EXAMPLE RULES FOR THE ENEXA USE CASE AT DATEV

A PREPRINT

Maria Schnödt-Fuchs, Martin Braun, Alex Goeßmann
DATEV eG

January 23, 2025

ABSTRACT

We demonstrate different processings of three example rules, which are representing accounting practices. First we formulate the rules as Horn clauses and SPARQL queries. Then we investigate their representation in graphical models to enable reasoning in soft logics. Finally we derive inference problems and implement MAP inference as a QUBO.

Contents

1 Formulation in Logic

1.1 Description Logic for KG

The atoms appearing in the rules are link existence variables connected with a Knowledge Graph. We distinguish between class membership links (unary predicates) and more generic links (binary predicates).

Classes for a Beleg x:

- Ausgangsrechnung(x) = A(x)

Classes for a Artikelposition z:

- Unterschrank(z) = S(z)
- Moebel(z) = M(z)
- Umsatzerloes(z) = U(z)

Classes for a Mandant y:

- Bautischlerei(y) = B(y)

Further links:

- versandterBeleg(y,x) = vB(y,x)
- hatLeistungserbringer(x,y) = hL(x,y)
- hatBelegzeile(x,z) = hB(x,z)

To ease the notation, we use the by equivalences specified abbreviations of the link existence variables.

1.2 Example Horn Clauses

Here want to process these three example rules:

r_1:

$$\forall z : Unterschrank(z) \rightarrow Moebel(z)$$

r_2:

$$\forall x, y : hatLeistungserbringer(x, y) \wedge versandterBeleg(y, x) \rightarrow Ausgangsrechnung(x)$$

r_3:

$$\forall x, y, z : Ausgangsrechnung(x) \wedge versandterBeleg(y, x) \wedge Bautischlerei(y) \wedge hatBelegzeile(x, z) \wedge Moebel(z) \\ \rightarrow Umsatzerloes(z)$$

A general Horn clause would be of the structure

$$\forall x_1, \ldots, x_n : Q_1(x_1, \ldots, x_n) \wedge \ldots \wedge Q_d(x_1, \ldots, x_n) \rightarrow P(x_1, \ldots, x_n)$$

where $Q_1, ..., Q_d$ and P are positive literals (atoms / link existence variables).

1.3 Rule representation by SPARQL queries

We can represent the Horn clauses using the SPARQL syntax for processing queries on Knowledge Graphs. First we represent link existance variables by triple patterns, where we distinguish between

- Unary: C(x) by

  ```
  ?x a C .
  ```

- Binary: r(x,y) by

  ```
  ?x r ?y .
  ```

Then we place the premises of the rules in $WHERE\{...\}$ and heads in $INSERT\{...\}$ statements.

The examples above are represented by the queries:

r_1:

```
INSERT {
    ?x a tev:Moebel .
}
WHERE {
    ?z a tev:Unterschrank .
}
```

r_2:

```
INSERT {
    ?x a tev:Ausgangsrechnung .
}
WHERE {
    ?x tev:hatLeistungserbringer ?y .
    ?y tev:versandterBeleg ?x .
}
```

r_3:

```
INSERT {
    ?z a tev:Umsatzerloes .
}
WHERE {
    ?x a tev:Ausgangsrechnung .
    ?y tev:versandterBeleg ?x .
    ?y a tev:Bautischlerei .
    ?x tev:hatBelegzeile ?z .
    ?z a tev:Moebel .
}
```

In general we can represent any Horn clause in SPARQL as:

```
INSERT {
    P(x_1,\ldots,x_n)
}
WHERE {
    Q_1(x_1,\ldots,x_n) .
    ...
    Q_d(x_1,\ldots,x_n) .
}
```

1.4 Rule representation by polynomials

To transform the rules into polynomials, we use the DNF and some calculus.

r_1:

$$\begin{aligned}
&f_1(S(z), M(z)) \\
&\quad = [S(z) \to M(z)] \\
&\quad = [\neg U(z) \vee M(z)] \\
&\quad = 1 - [U(z)](1 - [M(z)])
\end{aligned}$$

r_2:

$$\begin{aligned}
&f_2(hL(x,y), vB(y,x), A(x)) \\
&\quad = [hL(x,y) \wedge vB(y,x) \to A(x)] \\
&\quad = [\neg hL(x,y) \vee \neg vB(x,y) \vee A(x)] \\
&\quad = 1 - [hL(x,y)][vB(x,y)](1 - [A(x)])
\end{aligned}$$

r_3:

$$\begin{aligned}
&f_3(A(x), vB(y,x), B(y), hB(x,z), M(z), U(z)) \\
&\quad = [A(x) \wedge vB(y,x) \wedge B(y) \wedge hB(x,z) \wedge M(z) \to U(z)] \\
&\quad = [\neg A(x) \vee \neg vB(y,x) \vee \neg B(y) \vee \neg hB(x,z) \vee \neg M(z) \vee U(z)] \\
&\quad = 1 - [A(x)][vB(y,x)][B(y)][hB(x,z)][M(z)](1 - [U(z)])
\end{aligned}$$

Here, we denote by the brackets $[\cdot]$ the embedding of truth values true and false by 0 and 1.

A general Horn clause can be represented by the polynomial

$$\begin{aligned}
&f(x_1, \dots, x_n) \\
&\quad = [Q_1(x_1, \dots, x_n) \wedge \dots \wedge Q_d(x_1, \dots, x_n) \to P(x_1, \dots, x_n)] \\
&\quad = 1 - [Q_1(x_1, \dots, x_n)] \cdot \dots \cdot [Q_d(x_1, \dots, x_n)](1 - [P(x_1, \dots, x_n)]) \,.
\end{aligned}$$

2 Formulation as Graphical Models

2.1 Markov Logic Networks

Given weights w_1, w_2, w_3 representing a confidence notion of the corresponding formula, we define a corresponding MLN by the distribution

$$\begin{aligned}
&\mathbb{P}[A(x), S(z), M(z), U(z), B(y), vB(y,x), hL(x,y), hB(x,z)] \\
&\quad = \frac{1}{Z(w_1,w_2,w_3)} \exp[w_1 \cdot f_1(S(z), M(z)) + w_2 \cdot f_2(hL(x,y), vB(y,x), A(x)) \\
&\qquad\qquad + w_3 \cdot f_3(A(x), vB(y,x), B(y), hB(x,z), M(z), U(z))] \,.
\end{aligned}$$

Here Z is the partition function ensuring the normation of the probability distribution, defined as

$$\begin{aligned}
Z(w_1,w_2,w_3) = \sum_{\text{worlds}} &\exp[w_1 \cdot f_1(S(z), M(z)) + w_2 \cdot f_2(hL(x,y), vB(y,x), A(x)) \\
&+ w_3 \cdot f_3(A(x), vB(y,x), B(y), hB(x,z), M(z), U(z))] \,.
\end{aligned}$$

By worlds we refer to any assignment of the variables.

2.2 Bayesian Networks

$$\begin{aligned}
&\mathbb{P}[A(x), S(z), M(z), U(z), B(y), vB(y,x), hL(x,y), hB(x,z)] \\
&= \mathbb{P}[S(z)]\mathbb{P}[hL(x,y)]\mathbb{P}[vB(y,x)] \\
&\quad \cdot \mathbb{P}[M(z)|S(z)] \\
&\quad \cdot \mathbb{P}[A(x)|hL(x,y), vB(y,x)] \\
&\quad \cdot \mathbb{P}[U(z)|M(z), hB(x,z), A(x), B(y)]
\end{aligned}$$

Here the example rules are represented by conditional probabilities.

The Bayesian Network has been implemented in the repo https://git.zd.datev.de/enexa/experiments/bayesianbuchungsbeispiel/.

3 Inference

Given evidence $E = e$, where $E \subset X$, we can apply the Bayes rule to update the probability distribution.

We distinguish two types of inference:

- Determining the a posteriori distribution given evidence.
- Finding the most likely state of the a posteriori distribution.

3.1 MAP Inference as HUBO

$$\begin{aligned}
&\operatorname{argmax} \mathbb{P}_{\mathrm{MLN}}[A(x), S(z), M(z), U(z), B(y), vB(y,x), hL(x,y), hB(x,z)] \\
&= \operatorname{argmax} w_1 \cdot f_1(S(z), M(z)) + w_2 \cdot f_2(hL(x,y), vB(y,x), A(x)) \\
&\qquad + w_3 \cdot f_3(A(x), vB(y,x), B(y), hB(x,z), M(z), U(z))]
\end{aligned}$$

This is an HUBO of sixth order.

3.2 MAP Inference as QUBO

For rules two and three we need to introduce slack variables representing the premises:

- b_2 enforced to be the premise of r_2 using the formula
$$b_2 = \text{argmax}_{b_2\in\{0,1\}}\ b_2 \left([hL(x,y)] + [vB(y,x)] - \frac{3}{2} \right)$$
- b_3 enforced to be the premise of r_3 using the formula
$$b_3 = \text{argmax}_{b_3\in\{0,1\}}\ b_3 \left([A(x)] + [vB(y,x)] + [B(y)] + [hB(x,z)] + [M(z)] - \frac{9}{2} \right)$$

We can then represent the MAP Inference as an HUBO

$$\begin{aligned}\text{argmax}\ & w_1 \left(1 - [S(z)] + [S(z)][M(z)]\right)\right) \\ & + w_2 \left(1 - b_2 + b_2 \cdot [A(x)]\right) \\ & + w_3 \left(1 - b_3 + b_3 \cdot [U(z)]\right) \\ & + \tilde{w} \left(b_2([hL(x,y)] + [vB(y,x)] - \frac{3}{2}) + b_3([A(x)] + [vB(y,x)] + [B(y)] + [hB(x,z)] + [M(z)] - \frac{9}{2}) \right)\end{aligned}$$

Here the enforcing weight $\tilde{w}$ can be chosen, but needs to exceed w_2 and w_3 to ensure that b_2 and b_3 represent the premises of rules r_2 and r_3.

We can represent this quadratic polynomial as dependent on the vector

$$s = \begin{bmatrix} [A(x)] \\ [S(z)] \\ [M(z)] \\ [U(z)] \\ [B(y)] \\ [vB(y,x)] \\ [hL(x,y)] \\ [hB(x,z] \\ b_2 \\ b_3 \end{bmatrix} \in \{0,1\}^{10} \tag{1}$$

The associated problem matrix is then defined as

$$Q = \left[\begin{array}{cccccccc|cc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & w_2 & \tilde{w} \\ 0 & -w_1 & w_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \tilde{w} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & w_3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \tilde{w} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \tilde{w} & \tilde{w} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \tilde{w} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \tilde{w} \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -w_2 - \frac{3}{2}\tilde{w} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -w_3 - \frac{9}{2}\tilde{w} \end{array}\right].$$

Alternatively, one can also introduce a slack variable for the rule r_1 to have a consistent representation scheme in QUBO (see comment on generic rule representation below).

The MAP inference problem is then represented by the QUBO

$$\text{argmax}\ s^T Q s\,.$$

This problem has been implemented in the repo https://git.zd.datev.de/enexa/experiments/qubobuchungsbeispiel/. Therein we also investigate exact and approximative sampling algorithms.

For a generic rule system with d random variables and p rules

$$Q = \left[\begin{array}{c|c} E \in \mathbb{R}^{d \times d} & C \in \mathbb{R}^{d \times p} \\ 0 \in \mathbb{R}^{p \times d} & D \in \mathbb{R}^{p \times p} \end{array} \right]$$

Here, D is a diagonal matrix with entries for the ith rule by

$$-w_i - \tilde{w}(n_i - \frac{1}{2})$$

where n_i is the number of premises in the rule. C consist of vectors c_i for the ith rule, with entries $\tilde{w}$

The matrix E is always diagonal and representing the evidence. If no evidence has been provided, it is the zero matrix. If there is evidence, we encode it by $\pm\infty$ entries on the diagonal.

4 Outlook

We need to approach these questions:

- Can we represent the logic of accounting entirely by Horn clauses?
- Are Markov Logic Networks or Bayesian Networks preferable in the probabilistic representation of accounting rules?
- Where is the tradeoff between exact and approximative inference algorithms?