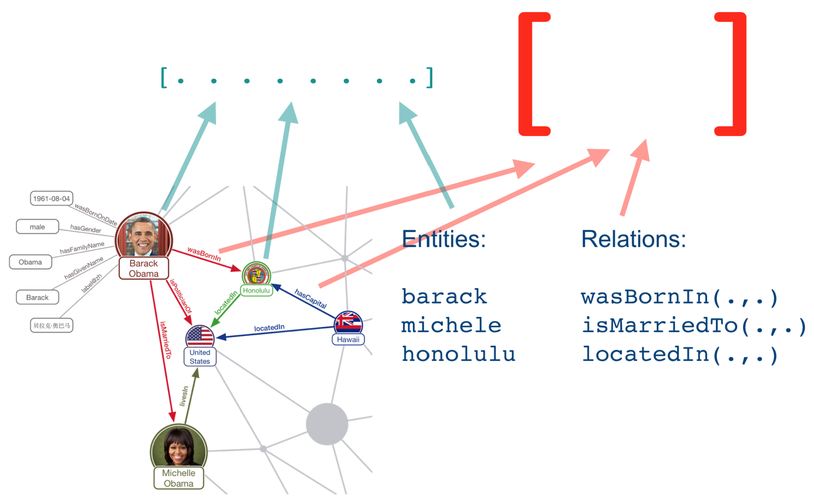
# Auto-encoding Logic Programs

**Sebastijan Dumančić**, Tias Guns, Wannes Meert, Hendrik Blockeel
{firstname,lastname}@cs.kuleuven.be, tias.guns@vub.be
KU Leuven, VUB, Belgium

## Problem

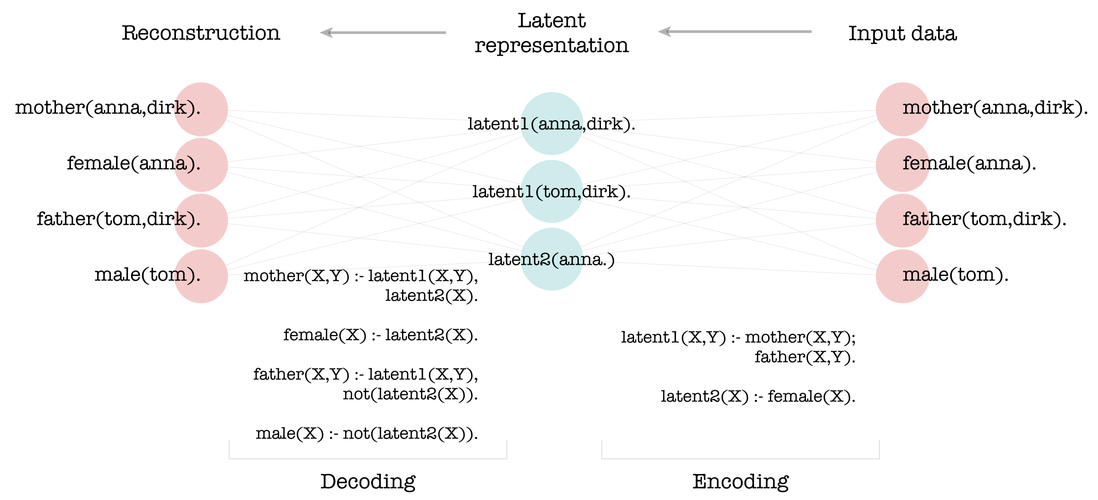
**Deep learning for relational data:** re-represent with vectors




**Can we do it in a more principled way by combining expressive knowledge representation frameworks with deep learning ideas?**

## An alternative

**Auto-encoding logic programs:** end-to-end *logical* latent representation learning
→ replace matrix computation with logic programs




## Learning ALPs: Constraint Optimization Perspective

### Mapping to CSP

**Initialize variables:**

//Encoder clauses:
var int: $ec_1$;
...
var int: $ec_N$;

//Decoder clauses:
var int: $dc_1$;
...
var int: $dc_M$;

**Encode reconstruction:**

$rf_1 \Leftrightarrow dc_1 \vee dc_3 \vee dc_5 3$
$rf_2 \Leftrightarrow dc_5 \vee dc_{14} \vee dc_{167} \vee dc_{67} \vee dc_{78}$
...
$rf_k \Leftrightarrow dc_m \vee dc_n \vee dc_l \vee dc_p$

**Objective function:**

$\max \sum_{i=1}^{F} w_i \times rf_i - \sum_{j=1}^{F'} w_j \times rf'_j$

**Connect encoder and decoder:**

//Encoder → Decoder
$ec_n \Leftrightarrow dc_k \vee dc_l \ \ldots \vee dc_m$;
...

//Decoder → Encoder
$dc_p \Leftrightarrow ec_x \wedge \ \ldots \wedge dc_y$;
...

**Impose bottleneck constraints:**

//Fact limit
$\sum_{i=1}^{N} w_i \times ec_i \leq C$

//Sparsity
$\frac{\sum_{i=1}^{N} w_i \times ec_i}{N} \leq \frac{\texttt{\#facts in original representation}}{\texttt{\#predicates in original representation}}$

**Impose semantic constraints:**

//Reconstruct each predicate
$p_1$ : $dc_m \vee dc_n \vee \ \ldots \vee dc_p \geq 1$
...
$p_K$ : $dc_m \vee dc_n \vee \ \ldots \vee dc_p \geq 1$

//Block immediate refinements
NOT($ec_i \wedge ec_j$)
...
NOT($ec_k \wedge ec_l$)

//Block syntactic variants
// –> decoder clauses reconstructing the same facts
$dc_m$ + $dc_n$ + ...+ $dc_o \leq 1$
...
$dc_x$ + $dc_y$ + ...+ $dc_z \leq 1$

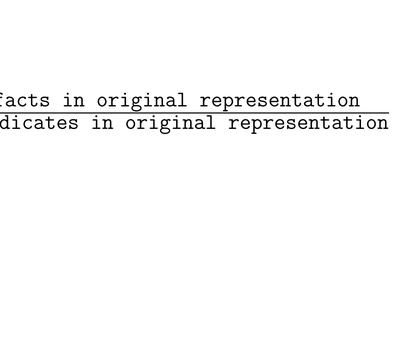
//Block signature variants
// –> decoder clauses reconstructing the same facts and having the same predicates in the body
$dc_k$ = ? $dc_n$ = false ... $dc_m$ = false

### Search

**Large neighbourhood search:** repeatedly run exhaustive search in a limited neighbourhood of the best found solution so far
→ fix values for a subset of variables
→ bounded by the number of backtrack steps
→ bounded by the search time

**Remember-forget** *strategy for constructing neighbourhood:*

- the solution is necessarily extremely sparse
- *remember* $R\%$ of selected variables in the current solution
- *forget* $F\%$ of encoder clauses not selected in the current solution



Given a set of facts

mother(anna,dirk). female(anna). father(tom,dirk). male(tom).

and the language bias (= syntactic restrictions defining the acceptable logical formulas)

Generate all candidate encoder and decoder clauses

latent1(X,Y) :- mother(X,Y);father(X,Y).
latent2(X) :- female(X).
mother(X,Y) :- latent1(X,Y),latent2(X).
mother(X,Y) :- latent1(X,Y), not(latent3(X)).
...

and associate a *decision variable* with each candidate clause

For each fact indicate which decoder clauses reconstruct it. For example, if the fact `mother(anna,dirk)` is reconstructed with `mother(X,Y) :- latent1(X,Y),latent2(X).` ($dc_1$) and `mother(X,Y) :- latent4(X,Y).` ($dc_2$) introduce a constraint $dc_1 \vee dc_2 \Leftrightarrow rf_i$

Select the subset of encoder/decoder clauses in order to maximize the number of reconstructed facts while minimizing the number of falsely reconstructed facts (not present in the original data)

Connect encoder and decoder clauses:

- if a decoder clause is selected, all of the necessary encoder clauses have to be selected as well
- if an encoder clause is selected, at least one of the decoder clauses using it has to be selected

Force the latent representation to be compressing, either by limiting the number of facts or enforcing sparsity in the latent representation

For each predicate $p$ in the original data, Impose a constraint saying that at least one decoder clause with the predicate $p$ in the head of the clause has to be selected

For each pair of encoder clauses for which one clause if *a refinement* of the other, impose a constraint stating that at most one can be selected (encouraging diversity)

For each set of decoder clauses that are *syntactic variants* – they reconstruct the same facts, impose a constraint stating that at most one of the decoder clauses can selected

For each set of decoder clauses that are *signature variants* – they reconstruct the same facts and are composed of the same predicates, remove all but one decoder clause from the set

### Preliminary results

**Question:** does learning from *latent representation* improves the performance of a model?
→ the preliminary experiments show that *a generative Markov Logic Network often performs better when learned on the latent representation*