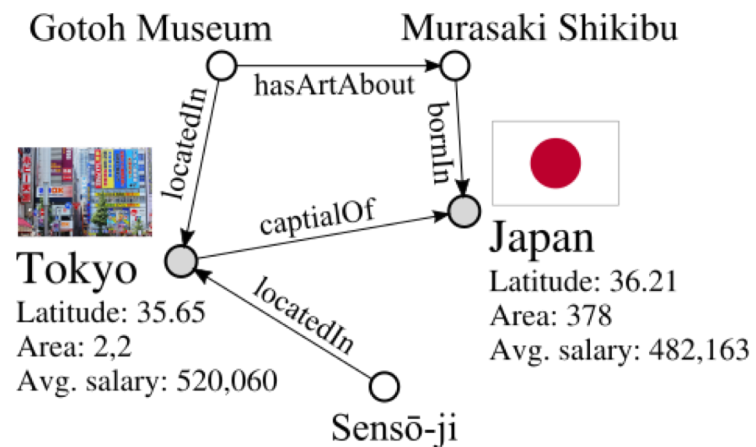


Learning from Multi-Modal and Graph-Structured Data

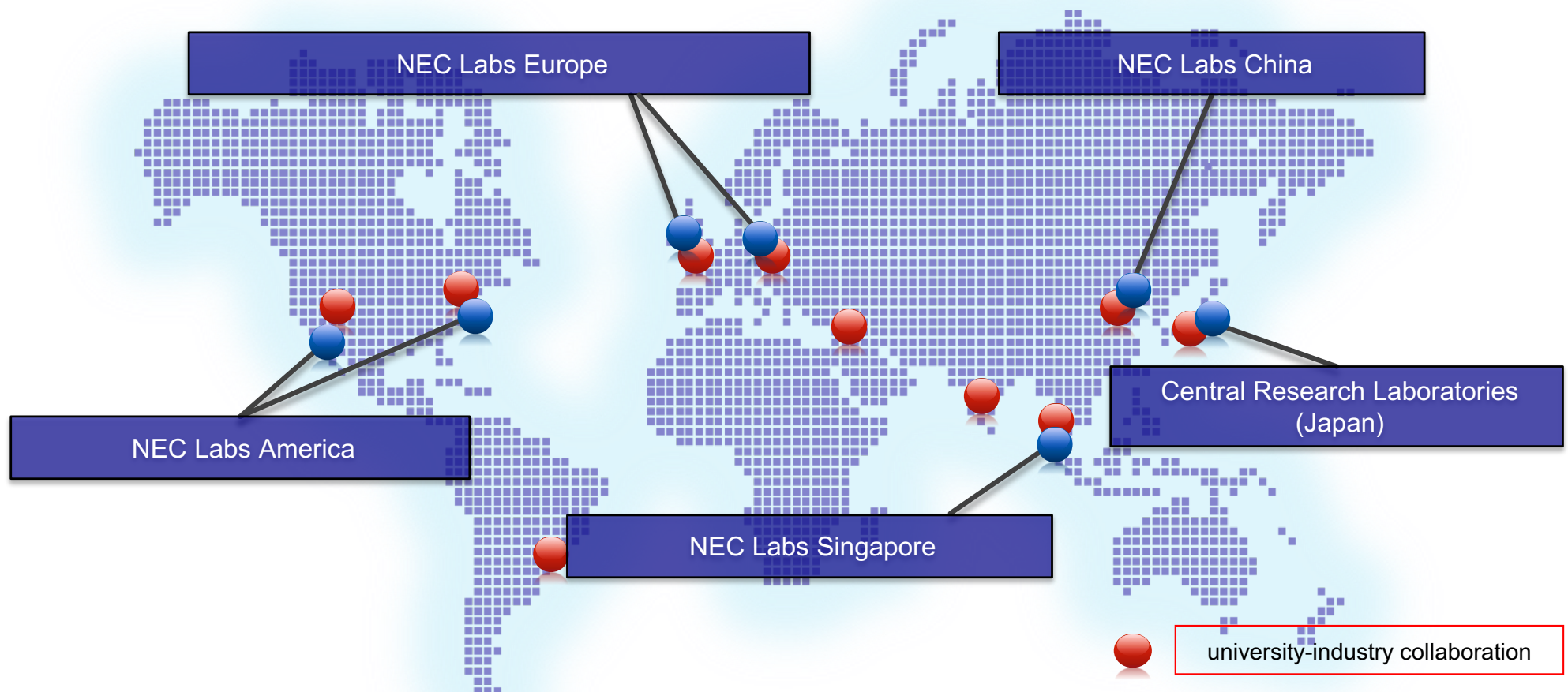
Alberto Garcia-Duran and Mathias Niepert

**NEC Labs Europe
Heidelberg**



NEC Global R&D Activities

- (1) Create new technologies and business directions through collaboration between labs and academia
- (2) Reinforce global open innovation, corporate with academia and industry partners



NEC Labs Europe: What do we do?

- ~ 80 researchers, ~80% PhDs, 20 nationalities

- Pure research lab, no product development

- Main objectives:

1. Research output for top tier conferences
2. Stable prototypes for technology transfer
3. Patent applications

- Product prototypes based on lab's research



Research Collaborations

■ NEC Japan (business units and central labs)

- Digital Health
- Retail
- Finance
- Networked Systems



■ EU Projects

- Exploration of applications not coming from NEC
- Opportunity to stay in touch with research community
- Understand trends and problems in the SME market



■ Third party Collaborations

- DKFZ
- University of Heidelberg medical school



Medizinische Fakultät Heidelberg

Systems and ML Research Group

14 ML Researchers

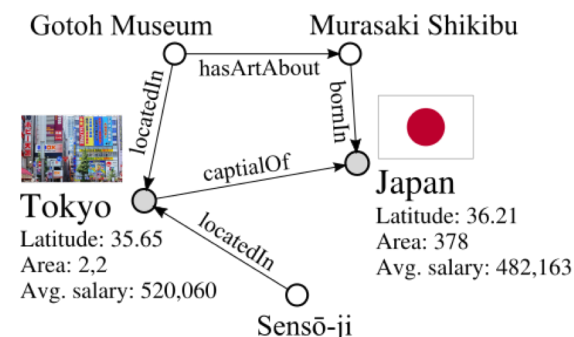
12 Systems Researchers



Main Research Themes

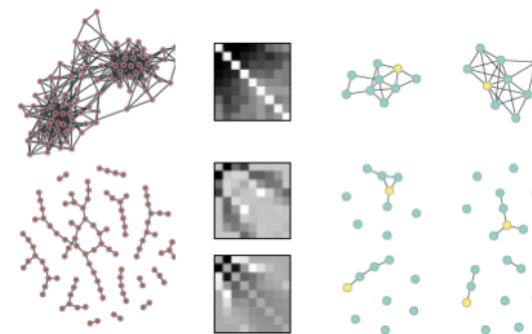
Multi-Modal Learning and Reasoning

- Combining different attribute types and modalities
- Knowledge graphs for multi-modal learning**
(combining deep learning and logical reasoning)



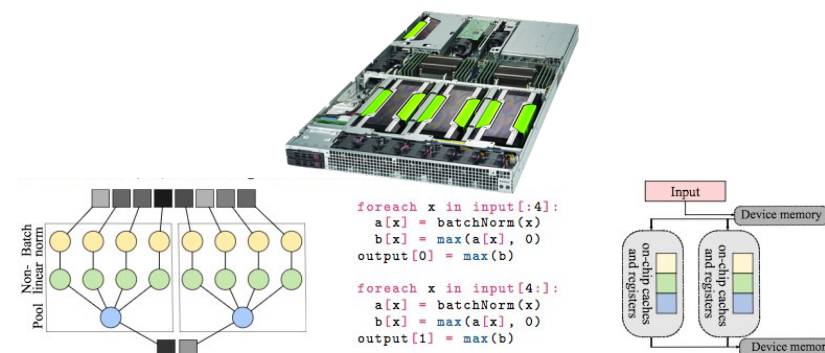
Graph-based Machine Learning

- Learning graph representations
- Unsupervised and semi-supervised learning**



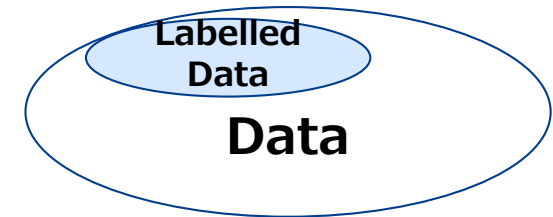
Systems and ML

- ML for Systems and Systems for ML
- CPU/GPU/network optimizations etc.
- Deep learning for data networks

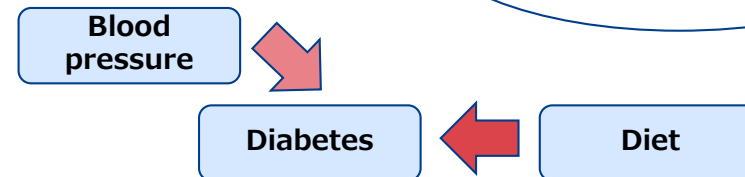


Technological Challenges

ML that works without much labelled data
(unsupervised and semi-supervised learning)



Interpretable and Explainable AI



Ability to combine different data modalities
(data integration, multi-modal learning)



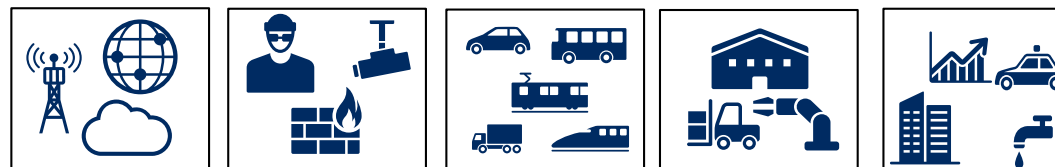
Latitude: 35.65

Tokyo
(Japanese: [\[to:kjo:\]](#) ([listen](#)), English: [/'toʊki.oʊ/](#)), officially **Tokyo Metropolis**,^[6] is the capital of [Japan](#) and one of its 47 [prefectures](#).^[7]

Efficiency and support of real time predictions
(network speed if required)

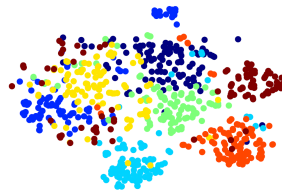
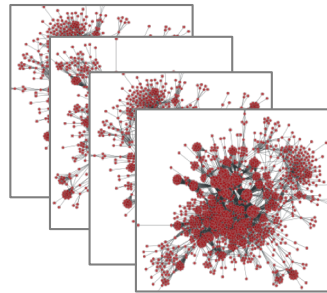


Applicable to several business use cases (horizontal technology)



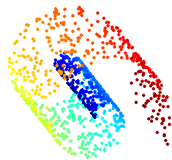
Graph-Based Machine Learning

Learn representations for entire graphs

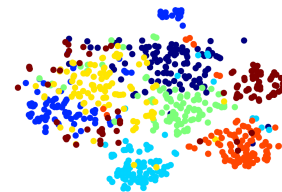
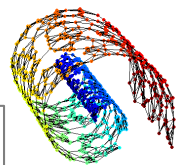


Graph classification/
regression problems

Learn representations for nodes

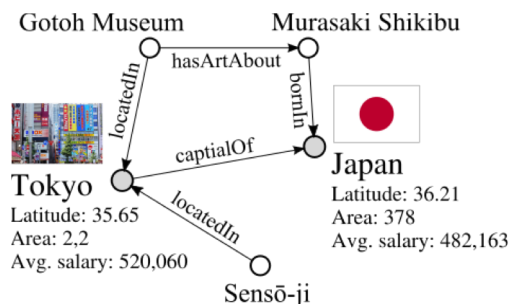
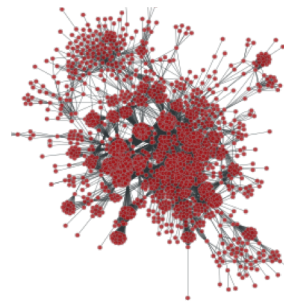


Induce graph



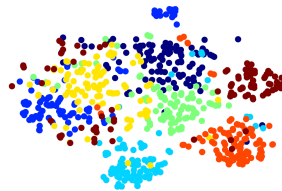
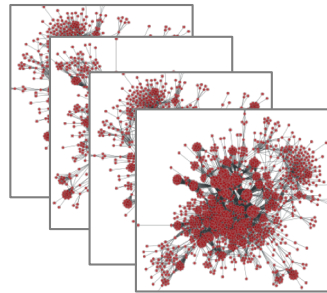
Node classification/
regression problems

Link prediction
problems

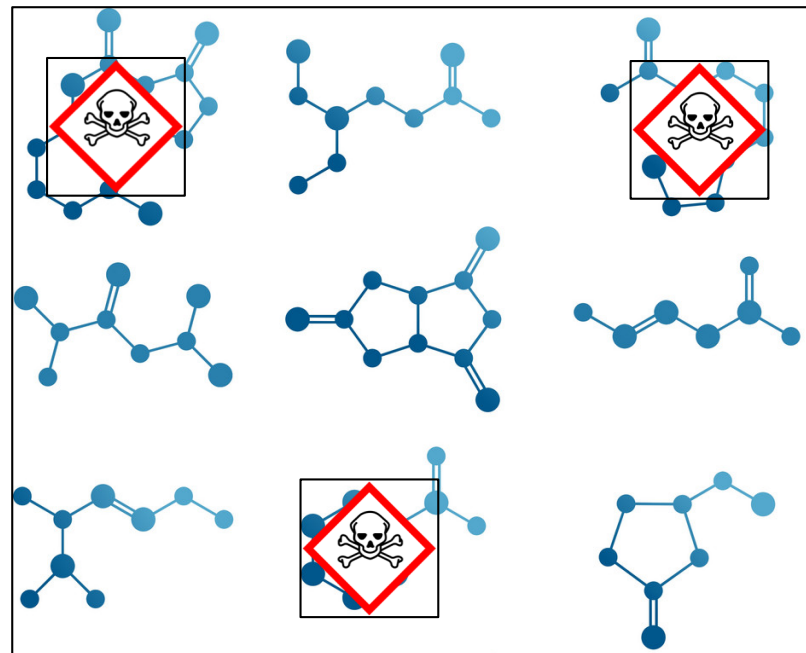


Example Applications – Drug Discovery

Learn representations for entire graphs

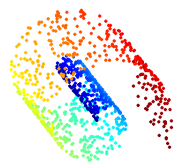


Graph classification/
regression problems

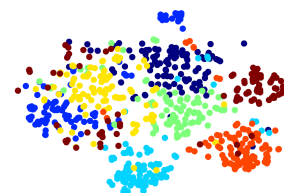
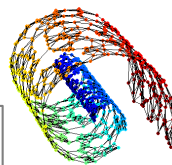


Example Applications – Patient Outcome Prediction

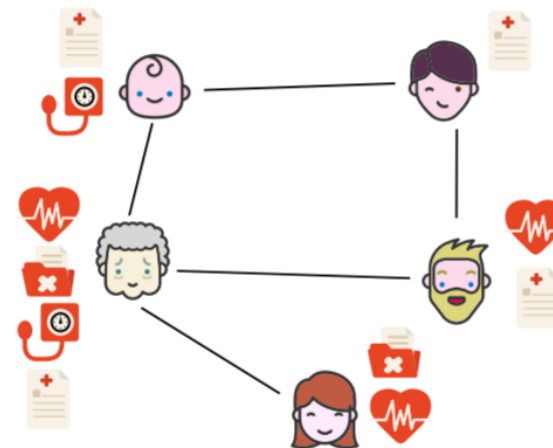
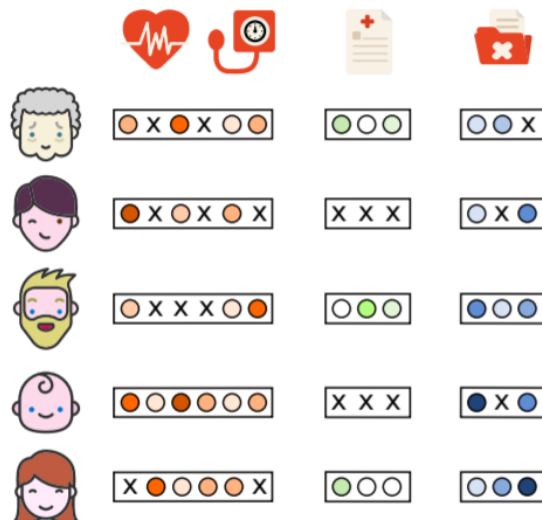
Learn representations for nodes



Induce graph

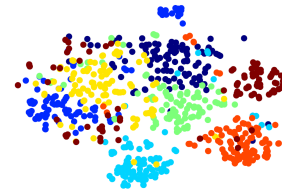
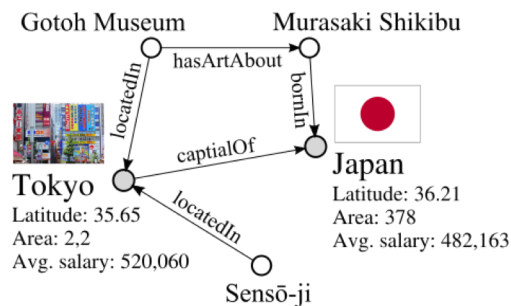


Node classification/
regression problems

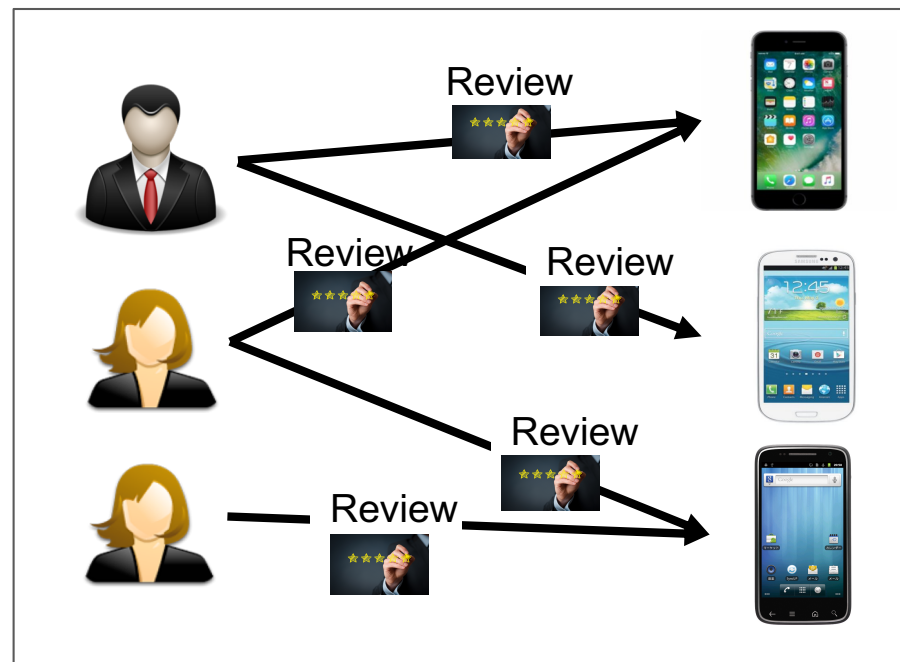


Example Applications – Recommender Systems

Learn representations for nodes

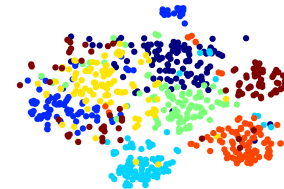
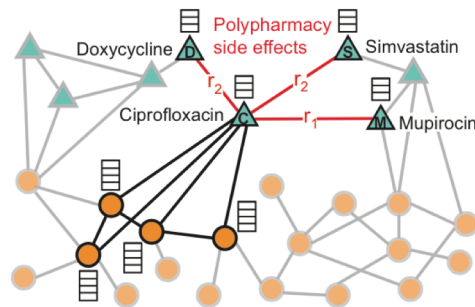


Link prediction problem

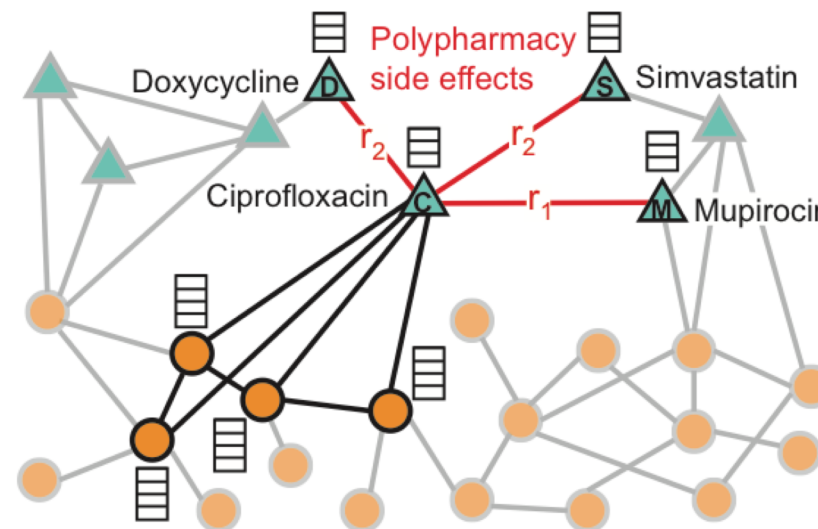


Example Applications – Polypharmacy Prediction

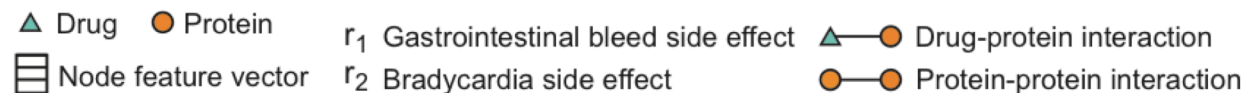
Learn representations for nodes



Link prediction problem

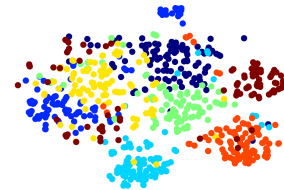
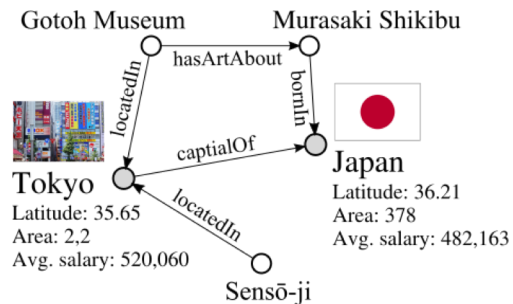


© SNAP 2018

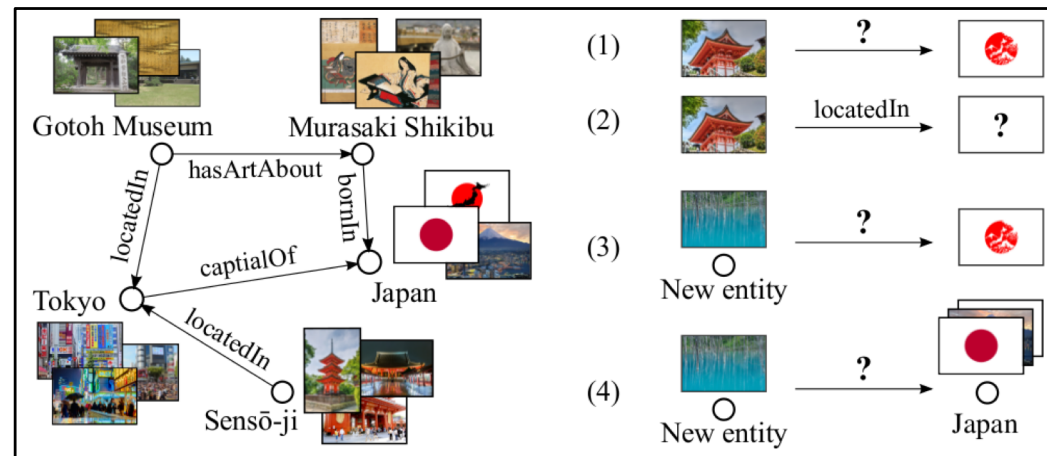
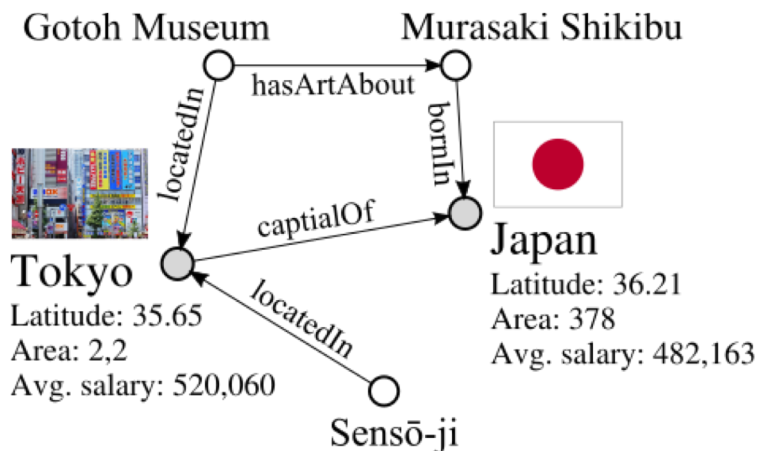


Example Applications – Knowledge Base Completion

Learn representations for nodes



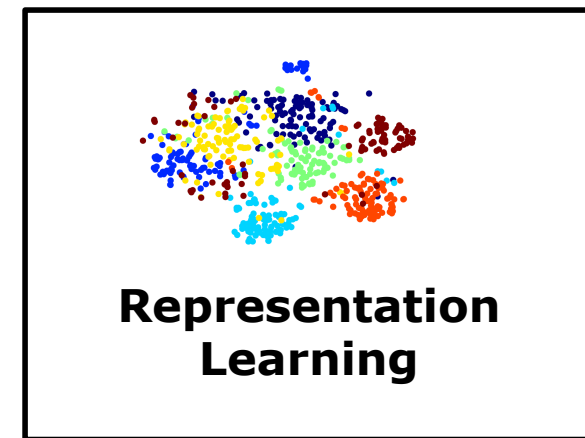
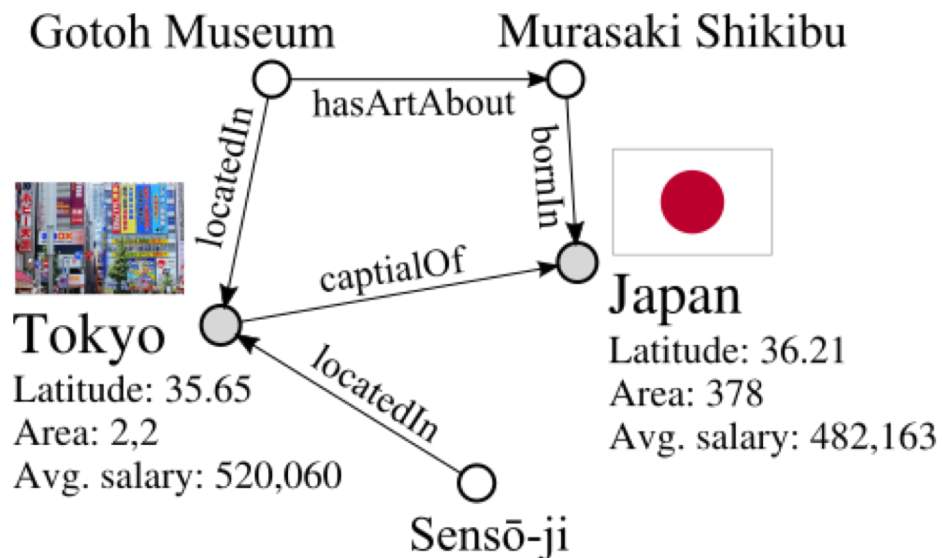
Link prediction problem



Outline of the First Part of our Lecture

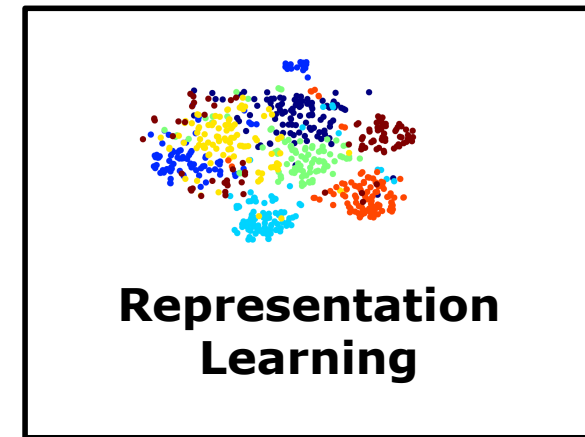
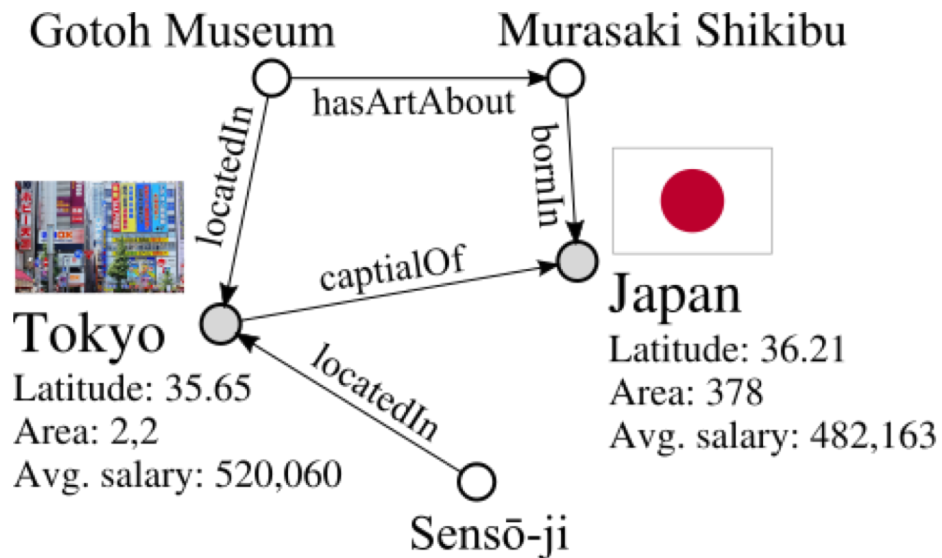
1. Basic Concepts
2. Two Perspectives on Learning from Graphs
 - Knowledge Graph = Tensor (KB completion, evaluation, etc.)
 - Learning from Local Structure (learning from paths and neighborhoods)
3. Some Practical Observations

What Problem Does the Lecture Address?

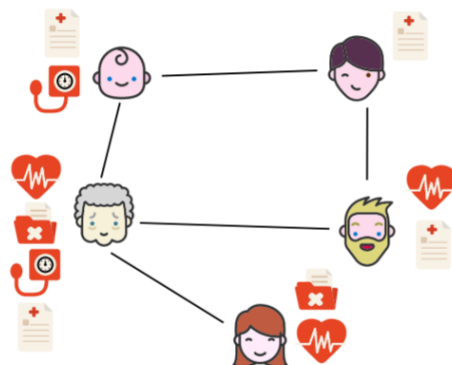


1. Properties of nodes
2. Properties of links
3. Properties of graphs
4. Prediction of missing links
5. More complex queries

What Problem Does the Lecture Address?



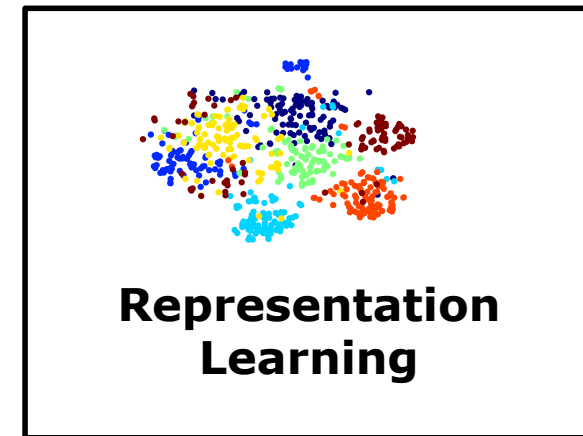
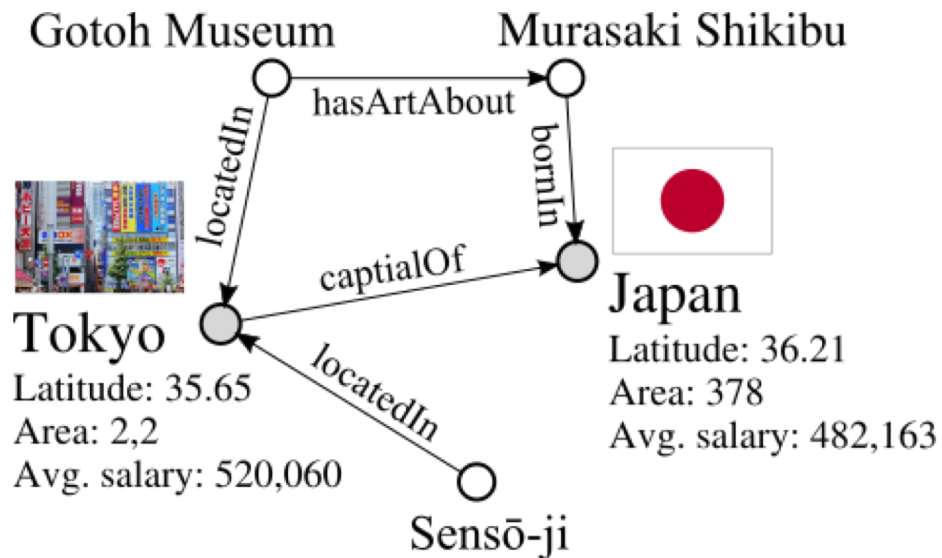
Answering queries



What is the diagnosis and outlook of bearded patient?

1. Properties of nodes
2. Properties of links
3. Properties of graphs
4. Prediction of missing links
5. More complex queries

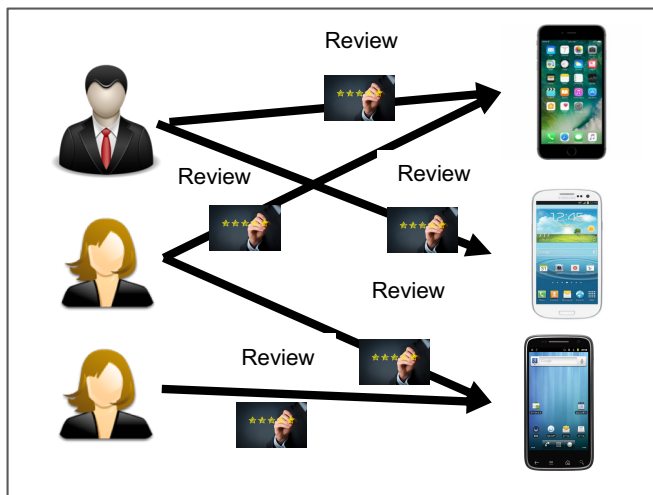
What Problem Does the Lecture Address?



Answering queries

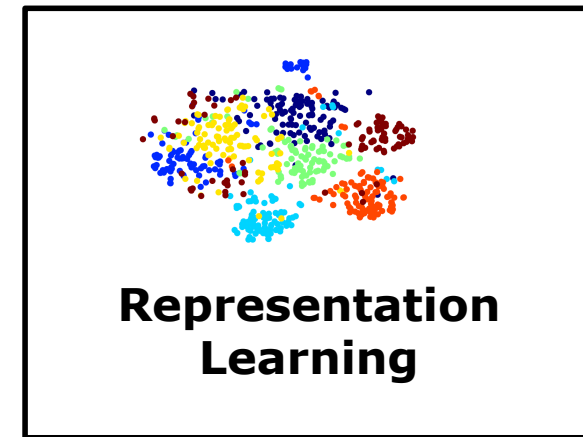
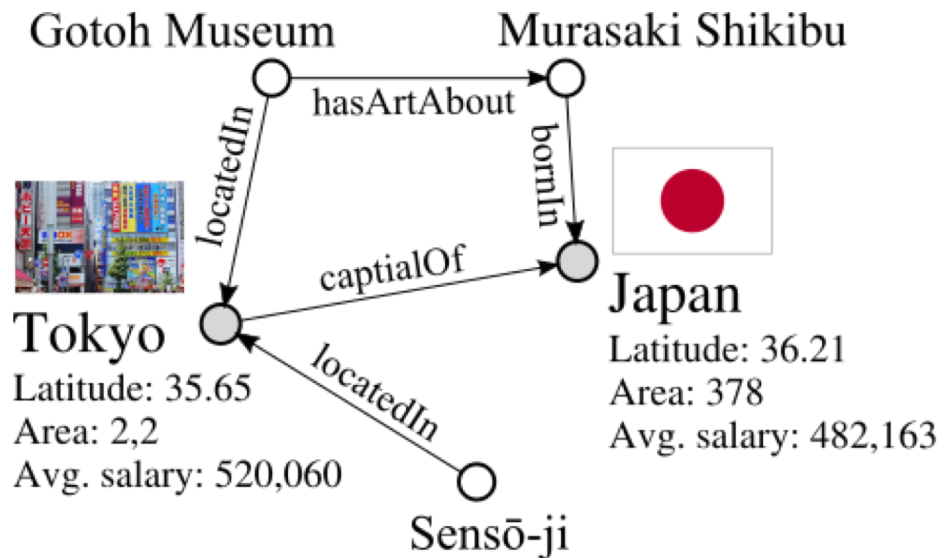


1. Properties of nodes
2. Properties of links
3. Properties of graphs
4. Prediction of missing links
5. More complex queries

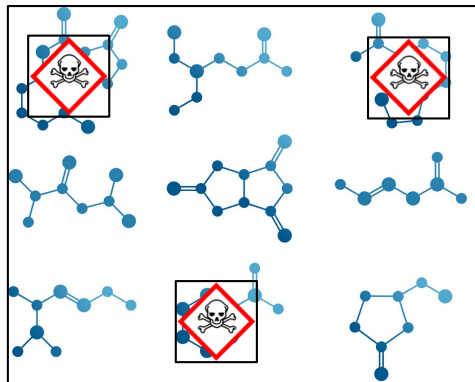


What's the rating
user A would give
to product 3?

What Problem Does the Lecture Address?

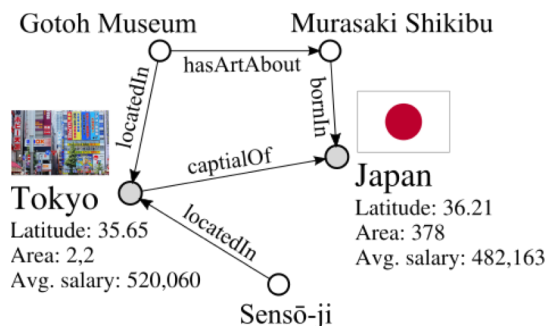
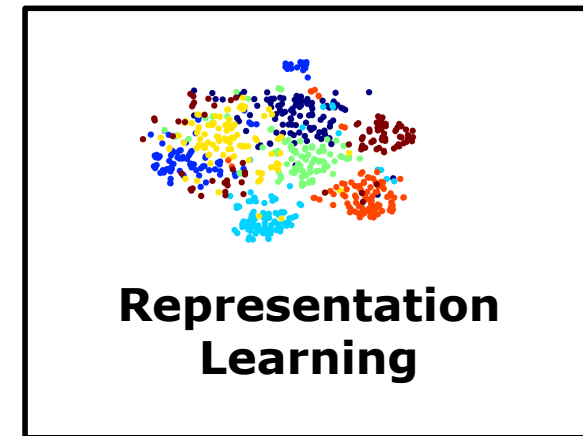
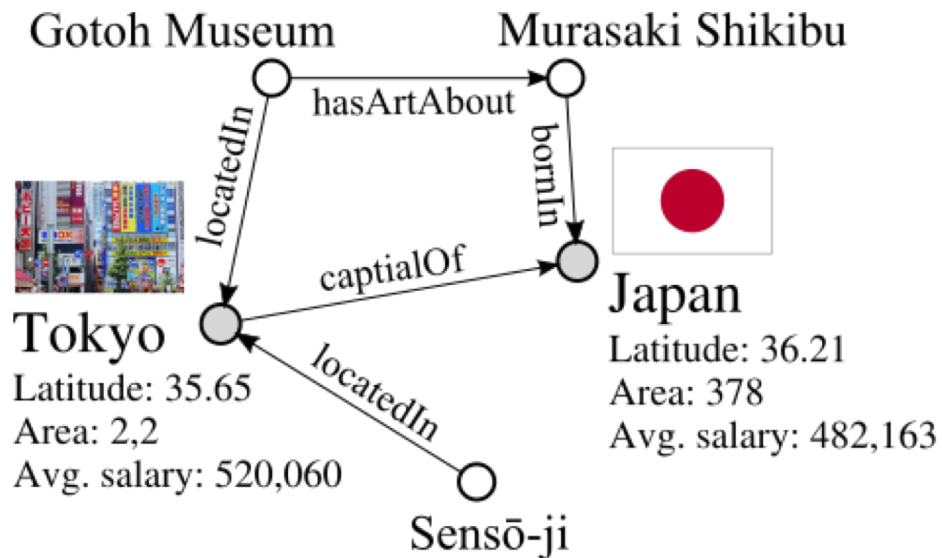


Answering queries



1. Properties of nodes
2. Properties of links
3. Properties of graphs
4. Prediction of missing links
5. More complex queries

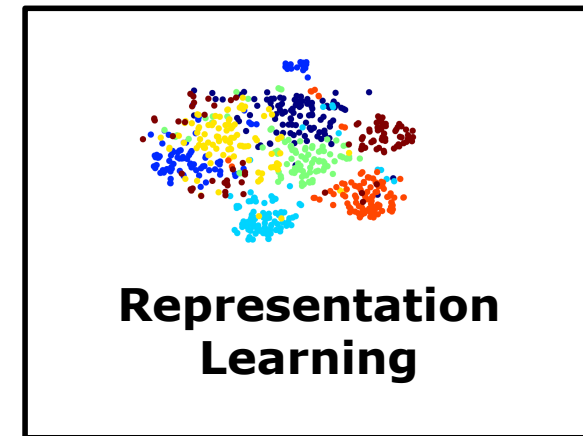
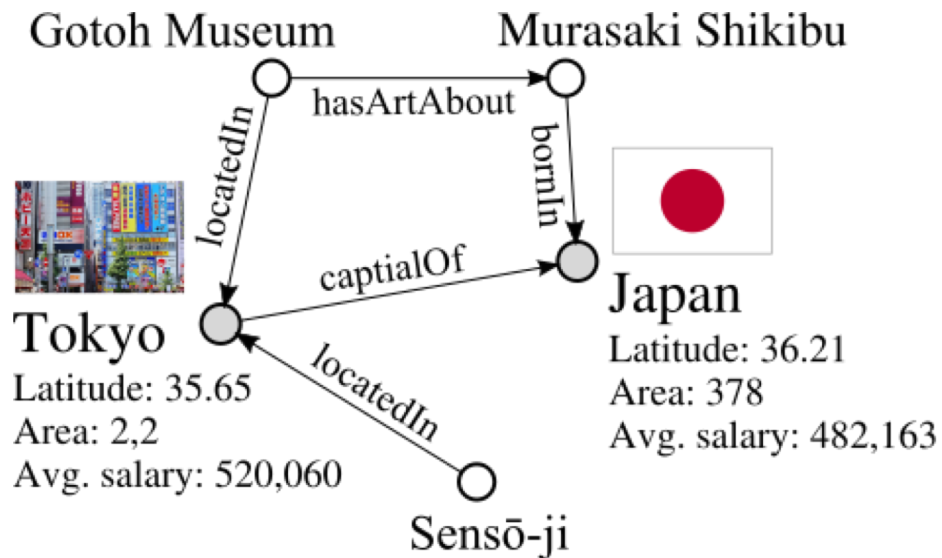
What Problem Does the Lecture Address?



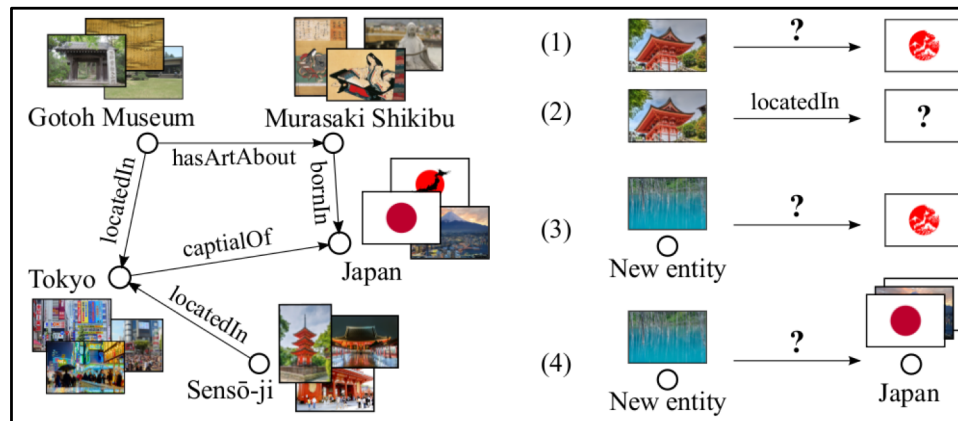
Where is Senso-ji located (besides Tokyo) ?

1. Properties of nodes
2. Properties of links
3. Properties of graphs
4. Prediction of missing links
5. More complex queries

What Problem Does the Lecture Address?



Answering queries



1. Properties of nodes
2. Properties of links
3. Properties of graphs
4. Prediction of missing links
5. More complex queries

A Quick Word Before We Start

- | SRL (ProbLog, Markov Logic, PSL, etc.) has been **successfully** used to learn from graph structured data
- | **Assumption:** other lectures have covered these topics
- | **Hope:** Combine concepts from SRL and representation learning to have advantages of both



Basic Graph Terminology

- Node identifier
- Node label

Gotoh Museum

Murasaki Shikibu



Tokyo

Latitude: 35.65
Area: 2,2
Avg. salary: 520

locatedIn

hasArtAbout

bornIn

capitalOf

locatedIn

Sensō-ji



Japan

Latitude: 36.21
Area: 378
Avg. salary: 482,163

- Relation type
- (Relation)
- Predicate

- Node
- Entity
- Object

- Node features
- Node attributes

Multi-relational graphs without additional node features can be represented as a list of triples **(h, r, t)**

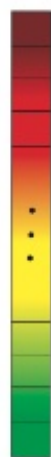
Head entity

Relation type

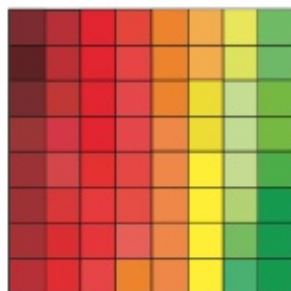
Tail entity

Vectors and Tensors

vector



matrix

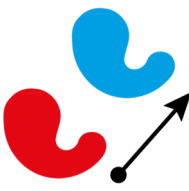


tensor



Vector Operations

Summation

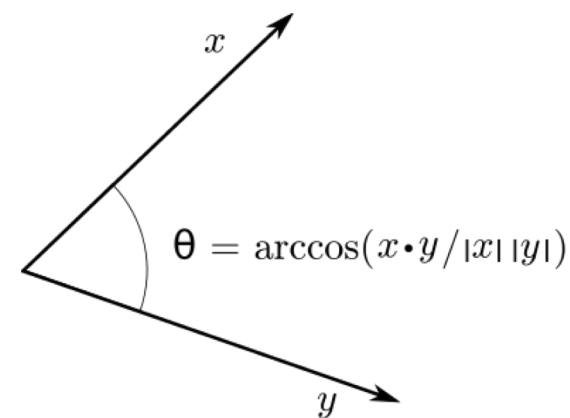
$$\begin{bmatrix} a \\ b \\ c \\ d \\ e \end{bmatrix} + \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{bmatrix} = \begin{bmatrix} a+1 \\ b+2 \\ c+3 \\ d+4 \\ e+5 \end{bmatrix}$$


Elementwise multiplication

$$\begin{bmatrix} a \\ b \\ c \\ d \\ e \end{bmatrix} * \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{bmatrix} = \begin{bmatrix} a1 \\ b2 \\ c3 \\ d4 \\ e5 \end{bmatrix}$$

Vector dot product

$$\begin{bmatrix} a \\ b \\ c \\ d \\ e \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{bmatrix} = a1 + b2 + c3 + d4 + e5$$



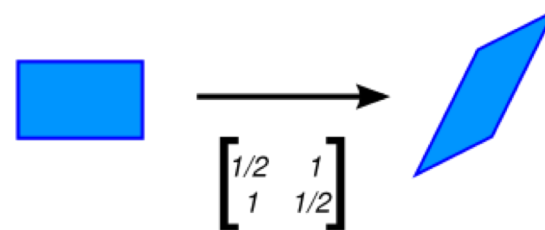
Matrix Operations

Elementwise multiplication (Hadamard product)

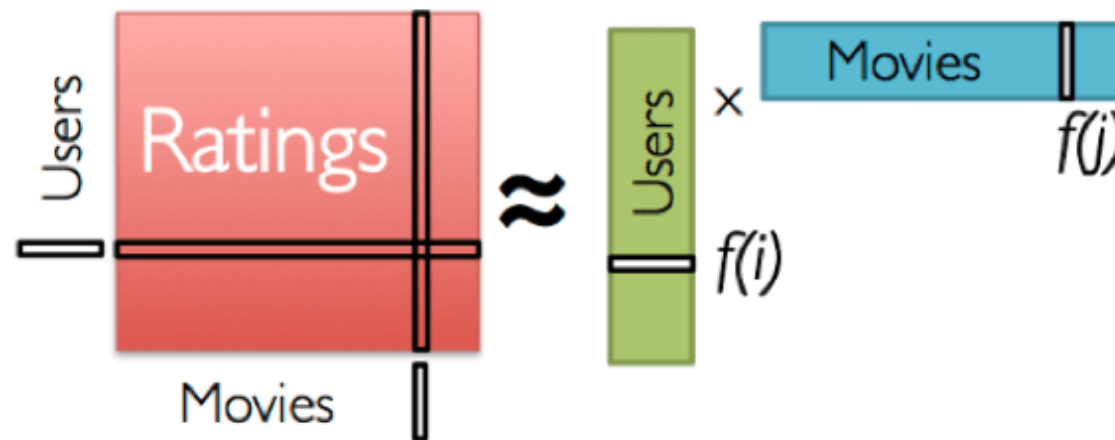
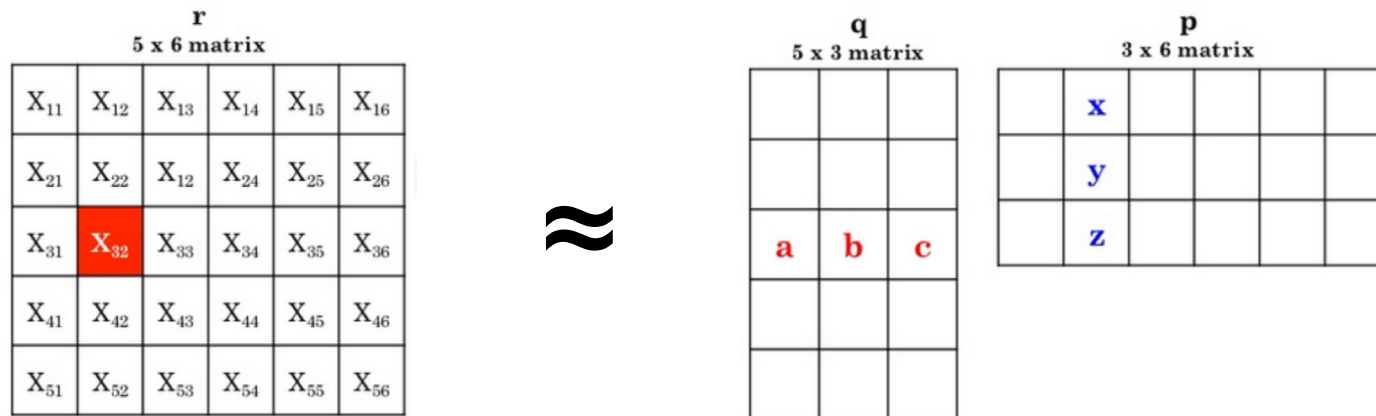
$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} * \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} = \begin{bmatrix} a1 & b2 & c3 \\ d4 & e5 & f6 \\ g7 & h8 & i9 \end{bmatrix}$$

Matrix multiplication

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \times \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} = \begin{bmatrix} a1+b4+c7 & a2+b5+c8 & a3+b6+c9 \\ d1+e4+f7 & d2+e5+f8 & d3+e6+f9 \\ g1+h4+i7 & g2+h5+i8 & g3+h6+i9 \end{bmatrix}$$

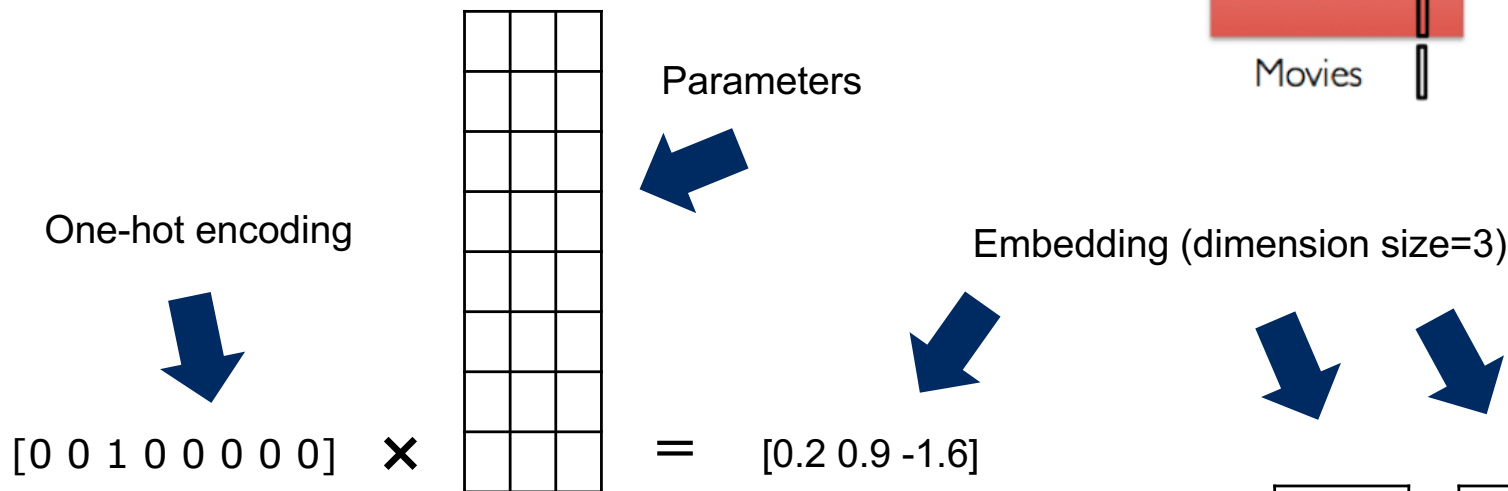


Matrix Factorization

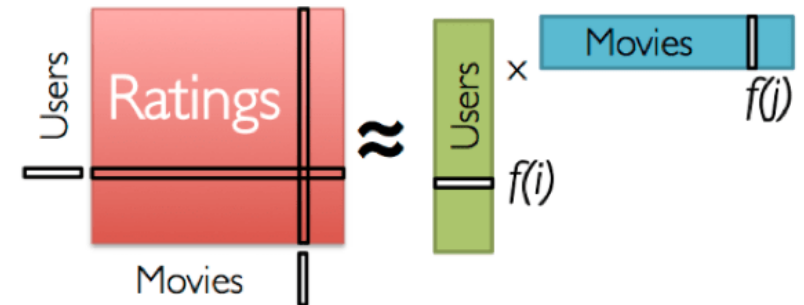


The Differential Programming Approach

- **Step 1:** Assume users and movies are represented with one-hot encoding and define **encoding** function f for users and movies



- **Step 2:** Define **scoring function** between user-movie pairs
- **Step 3:** Define a **loss** between scorings and actual existing user ratings
- **Step 4:** Apply **gradient decent** to train the model "end-to-end"



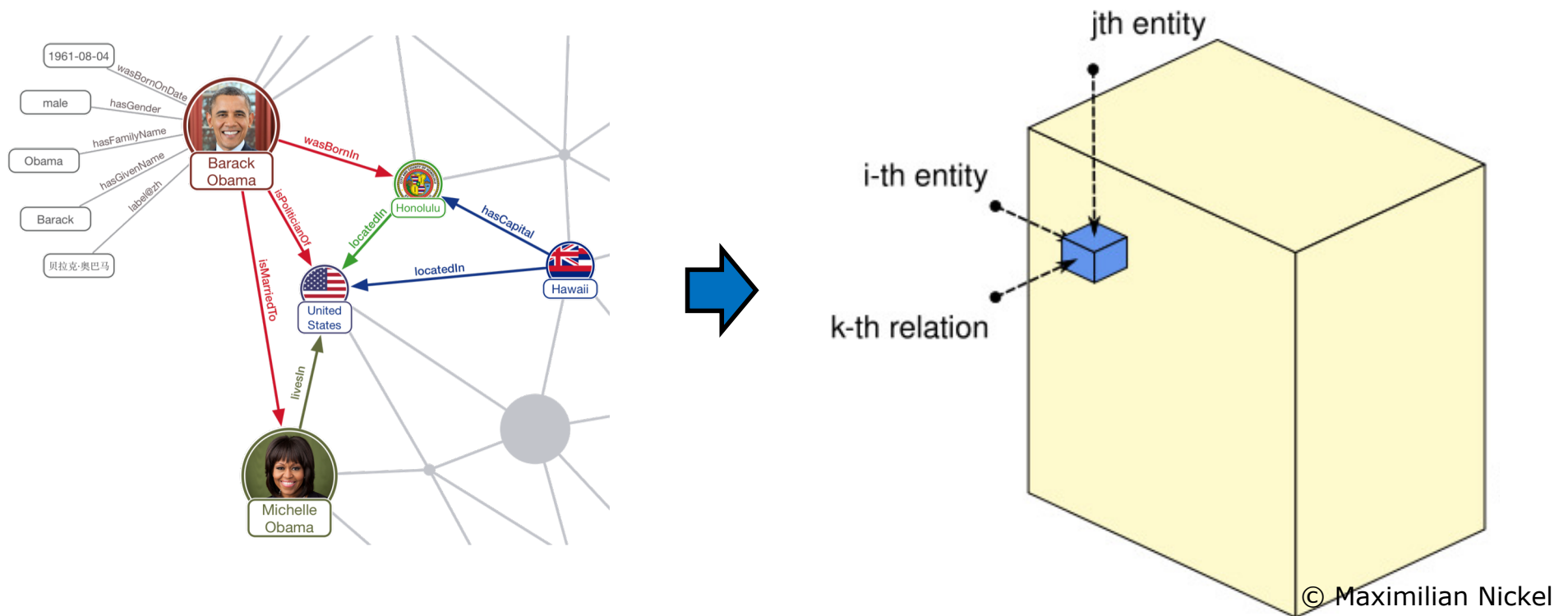
$$\text{Score} = \begin{bmatrix} 0.2 \\ 0.9 \\ -1.6 \end{bmatrix} \cdot \begin{bmatrix} 0.8 \\ -1.2 \\ 0.5 \end{bmatrix} = [-1.72]$$

$$\text{Loss} = (-1.72 - 3)^2$$

Observed rating

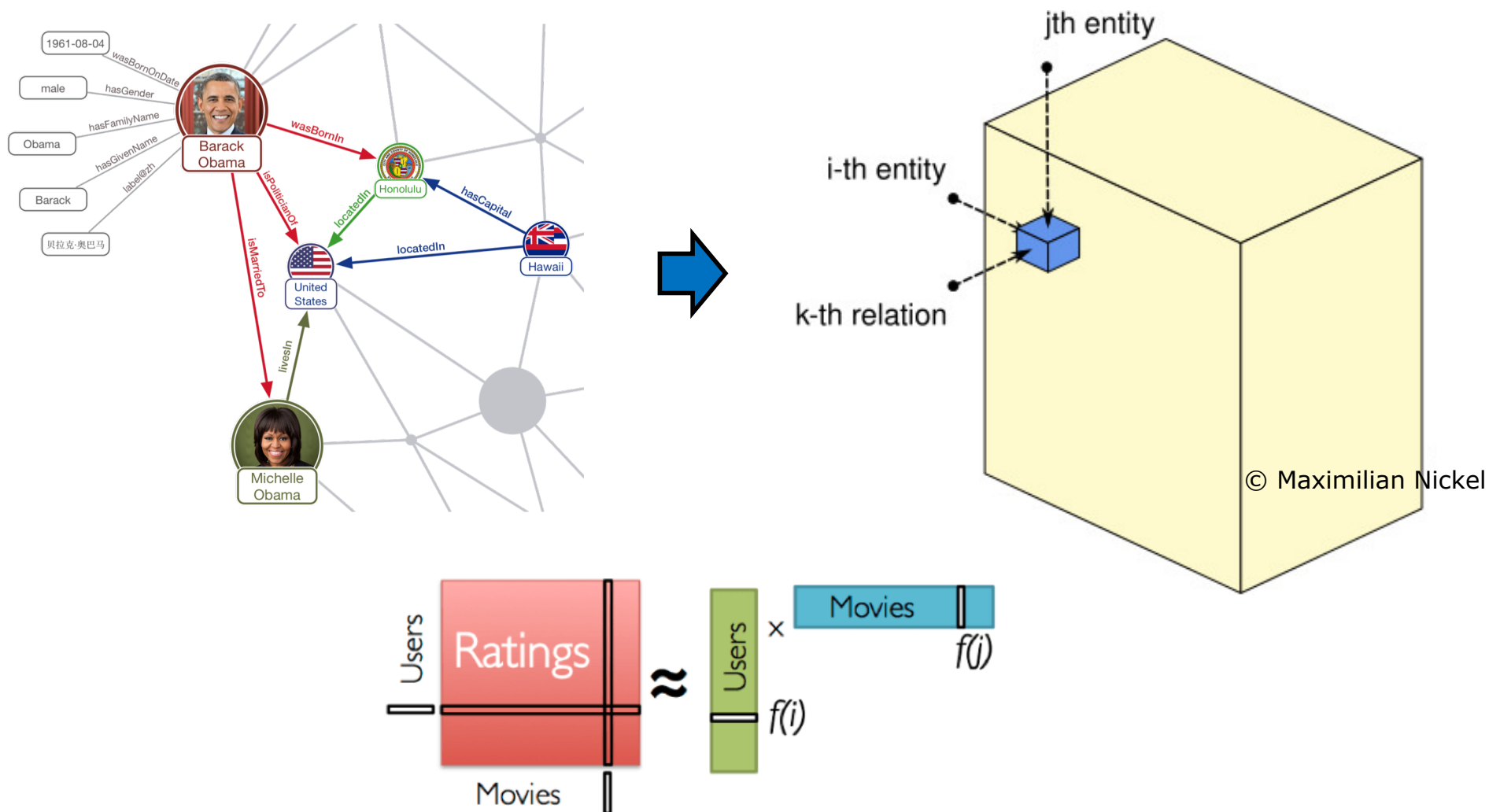
Two Perspectives on Learning from Graph Data

1. The multi-relational graph as a **3D tensor**

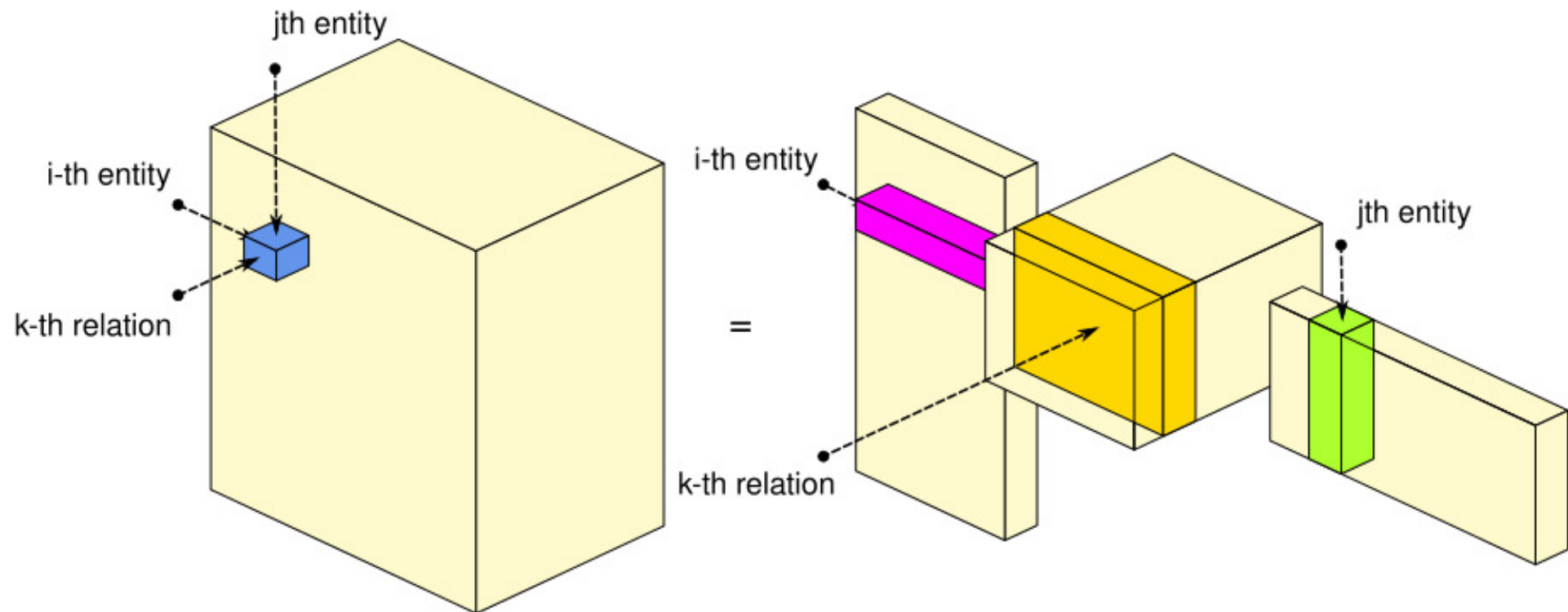


Two Perspectives on Learning from Graph Data

1. The multi-relational graph as a **3D tensor**

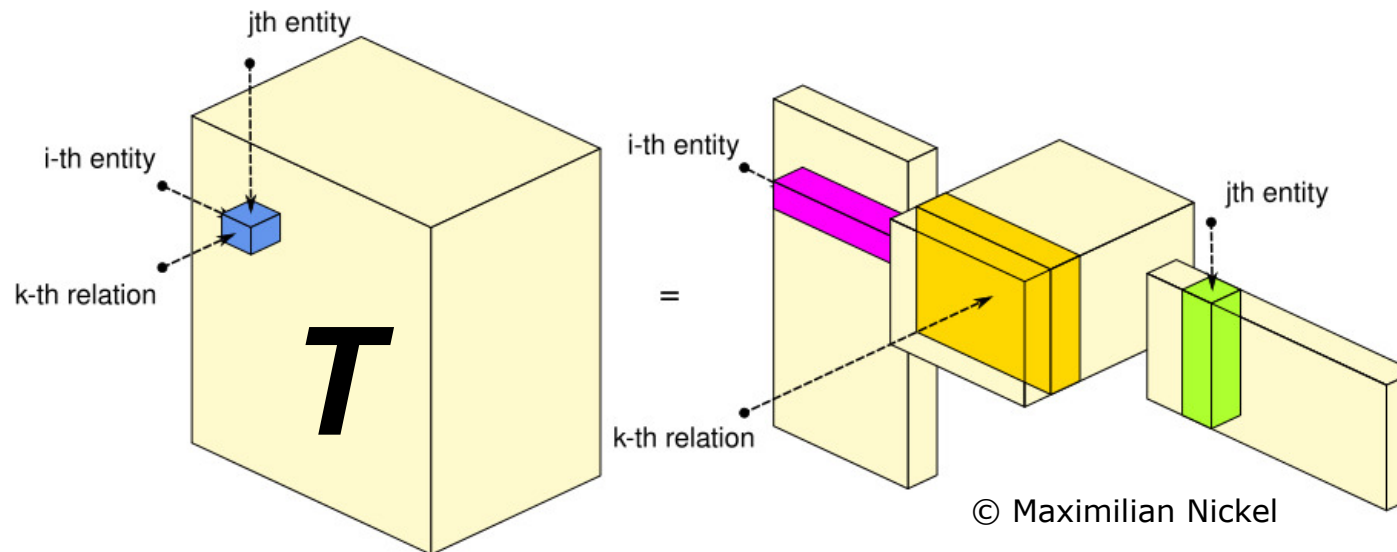


1. The multi-relational graph as a **3D tensor**


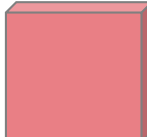


© Maximilian Nickel

Nickel et al, A Three-Way Model for Collective Learning on Multi-Relational Data, 2011



- **Step 1:** Choose the representation (encoding) for entities and relations

Entities: $e_i =$  Relation types: $w_r =$ 

- **Step 2:** Choose scoring function for triples (h, r, t) = coordinates in the 3D tensor

$$s(h, r, t) = e_h^T \cdot w_r \cdot e_t$$

- **Step 3:** Choose loss function

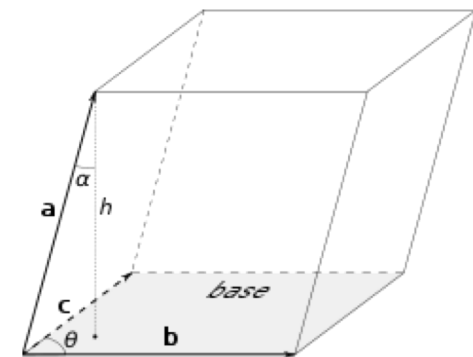
$$\sum_{h,r,t} (T_{\{h,r,t\}} - s(h,r,t))^2$$

- **DistMult:** top performing KB embedding method
- Simplifies RESCAL; relation matrix only non-zero in diagonal

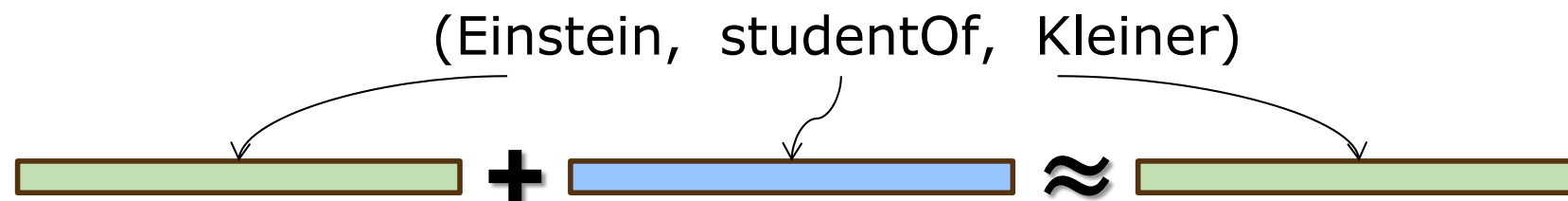
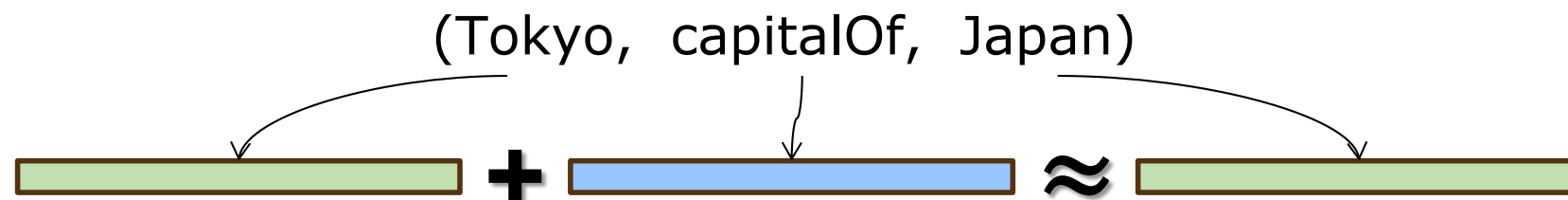
Triple: (h, r, t)

$$s(\mathbf{e}_h, \mathbf{e}_t, \mathbf{e}_r) = (\mathbf{e}_h * \mathbf{e}_t) \cdot \mathbf{e}_r$$

- **Geometric interpretation:** Absolute value is the volume of the 3D parallelepiped spanned by the three vectors

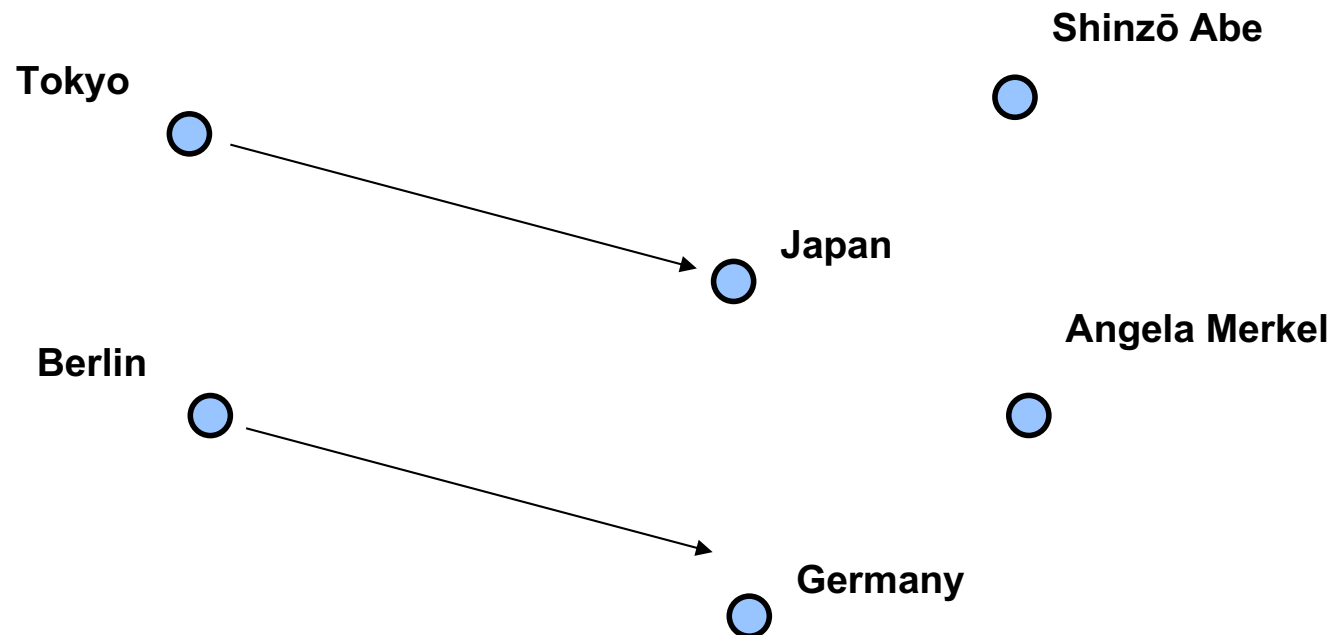


TransE learns embeddings of entities and relations



...

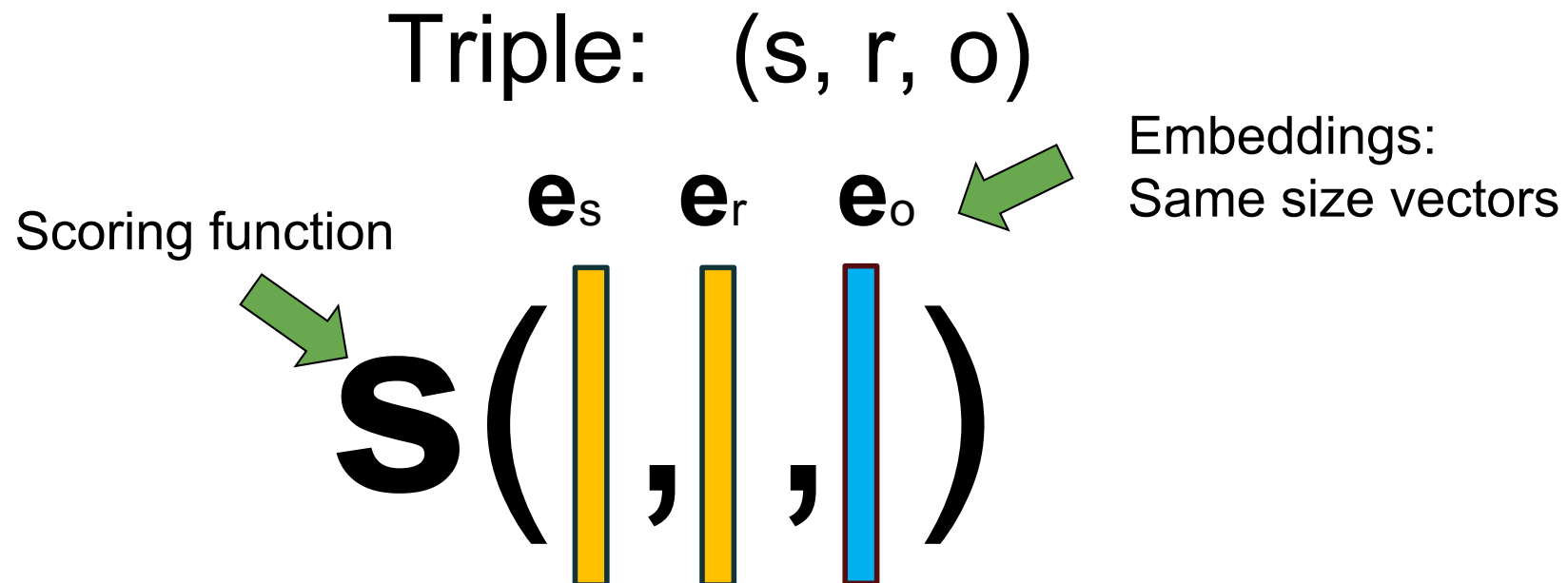
TransE learns embeddings of entities and relations



Geometric interpretation: Relation vector translates (moves) head entity embedding to tail entity embedding

Knowledge Graph Representations

- Many alternative scoring functions have been proposed



| Model | Scoring Function | Relation parameters |
|-------------------------------|--|--|
| RESCAL (Nickel et al., 2011) | $e_s^T W_r e_o$ | $W_r \in \mathbb{R}^{K^2}$ |
| TransE (Bordes et al., 2013b) | $\ e_s + w_r - e_o\ _p$ | $w_r \in \mathbb{R}^K$ |
| NTN (Socher et al., 2013) | $u_r^T f(e_s W_r^{[1..D]} e_o + V_r \begin{bmatrix} e_s \\ e_o \end{bmatrix} + b_r)$ | $W_r \in \mathbb{R}^{K^2 D}, b_r \in \mathbb{R}^K$ $V_r \in \mathbb{R}^{2KD}, u_r \in \mathbb{R}^K$ |
| DistMult (Yang et al., 2015) | $\langle w_r, e_s, e_o \rangle$ | $w_r \in \mathbb{R}^K$ |
| HolE (Nickel et al., 2016b) | $w_r^T (\mathcal{F}^{-1}[\mathcal{F}[e_s] \odot \mathcal{F}[e_o]])$ | $w_r \in \mathbb{R}^K$ |
| ComplEx | $\text{Re}(\langle w_r, e_s, \bar{e}_o \rangle)$ | $w_r \in \mathbb{C}^K$ |

Trouillon et al. 2016

Loss Functions

- Combines list of true triples with scoring function into a differentiable loss function
- Challenge:** open-world assumption → only positive examples
- Several losses have been proposed

- Margin based loss

$$\max(0, \gamma + s(h, r, t) - s(h, r, t'))$$

Positive margin

Randomly corrupted tail (or head)

- Softmax-based loss

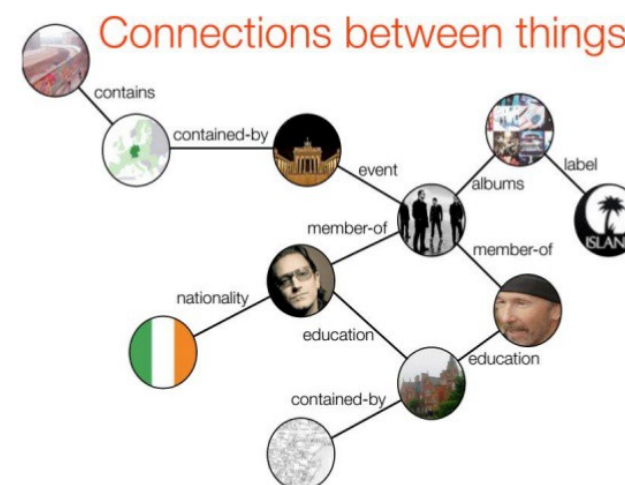
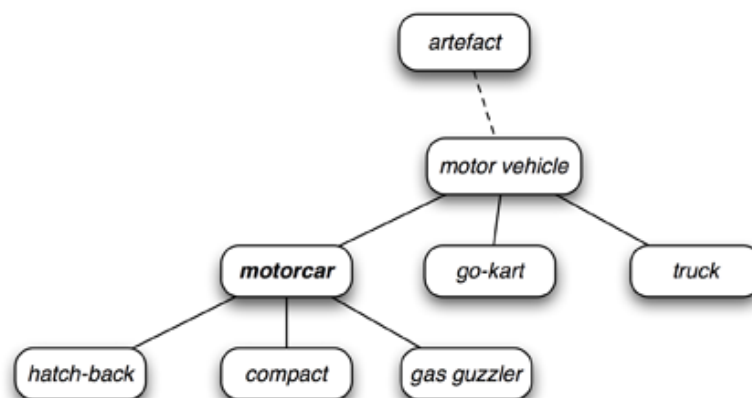
$$-\log\left(\frac{\exp(s(h, r, t))}{\sum_{triple \in \mathcal{C}} \exp(s(triple))}\right)$$

$\mathcal{C} = N$ corrupted triples

Evaluating KB Completion Methods

There are several benchmark data sets

- FB15k
- FB15k-237
- FB122
- WN18
- ...



| Data set | FB15k | FB15k-num | FB15k-237 | FB15k-237-num | WN18 | FB122 |
|---------------------|---------|-----------|-----------|---------------|---------|--------|
| Entities | 14,951 | 14,951 | 14,541 | 14,541 | 40,943 | 9,738 |
| Relation types | 1,345 | 1,345 | 237 | 237 | 18 | 122 |
| Training triples | 483,142 | 483,142 | 272,115 | 272,115 | 141,442 | 91,638 |
| Validation triples | 50,000 | 5,156 | 17,535 | 1,058 | 5,000 | 9,595 |
| Test triples | 59,071 | 6,012 | 20,466 | 1,215 | 5,000 | 11,243 |
| Relational features | 90,318 | 90,318 | 7,834 | 7,834 | 14 | 47 |

Evaluation Procedure

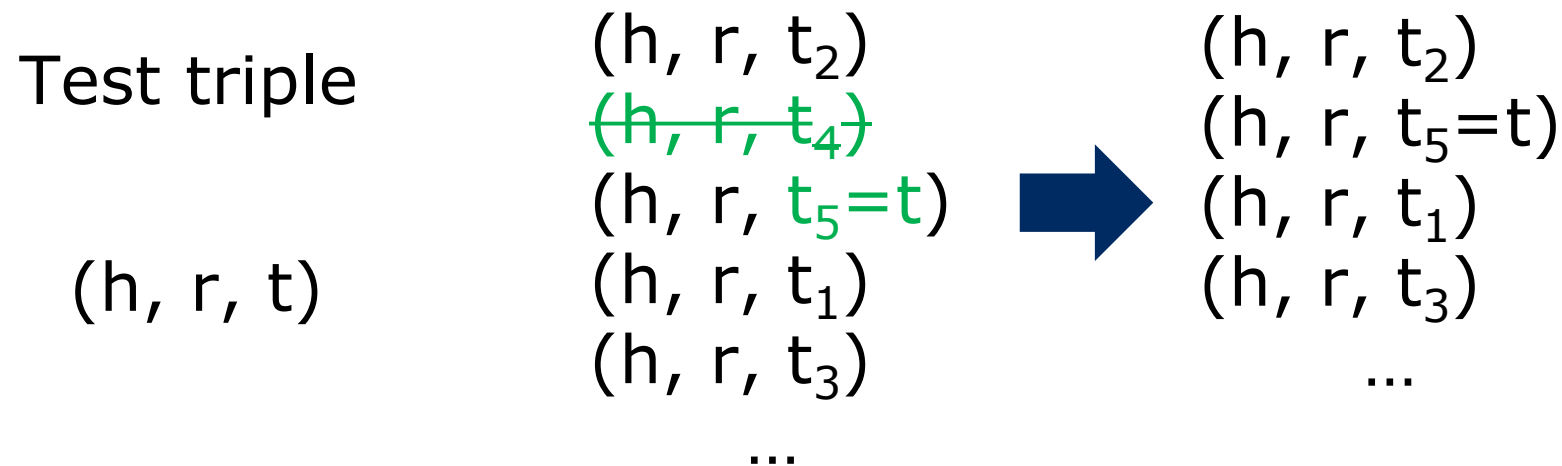
- Most knowledge graph benchmarks come with a predefined $\sim 80/10/10$ split of the triples
- Train the model on the training triples, tune hyperparameters on the validation triples, report metrics on the test triples

| Test triple | Substitute tail | Compute scores and rank |
|-------------|-----------------|-------------------------|
| (h, r, t) | (h, r, t_1) | (h, r, t_2) |
| | (h, r, t_2) | (h, r, t_4) |
| | (h, r, t_3) | $(h, r, t_5=t)$ |
| | (h, r, t_4) | (h, r, t_1) |
| | (h, r, t_5) | (h, r, t_3) |
| | ... | ... |

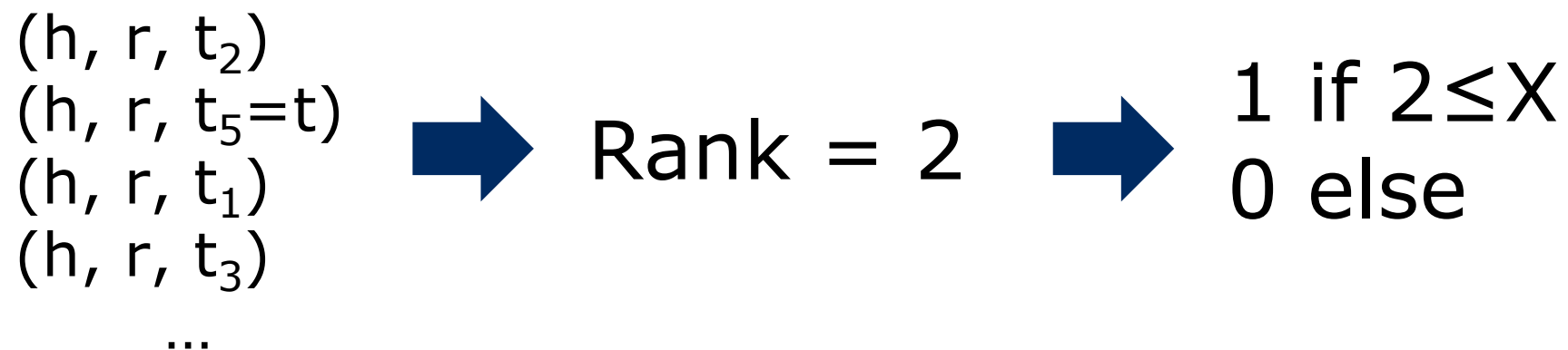
- Apply quality measures for rankings

Evaluation Metrics – Hits@X

Usually, other correct completions are removed



Hits@X: “Is the correct entity among the top X ranked entities?”



Evaluation Metrics –Mean Rank (MR)

Mean rank (MR): “The mean of the ranks of correct entity.”

Test triple

(h, r, t)

(h, r, t_2)

$(h, r, t_5=t)$

(h, r, t_1)

(h, r, t_3)

...



Rank = 2


Compute **average** of all ranks
for all test triples

Evaluation Metrics – Mean reciprocal rank

Mean reciprocal rank (MRR): “The mean of the ranks of correct entity.”

Test triple: (h, r, t)

(h, r, t₂)
(h, r, t₅=t)
(h, r, t₁)
(h, r, t₃)
...



Rank = 2

Compute average of the **reciprocal** of the rank of correct entity

$$MRR = \sum_{t \in Test} \frac{1}{rank(t)}$$

Some Recent Results

| Method | Filtered | | | | | | Extra features |
|---|------------|-------------|--------------|-------------|-------------|--------------|----------------|
| | WN18 | | | FB15k | | | |
| | MR | H10 | MRR | MR | H10 | MRR | |
| SE (Bordes et al., 2011) | 985 | 80.5 | - | 162 | 39.8 | - | None |
| Unstructured (Bordes et al., 2014) | 304 | 38.2 | - | 979 | 6.3 | - | |
| TransE (Bordes et al., 2013) | 251 | 89.2 | - | 125 | 47.1 | - | |
| TransH (Wang et al., 2014) | 303 | 86.7 | - | 87 | 64.4 | - | |
| TransR (Lin et al., 2015b) | 225 | 92.0 | - | 77 | 68.7 | - | |
| CTransR (Lin et al., 2015b) | 218 | 92.3 | - | 75 | 70.2 | - | |
| KG2E (He et al., 2015) | 331 | 92.8 | - | 59 | 74.0 | - | |
| TransD (Ji et al., 2015) | 212 | 92.2 | - | 91 | 77.3 | - | |
| lppTransD (Yoon et al., 2016) | 270 | 94.3 | - | 78 | 78.7 | - | |
| TranSparse (Ji et al., 2016) | 211 | 93.2 | - | 82 | 79.5 | - | |
| TATEC (Garcia-Duran et al., 2016) | - | - | - | 58 | 76.7 | - | |
| NTN (Socher et al., 2013) | - | 66.1 | 0.53 | - | 41.4 | 0.25 | |
| HolE (Nickel et al., 2016) | - | 94.9 | 0.938 | - | 73.9 | 0.524 | |
| STransE (Nguyen et al., 2016) | 206 | 93.4 | 0.657 | 69 | 79.7 | 0.543 | |
| ComplEx (Trouillon et al., 2017) | - | 94.7 | 0.941 | - | 84.0 | 0.692 | |
| ProjE wlistwise (Shi and Weniger, 2017) | - | - | - | 34 | 88.4 | - | |
| IRN (Shen et al., 2016) | 249 | 95.3 | - | 38 | 92.7 | - | |
| RTransE (García-Durán et al., 2015) | - | - | - | 50 | 76.2 | - | Path |
| PTransE (Lin et al., 2015a) | - | - | - | 58 | 84.6 | - | |
| GAKE (Feng et al., 2015) | - | - | - | 119 | 64.8 | - | |
| Gaifman (Niepert, 2016) | 352 | 93.9 | - | 75 | 84.2 | - | |
| Hiri (Liu et al., 2016) | - | 90.8 | 0.691 | - | 70.3 | 0.603 | |
| R-GCN+ (Schlichtkrull et al., 2017) | - | 96.4 | 0.819 | - | 84.2 | 0.696 | |
| NLFeat (Toutanova and Chen, 2015) | - | 94.3 | 0.940 | - | 87.0 | 0.822 | Text |
| TEKE_H (Wang and Li, 2016) | 114 | 92.9 | - | 108 | 73.0 | - | |
| SSP (Xiao et al., 2017) | 156 | 93.2 | - | 82 | 79.0 | - | |
| DistMult (orig) (Yang et al., 2015) | - | 94.2 | 0.83 | - | 57.7 | 0.35 | None |
| DistMult (Toutanova and Chen, 2015) | - | - | - | - | 79.7 | 0.555 | |
| DistMult (Trouillon et al., 2017) | - | 93.6 | 0.822 | - | 82.4 | 0.654 | |
| Single DistMult (this work) | 655 | 94.6 | 0.797 | 42.2 | 89.3 | 0.798 | |
| Ensemble DistMult (this work) | 457 | 95.0 | 0.790 | 35.9 | 90.4 | 0.837 | |

Methods generally **sensitive to hyperparameters** such as loss, number of negative examples, embedding dim, etc.

Well-tuned simple methods outperform more complex models

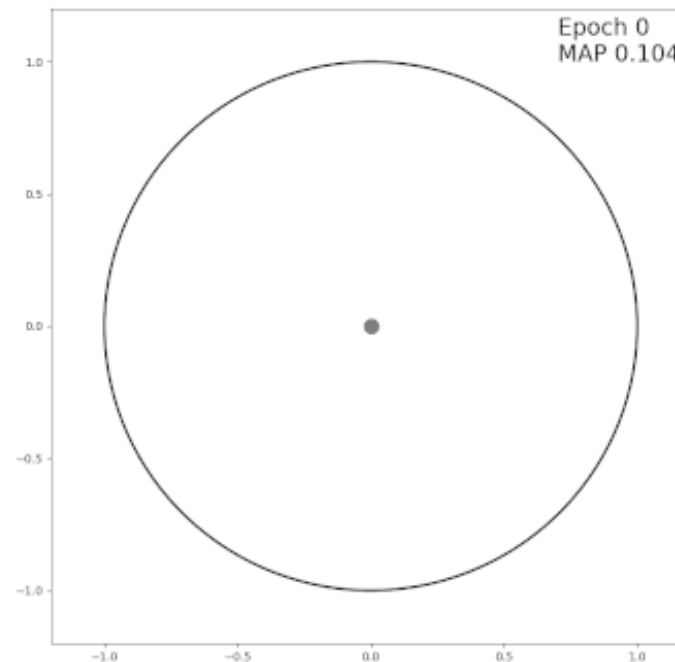
Kadlec et al., Knowledge Base Completion: Baselines Strike Back, 2017

Survey Paper for Learning with KGs

- Nickel et al., A Review of Relational Machine Learning for Knowledge Graphs. Proc. IEEE, 2015.

Recent Developments

- Hyperbolic embeddings (Nickel et al. 2017)
- Useful for hierarchical knowledge graphs



<https://hazyresearch.github.io/hyperE/>

Knowledge Graph Embeddings

What do they actually learn?

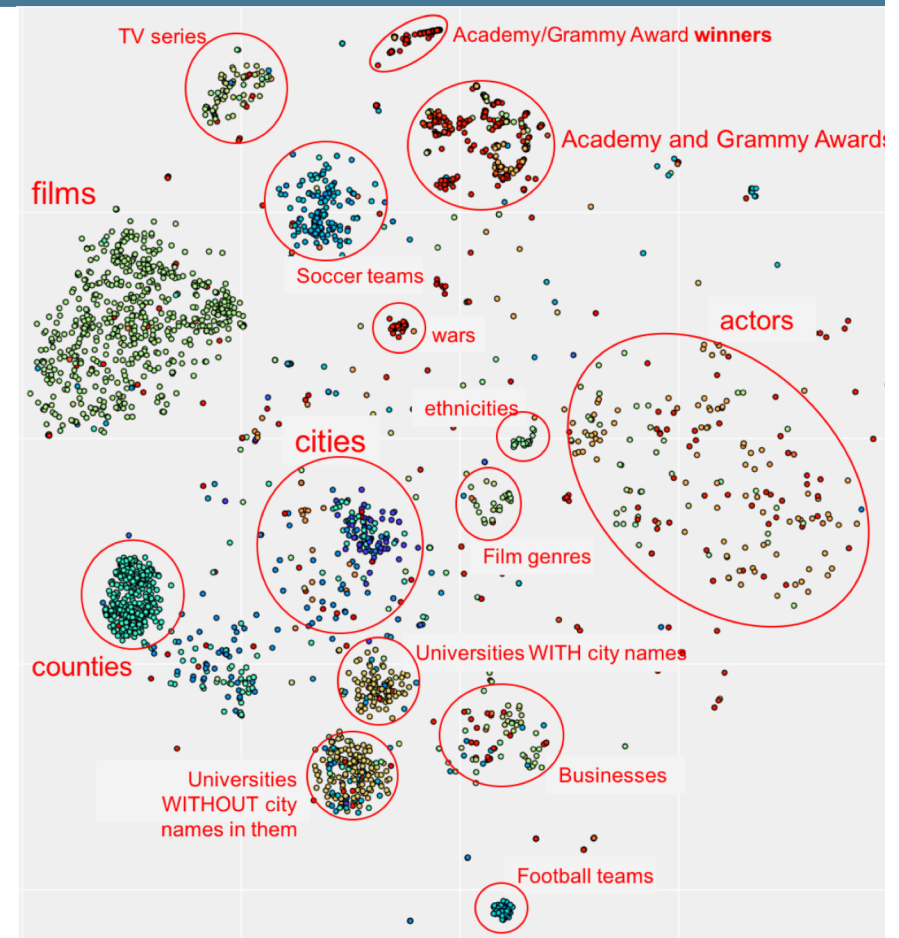
- Fine grained **latent types** of entities
- Latent representation of relation types

What do they not learn?

- Relational model with **constants**
- E.g., relation true if married to PersonX

Majority of KB embedding approaches are outperformed by simple relational baselines

- First observed by Toutanova et al, 2015
- Holds true for dense KBs (e.g. FB15k) but not for sparser ones (e.g., FB15k-237)
- Embedding methods outperform purely relational models on sparse KBs



© Corby Rosset

Alternative Matrix Representations

Universal Schema (Riedel et al., 2013)

Text documents: relations from dependency parses

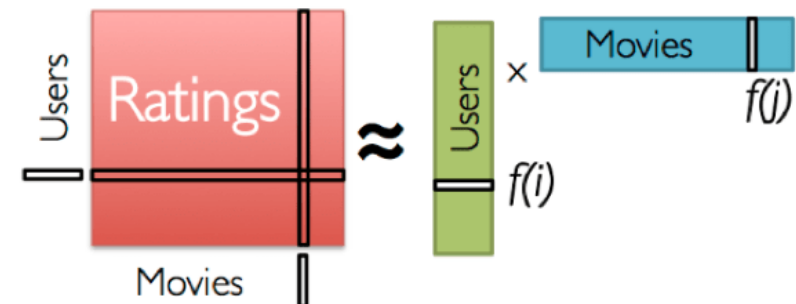
Pairs of entities

Relation types

| | President of | Prime of | Chancellor of | Chief Executive | Leader of | Header of state | HeadOf | TopMember |
|--------------------|--------------|----------|---------------|-----------------|-----------|-----------------|--------|-----------|
| Obama, U.S. | Y | | | Y | | | Y | |
| Merkel, Germany | | | Y | | Y | Y | Y | |
| S Harper, Canada | | Y | | | Y | | Y | |
| V Putin, Russia | Y | | | | Y | Y | Y | |
| Larry Page, Google | | | | Y | | | Y | Y |
| V. Rometty, IBM | Y | | | Y | Y | | | Y |
| Tim Cook, Apple | | | | Y | | | Y | Y |
| E Grimson, MIT | | | Y | | | | Y | |

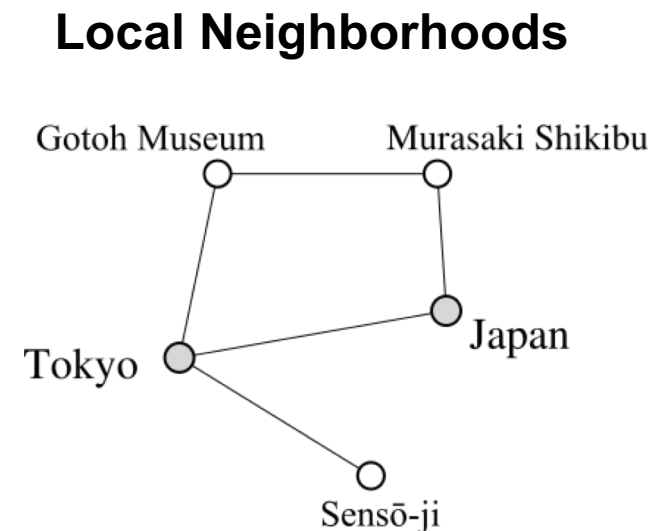
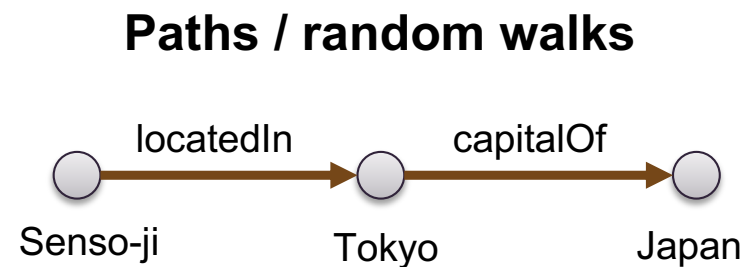
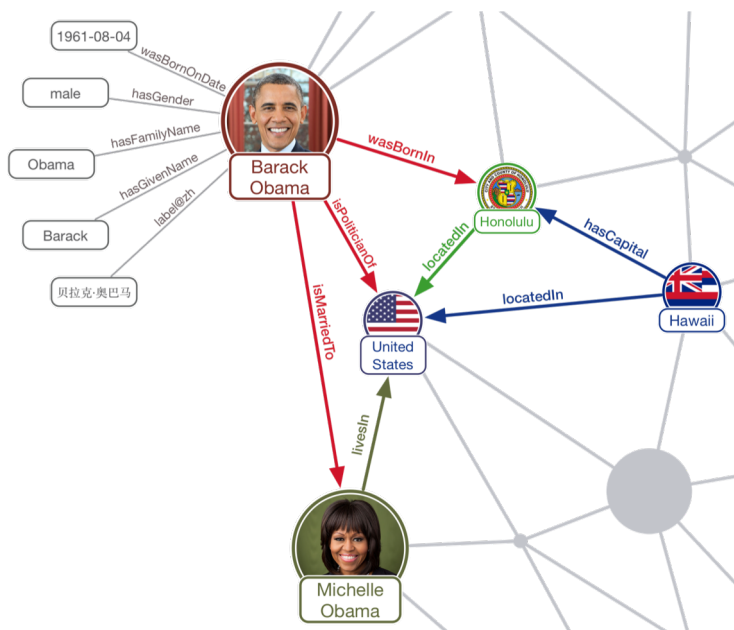
© Riedel et al.

- Also used in conjunction with rule mining approaches (Voelker and Niepert, 2011)
- More later ...



Two Perspectives on Learning from Graph Data

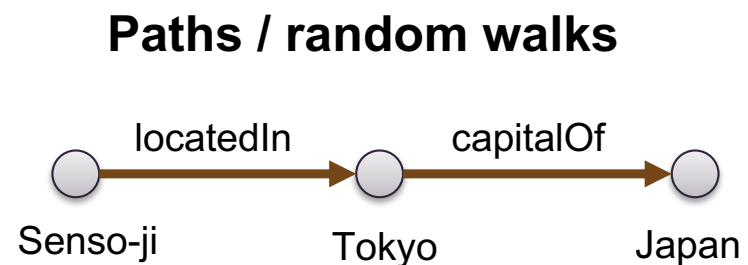
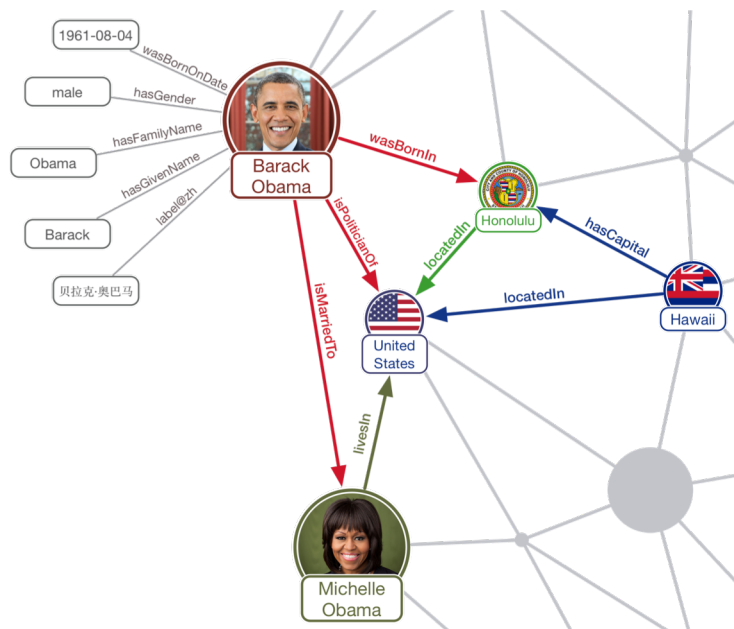
2. Learning from Local Graph Structures



NB: Learning from local structures can capture global properties through a recursive propagation process between nodes

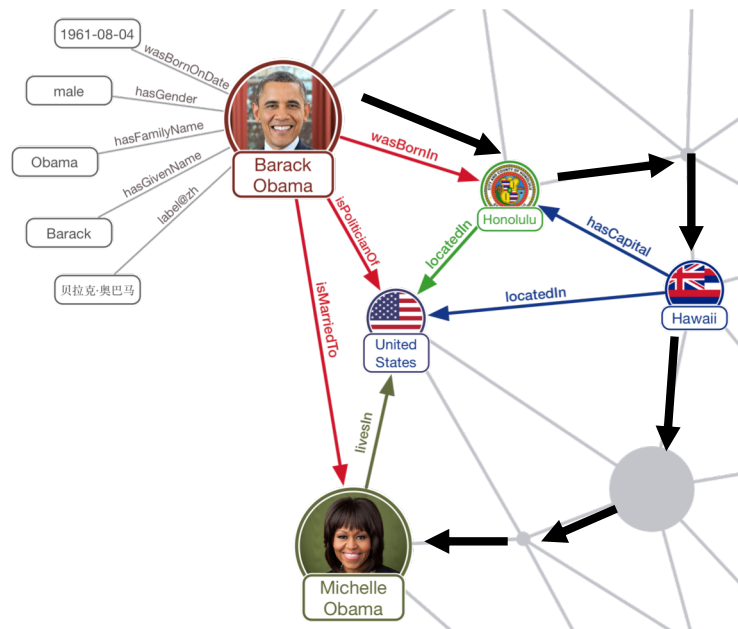
Learning From Random Walks and Paths

- Basic idea: **Mine frequent paths** in the graph and use these paths as features for some learning method

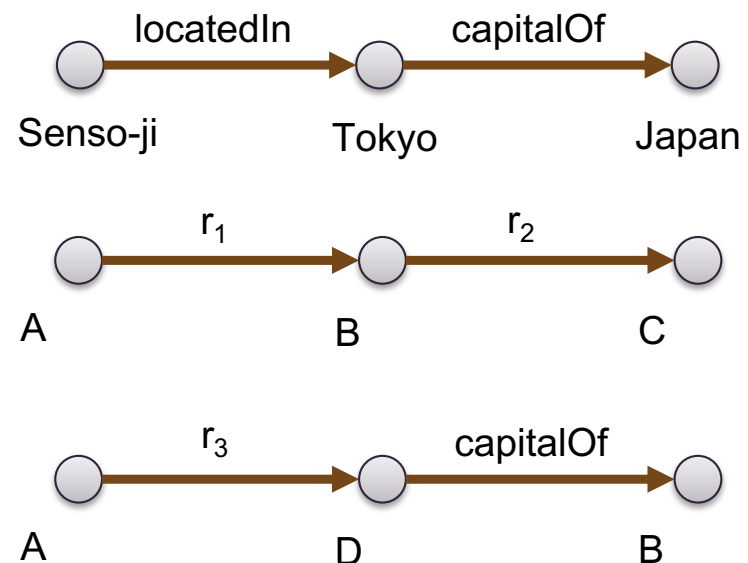


Methods for Path Extraction

Perform a large number of **Random Walks**



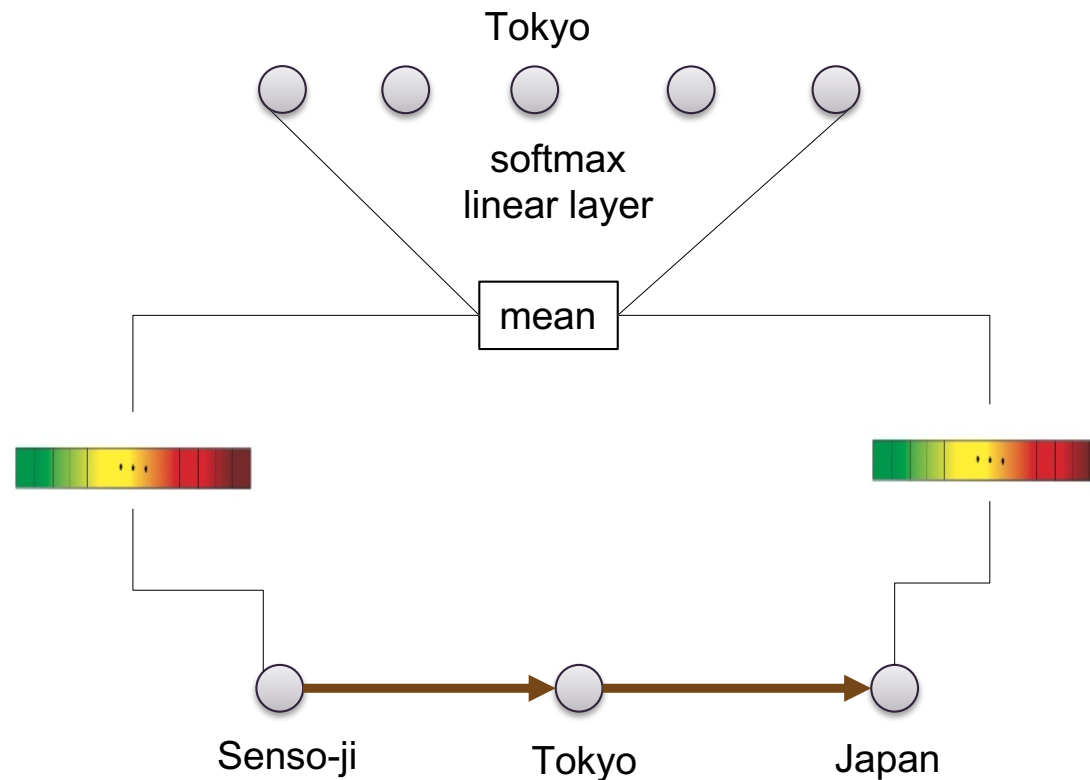
Paths / random walks



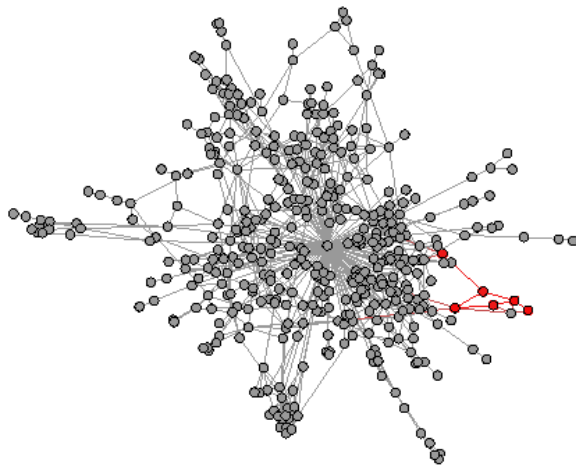
Keep the paths most frequently encountered

Methods for Learning from Single-Relational Paths

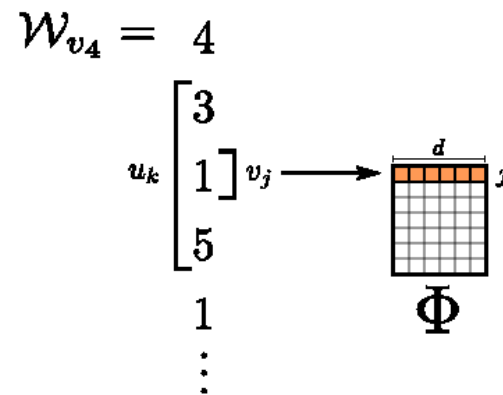
- Interpret every walk as a sentence (sequence of nodes visited)
- Train word embedding method such as Word2vec



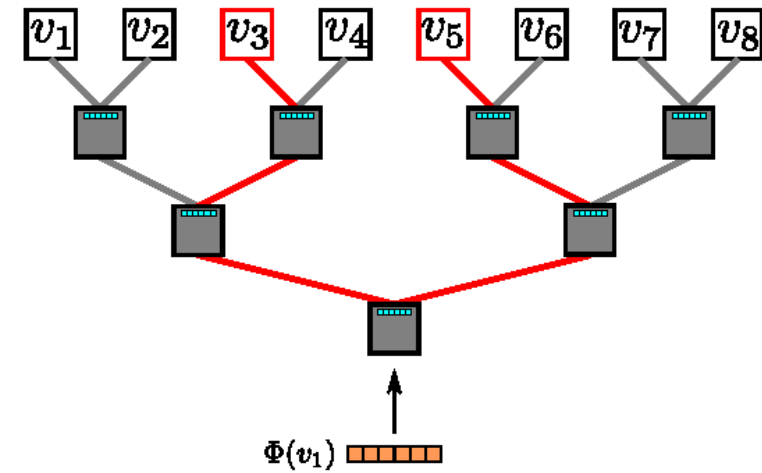
Continuous bag of words



(a) Random walk generation.



(b) Representation mapping.



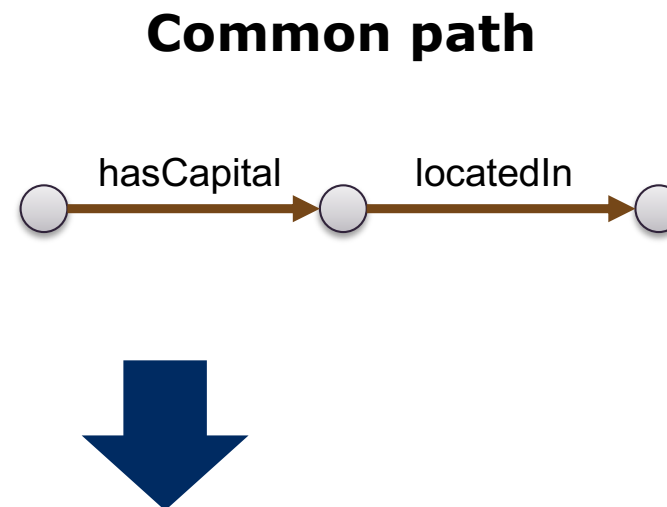
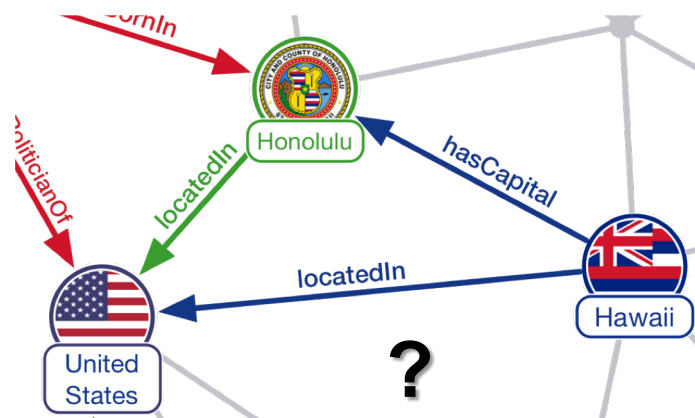
(c) Hierarchical Softmax.

↑
Skip-gram
model

Results in node embeddings to be used for other tasks

Methods for Learning from Multi-Relational Paths

- Interpret every walk as a logical rule:
“If path is present, then set feature to 1”
- Combine these features with simple classifier such as logistic regression



Good feature to predict “locatedIn”

Lao and Cohen, Path Ranking Algorithm, 2010

Other Methods for Mining Path-Like Features (I)

- Create table with one row per entity pair and one column per relational type (coined “universal schema” in IE context)
- Perform association rule mining (Voelker and Niepert, 2011)

Text documents: relations from dependency parses

Pairs of entities

Relation types

| | President of | Prime of | Chancellor of | Chief Executive | Leader of | Header of state | HeadOf | TopMember |
|--------------------|--------------|----------|---------------|-----------------|-----------|-----------------|--------|-----------|
| Obama, U.S. | Y | | | Y | | | Y | |
| Merkel, Germany | | | Y | | Y | Y | Y | |
| S Harper, Canada | | Y | | | Y | | Y | |
| V Putin, Russia | Y | | | | Y | Y | Y | |
| Larry Page, Google | | | | Y | | | Y | Y |
| V. Rometty, IBM | Y | | | Y | Y | | | Y |
| Tim Cook, Apple | | | | Y | | | Y | Y |
| E Grimson, MIT | | | Y | | | | Y | |

© Riedel et al.

- PresidentOf(A, B) \rightarrow HeadOf(A, B) etc.
- Only Horn clauses with same two variables per relation

Other Methods for Mining Path-Like Features (II)

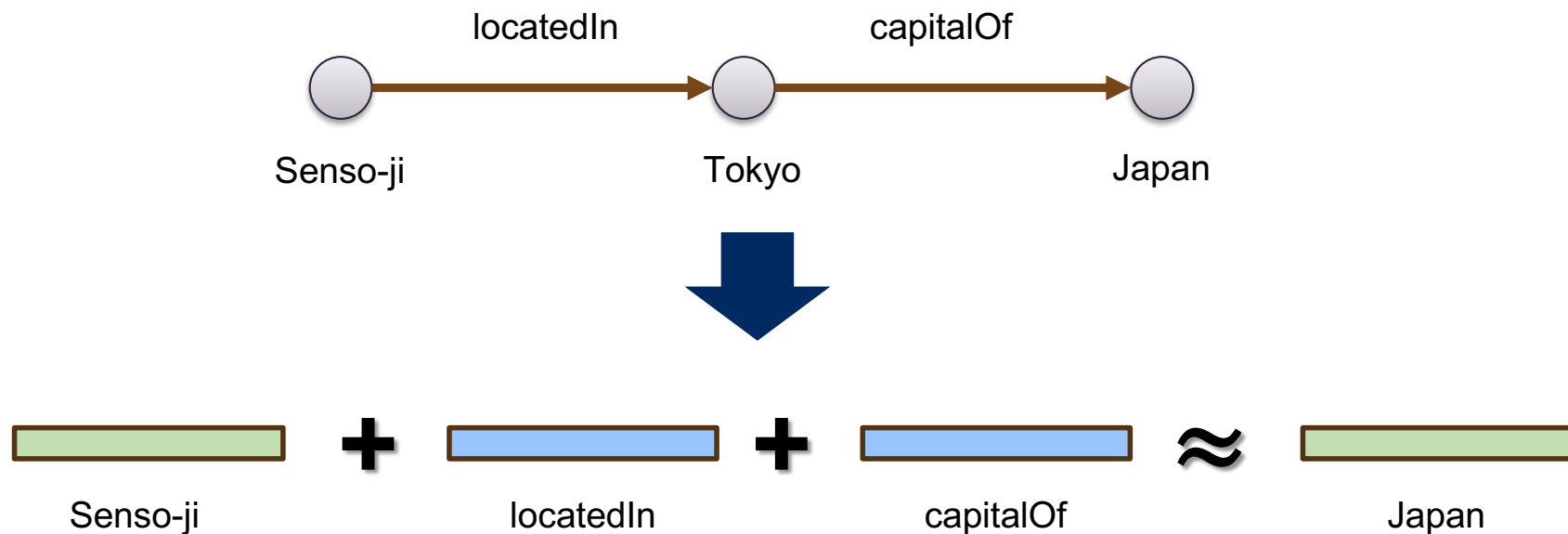
- **AMIE** (Galárraga et al. 2013) generalizes prior work by mining **closed horn rules** such as $R(A, B) \wedge R(B, C) \rightarrow R(A, C)$
- Closed rule: all variables appear at least in two relations
- Highly optimized for large knowledge graphs
- KBLRN (more later) uses this as the core rule miner

| Dataset | # of facts | Settings | Latest runtime |
|----------------------------|------------|---------------------|----------------|
| <u>YAGO2</u> | 948048 | Default | 28.19s |
| <u>YAGO2</u> | 948048 | Support 2 facts | 3.76 min |
| <u>YAGO2 sample</u> | 46654 | Support 2 facts | 2.90s |
| <u>YAGO2</u> | 948048 | Default + constants | 9.93 min |
| <u>YAGO2s</u> | 4122426 | Default | 59.38 min |
| <u>DBpedia 2.0</u> | 6704524 | Default | 46.88 min |
| <u>DBpedia 3.8</u> | 11024066 | Default | 7h 6 min |
| <u>Wikidata (Dec 2014)</u> | 11296834 | Default | 25.50 min |

<https://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/amie/>

Other Methods to Learn from Known Rules (I)

- Extend existing KG embedding methods to learn from longer paths (Guu et al. 2015)

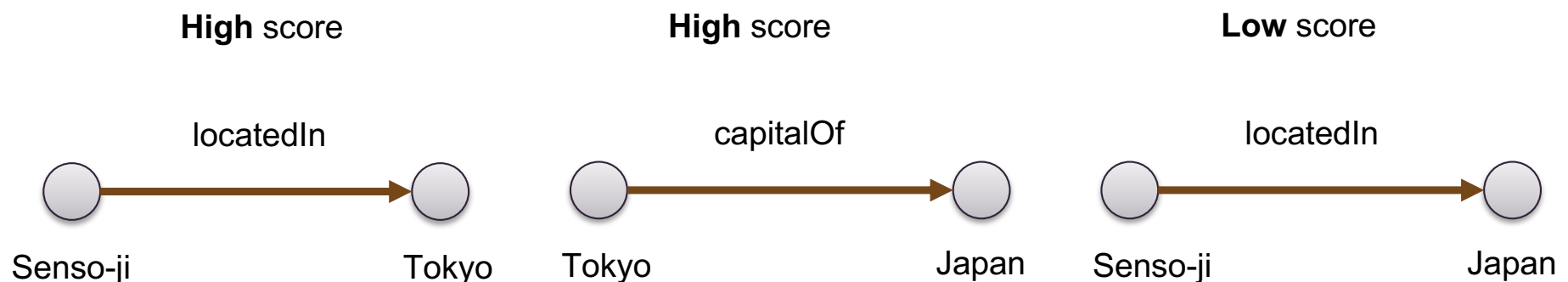


Other Methods to Learn from Known Rules (II)

- Use known rules to generate adversarial examples (Minervini et al. 2017)
- If existing KB completion model maintain fact representations that contradict a known rule, backpropagate to make contradiction less likely

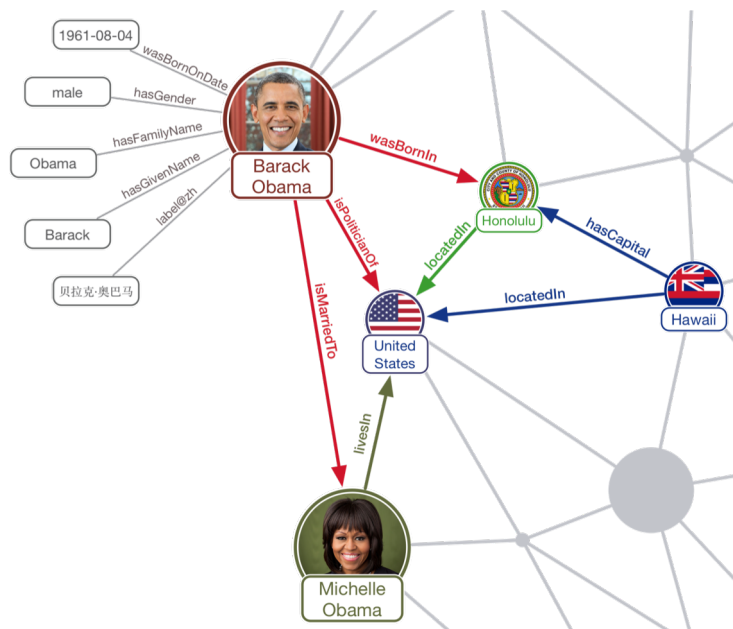
Example

- Rule:** $(A, \text{locatedIn } B) \text{ and } (B, \text{capitalOf}, C) \rightarrow (A, \text{locatedIn}, C)$
- Adversarial example:

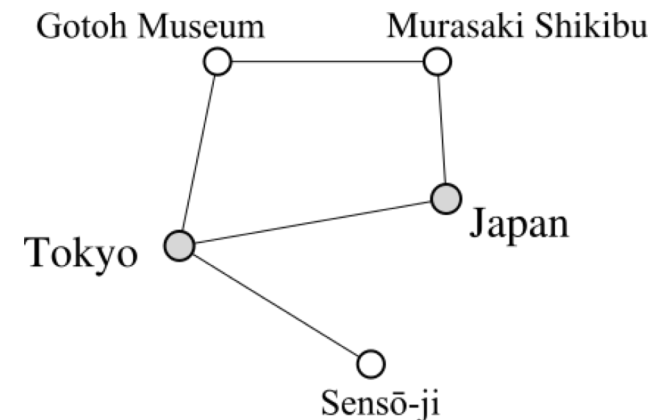


Two Perspectives on Learning from Graph Data

2. Learning from Local Graph Structures



Local Neighborhoods



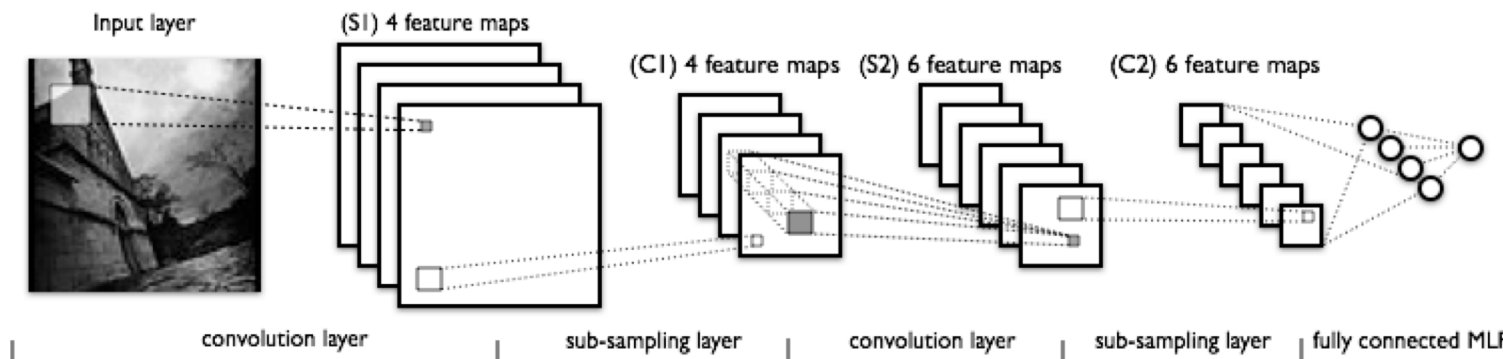
NB: Learning from local structures can capture global properties through a recursive propagation process between nodes

Representation Learning for Knowledge Graphs

Observation: Effective representations are often composed bottom-up from **local** representations

- Weight sharing
- Hierarchical features
- Model tractability

Example: Convolutional neural networks

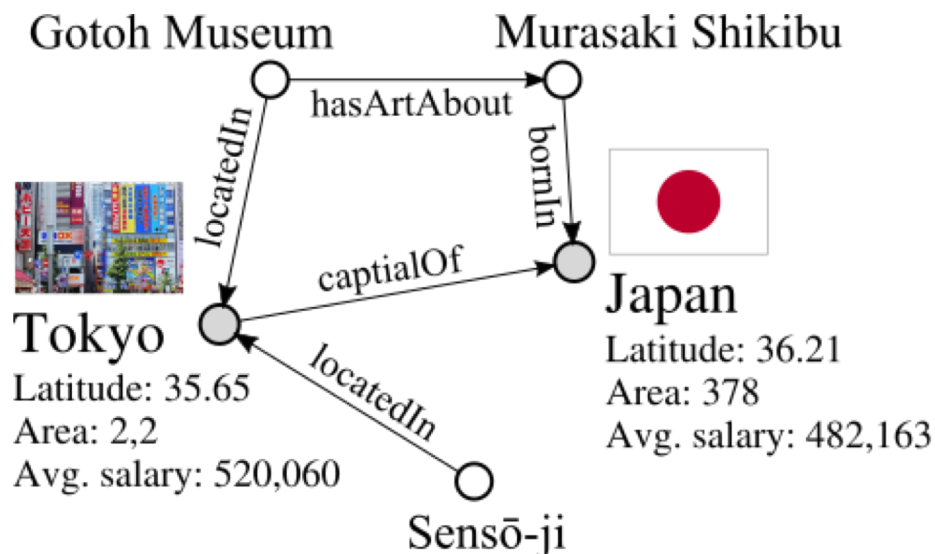


© Yann LeCun

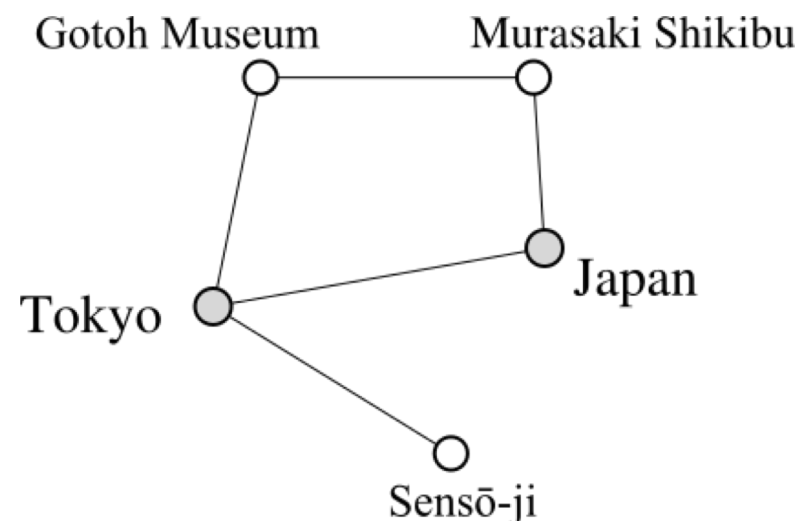
Question: What is a suitable notion of **locality** in knowledge graphs?

Gaifman Locality

Knowledge graph



Gaifman graph



Local sentences are sentences whose quantifiers range over r -neighborhoods of the Gaifman graph

Gaifman's Locality Theorem:

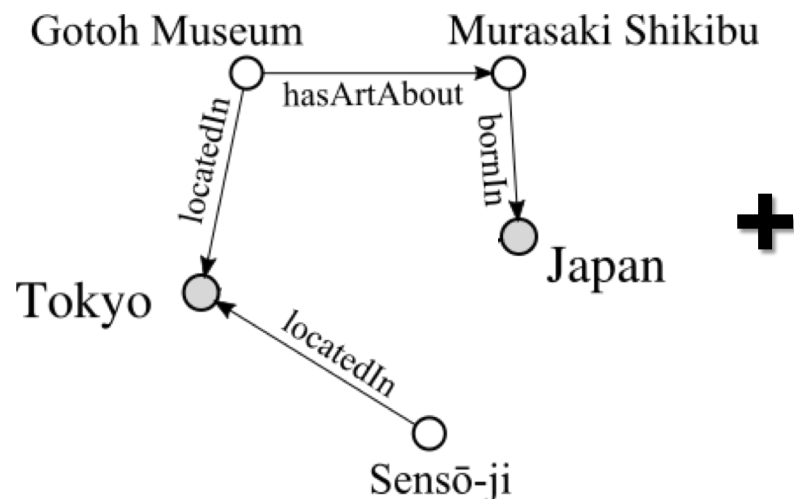
"Every first-order sentence can be written as a Boolean combination of **local sentences**."

The goal is to learn representations of r -neighborhoods for which the query evaluates to **true** and **false**

Query: (H?, capitalOf, T?)

Basic idea: Sample local neighborhoods where **query is true** and where **query is false** and use as **training data**

Sampled local neighborhood



Relational Features

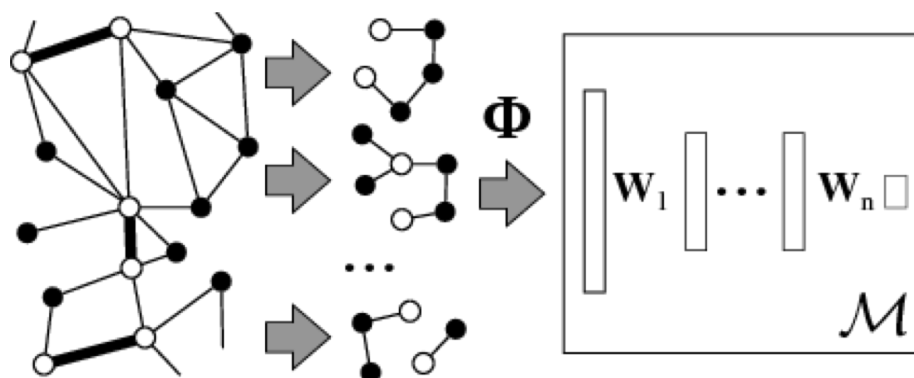
Φ

| | | | |
|-----|--|---|---|
| 1. | capitalOf(H, T) | → | 0 |
| 2. | capitalOf(T, H) | → | 0 |
| 3. | locatedIn(H, x) | → | 0 |
| 4. | locatedIn(x, H) | → | 0 |
| 5. | bornIn(T, x) | → | 2 |
| 6. | bornIn(x, T) | → | 0 |
| 7. | $\exists x, y : \text{locatedIn}(x, H) \wedge \text{hasArtAbout}(x, y) \wedge \text{bornIn}(y, T)$ | → | 1 |
| ... | | → | 1 |

...

Training Gaifman Models

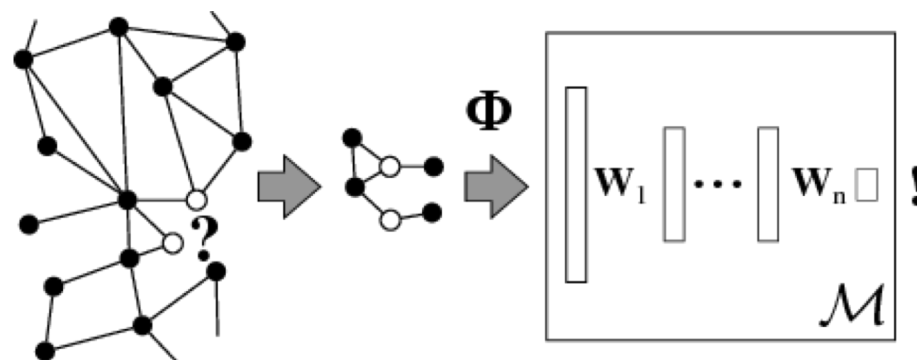
1. Given a target query (Tokyo, capitalOf, y?)
2. Sample a number of bounded-size neighborhoods of pairs (A, B) for which (A, capitalOf, B) holds
3. Sample a number of bounded-size neighborhoods of corrupted pairs (A', B')
4. Evaluate **relational features** to generate vector representation
5. Train a (deep) neural network model



Inference in Gaifman Models

- Inference is performed by generating one (or more) neighborhoods and querying the trained Gaifman model

(Tokyo, capitalOf, x?)



Discriminative Gaifman Models

Possible training objective

Vector representation of neighborhood resulting from relational features

Probability returned by the Gaifman model

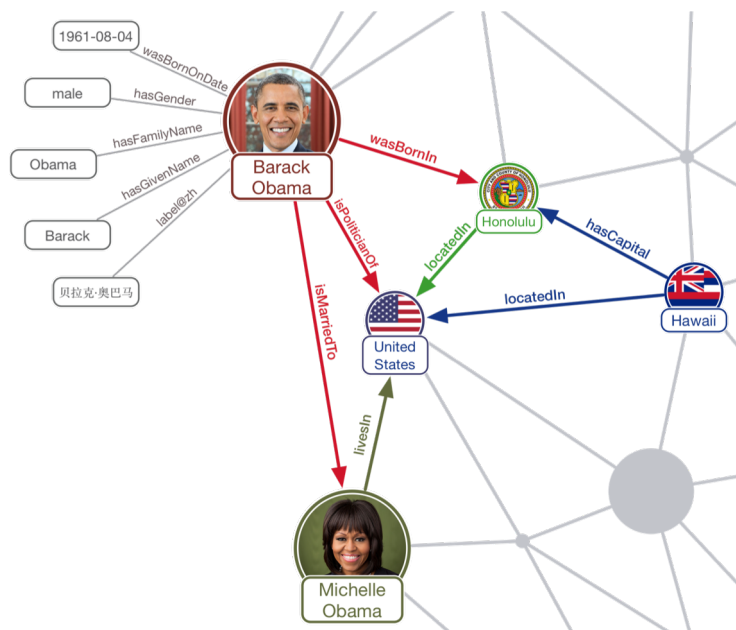
$$\mathcal{L} = - \left[\sum_{\mathbf{N} \in \mathcal{N}} \log p_{\mathcal{M}}(\mathbf{v}_{\mathbf{N}}) + \sum_{\tilde{\mathbf{N}} \in \tilde{\mathcal{N}}} \log(1 - p_{\mathcal{M}}(\mathbf{v}_{\tilde{\mathbf{N}}})) \right]$$

Sampled positive and negative Gaifman neighborhoods

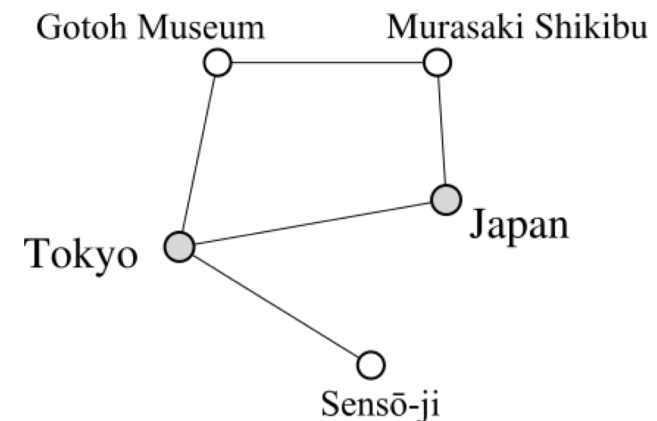
The diagram illustrates the training objective for Discriminative Gaifman Models. It features a central equation for the loss function \mathcal{L} . Above the equation, two green arrows point downwards to the terms $\log p_{\mathcal{M}}(\mathbf{v}_{\mathbf{N}})$ and $\log(1 - p_{\mathcal{M}}(\mathbf{v}_{\tilde{\mathbf{N}}}))$. The first arrow is labeled 'Probability returned by the Gaifman model' and the second is labeled 'Vector representation of neighborhood resulting from relational features'. Below the equation, two green arrows point upwards from the text 'Sampled positive and negative Gaifman neighborhoods' to the summation indices $\mathbf{N} \in \mathcal{N}$ and $\tilde{\mathbf{N}} \in \tilde{\mathcal{N}}$.

Two Perspectives on Learning from Graph Data

2. Learning from Local Graph Structures



Local Neighborhoods

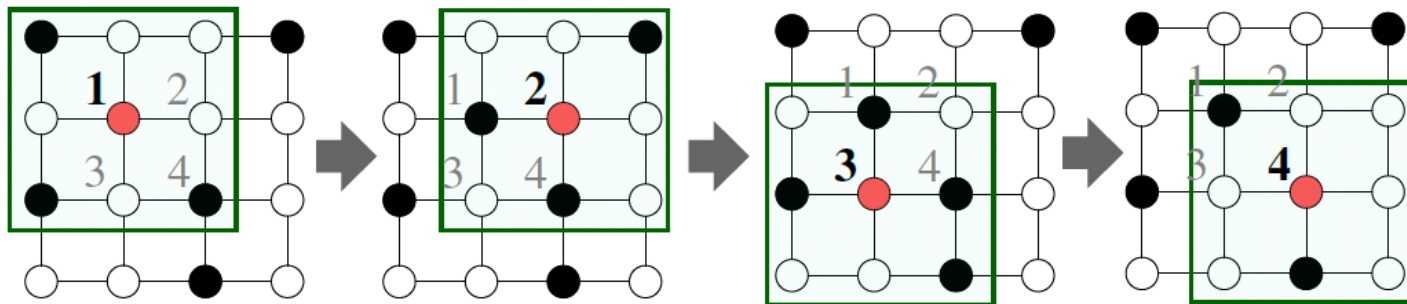


NB: Learning from local structures can capture global properties through a recursive propagation process between nodes

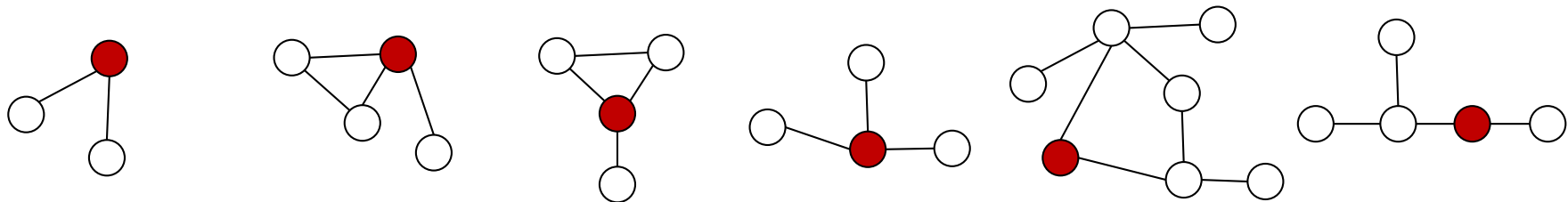
Strengths of CNNs

- Implicit feature hierarchy based on **local features**
- Parameter sharing** across data points

Straightforward for regular graphs

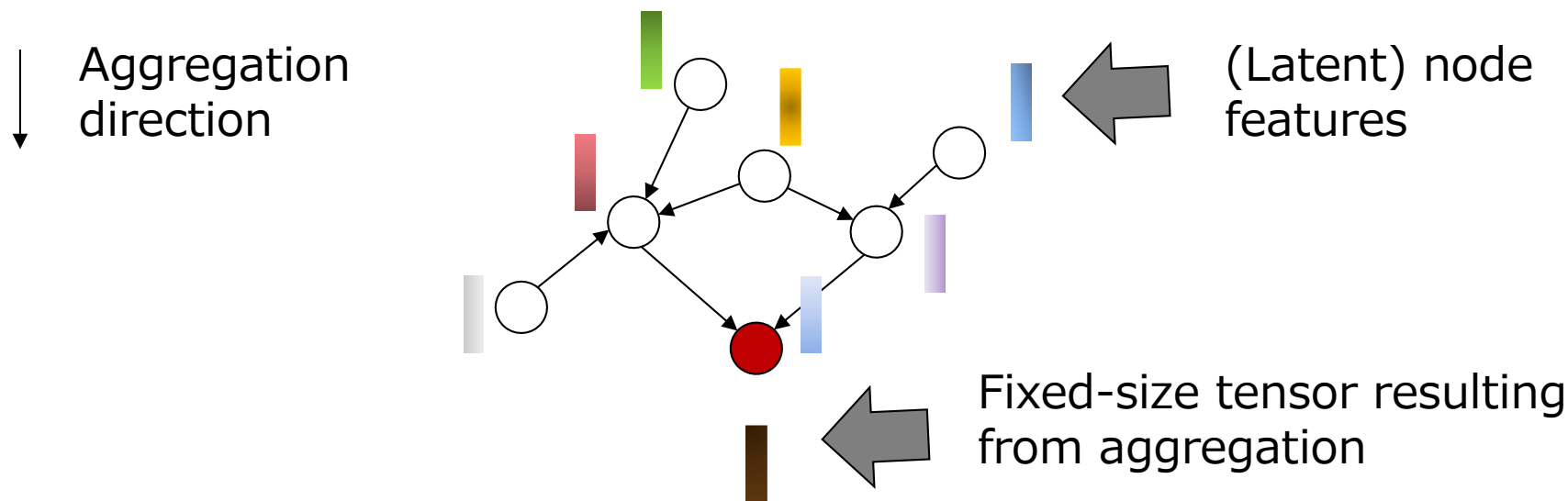


Challenging for irregular graphs



The Big Question of Graph CNNs

- How do we **aggregate neighborhood information** into **fixed-size** representations? → requirement for **weight sharing**



- Feature transformations are applied **locally** for each node on its neighborhood
- Requires ability to work with **highly heterogeneous** neighborhood structures

A Spectrum Of Methods

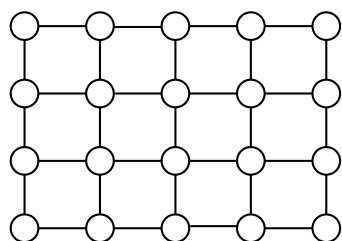
Patchy [ICML 2016]
Neighborhood
Normalization



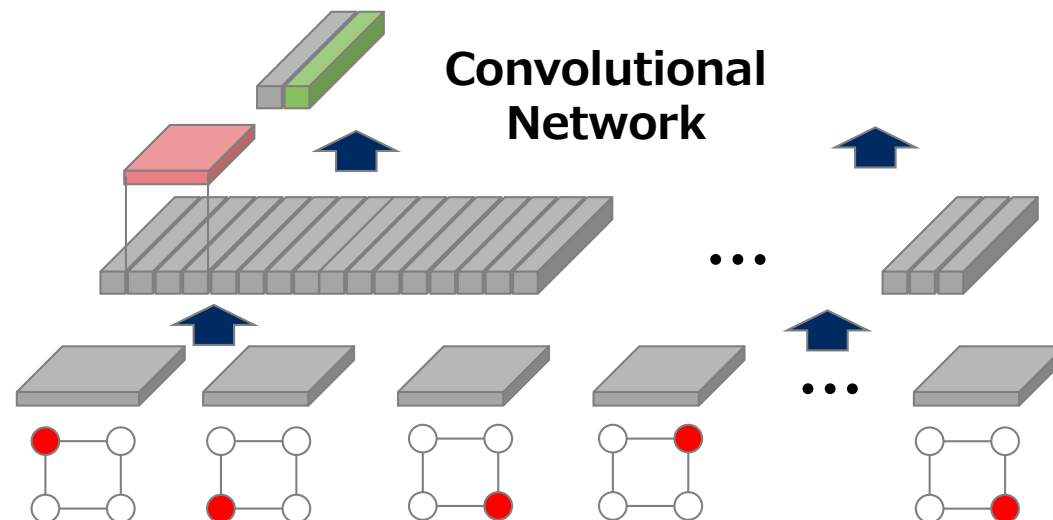
High variance
Low bias

Image CNN

- Grid graph required (spatial order)
- Works only for images

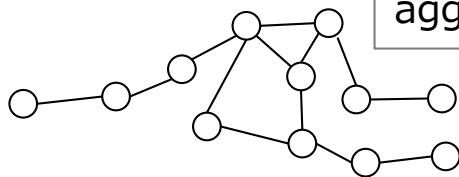


Standard CNN
moves over image

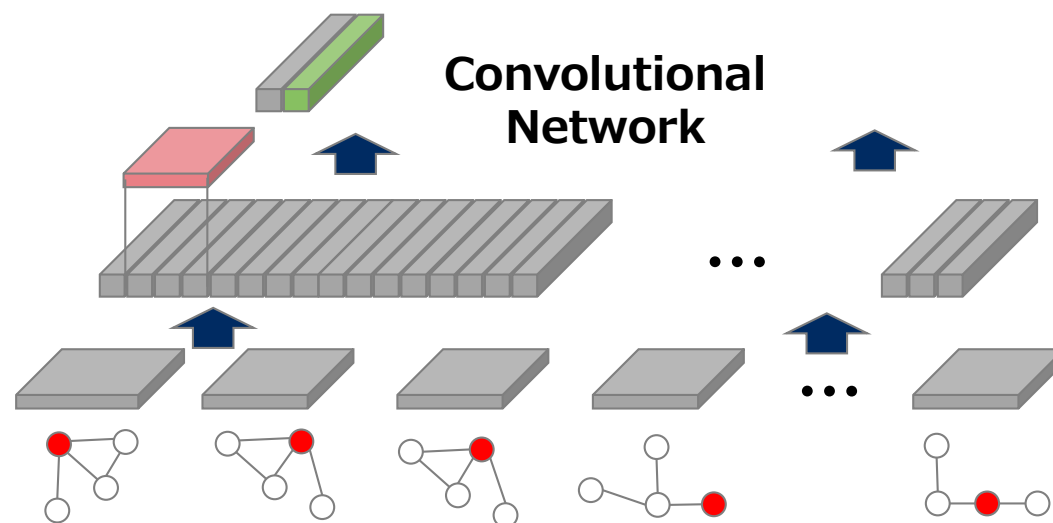


Graph CNN

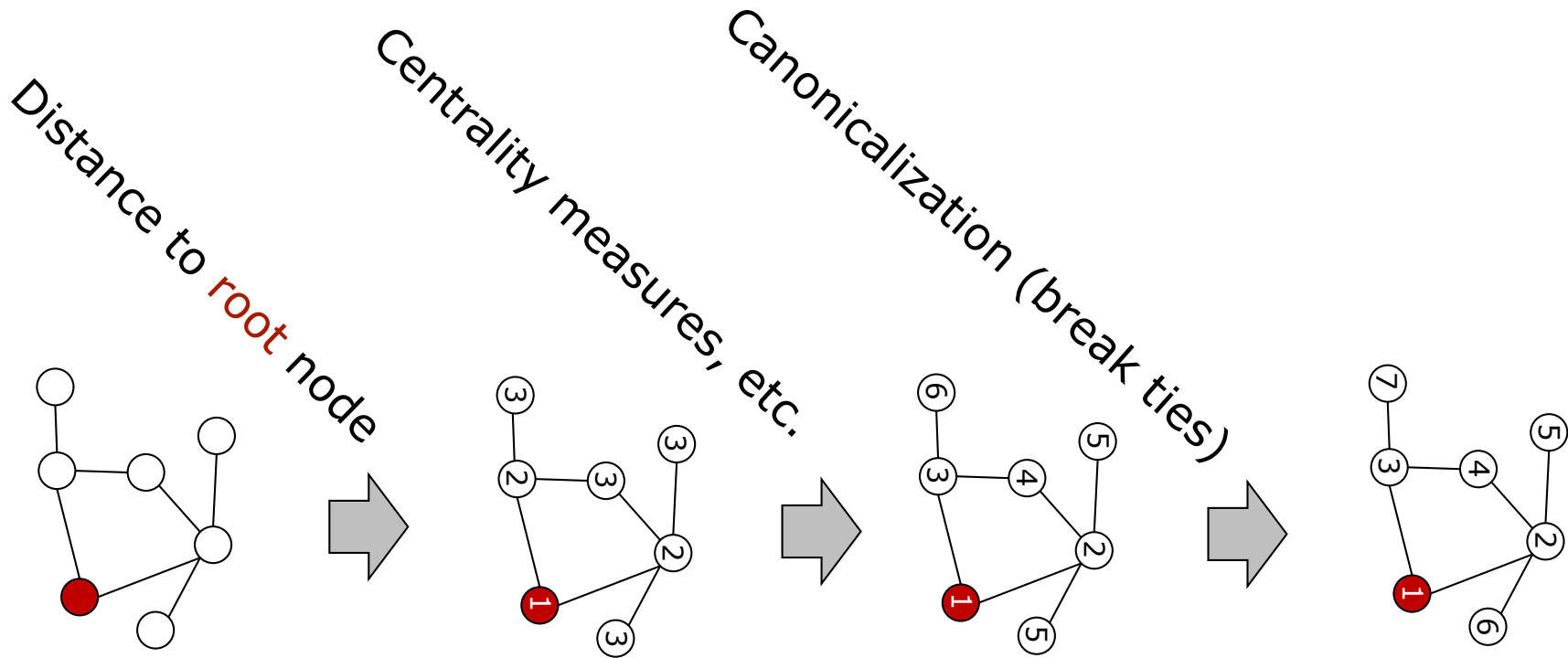
- Arbitrary input graph
- Node attributes
- Edge attributes

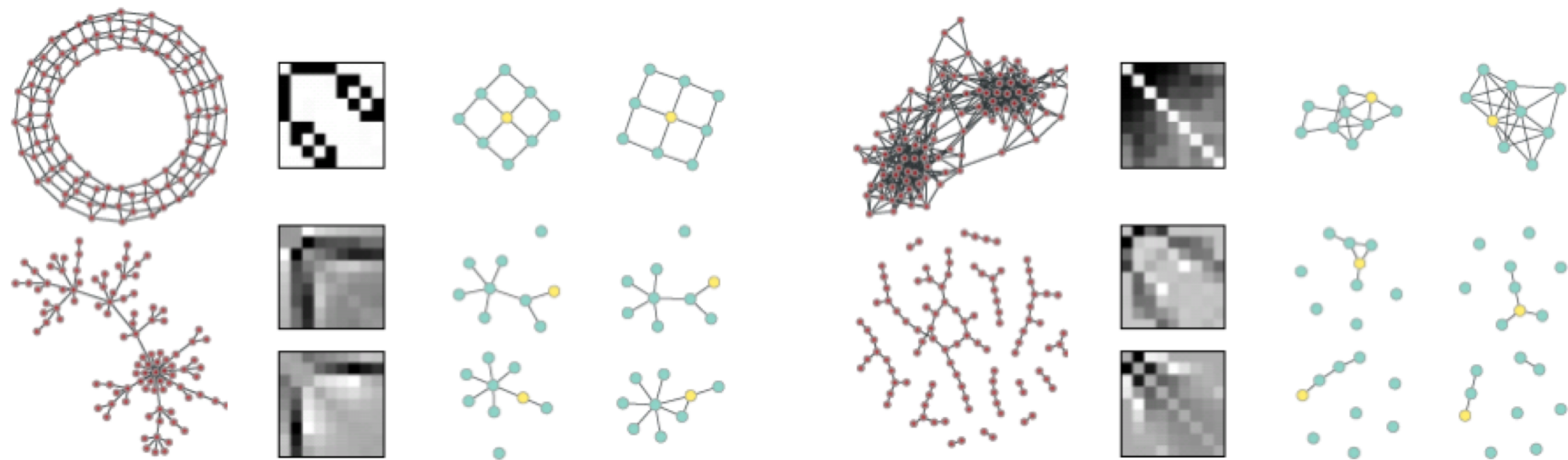


What are good local
aggregations?



Niepert et al, 2016





motifs *learned by the model*

small instances of input graphs

A Spectrum Of Methods

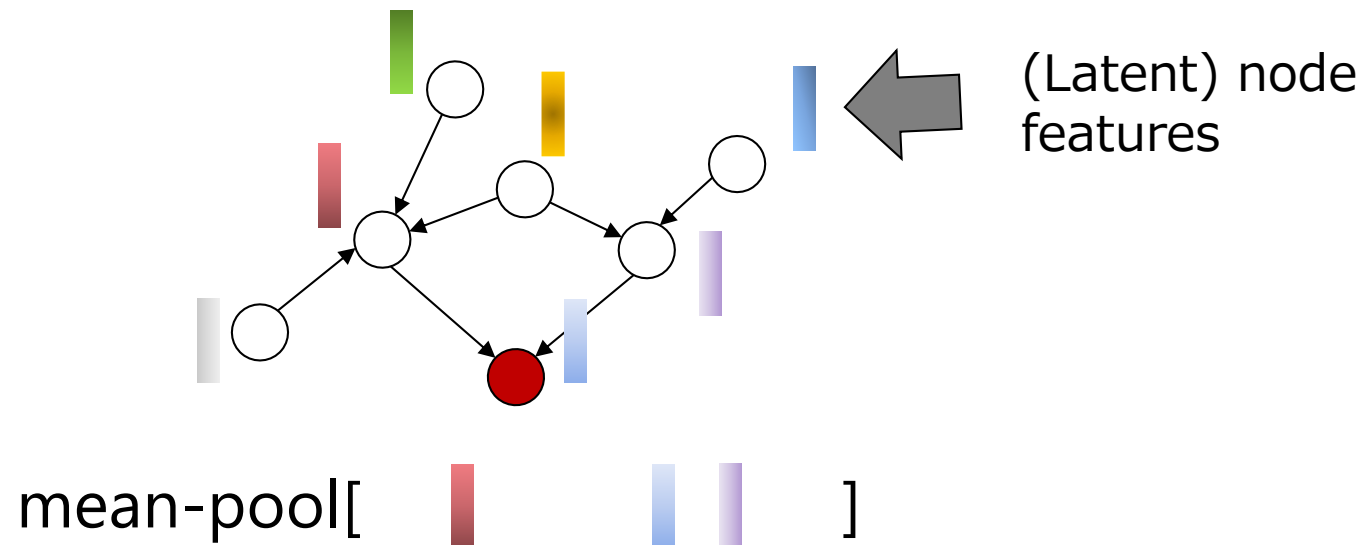
Patchy [ICML 2016]
Neighborhood
Normalization

GCN [ICLR 2017]
Average Pooling

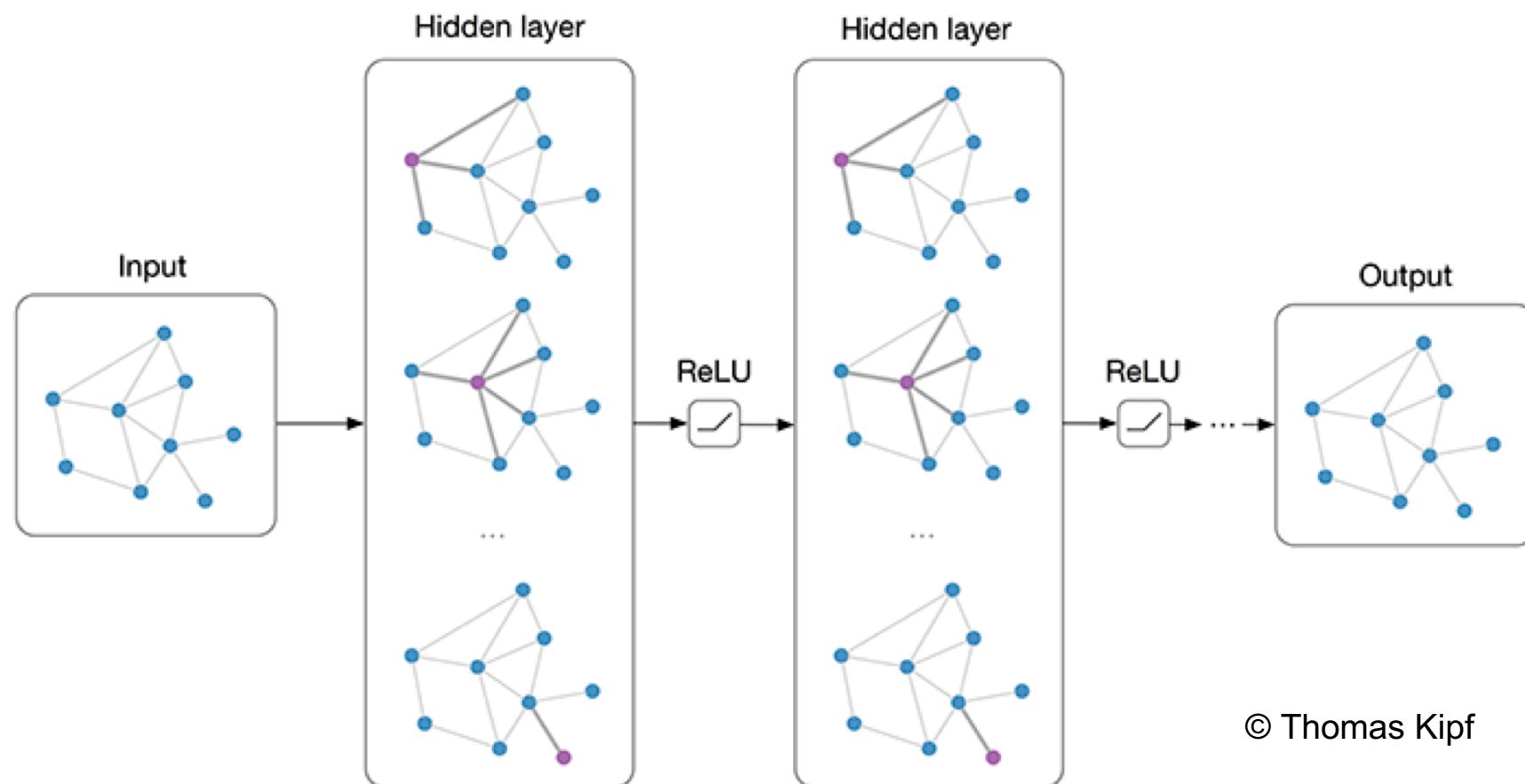


High variance
Low bias

Low variance
High bias



- Compute a **weighted sum** of the node features where weights are determined by **global node adjacency** information
- Essentially **average pooling** of the (latent) node features



© Thomas Kipf

A Spectrum Of Methods

Patchy [ICML 2016]
Neighborhood
Normalization

MoNet [CVPR 2017]
Coll. of weighted sums

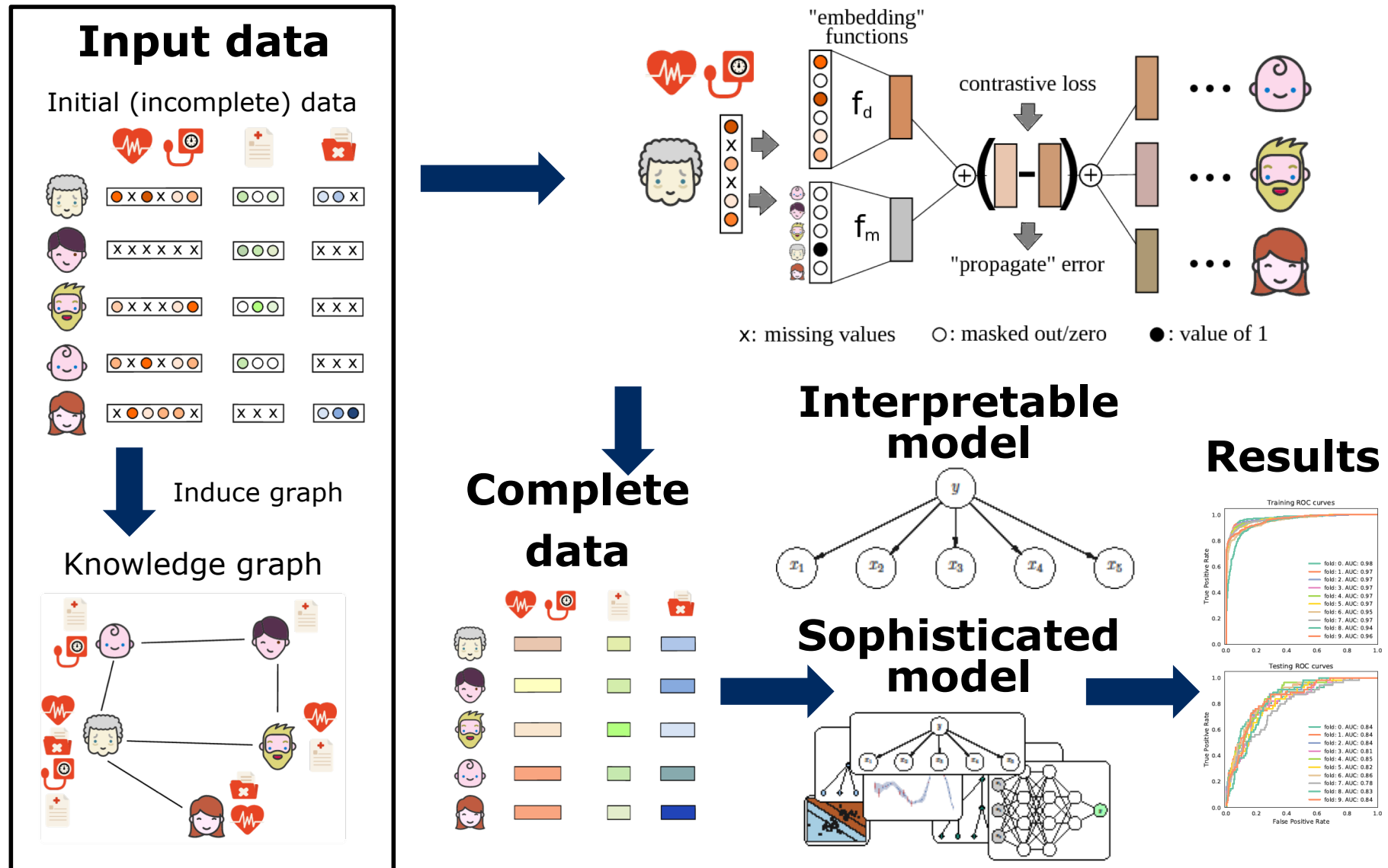
GCN [ICLR 2017]
Average Pooling



Generalization: MoNet (fixed number of weighted sums) \sim Gaussian mixture model but with fixed number of Gaussian kernels

Recent Methods for Learning from Neighborhoods

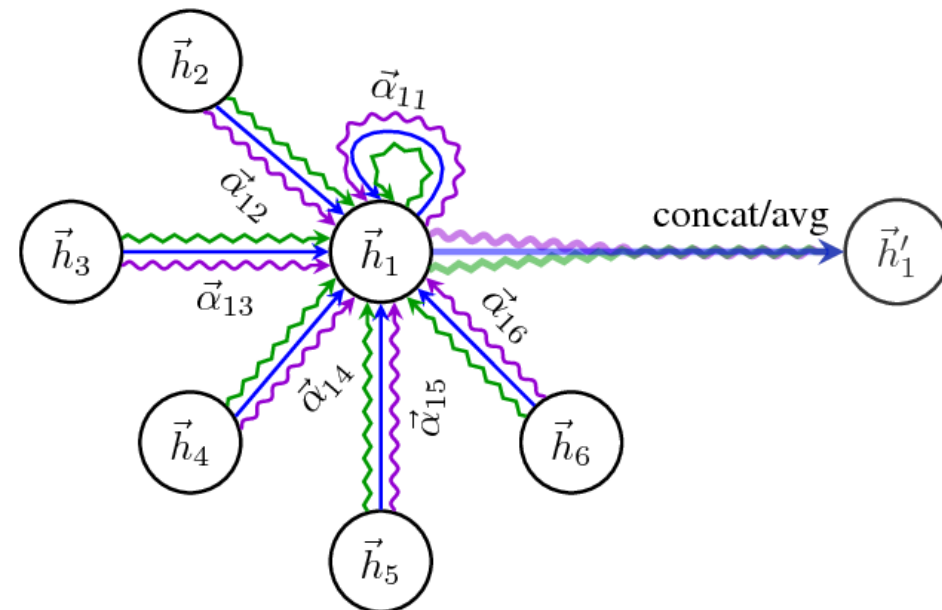
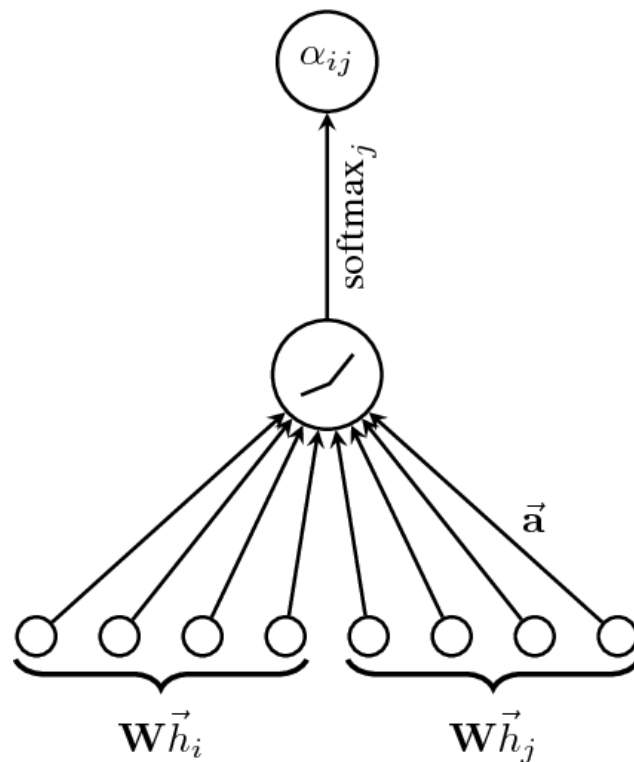
Embedding Propagation (unsupervised, multi-modal, missing data)



Recent Methods for Learning from Neighborhoods (II)

- Graph Attention Networks (extend idea of attention to graphs)
- Special case of MoNet

Petar Veličković, 2018



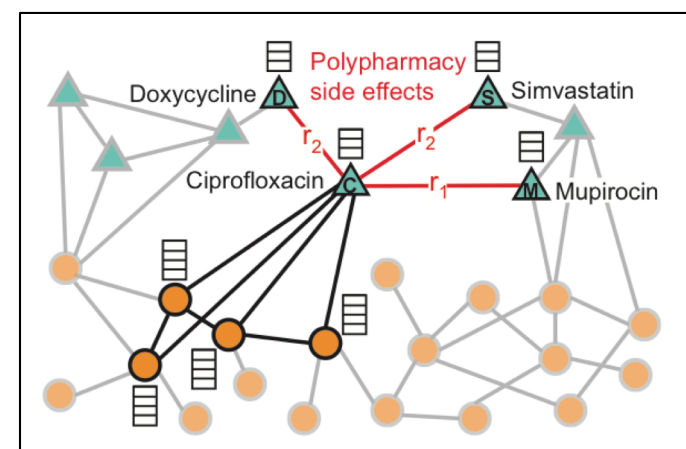
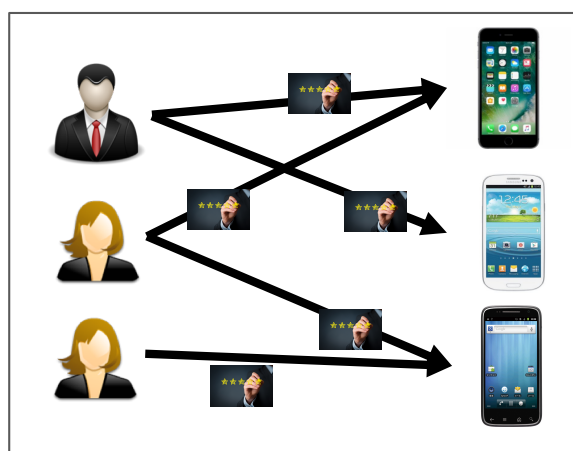
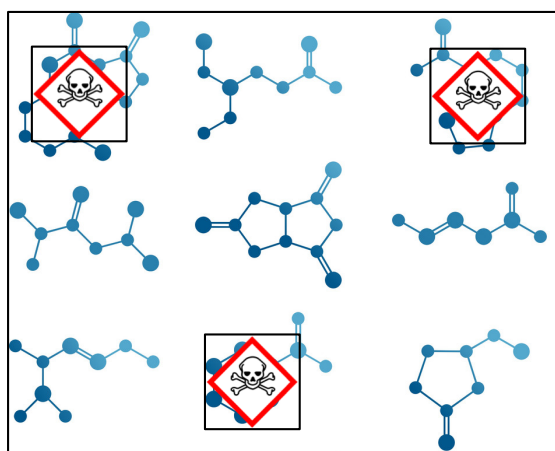
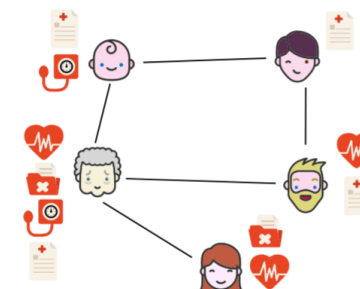
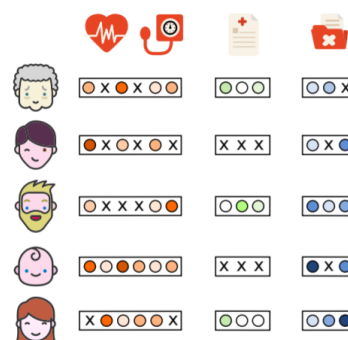
Some Practical Observations

1. Embedding methods learn latent type representations
2. Dichotomy of relational features: either work perfectly or fail completely (random) → known to be brittle
3. Relational features outperform embedding methods on KBs with dense relational structure
4. Embedding methods outperform relational methods in more sparsely connected KGs; don't depend on quality of rules
5. Combinations of the two are more robust and perform better (Motivation for second part of the lecture)

| Query | Correct entity | Rank | |
|--------------------------|----------------|------|-----|
| | | TE | GM |
| nationality(?, US) | W. H. Macy | 2 | 233 |
| born_here(HK, ?) | W. Chau-sang | 5 | 135 |
| contains(?, Curtis-Inst) | USA | 32 | 1 |
| children(?, H. Roshan) | R. Roshan | 26 | 1 |

Some Practical Observations

- KG embedding methods are very versatile
- We have successfully used it for
 - Product Recommendation
 - Polypharmacy Predictions
 - Patient Outcome Prediction
 - Drug Discovery Problems



Some Practical Observations

- **Simple** methods tend to be more robust, that is, generalize better in several application domains
- Including more modalities (text, images, numerical features) improves results (motivation for second part of lecture)
- However, improvement over **simple** methods is modest for the typical knowledge base completion benchmarks
- Important in industrial applications to be able to incorporate “relational features”, that is, known domain-specific rules
- In industrial applications, there is inherent value in methods that allow one to **understand the rules used for prediction** → advantage of methods that do not learn a *purely latent* representation (motivation for KBLRN)
- Most industrial applications involve relational data and/or text data (images and other modalities are more rare)