III-Logics
Markov Logic Networks

Foundations of Neuro-Symbolic AI

Alex Goessmann

University of Applied Science Würzburg-Schweinfurt

Summer Term 2026

From Logics to Probability

Why is logical reasoning not enough?

- Evidence is incomplete: Not all conditions to rules might be known
- Evidence is uncertain: There might be errors in the data

Enhancement of semantics (upgrade of epistemologic assumptions):

- Logic: Possible/impossible
- Probability: Numeric uncertainties beyond possible/impossible

Judea Pearl in Probabilistic Reasoning in Intelligent Systems (1988):

Research [...] should bring together logics aptitude for handling the visible and probabilities ability to summarize the invisible.

Tensors as a common framework

Propositional Logics

Representation of semantics by

$$f : \bigtimes_{k\in[d]} [2] \rightarrow [2] = \{0, 1\}$$

Tensor Networks by formula compositions

Contractions decide entailment by checking

$$\mathcal{C}(\{\mathcal{KB}, \beta^f\}\{X_f\}) \parallel \epsilon_1$$

Probabilities

Representation of uncertainties by

$$\mathbb{P} : \bigtimes_{k\in[d]} [m_k] \rightarrow [0, 1]$$

Tensor Networks by conditional independencies

Contractions answer probabilistic queries

$$\mathbb{P}^X = \langle\mathbb{P}\rangle_{[X]}$$

Probabilistic Interpretations of Logics

Uniform distribution of the models of f is a distribution

$$\mathbb{P}^f = \frac{1}{\langle\{f\}\rangle_{[\varnothing]}} f$$

where f is a hard constraint: States $x_0, \dots, x_{d-1} \in \bigtimes_{k\in[d]}[2]$ with $f(x_0, \dots, x_{d-1}) = 0$ have vanishing probability.

We build a soft constraint by a weight $\theta \in \mathbb{R}$ tuning the quotient of probabilities

$$\exp\left[\theta \cdot f\right](x_0, \dots, x_{d-1}) = \begin{cases} 1 & \text{if } f(x_0, \dots, x_{d-1}) = 0 \\ \exp\left[\theta\right] & \text{if } f(x_0, \dots, x_{d-1}) = 1 \end{cases}$$

and normate to get a probability tensor

$$\mathbb{P}^{(f,\theta)} = \frac{1}{\langle\{\exp\left[\theta \cdot f\right]\}\rangle_{[\varnothing]}} \exp\left[\theta \cdot f\right] .$$

Collection of weighted formulas

Definition (Markov Logic Network)

The Markov Logic Network to as set of formulas $\mathcal{F}$ weighted by $\theta : \mathcal{F} \to \mathbb{R}$ is the distribution

$$\mathbb{P}^{(\mathcal{F},\theta)} = \frac{1}{\mathcal{Z}\left(\mathcal{F},\theta\right)} \left\langle \exp\left[\theta_f \cdot f\right] \, : \, f \in \mathcal{F} \right\rangle_{\left[f^0,\ldots,f^{d-1}\right]}$$

where

$$\mathcal{Z}\left(\mathcal{F},\theta\right) = \left\langle \left\{ \exp\left[\theta_f \cdot f\right] \, : \, f \in \mathcal{F} \right\} \right\rangle_{\left[\varnothing\right]}$$

is called the partition function.

Tensor Network Representation

Markov Logic Networks are represented by

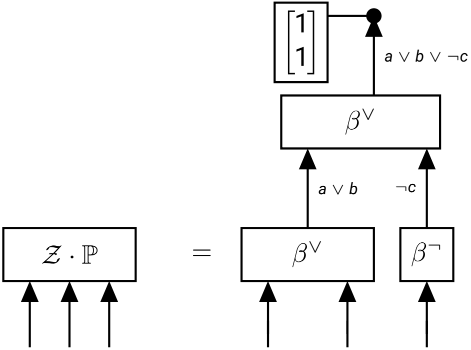
-

Markov Logic Networks combine logical and probabilistic approaches:

- Encoding of logical connectives are factors of the graphical model
- Further factors implement hard and soft constraints

Example: Distribution over the atomic variables a, b, c where

- All worlds of equal probability ($\mathbb{P} = \frac{1}{2^3}\mathbb{I}$)



Tensor Network Representation

Markov Logic Networks are represented by

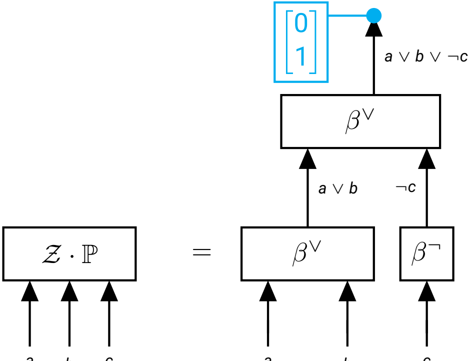
-

Markov Logic Networks combine logical and probabilistic approaches:

- Encoding of logical connectives are factors of the graphical model
- Further factors implement hard and soft constraints

Example: Distribution over the atomic variables a, b, c where

- $c \Rightarrow (a \vee b)$ is always true



Tensor Network Representation

Markov Logic Networks are represented by

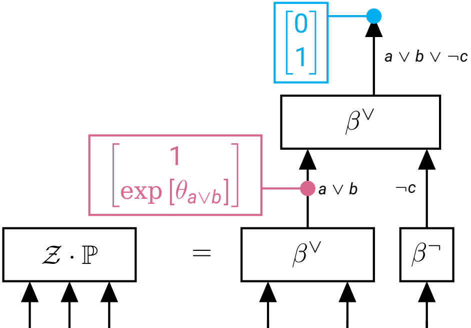
-

Markov Logic Networks combine logical and probabilistic approaches:

- Encoding of logical connectives are factors of the graphical model
- Further factors implement hard and soft constraints

Example: Distribution over the atomic variables a, b, c where

- $c \Rightarrow (a \vee b)$ is always true
- $a \vee b$ is likely to be true, tuned by a weight $\theta_{a \vee b}$



Infering Markov Logic Networks

Logical: Model counts by tensor network contractions

- Global contractions: Model-theoretic Entailment
- Local contractions: Constraint Propagation

Probabilistic: Conditional probabilities by tensor network contractions

- Exact Inference: Belief Propagation
- Approximate Inference: Gibbs Sampling, Loopy Belief Propagation

Tradeoff

The size of tensor network contractions is a tradeoff between

Completeness/Exactness $\leftrightarrow$ Efficiency (Demand of the contraction)

Application of Markov Logic Networks

(Ante-hoc & globally) Explainable Generative Models

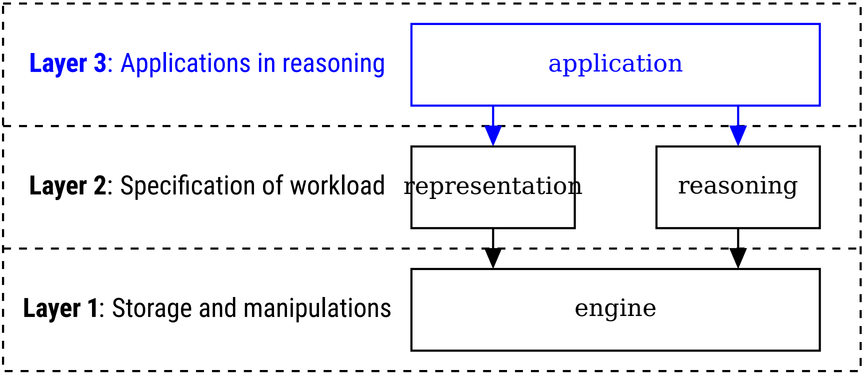
- Human-machine interface based on interpretable logical sentences
- Expert evaluation and manipulation of models
- Generation of synthetic data (e.g. for test purposes)

Logic and Probabilistic Programming

- Generation by declarative programming or learned from data
- Prediction of unseen variables given evidence
- Explainability built-in by logics
- Uncertainty assessment built-in by probabilities

Implementation in tnreason: Subpackage application

The subpackage application implements generalizations of Markov Logic Networks and corresponding reasoning tasks.



Extension of Script Language $\sigma^{\cdot}$

Formulas with hard logical interpretation are stored as before in fact dictionaries:

$$\{\text{key}(f) : \sigma^{f} \text{ for } f \in \mathcal{F}\}$$

Formulas with soft logical interpretation (as in Markov Logic Networks) are stored in weighted formulas dictionaries:

$$\{\text{key}(f) : \sigma^{f} + [\theta_f] \text{ for } f \in \mathcal{F}\}$$

Hybrid Knowledge Bases

Probability distributions, which are specified by propositional formulas are captured by the class

knowledge.HybridKnowledgeBase

initialized with arguments

- facts: Dictionary of propositional formulas stored as σ^f representing hard logical constraints
- weightedFormulas: Dictionary of propositional formulas stored as σ^f+[θ_f] representing soft logical constraints
- evidence: Dictionary of atomic formulas, where key are the formulas in string representation and values the certainty in $[0, 1]$ (float or int) of the atom being true
- categoricalConstraints: Dictionary of categorical constrained, which values are lists of atomic formulas stored as strings σ^f