Embedding-Based Techniques

MATRICES, TENSORS, AND NEURAL NETWORKS
Probabilistic Models: Downsides

Limitation to Logical Relations

- Representation restricted by manual design
  - Clustering? Asymmetric implications?
  - Information flows through these relations
- Difficult to generalize to unseen entities/relations

Computational Complexity of Algorithms

- Complexity depends on explicit dimensionality
  - Often NP-Hard, in size of data
  - More rules, more expensive inference
- Query-time inference is sometimes NP-Hard
- Not trivial to parallelize, or use GPUs

Embeddings

- Everything as dense vectors
- Can capture many relations
- Learned from data

- Complexity depends on latent dimensions
- Learning using stochastic gradient, back-propagation
- Querying is often cheap
- GPU-parallelism friendly
Two Related Tasks

Relation Extraction

Graph Completion
Two Related Tasks

Relation Extraction

Graph Completion
John was born in Liverpool, to Julia and Alfred Lennon.
John was born in Liverpool, to Julia and Alfred Lennon.
“Distant” Supervision

No direct supervision gives us this information.

**Supervised:** Too expensive to label sentences

**Rule-based:** Too much variety in language

Both only work for a small set of relations, i.e. 10s, not 100s
Relation Extraction as a Matrix

John was born in **Liverpool**, to **Julia** and **Alfred Lennon**.

<table>
<thead>
<tr>
<th>Entity Pairs</th>
<th>was born in</th>
<th>was born to</th>
<th>and</th>
<th>birthplace(X,Y)</th>
<th>spouse(X,Y)</th>
</tr>
</thead>
<tbody>
<tr>
<td>John Lennon, Liverpool</td>
<td></td>
<td></td>
<td>1</td>
<td>?</td>
<td></td>
</tr>
<tr>
<td>John Lennon, Julia Lennon</td>
<td></td>
<td></td>
<td>1</td>
<td>?</td>
<td></td>
</tr>
<tr>
<td>John Lennon, Alfred Lennon</td>
<td></td>
<td></td>
<td>1</td>
<td>?</td>
<td></td>
</tr>
<tr>
<td>Julia Lennon, Alfred Lennon</td>
<td></td>
<td></td>
<td></td>
<td>?</td>
<td></td>
</tr>
<tr>
<td>Barack Obama, Hawaii</td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Barack Obama, Michelle Obama</td>
<td></td>
<td></td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

Universal Schema, Riedel et al, NAACL (2013)
Matrix Factorization

\[ P(R(i, j)) = \sigma(p_{i,j} \cdot r_R) \]
Training: Stochastic Updates

Pick an observed cell, \( R(i, j) \):

- Update \( p_{ij} \) & \( r_{R} \) such that \( R(i, j) \) is higher

Pick any random cell, assume it is negative:

- Update \( p_{xy} \) & \( r_{R'} \) such that \( R'(x, y) \) is lower
Relation Embeddings

$p_{Barack,Michelle}$
$p_{spouse}$
$p_{George,Laura}$
$r_{bornIn}$
$r_{was-born-in}$
$r_{livedIn}$
$r_{is-native-of}$
Embeddings ~ Logical Relations

Relation Embeddings, w
- Similar embedding for 2 relations denote they are paraphrases
  - is married to, spouseOf(X,Y), /person/spouse
- One embedding can be contained by another
  - w(topEmployeeOf) ⊂ w(employeeOf)
  - topEmployeeOf(X,Y) → employeeOf(X,Y)
- Can capture logical patterns, without needing to specify them!

Entity Pair Embeddings, v
Similar entity pairs denote similar relations between them
Entity pairs may describe multiple “relations”
  independent foundedBy and employeeOf relations
## Similar Embeddings

<table>
<thead>
<tr>
<th>Time, Inc</th>
<th>Amer. Tel. and Comm.</th>
<th>1</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Volvo</td>
<td>Scania A.B.</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Campeau</td>
<td>Federated Dept Stores</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Apple</td>
<td>HP</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Successfully predicts “Volvo owns percentage of Scania A.B.” from “Volvo bought a stake in Scania A.B.”
Implications

X historian at Y → X professor at Y

X professor at Y → X historian at Y

(Freeman, Harvard) → (Boyle, Ohio State)

Kevin Boyle
Ohio State

R. Freeman
Harvard

Learns asymmetric entailment:
PER historian at UNIV → PER professor at UNIV

But,
PER professor at UNIV ↳ PER historian at UNIV

From Sebastian Riedel
Two Related Tasks

Relation Extraction

Graph Completion
Graph Completion

- Alfred Lennon
  - childOf: Julia Lennon
  - livedIn: "born in __, to"

- John Lennon
  - childOf: Julia Lennon
  - livedIn: "born in __, to"
  - birthplace: Liverpool

- Liverpool
  - "was born in" John Lennon

- Julia Lennon
  - "was born to" Alfred Lennon
  - "and" John Lennon

- "born in __, to" Julia Lennon
Graph Completion

- Liverpool
- John Lennon
- Alfred Lennon
- Julia Lennon

- birthplace
- livedIn
- childOf
- spouse
Tensor Formulation of KG

Does an unseen relation exist?
Factorize that Tensor

\[ S(r(a, b)) = f(v_r, v_a, v_b) \]
Many Different Factorizations

**CANDECOMP/PARAFAC- Decomposition**

\[ S(r(a, b)) = \sum_{k} R_{r,k} \cdot e_{a,k} \cdot e_{b,k} \]

**Tucker2 and RESCAL Decompositions**

\[ S(r(a, b)) = (R_{r} \times e_{a}) \times e_{b} \]

**Model E**

\[ S(r(a, b)) = R_{r,1} \cdot e_{a} + R_{r,2} \cdot e_{b} \]

**Holographic Embeddings**

\[ S(r(a, b)) = R_{r} \times (e_{a} \star e_{b}) \]

Not tensor factorization (per se)

Translation Embeddings

\[
S(r(a, b)) = -\|e_a + R_r - e_b\|^2_2
\]

**TransH**

\[
S(r(a, b)) = -\|e_a^\perp + R_r - e_b^\perp\|^2_2
\]

\[
e_a^\perp = e_a - w_r^T e_a w_r
\]

**TransR**

\[
S(r(a, b)) = -\|e_a M_r + R_r - e_b M_r\|^2_2
\]

---

Parameter Estimation

Observed cell: increase score
\[ S(r(a, b)) \]

Unobserved cell: decrease score
\[ S(r'(x, y)) \]
Matrix vs Tensor Factorization

- Vectors for each entity pair
- Can only predict for entity pairs that appear in text together
- No sharing for same entity in different entity pairs

- Vectors for each entity
- Assume entity pairs are “low-rank”
  - But many relations are not!
  - Spouse: you can have only ~1
- Cannot learn pair specific information
What they can, and can’t, do..

- **Red**: deterministically implied by **Black**
  - needs *pair-specific* embedding
  - Only F is able to generalize
- **Green**: needs to estimate entity types
  - needs *entity-specific* embedding
  - Tensor factorization generalizes, F doesn't
- **Blue**: implied by **Red** and **Green**
  - Nothing works much better than random

Joint Extraction+Completion

- Joint Model
- Relation Extraction
- Graph Completion

Surface pattern

Relation

relation

relation

relation

relation
So far, we’re learning vectors for each entity/surface pattern/relation.

But learning vectors independently ignores “composition”

### Composition in Surface Patterns
- Every surface pattern is not unique
- Synonymy: A is B’s spouse.
  A is married to B.
- Inverse: X is Y’s parent.
  Y is one of X’s children.
- Can the representation learn this?

### Composition in Relation Paths
- Every relation path is not unique
- Explicit: A parent B, B parent C
  A grandparent C
- Implicit: X bornInCity Y, Y cityInState Z
  X “bornInState” Z
- Can the representation capture this?
Composing Dependency Paths

... was born to ...

... ‘s parents are ...

\(\text{parentsOf} \)

(never appears in training data)

But we don’t need linked data to know they mean similar things...

Use neural networks to produce the embeddings from text!

Composing Relational Paths

Review: Embedding Techniques

Two Related Tasks:
- Relation Extraction from Text
- Graph (or Link) Completion

Relation Extraction:
- Matrix Factorization Approaches

Graph Completion:
- Tensor Factorization Approaches

Compositional Neural Models
- Compose over dependency paths
- Compose over relation paths
Tutorial Overview

Part 1: Knowledge Graphs

Part 2: Knowledge Extraction

Part 3: Graph Construction

Part 4: Critical Analysis