CSE 579: Knowledge Representation & Reasoning

Module 6: KRR with Uncertainty

Ali Altunkaya
Spring 2023

Outline

1. Basics of Probability
    1. Random Variables & Possible Worlds
    2. Joint Distribution & Marginalization
    3. Conditioning
    4. Bayes Rule & Product Rule
    5. Marginal Independence & Conditional Independence
2. Bayesian Networks
3. Markov Logic
4. Language LP^{MLN}

Overview of Module 6

Overview

- Basics of Probability
- What is Bayesian Networks? How Bayesian Networks utilize conditional independence?
- Markov Logic (combines FOL Logic + Probability)
- LPMLN (combines ASP + Markov Logic) is a formalism, a probabilistic extension of ASP.
- How to compute probabilistic stable models of LPMLN.
  - first turn LPMLN into ASP program
  - then apply LPMLN inference on probabilistic reasoning

http://reasoning.eas.asu.edu/lpmln/Tutorial.html

1. Basics of Probability

1-) Basic Axioms:

- $0 \leq P(A) \leq 1$ → A probability is between 0 and 1.
- P(tautology) = 1
- $P(A \vee B) = P(A) + P(B)$ → If A and B are "mutually exclusive" which means they cannot happen at the same time.

2-) Random Variable & Possible Worlds:

- A random variable X, can have a set of values it can take with probability $p(X_i)$.
  - $\sum p(X_i) = 1$
- A possible world w, specifies an assignment to each random variable in the world. (very similar to interpretations)
  - For example, a dice can take six values:

$$\sum_{i=1}^{6} p(X_i) = 1$$

1. Basics of Probability

3-) Joint Probability Distribution:

- $P(X \wedge Y)$ → Joint probability: if X and Y happens at the same time.
- Joint probability distribution, contains all possible pairs of outputs for X and Y
  - $\sum p(X=x_i, Y=y_j) = 1$

4-) Marginalized Probability: Given joint probability distribution, we can compute distributions over smaller sets of variables.

- Summing out a dimension in the JPD table.

1. Basics of Probability

5-) Conditional Probability: Probability of an event, given a condition.

- P(temp=hot | weather=sunny) → what is the probability of temp=hot, given that weather is sunny.
- $P(t=hot \mid w=sunny) = P\ (t=hot \wedge w=sunny) / P(w=sunny)$
- $P(h|e) = P\ (h \wedge e) / P(e)$

6-) Bayes Rule: Bayes rule utilizes conditional probability.

- $P(h \mid e) = P(h \wedge e) / P(e)$ can be written as below in two forms:
- $P\ (h \wedge e) = P(h \mid e)\ P(e)$
- $P\ (e \wedge h) = P(e \mid h)\ P(h)$
- Since $P\ (h \wedge e) = P\ (e \wedge h)$ then:
  - $P(h \mid e)\ \ P(e) = P(e \mid h)\ P(h)$
  - $P(h \mid e) = P(e \mid h)\ P(h) / P(e)$ → Bayes Rules

1. Basics of Probability

7-) Product Rule (aka Chain Rule): How to compute joint probability of many random variables using conditional probabilities.

- $p(f_1 \wedge f_2) = p(f_1)\ p(f_2 \mid f_1)$
- $p(f_1 \wedge f_2 \wedge f_3) = p(f_1)\ p(f_2 \mid f_1)\ p(f_3 \mid f_1, f_2)$
- $p(f_1 \wedge f_2 \ldots\ f_n) = p(f_1)\ p(f_2 \mid f_1)\ p(f_3 \mid f_1, f_2) \ldots\ldots\ldots p(f_n \mid f_1, f_2, \ldots, f_{n-1})$

8-) Marginal Independence: If value of Y has no effect on value of X then

- $P(X=x_i, Y=y_j) = P(X=x_i)$

9-) Conditional Independence: If X is independent of Y given Z, then

- $P(X=x_i \mid Y=y_j, Z=z_k) = P(X=x_i \mid Z=z_k)$

10-) Exploiting Conditional Independence: Product (Chain) Rule doesn't help us to reduce the size of joint-distribution table, which is 2^n.

- Using conditional independence, we can reduce the size of JD table and can gain compactness.
- For example: if the probability of next letter depends only the current letter.
- Another example will be covered in Bayesian Networks.

2. Bayesian Networks

Bayesian Networks have two components:

1) DAG (Directed Acyclic Graph) where

- each node corresponds to a random variable
- each edge indicates a direct influence between random variables using CPD table

2) CPD (Conditional Probability Distribution) table shows the probability of a random variable given its parents.

- $P(X_i \mid Parent(X_i))$ where $Parent(X_i)$ are parents of X_i .
- Independence Assumption: Each random variable is independent of its non-descendants, given its parents.
- We can achieve compact representation in computing chain rule on Bayesian Networks, due to Independence Assumption.
  - $P(X_1, X_2, \ldots, X_n) = \prod P(X_i \mid Parent(X_i))$

2. Bayesian Networks

Types of Inferences on Bayesian Networks:

- Diagnostic
- Predictive
- Mixed
- Inter-causal
- Etc...

We can compute all types of inferences using the same Chain Rule and Conditional probability as below:

- $P(X_1,X_2, \ldots, X_n) = \prod P(X_i \mid Parent(X_i))$

3. Markov Logic

Combines Logic and Probability. Because we need it, for example as seen below, not all smokers get cancer.

- Cancer(x) ← Smokes(x)

So we assign weight to each rule. We give more weight to important rules/formulas.

- Head ← Body (hard rule)
- w Head ← Body (soft rule, because there is a weight for each rule)

Log probability of a world is proportional with the sum of weights.

- log P(world) ↔ ∑ weights of rules see example below
  - log P(0.2 * 0.4) ↔ ∑ (0.2 + 0.4) equals 3.6 ↔ 0.6
  - log P(0.2 * 0.4 * 0.2) ↔ ∑ (0.2 + 0.4 + 0.2) equals 5.9 ↔ 0.8
- It can be also written as below
  - P(world) ↔ exp(∑ weights of rules)

4. LPMLN

LPMLN is a formalism, which combines ASP with Markov Logic Networks.

Syntax:

- Each rule has a weight, which indicates the importance of a rule. It is either:
  - a real number such as 1, 2, 3 etc. or
  - an infinite weight (α)
- Variables are grounded with the domain values.

Semantics:

- Soft stable model: Each interpretation (or a model) has a weight too, using the sum of log probabilities.
- We can calculate the probability of each model, and make ranking. So we can see which model is better or more important.

Reward-based weight is proportional with Penalty-based weight:

- Reward-based weights: Counting rules that are true under the interpretation.
- Penalty-based weights: Counting rules that are false under the interpretation. Also add minus.

4. LPMLN

What is the difference between MLN and LPMLN?

- MLN (FOL + Probabilities)
- LPMLN (ASP + Probabilities)

LPMLN relation to other Languages.

1-) ASP → LPMLN:

- ASP can be considered as a special case of LPMLN where each rule weight is infinite.
- If Π has at least one deterministic stable model in ASP, then all probabilistic stable models of P(Π) has the same probability in LPMLN.
- If Π has no deterministic stable model in ASP, then P(Π) can still have some probabilistic stable models in LPMLN.

2-) MLN → LPMLN:

- We can convert MLN program into LPMLN program, by adding choice rule for every ground atom. So both programs will have the same stable models.
  - Since we add choice rule, we can easily satisfy “minimal model” stuff...

4. LPMLN

LPMLN relation to other Languages.

3-) LPMLN → MLN:

- We can convert LPMLN program into MLN program via "completion" process (converting ASP program into Propositional Formula) in two steps:
  - For each atom, add a new rule, by changing the direction of implication, collecting all bodies in the same rule, and head is the atom.
- Completion process does not work on non-tight programs (if there is a cycle)
  - Positive dependency graph: only use positive atoms (no negation).
  - Loop is a path of length>0 in the Positive dependency graph.
  - A tight-program has no loops.
- For any tight LPMLN program Π, the probability of its stable models are the same with the completion of Π under MLN semantics.

4-) LPMLN → ASP: We can convert LPMLN program to ASP using weak constraints. Using levels and ranking. Then

- For any LPMLN program Π, the most probable stable models of Π are the same with the stable models of ASP(Π).

4. LPMLN

LPMLN inference examples:

- Finding the most probable stable models of a program.
- Finding marginal probability of a query on the stable models.
- Finding conditional probability of a query on the stable models.

- Representing Bayesian Networks in LPLMN
  - Encoding CPT (Conditional Probability Table) in clingo code
  - Encoding DAG (Directed Acyclic Graph) in clingo code

- Representing Probabilistic Graphs in LPMLN

Thanks

&

Questions