-valued), and $x_{\{k\}}$ represent the state of the k^{th} clique.

Hammersley-Clifford Theorem: every Markov Network may be represented by clique potentials.

Log-linear models

Clique potentials are represented by the sum of weighted features.

$$P(X = x) = \frac{1}{Z} \Sigma_j w_j (f_j(x))$$

We will usually consider binary features.

Markov Logic

A Markov logic network L is a set of pairs (F_i, w_i) where F_i is a formula in first-order logic and w_i is a real number. Together with a finite set of constants $C = \{c_1, \ldots, c_k\}$, it defines a Markov network $M_{L,C}$ defined as follows:

- $M_{L,C}$ contains one binary node for each possible grounding of each predicate appearing in L. The value of the node is 1 if the ground atom is true, 0 otherwise;
- $M_{L,C}$ contains one binary feature for each possible grounding of each formula F_i in L. The value of this feature is 1 if the ground formula is true, and 0 otherwise. The weight of the feature is the weight of the associated formula.

Probability distribution for Markov Logic Networks

$$P(X = x) = \frac{1}{Z} \exp(\Sigma_i w_i n_i(x))$$

where $n_i(x)$ is the number of true groundings of F_i in x.

An example of Markov Logic Network

Represent the MLN associated with the following formulas and the domain $\{A, B\}$:

- $\forall x$ smoke$(x) \Rightarrow$ Cancer(x) with weight 1.5
- $\forall x \forall y$ friend$(x, y) \wedge$ smoke$(x) \Rightarrow$ smoke(y) with weight 1.1

Assumptions

- C is of finite size
- different constants refer to different objects
- the only objects in the domain are those representable using the constant and function symbols in (L, C)
- for each function appearing in L, the value of that function applied to every possible tuple is known, and is an element of C

What to think about these assumptions?

Link with classical entailment

Consider the following formula: $\forall x R(x) \Rightarrow S(x)$ of weight w and a single constant A.

- what are the probabilities of each world?
- what is the probability of $S(A)$ knowing $R(A)$?

Any observation? Prove this in the general case of satisfiable knowledge bases.

Inference of Most Probable Explanations

We are given a set x of observations (truth value of some ground atoms).
What is the state of the world that is the most probable given these observations?

WalkSat and MaxWalkSat

Selman, Kautz, and Cohen, 1996

Intutition: start from a random assignment of the variables. Picks randomly an unsatisfied clause, and pick either randomly an atom to flip, or choose the one maximizing the sum of the weight of the satisfied clauses. Repeat the picking and flipping process. Repeat the whole process.

WalkSat and MaxWalkSat

Selman, Kautz, and Cohen, 1996

Intutition: start from a random assignment of the variables. Picks randomly an unsatisfied clause, and pick either randomly an atom to flip, or choose the one maximizing the sum of the weight of the satisfied clauses. Repeat the picking and flipping process. Repeat the whole process.

Why is randomness used here?

A strong limit of the approach

- the grounding step makes the size of the formula exponential
- this has an impact on the running time and on the quality of the approximation

Possible approaches:

- lazy reasoning
- lifted reasoning

Lifted reasoning

Instead of grounding formulas to get back to propositional logic, work directly on first-order formulas. Most of the probabilistic approaches may be lifted (to some extent). We study here an example of lifting for MaxWalkSat.

- first a "full lift" for monadic first-order logic
- adaptation to first-order logic

Monadic FO

We use only predicates of arity 1.
Example:

$$\mathrm{Smokes}(x) \rightarrow \ \mathrm{Cancer}(x)$$

From First Order to Monadic

Remember our assumptions about the fixed given domain. How can you go from first-order logic to Monadic logic?

Lifted reasoning for Monadic FO

Sarkhel, Gogate, AAAI'13

Property

If two worlds ω_1 and ω_2 of a monadic MLN are such that:

- they differ only on truth assignments to the grounding of a predicate S
- the number of true groundings of S is the same in both ω_1 and ω_2

then ω_1 and ω_2 are equiprobable.

Lifted reasoning for Monadic FO

Sarkhel, Gogate, AAAI'13

Property

If two worlds ω_1 and ω_2 of a monadic MLN are such that:

- they differ only on truth assignments to the grounding of a predicate S
- the number of true groundings of S is the same in both ω_1 and ω_2

then ω_1 and ω_2 are equiprobable.

Any comment?

Some applications of the framework

- entity resolution
- information extraction
- co-reference resolution

More details on these next week.