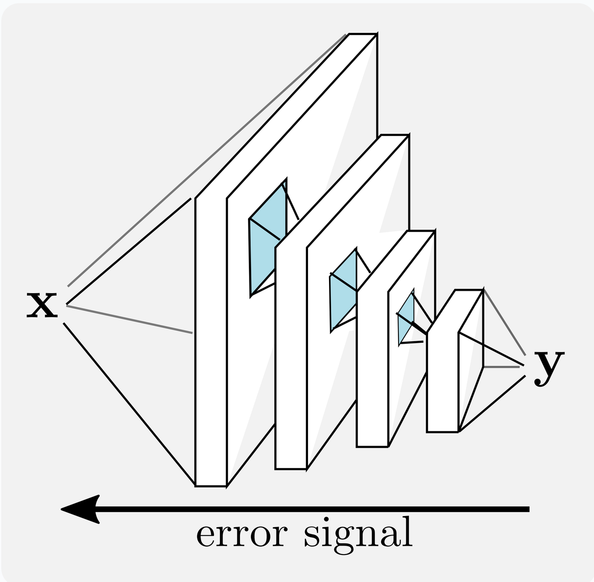
The Knowledge Graph for End-to-End Learning on Heterogeneous Knowledge

VU UNIVERSITY AMSTERDAM

Xander Wilcke, Peter Bloem, en Victor de Boer
{w.x.wilcke, p.bloem, v.de.boer}@vu.nl

Introduction

- In modern machine learning, **manual feature engineering has given way to end-to-end learning**.
- With end-to-end learning
  - every step in the machine learning pipeline is differentiable and can thus be tuned
  - we can incorporate feature engineering into the machine learning model and let it **learn relevant features automatically**
  - we minimize bias otherwise introduced by the adding, removing, or transformation of data



- However, **current end-to-end models are unsuited for learning on *heterogeneous knowledge***

Information of different types and from different domains

We argue [1] that
to enable true end-to-end learning on *heterogeneous knowledge* we must

a) adopt the *knowledge graph* as the default data model for this kind of knowledge, and

b) develop end-to-end models which can directly consume *knowledge graphs*

The Knowledge Graph

- Knowledge is encoded using **binary statements**
- Statements are of the form:

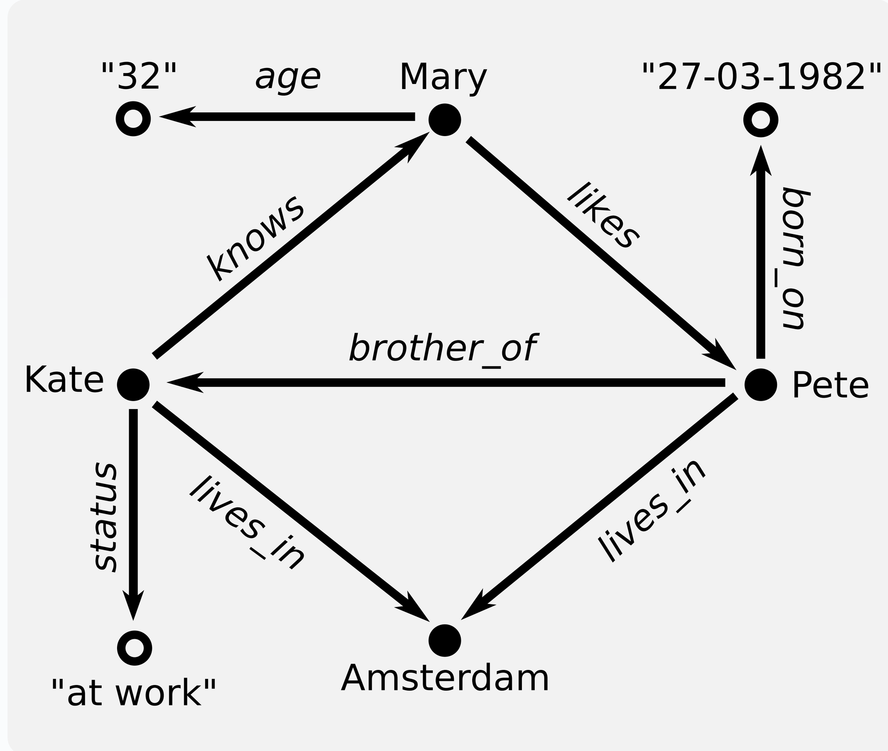
These are also called triples!

(*subject*, *predicate*, *object*)

- Subjects: entities ("things") to and from which can be linked
- Objects: entities or literals ("attributes") holding a raw value
- Predicates: relations between subjects and objects

Knowledge graphs can be represented more intuitively as a graph

- Background knowledge is similarly encoded and integrates naturally
- Any two knowledge graphs can be integrated instantaneously if they share (at least) a single subject
- A global network of interlinked and open knowledge graphs already exists, and is called the Linked Open Data cloud.



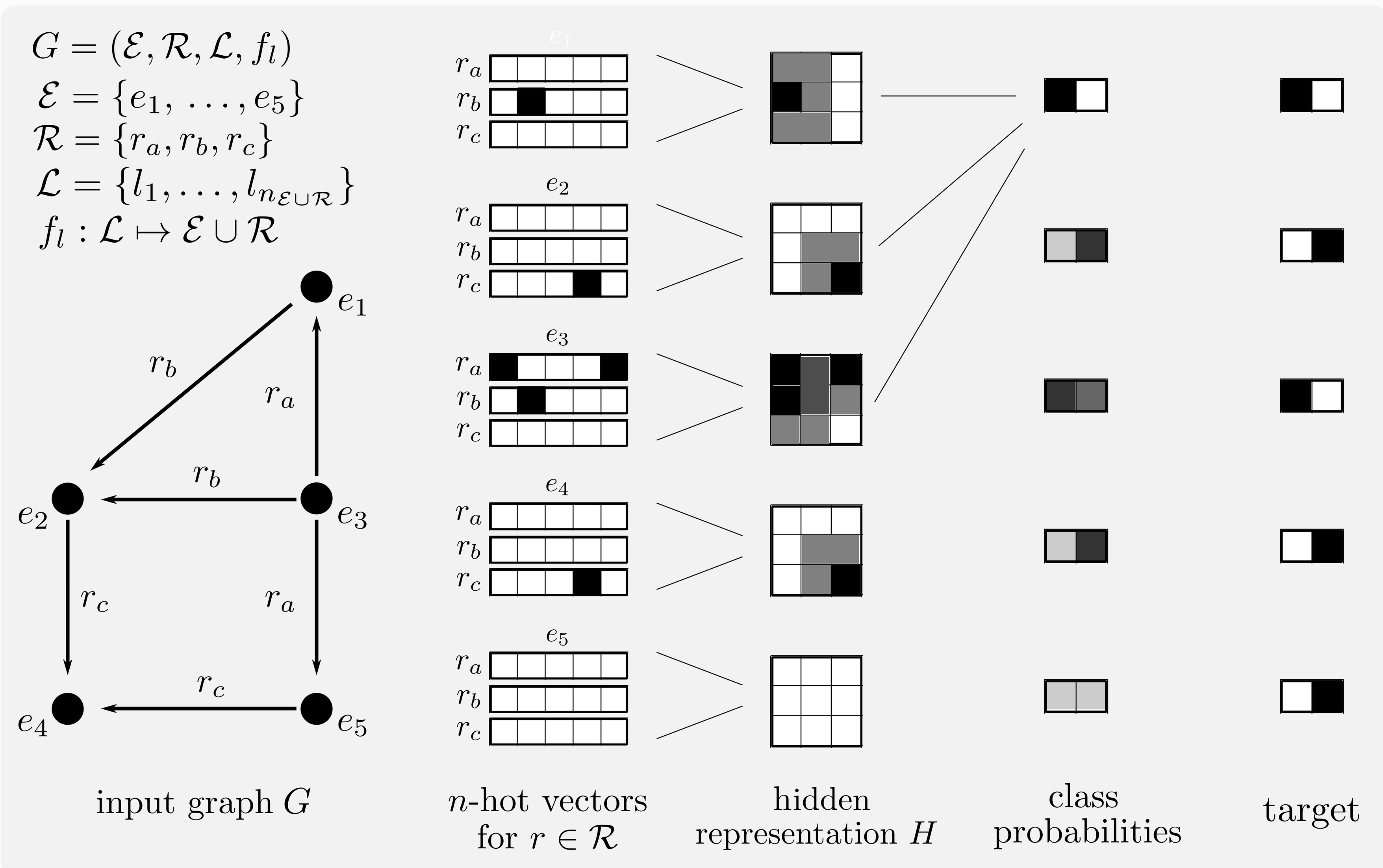
Example of a knowledge graph which depicts three individuals, two of which live in Amsterdam. A single attribute is given for each of them, each of a different data type.

End-to-End Learning on Knowledge Graphs

Graph convolutions

- generalize convolutional filters to graphs
- allow for end-to-end learning on knowledge graph

- **Advantages** to data scientists:
  - A single uniform data model across nearly all domains and tasks
  - Data sets expressed as knowledge graphs are task independent
  - Any method that is tailored to knowledge graphs can consume all knowledge graphs without preprocessing them first
  - Integration and harmonization of data sets requires minimal effort
  - A huge collection of knowledge graphs already exists, and is freely available on the Linked Open Data cloud



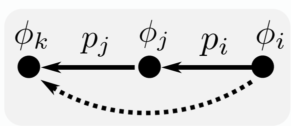
Example of how a *graph convolution network* can be applied to knowledge graphs. For simplification, only outward relations are considered.

- The Relational Graph Convolution Network (RGCN) [2]:
  - is an adaptation of graph convolutions to relational graphs
  - holds internal representations of in- and outward links and loops
  - learns from the graph's structure (relations between vertices)
  - can be applied to knowledge graphs

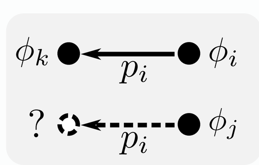
[2] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. *Modeling relational data with graph convolutional networks.* arXiv preprint arXiv:1703.06103, 2017.

Challenges

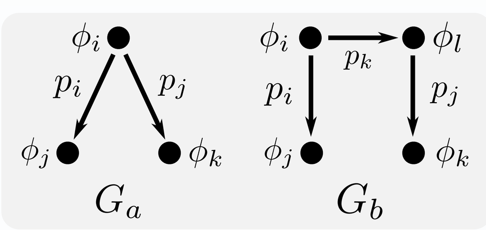
- End-to-End Learning on knowledge graphs is still very experimental and still has many unsolved challenges
- We identify **four major challenges** [1]:
  - **Dealing with implicit knowledge**
    A wealth of knowledge is implied through the interplay of assertion knowledge and background knowledge, e.g. by transitivity. By exploiting this knowledge, machine learning models can learn from much richer data



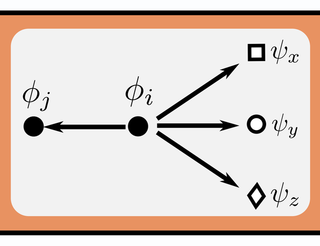
  - **Dealing with incomplete knowledge**
    Real-world knowledge is often incomplete: there may be more entities for which certain attributes are missing, than for which they are present.
    A machine learning model should cope with missing values natively.



  - **Dealing with differently-structured knowledge**
    While knowledge graphs encode knowledge uniformly. different modelling choices do lead to different graph topologies. End-to-end methods can cope with this to an extent, but learning models should still take this into account to aid convergence.

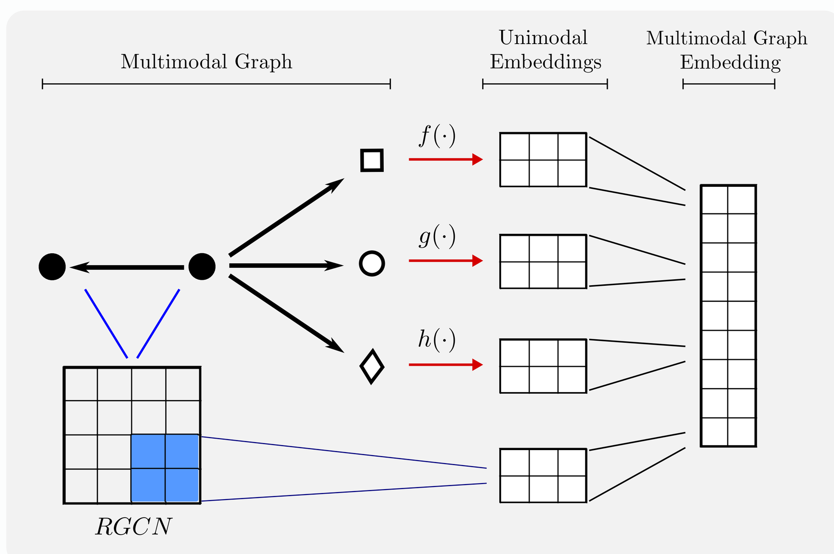


  - **Dealing with multi-modal knowledge**
    Heterogeneous knowledge is multi-modal by nature. In knowledge graphs, knowledge that is not a "thing" is encoded as a literal of a certain data type. Dealing with these is the main focus of our current research

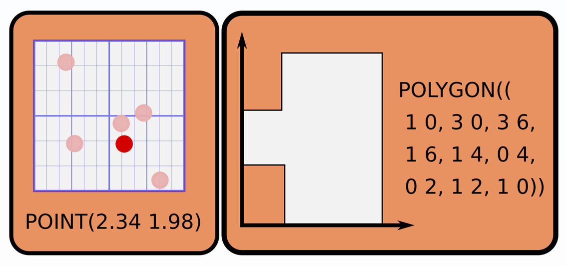


Multi-modal Graph Embeddings

Our proposed model extends the RGCN with modules dedicated to different modalities (the different oval shapes in the figure), each one dealt with accordingly and projected into the same multi-modal embedding space.



- Multi-modal learning on knowledge graphs has been left largely unaddressed [1].
- Most **present methods solely learn from graphs' structure**: literals are either omitted completely or are stripped from their values and treated as non-literals.
- To achieve multi-modal learning, we must
  - treat literals and non-literals as separate cases
  - address each data type separately and accordingly
  - project the different modalities into a **joint representation space**
- **Special attention is given to spatial information** which is an intrinsic aspect of all physical entities, and which enables us to perform spatially-oriented learning tasks.



NETWORK INSTITUTE

SPIN LAB SPATIAL INFORMATION LABORATORY

[1] Xander Wilcke, Peter Bloem, and Victor de Boer. *The Knowledge Graph as the Default Data Model for Learning on Heterogeneous Knowledge*. Data Science, 1(1-2):39–57, 2017. DOI:10.3233/DS-170007.