Machine Learning with Knowledge Graphs

Volker Tresp

Siemens Corporate Technology
Ludwig Maximilian University of Munich

Joint work with Maximilian Nickel
With contributions from Xueyan Jiang and Denis Krompass
Prelude

- My background is in Machine Learning and I got involved in Semantic Web projects maybe 6 years ago
- Learning about the Semantic Web clarified my thinking about many things dramatically

- Immediate love affair with RDF
  - Nothing is ever wrong
  - No contradictions
Prelude

- My background is in Machine Learning and I got involved in Semantic Web projects maybe 6 years ago
- Learning about the Semantic Web clarified my thinking about many things dramatically

- Immediate love affair with RDF
  - Nothing is ever wrong
  - No contradictions
Overview

- Why Machine Learning needs Knowledge Graphs
- Statistical Relational Learning
- Learning with the YAGO Knowledge Graph
- Towards Relevant Use Cases
What is Machine Learning?

Machine Learning versus Statistics versus Data Mining

- Statistics focuses on interpretable parameters
- Data mining focuses on the discovery of meaningful patterns
- Machine Learning focuses on prediction accuracy
Classification

Classification is the work horse of machine learning

- Predict class memberships for many objects
  - Very powerful
  - Surprisingly general
Typical Classifiers

Predicting class $k$ for input $z_l$:

$$P(x^k(z_l) = 1) \leftarrow f^k(z_l)$$

Fixed basis functions:

$$f^k(z_l) = \sum_{m=1}^{M} w^k_m b_m(z_l)$$

Kernels:

$$f^k(z_l) = \sum_{n=1}^{N} v^k_m k(z_l, z_n)$$

Really the same things; deep learners would call the shallow

Neural Networks:

$$f^k(z_l) = \text{NN}_{\text{deep}}(z_l)$$

- 10 layers with 1000 neurons per layer
- Currently the hottest thing!
Scientists See Promise in Deep-Learning Programs

- Google, Microsoft, Facebook, Baidu are all investing heavily in deep learning

Using an artificial intelligence technique inspired by the way the brain recognizes patterns, technology companies are reporting startling gains in fields as diverse as computer vision, speech recognition and the identification of promising new drugs.

The advances have led to enthusiasm among researchers to perform human-like activities like seeing, listening and thinking. They offer the promise of machines that converse with humans and perform tasks like driving cars and working in factories, raising the specter of automated robots that could replace human workers.
Detecting Cats in Images

- Best performing in detecting cats in images and videos (Andrew Ng)

How Many Computers to Identify a Cat? 16,000
Where from here?

- A deep learning network sees more cats than any child but is not as good at this task
- Deep Learning community: we need better unsupervised learning to pre-structure the network

Maybe we would say: we need background knowledge
Also: we do not just want to detect cats!
Challenges

Predict all classes: „This is a cat!“ „This is a dog!“ „This is a house!“ …

Recognize specific entities: „This my cat Max!“
[In our experiments $10^7$]

Predict all attributes: „Max is evil!“

Predict all relationships: „Max likes Mary!“
[In our experiments $10^{14}$] [ #of synapses]
Vision

„You must be president Obama!“
„How is your wife Michelle?“
γλαύκας εἰς Ἀθήνας κομίζειν
Requirement: Understanding of the World

- We need to know about the entities, attributes and classes in the world, and the various relationships that do or might exist between those

- We need ontologies!
Biomedical Ontologies

International Statistical Classification of Diseases and Related Health Problems (ICD)

- Used extensively in billing

SNOMED Clinical Terms (SNOMED CT)

- A systematically organized computer processable collection of medical terms providing codes, terms, synonyms and definitions used in clinical documentation and reporting.
- Application: EHR

RadLex

- Unified language of radiology terms for standardized indexing and retrieval of radiology information resources

Open Biomedical Ontologies (OBO)

- Controlled vocabularies for shared use across different biological and medical domains
- Gene Ontology (GO) is a part (genes and gene products)

Example GO term

<table>
<thead>
<tr>
<th>id: GO:0000016</th>
</tr>
</thead>
<tbody>
<tr>
<td>name: lactase activity</td>
</tr>
<tr>
<td>namespace: molecular_function</td>
</tr>
<tr>
<td>def: &quot;Catalysis of the reaction: lactose + H2O = D-glucose + D-galactose.&quot; [EC:3.2.1.108]</td>
</tr>
<tr>
<td>synonym: &quot;lactase-phlorizin hydrolase activity&quot; BROAD [EC:3.2.1.108]</td>
</tr>
<tr>
<td>synonym: &quot;lactase galactohydrolase activity&quot; EXACT [EC:3.2.1.108]</td>
</tr>
<tr>
<td>xref: EC:3.2.1.108</td>
</tr>
<tr>
<td>xref: MetaCyc:LACTASE-BXH</td>
</tr>
<tr>
<td>xref: Reactome:20536</td>
</tr>
<tr>
<td>ia_a: GO:0004553 ! hydrolase activity, hydrolyzing O-glycosyl compounds</td>
</tr>
</tbody>
</table>
For the First Time there Exist Sizable General Ontologies: DBpedia, YAGO, Freebase, Knowledge Graph


Bollacker, Evans, Paritosh, Sturge, Taylor, 2008

Suchanek, Kasneci, Weikum: 2007
Linked Open Data (Semantic Web)
**Triple Graphs**

- Dog1 is an animal
- Cat1 is a cat
- Cats are animals
- Zoos host animals
- Zoo1 hosts the Cat2

---

**In english**

- Max likes Mary

**The graph**

- RDF/turtle

```turtle
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix ex: <http://example.org/> .
@prefix zoo: <http://example.org/zoo/> .
ex:dog1   rdf:type    ex:animal .
ex:cat1   rdf:type    ex:cat .
ex:cat    rdfs:subClassOf ex:animal .
zoo:host  rdfs:range  ex:animal .
ex:zoo1   zoo:host    ex:cat2 .
```

---

**RDF/Special terms**

- `ex:dog1`
- `ex:cat1`
- `ex:cat`
- `ex:zoo1`

**RDFS/Special terms**

- `rdf:type`
- `rdfs:subClassOf`
- `rdfs:range`
- `zoo:host`
Knowledge Bases are Triple Graphs

- Linked Open Data (LOD) and large ontologies like DBpedia, Yago, Knowledge Graph are graph-based knowledge representations using light-weight ontologies, and are accessible to machine learners.

- They are all triple oriented and more or less follow the RDF standard
  - RDF: Resource Description Framework
Overview

- Why Machine Learning needs Knowledge Graphs
- **Statistical Relational Learning**
- Learning with the YAGO Knowledge Graph
- Towards Relevant Use Cases
Canonical Relational Machine Learning Task

\[ \langle e_i, r^k, e_j \rangle \text{ true or false?} \]

\[ P(\langle e_i, r^k, e_j \rangle = 1) \leftarrow f^k(z_l) \]

- So, very simple, we build one classifier for each relation type \( k \) and we are done

- But what is the input \( z_l \) ?
I. Relational Learning with *Known* Features

features (age, sex, *features derived from a neighborhood of the entity in the environment of the RDF-graph*)

$$\langle e_i, r^k, e_j \rangle \rightarrow \begin{cases} f^k(z_{l(i,j)}) & (a_{i,1}, a_{j,2}, \ldots, a_{i,r})^T \\ z_{l(i,j)} = (a_{i,1}, a_{j,2}, \ldots, a_{i,r}, a_{j,1}, a_{j,2}, \ldots, a_{j,r})^T & (a_{j,1}, a_{j,2}, \ldots, a_{j,r})^T \end{cases}$$

$$f^k(z_l) = \sum_{m=1}^{M} w^k_m b_m(z_l)$$

$$f^k(z_l) = \sum_{n=1}^{N} v^k_n k(z_l, z_n)$$

$$f^k(z_l) = \text{NN}_{\text{deep}}(z_l)$$

Popular in learning from the Semantic Web
II. Relational Learning with *Latent* Features

Same, but features are treated as *latent (unknown) variables*

\[
\begin{align*}
\langle e_i, r^k, e_j \rangle \quad &\rightarrow \quad (a_{i,1}, a_{j,2}, \ldots, a_{i,r})^T \\
\chi^k(z_{l(i,j)}) \quad &\rightarrow \quad (a_{j,1}, a_{j,2}, \ldots, a_{j,r})^T \\
z_{l(i,j)} = (a_{i,1}, a_{j,2}, \ldots, a_{i,r}, a_{j,1}, a_{j,2}, \ldots, a_{j,r})^T \\
\end{align*}
\]

\[
f^k(z) = \sum_{m=1}^{M} W^k_m b_m(z_l)
\]
With Latent Features We Get **Collective Learning**

- Information can globally propagate in the network of random variables
- Thus one can learn that: *Jack is rich since the father of his father is rich*
But what are good basis functions?
- We need to represent the interactions between all feature components
- Binary interactions

\[ f^k(z_l) = \sum_{s=1}^{r} \sum_{t=1}^{r} w_{s,t}^k b_{s,t}(z_l) \]

\[ b_{s,t}(z_l) = a_{i,s} a_{j,t} \]
Mapping to a Tensor Factorization Problem

\[ f^k(z_l) = \sum_{s=1}^{r} \sum_{t=1}^{r} w^k_{s,t} a_{i,s} a_{j,t} = a^T_k R_k a_j \]

\[ (R_k)_{s,t} = w^k_{s,t} \]

- Here, \( R_k \) is a \( r \times r \) matrix
- We can take the matrices for the different relations \( R_1, R_2, R_3, \ldots \)
on to of each other and obtain the core tensor \( R \)
- In tensor notation: We factorize the tensor \( X \)

\[ X \leftarrow R \times_1 A \times_2 A \]

\[ (X)_{i,j,k} = x^k(z_l(i,j)) \]
RESCAL Factorization

\[ x_{ijk} = \begin{cases} 
1, & \text{if triple (i-th entity, k-th relation, j-th entity) exists} \\
0, & \text{otherwise}
\end{cases} \]
Cost Functions

Frobenius norm

\[
\arg\min_{A,R} \|X - R \times_1 A \times_2 A\|^2 + \lambda_A \|A\|^2 + \lambda_R \|R\|^2
\]

Probabilistic View

\[
P (X | A, R) = \prod_{i=1}^{n} \prod_{j=1}^{n} \prod_{k=1}^{m} P (x_{ijk} | a_i^T R_k a_j)
\]

\[
a_i \sim \mathcal{N} (0, \sigma_{A_i}^2 I)
\]

Gaussian

\[
x_{ijk} \sim \mathcal{N} (a_i^T R_k a_j, \sigma^2)
\]

Bernoulli

\[
x_{ijk} \sim \text{Bernoulli}(a_i^T R_k a_j)
\]
Iterative Update

- Most efficient: Alternating Least Squares (ALS)
  - Can exploit data sparsity
  - (stochastic gradient descent, …)

\[
A \leftarrow \left( \sum_{k=1}^{m} X_k A R_k^T + X_k^T A R_k \right) \left( \sum_{k=1}^{m} B_k + C_k + \lambda_A I \right)^{-1}
\]

\[
B_k = R_k A^T A R_k^T, \quad C_k = R_k^T A^T A R_k
\]

\[
\text{vec} \left( R_k \right) \leftarrow \left( Z^T Z + \lambda_R I \right)^{-1} Z^T \text{vec} \left( X_k \right)
\]

\[
Z = A \otimes A
\]
RESCAL for Different \(-\)arities

**Unary Relations**

\[
P(r_k(e_i)) \leftarrow r_k^T a_i = \sum_{n=1}^{r} r_{k,n} a_{i,n}
\]

**Binary Relations**

\[
P(r_k(e_i, e_j)) \leftarrow a_i^T R_k a_j = \sum_{n_1=1}^{r} \sum_{n_2=1}^{r} R_{k,n_1,n_2} a_{i,n_1} a_{j,n_2}
\]

**Ternary Relations**

\[
P(r_k(e_i, e_j, e_l)) \leftarrow \sum_{n_1=1}^{r} \sum_{n_2=1}^{r} \sum_{n_3=1}^{r} R_{k,n_1,n_2,n_3} a_{i,n_1} a_{j,n_2} a_{l,n_3}
\]
RESCAL for Binary Relations
Scalability

Graphs showing scalability with respect to different parameters.

- Time in seconds/iteration vs. number of entities, number of predicates, and number of known facts.
- Time in seconds/iteration vs. number of attributes, comparing entities vs. coupled.
- Time in seconds/iteration vs. number of latent components, comparing scalable vs. non-regularized vs. naive.
**Leading Performance in Link prediction on benchmark data sets**

**Kinship:** multiple kinship relations between members of the Alyawarra tribe in central Australia (10,790 kinship relationships (facts) between 104 persons over 26 relations)

**UMLS:** The UMLS data set consists of a small semantic network which is part of the Unified Medical Language System (UMLS) ontology. 6,752 relationships (facts) between 135 concepts over 49 relations

**Nations:** The Nations data set describes political interactions of countries between 1950 and 1965. It contains information such as military alliances, trade relationships or whether a country maintains an embassy in a particular country. 2,024 relationships between 14 countries over 56 dyadic relations

**Predicting relationships:** "Max likes Mary"
**Cora Data: Entity Resolution**

- 1295 publication records, where each publication is the subject of a relationship to its first author, a relationship to its title, and a relationship to its publication venue.
- Task: identify which authors, entities and venues refer to identical entities.

**Recognizing specific entities:**

> „This my cat Max!“
Overview

- Why Machine Learning needs Knowledge Graphs
- Statistical Relational Learning
- Learning with the YAGO Knowledge Graph
- Towards Relevant Use Cases
**Yago2 Core Ontology**

<table>
<thead>
<tr>
<th>Yago2 Core Ontology</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Resources</td>
<td>2.6 million</td>
</tr>
<tr>
<td>Number of Classes</td>
<td>340,000</td>
</tr>
<tr>
<td>Number of Predicates</td>
<td>87</td>
</tr>
<tr>
<td>Number of Known Facts</td>
<td>33 million</td>
</tr>
</tbody>
</table>

*The tensor has $10^{14}$ entries!*

*Siemens – MPII cooperation*
Classification: Type Prediction

<table>
<thead>
<tr>
<th>Type</th>
<th>Number of entities</th>
</tr>
</thead>
<tbody>
<tr>
<td>wordnet:person</td>
<td>884,261</td>
</tr>
<tr>
<td>wordnet:location</td>
<td>429,828</td>
</tr>
<tr>
<td>wordnet:movie</td>
<td>62,296</td>
</tr>
</tbody>
</table>

Table 3.9: Link-prediction experiments on YAGO2.

<table>
<thead>
<tr>
<th></th>
<th>wordnet:person</th>
<th>wordnet:location</th>
<th>wordnet:movie</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random</td>
<td>0.32</td>
<td>0.18</td>
<td>0.06</td>
</tr>
<tr>
<td>Setting a)</td>
<td>0.99</td>
<td>1.0</td>
<td>0.75</td>
</tr>
<tr>
<td>Setting b)</td>
<td>0.96</td>
<td>0.98</td>
<td>0.51</td>
</tr>
<tr>
<td>With attributes</td>
<td>-</td>
<td>-</td>
<td>0.85</td>
</tr>
</tbody>
</table>

Predicting concepts: “This is a cat”

a) Only those rdf:type triples that include the class C that should be predicted were removed from the test fold. All other type triples, including subclasses of C, are still present in the data.

b) All rdf:type triples were deleted in the test fold.
Writer’s Nationality: Demonstrating Collective Learning

(a) Collective learning example on YAGO. The objective is to learn the correlation between France and French Writer from examples like Emile Zola.

(b) Results for link prediction on YAGO2 writers data set over ten-fold cross-validation.
Learning a Taxonomy (-> Ontology)

- IIMB 2010 benchmark provided by the Ontology Alignment Evaluation
- Around 1400 entities of a movie domain
- 5 distinct top-level concepts
- On the top level: every concept is represented by a sufficient number of entities, while e.g. some level 2 movie concepts only include two or three entities and therefore are hard to recognize.

Table 3.10.: F-measure for selected concepts and weighted F-measure for all concepts per subclass-level

<table>
<thead>
<tr>
<th>Level 1</th>
<th>Level 2</th>
<th>Level 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Locations 0.95</td>
<td>City 0.99</td>
<td>Capital 0.99</td>
</tr>
<tr>
<td>Films 1.0</td>
<td>Anime 0.67</td>
<td>Director 0.78</td>
</tr>
<tr>
<td>Creature 1.0</td>
<td>Character 0.73</td>
<td>Character Creator 0.53</td>
</tr>
<tr>
<td>Budget 1.0</td>
<td>Person 1.0</td>
<td>Actor 0.98</td>
</tr>
<tr>
<td>Language 1.0</td>
<td>Country 0.80</td>
<td></td>
</tr>
<tr>
<td>All 0.982</td>
<td>All 0.852</td>
<td>All 0.947</td>
</tr>
</tbody>
</table>
Extensions: Nonnegative RESCAL

Nonnegative RESCAL (Krompass, Nickel, Tresp)
- sparse solutions with clustering properties
Extensions: Proofs and Bounds

- Analysis of generalization bounds when order of the tensor match or do not match
- Matricization results in a loss of generalization performance

Overview

- Why Machine Learning needs Knowledge Graphs
- Statistical Relational Learning
- Learning with the YAGO Knowledge Graph
- Towards Relevant Use Cases
**Within the domain:**
- Prediction of triples
- Classification (defining type)
- Clustering
- Taxonomy Learning
- Entity Resolution
- Visualization
- Querying
  - Who wants to be Trelenas friends
  - Can be generalized towards more complex probabilistic queries

*(Krompass, Nickel, Tresp, ISWC 2014)*

**Outside of the domain (new entities):**
- Calculate the latent factors for the new entity
- Can do all of the tasks above
- Object recognition becomes entity resolution
- Formulate the new object as a query
- Object recognition as a query
- Queries can become complex
Clinical Data Intelligence

Goals
- Personalized medicine: modeling the patient in her/his full complexity -> patient specific recommendations
- Global modeling of the clinical data / clinical decision processes: clinical ontology (concepts and instances)

Use Cases
- All data from all patients
- Breast cancer
- Nephrology
- Data from clinical studies

Challenges
- Ontologies
- Complex relational data (patient in a clinic)
- Representing time; sequential data
- Decision modeling: decision optimization (confounders, causality)
- Including unstructured data (reports, images)
- Including OMICS data
Predicting Diagnoses and Procedures

Figure 1: Data from 10000 patients were used. We considered 2331 possible diagnoses, 1634 possible procedures, 2721 possible lab results, 209 possible therapies and 281 general patient data. In total the data contained 5.9 million facts. We predicted the next decision (diagnosis, procedure) as a function of the information available for each patient. Plotted is the NDCG score (a popular score for evaluating ranking results [11]) as a function of the information available for each patient (a large number is desirable). An event corresponds to an instance in time where patient data is recorded. With increasing information, the prediction improves. We see plots for different approximation ranks: the highest rank gives best scores which reflects the high degree of data complexity.
Machine Learning with Images and Ontologies

Linking textual descriptions in radiology reports to medical images
References and Related Work

RESCAL

Extensions from other groups
- Richard Socher, Danqi Chen, Christopher D. Manning, Andrew Y. Ng. Reasoning With Neural Tensor Networks for Knowledge Base Completion, NIPS, 2013

SUNS (First application of factorization approaches to relational Semantic Web domains)
- Volker Tresp, Yi Huang, Markus Bundschus, and Achim Rettinger. Materializing and querying learned knowledge. IRMLeS, 2009

Triplerank (Application of PARAFAC for ranking; no collective learning)

Factorization Machines
- S. Rendle et al.: Different factorization approaches for preference prediction and relational learning (2009 and later)

Knowledge Vault (Google Team)
Conclusions

- Knowledge Graphs
  - First time: large general ontologies available
  - Useful for solving machine learning tasks

- Relational Machine Learning with RESCAL
  - Scalable relational learning with very competitive performance
  - Collective Learning
  - We are working on many improvements/extension

- RESCAL Learning with the YAGO Knowledge Graph
  - Experimental results in a number of relational learning tasks

- Towards Relevant Use Cases
  - Text understanding
  - Image understanding
  - Clinical data