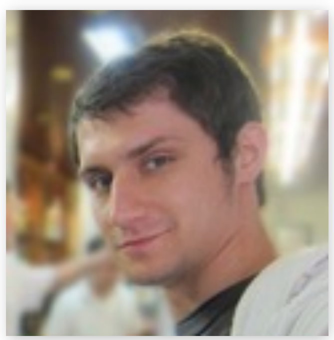
Embedding Probabilistic Logic for Machine Reading

aka Towards Two-Way Interaction with Reading Machines

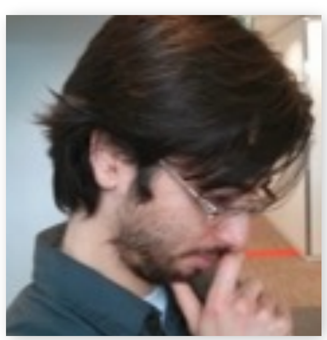
- **Sebastian Riedel (University College London)**

Collaborators

Tim Rocktäschel
UCL



Matko Bosnjak
UCL



Ivan Sanchez
UCL

Sameer Singh
UWash

Limin Yao
UMass Amherst (Twitter)

Andrew McCallum
UMass Amherst

Ben Marlin
UMass Amherst

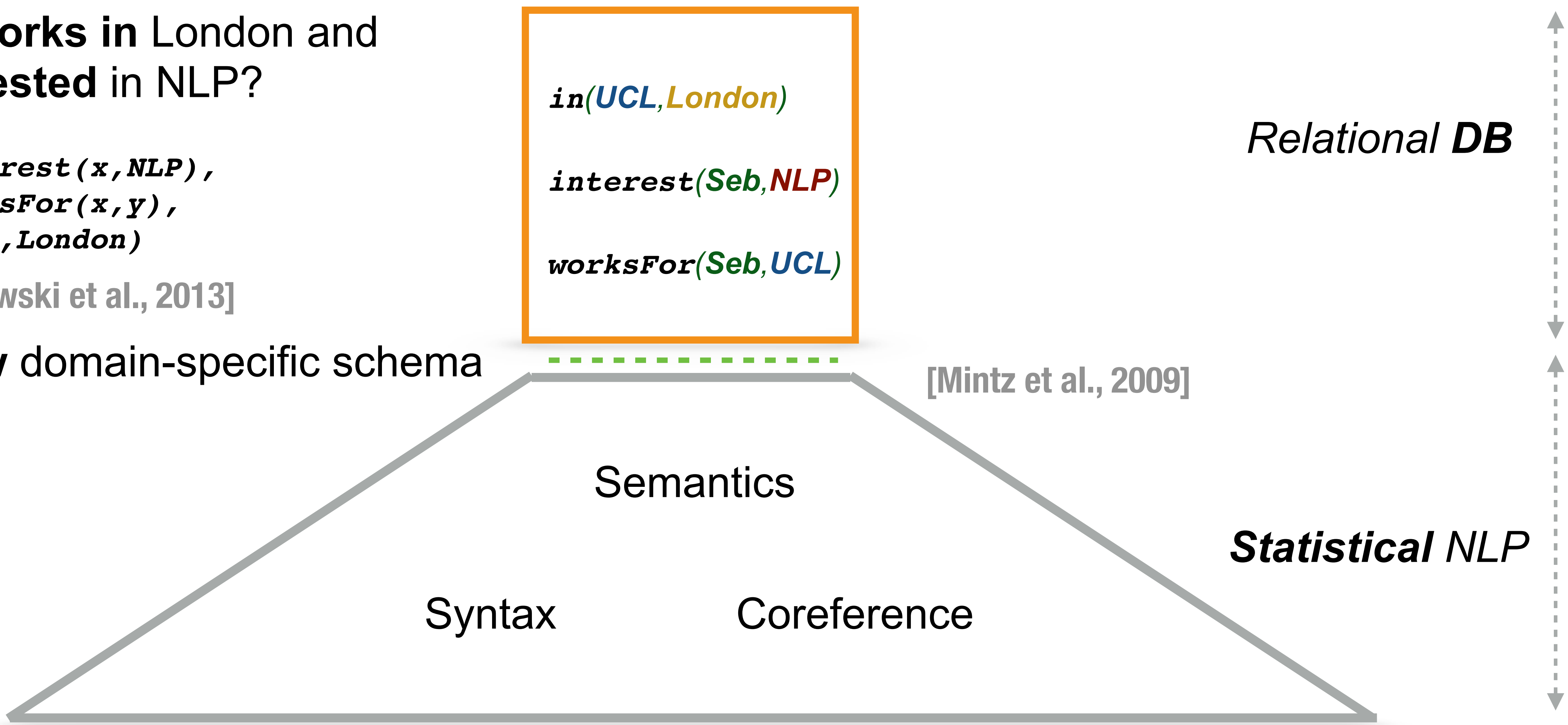
Machine Reading

“Who **works in** London and is **interested** in NLP?

```
interest(x,NLP),
worksFor(x,y),
in(y,London)
```

[Kwiatkowski et al., 2013]

Narrow domain-specific schema



*”**Sebastian Riedel** works in the area of **NLP** and is now Lecturer at **UCL**“*

Machine Reading

[Riedel et al., 2013]

“Who **works in** London and is **interested** in NLP?

```
interest(x,NLP),
worksFor(x,y),
in(y,London)
```

`in`(*UCL*,*London*)

`works-in-area-of`(*Seb*,*NLP*)

`lecturer-at`(*Seb*,*UCL*)

Semantics

Relational ***DB***

Wide *universal schema*

Syntax

Coreference

Statistical *NLP*

*”**Sebastian Riedel** works in the area of **NLP** and is now Lecturer at **UCL**“*

Semantics as Reasoning

[Riedel et al., 2013]

“Who **works in** London and is **interested** in NLP?

```
interest(x,NLP),
worksFor(x,y),
in(y,London)
```

```
in(UCL,London)

works-in-area-of(Seb,NLP)

lecturer-at(Seb,UCL)

worksFor(x,y):
 faculty-at(x,y)

interest(x,y):
 works-in-area-of(x,y)[0.9]

faculty-at(x,y):
 lecturer-at(x,y)
```

Statistical *Relational* ***Learner*** *and* ***Reasoner***

Wide *universal schema*

Syntax

Coreference

Statistical *NLP*

*”**Sebastian Riedel** works in the area of **NLP** and is now Lecturer at **UCL**“*

Benefit: Transitive Reasoning

"Who **works in** London and is **interested** in NLP?

```
interest(x,NLP),
worksFor(x,y),
in(y,London)
```

```
in(UCL,London)
works-in-area-of(Seb,NLP)
lecturer-at(Seb,UCL)
worksFor(x,y):
  faculty-at(x,y)
interest(x,y):
  works-in-area-of(x,y)[0.9]
faculty-at(x,y):
  lecturer-at(x,y)
```

***Statistical** Relational*
Learner** and **Reasoner

Wide *universal schema*

Syntax

Coreference

***Statistical** NLP*

*"**Sebastian Riedel** works in the area of **NLP** and is now Lecturer at **UCL**"*

Benefit: More Coverage

"Who is **faculty** in London and interested in NLP?

```
interest(x,NLP),
worksFor(x,y),
in(y,London)
```

```
in(UCL,London)
works-in-area-of(Seb,NLP)
lecturer-at(Seb,UCL)
worksFor(x,y):
 faculty-at(x,y)
interest(x,y):
 works-in-area-of(x,y)[0.9]
faculty-at(x,y):
 lecturer-at(x,y)
```

Statistical** Relational **Learner** and **Reasoner

Wide *universal schema*

Syntax Coreference

***Statistical** NLP*

*"**Sebastian Riedel** works in the area of **NLP** and is now Lecturer at **UCL**"*

Benefit: Code Reuse

"Who **lives in** London
and is interested in NLP?

```
interest(x,NLP),
worksFor(x,y),
in(y,London)
```

```
in(UCL,London)

works-in-area-of(Seb,NLP)

lecturer-at(Seb,UCL)

worksFor(x,y):
 faculty-at(x,y)

interest(x,y):
 works-in-area-of(x,y)[0.9]

livesIn(x,z):
 worksFor(x,y),
 locatedIn(y,z) [0.6]
```

***Statistical** Relational*
Learner** and **Reasoner

[Lao et al., 2011]

Wide *universal schema*

Syntax

Coreference

***Statistical** NLP*

*"**Sebastian Riedel** works in the area of **NLP** and is now Lecturer at **UCL**"*

Joint Inference

“Who **lives in** London
and is interested in NLP?

`interest(x,NLP),`
`worksFor(x,y),`
`in(y,London)`

Wide *universal schema*

`in`(UCL,London)

`works-in-area-of`(Seb,NLP)

`lecturer-at`(Seb,UCL)

`worksFor`(x,y):
`faculty-at`(x,y)

`interest`(x,y):
`works-in-area-of`(x,y)`[0.9]`

`livesIn`(x,z):
`worksFor`(x,y),
`locatedIn`(y,z) `[0.6]`

Syntax

Coreference

***Statistical** Relational*
Learner** and **Reasoner

***Statistical** NLP*

*”**Sebastian Riedel** works in the area of **NLP** and is now Lecturer at **UCL**“*

Reasoner and Learner

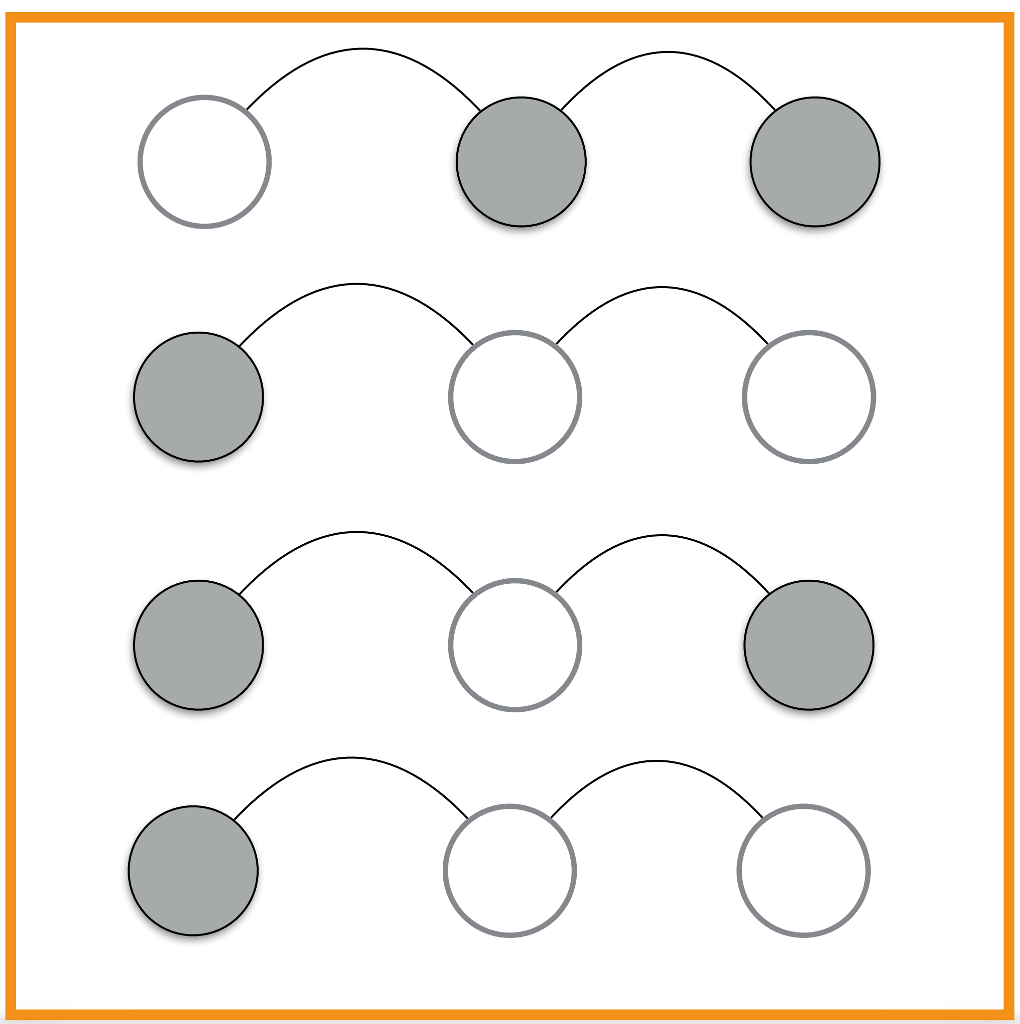
***Statistical** Relational*
Learner** and **Reasoner

?

Probabilistic Logics

Use (weighted) logics to define graphical models

`lecturer-at` `prof-at` `works-for`



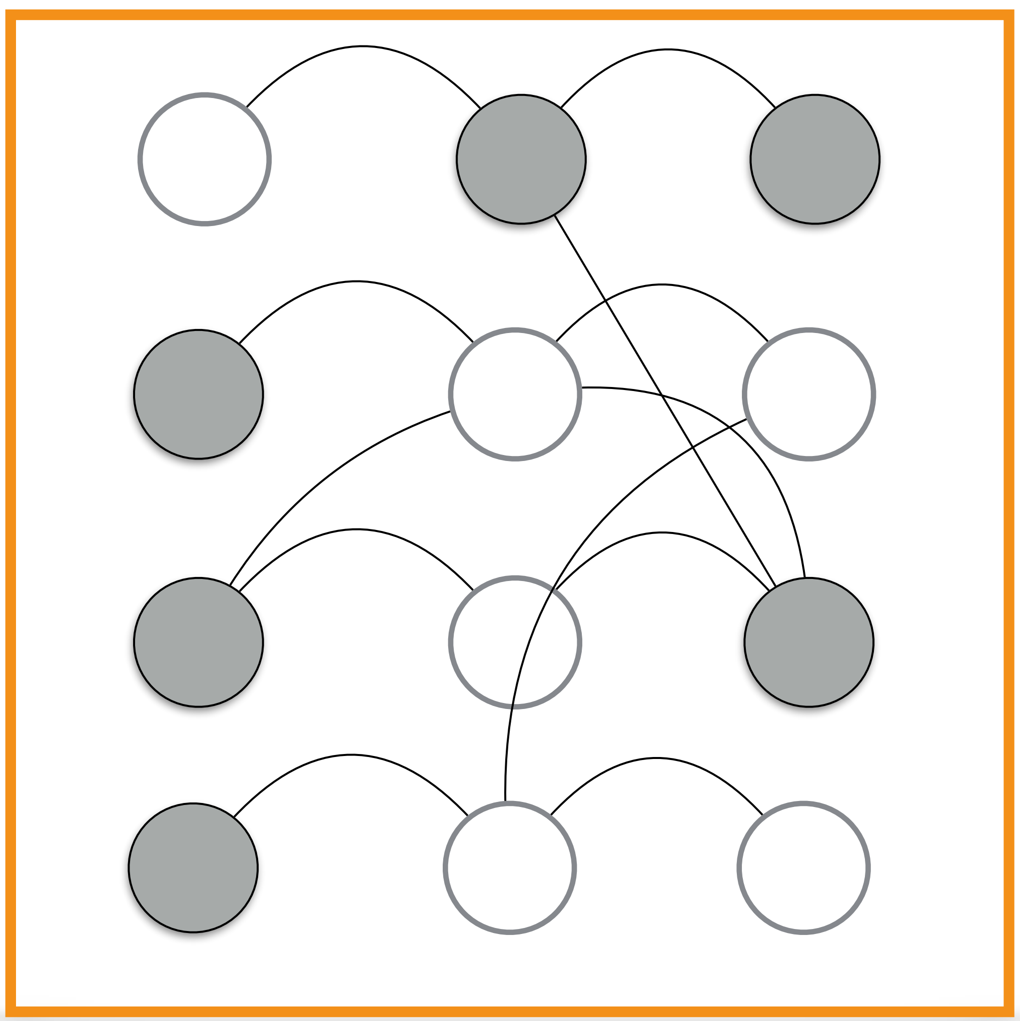
Examples

- **Markov Logic**
  [Richardson and Domingos, 2006]
- **Bayesian Logic Programs**
  [Kersting , 2007]

Probabilistic Logics

Use (weighted) logics to define graphical models

lecturer-at ***prof-at*** ***works-for***



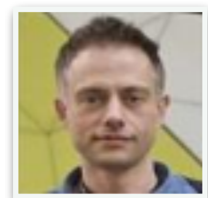
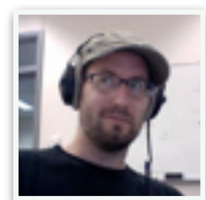
Problems

- **Inference**
- **Rule Learning**

Matrix Factorization

Think of database as a matrix or tensor

lecturer-at	prof-at	works-for
	1	1
1		
1		1
1		

Matrix Factorization

Embed entity (pairs) in low dimensional vector spaces

lecturer-at	prof-at	works-for		
	1	1	?	?
1			?	?
1		1	?	?
1			?	?

Matrix Factorization

Embed relations in low dimensional vector spaces

$$
\begin{bmatrix} & 1 & 1 \\ 1 & & \\ 1 & & 1 \\ 1 & & \end{bmatrix}
\quad
\begin{bmatrix} ? & ? \\ ? & ? \\ ? & ? \\ ? & ? \end{bmatrix}
\quad
\begin{array}{ccc} \texttt{lecturer-at} & \texttt{prof-at} & \texttt{works-for} \\ \hline ? & ? & ? \\ ? & ? & ? \end{array}
$$

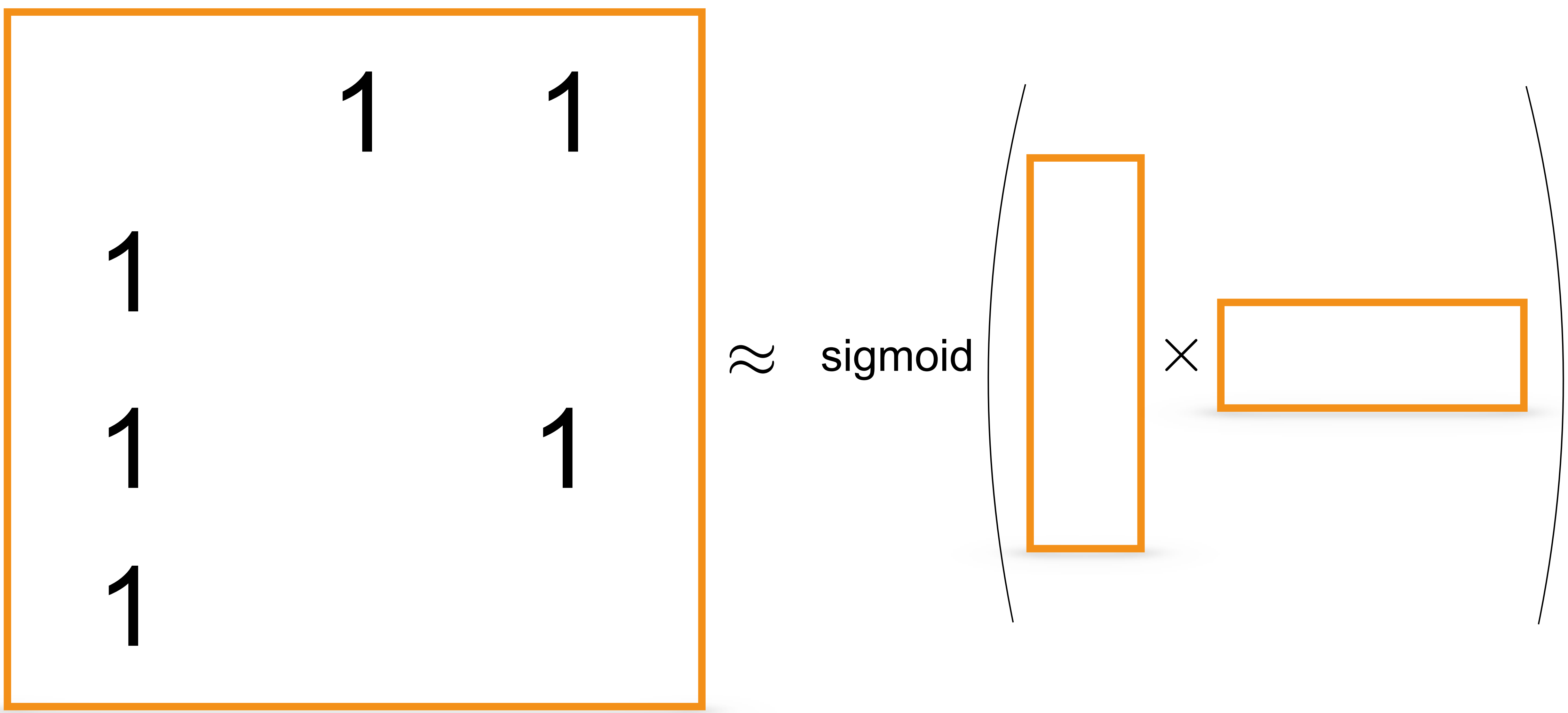
Matrix Factorization

Find a matrix-matrix product that approximates observed DB

$$
\begin{bmatrix} & 1 & 1 \\ 1 & & \\ 1 & & 1 \\ 1 & & \end{bmatrix}
\approx
\begin{bmatrix} ? & ? \\ ? & ? \\ ? & ? \\ ? & ? \end{bmatrix}
\times
\begin{array}{ccc}
\texttt{lecturer-at} & \texttt{prof-at} & \texttt{works-for} \\
\left[\begin{matrix} ? \\ ? \end{matrix}\right. & \begin{matrix} ? \\ ? \end{matrix} & \left.\begin{matrix} ? \\ ? \end{matrix}\right]
\end{array}
$$

Matrix Factorization

Or a non-linear function of this product



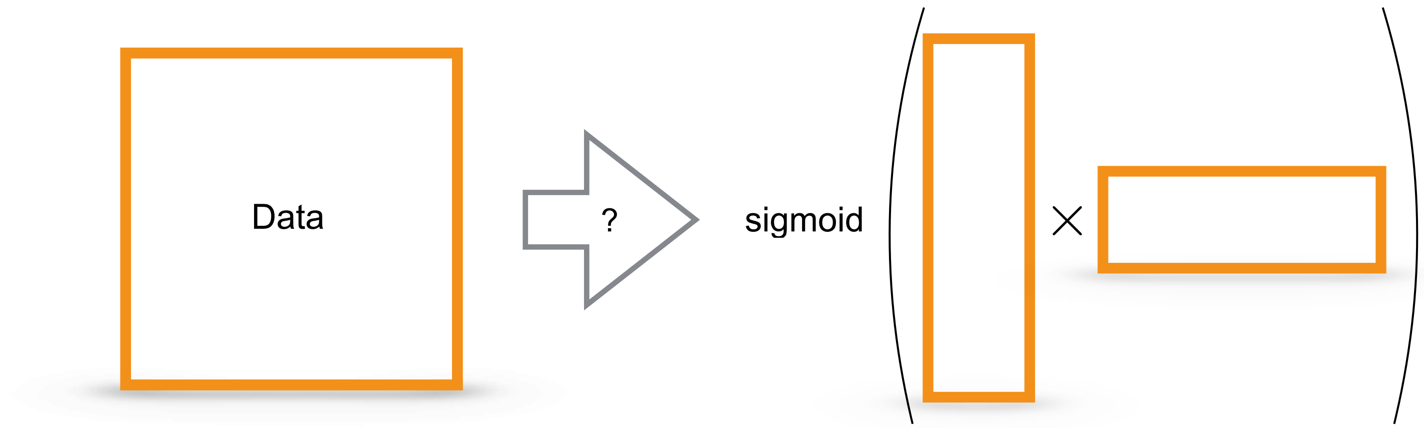
Matrix Factorization

Low rank forces some 0 cells to become non-zero => prediction

$$\begin{bmatrix} & 1 & 1 \\ 1 & & .9 \\ 1 & & 1 \\ 1 & & .9 \end{bmatrix} \approx \text{sigmoid}\left(\boxed{} \times \boxed{} \right)$$

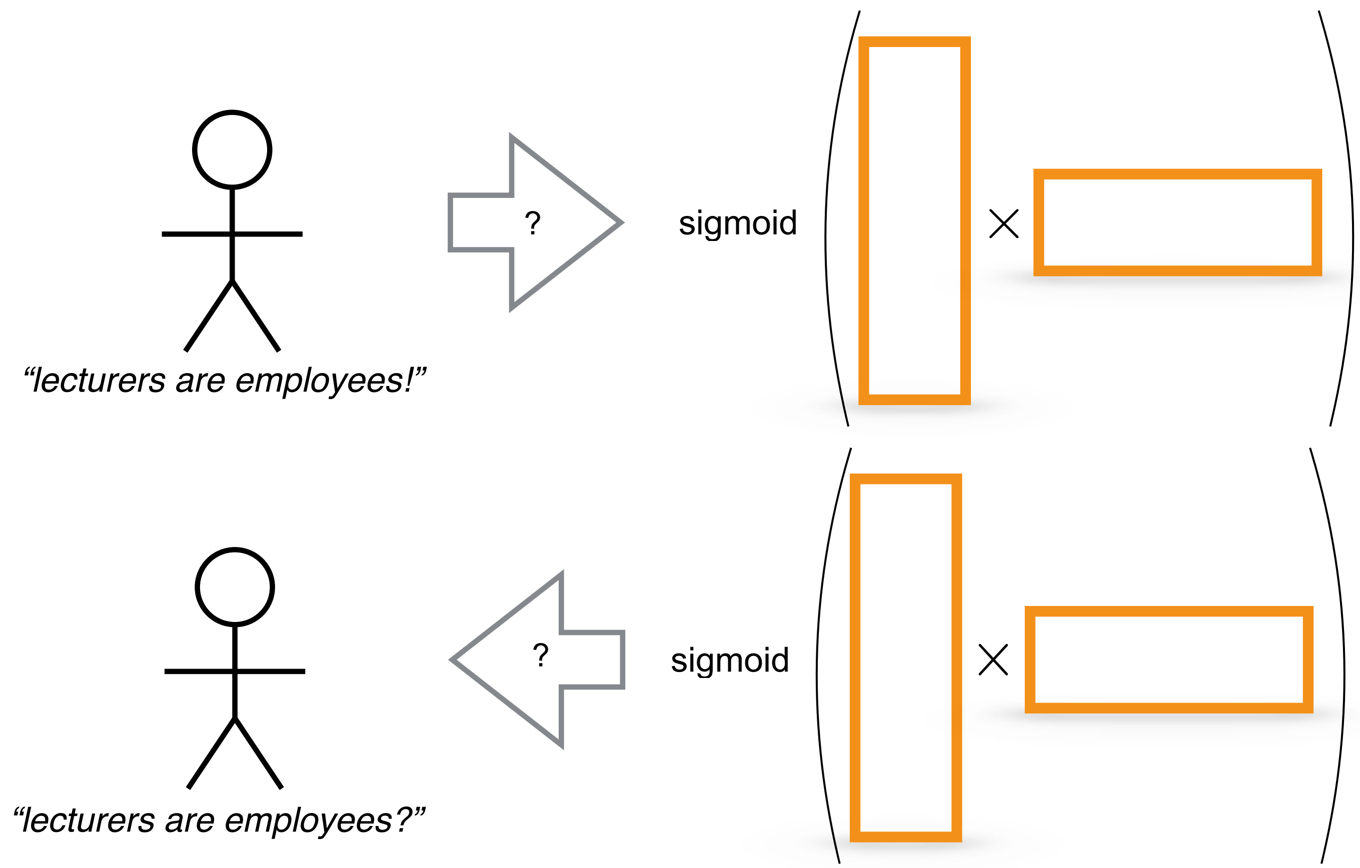
[Nickel, Bordes, …]

Overview



- **Matrix Factorization** Models

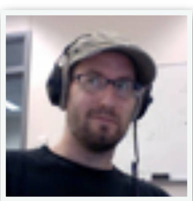
“Talking to Reading Machines”



- **Injecting** Knowledge
- **Extracting** Knowledge

Universal Schema Matrix

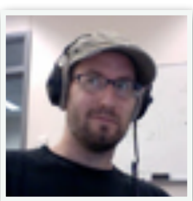
Schema contains structured and unstructured (~OpenIE) relations

	X-*is-professor-at*-**Y**	**X**-*museum-at*-**Y**	**X**-*teaches-history-at*-**Y**	*employee*(**X**,**Y**)
	1	1		1
	1			
	1		1	
				

Goal: Learn to *Complete*

Schema contains structured and unstructured (~OpenIE) relations

Freebase

	X-*is-professor-at*-**Y**	**X**-*museum-at*-**Y**	**X**-*teaches-history-at*-**Y**	*employee*(**X**,**Y**)
	1	1	?	1
 UCL	1	?	?	?
 UNIVERSITY OF OXFORD	1	?	1	?
	?	?	?	?

Model N: Baseline Classifier

[Mintz et al 2009,...]

Standard supervised relation extractor ...

	X-*is-professor-at*-**Y**	**X**-*museum-at*-**Y**	**X**-*teaches-history-at*-**Y**	*employee*(**X**,**Y**)
	1	**1**		$y^{x,y}_{\mathrm{emp}}$

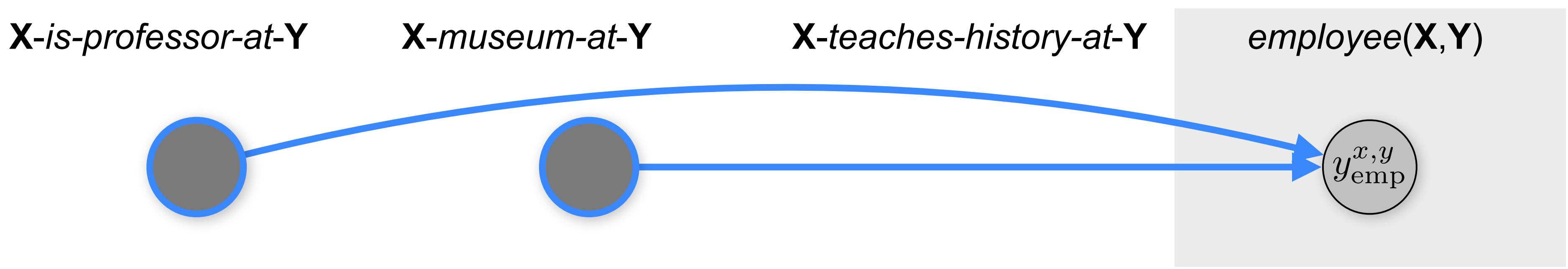
training data

$$p(y^{x,y}_{\mathrm{emp}} = 1 | \qquad\qquad)$$

Model N: Classifier

[Mintz et al 2009,...]

Standard supervised relation extractor ...



observed

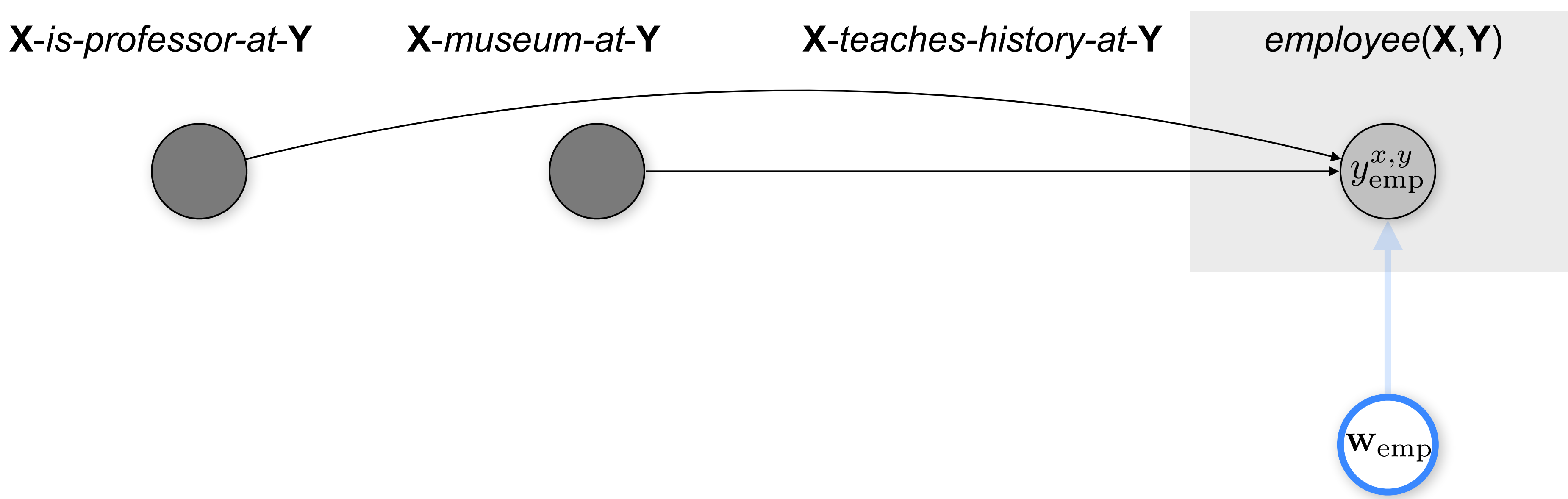
training data

$$p(y^{x,y}_{\mathrm{emp}} = 1 | \underline{\mathbf{f}^{x,y}_{\mathrm{emp}}} \qquad)$$

Model N: Classifier

[Mintz et al 2009,...]

Standard supervised relation extractor ...



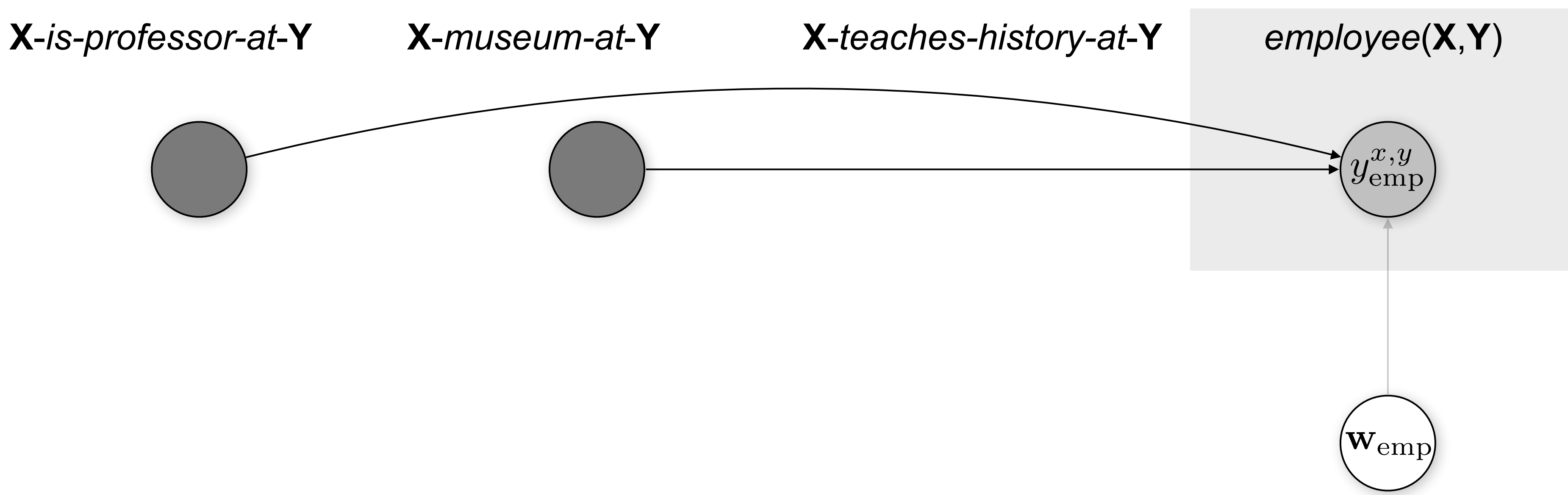
- *observed*
- *training data*
- *parameter*

$$p(y^{x,y}_{\text{emp}} = 1 | \mathbf{f}^{x,y}_{\text{emp}}, \underline{\mathbf{w}_{\text{emp}}})$$

Model N: Classifier

[Mintz et al 2009,...]

Standard supervised relation extractor ...

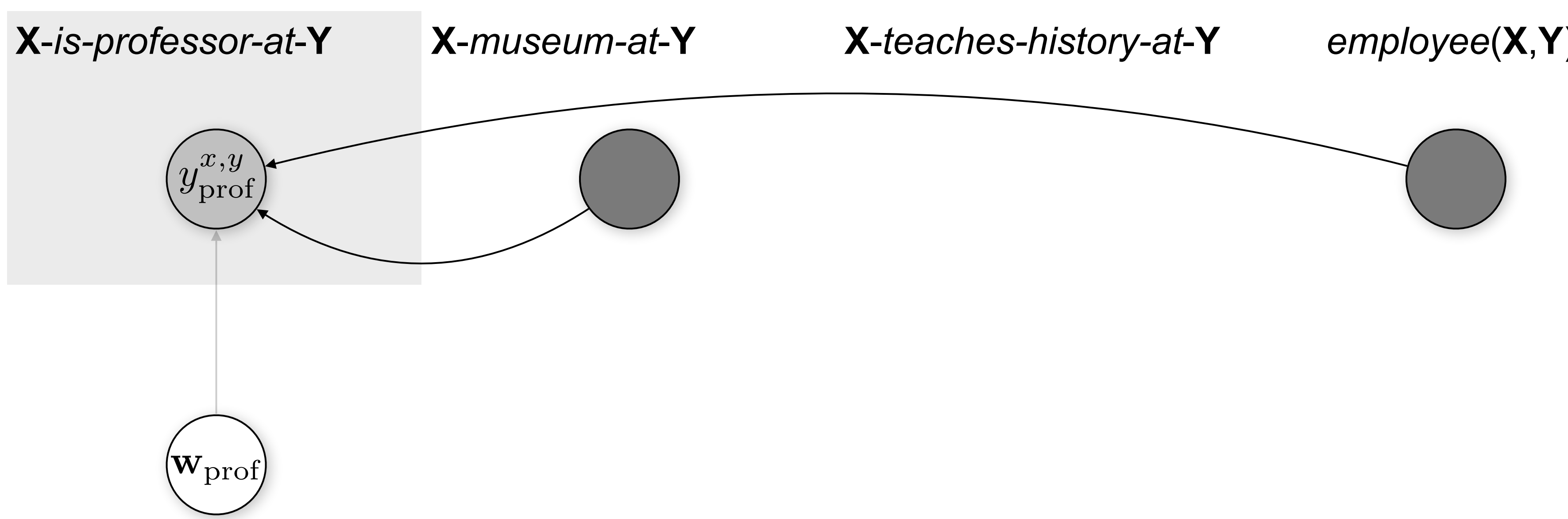


- *observed*
- *training data*
- *parameter*

$$p(y^{x,y}_{\mathrm{emp}} = 1 | \mathbf{f}^{x,y}_{\mathrm{emp}}, \mathbf{w}_{\mathrm{emp}}) \propto \exp[\underline{< \mathbf{f}^{x,y}_{\mathrm{emp}}, \mathbf{w}_{\mathrm{emp}} >}]$$

Model N: Classifier

... for each pattern



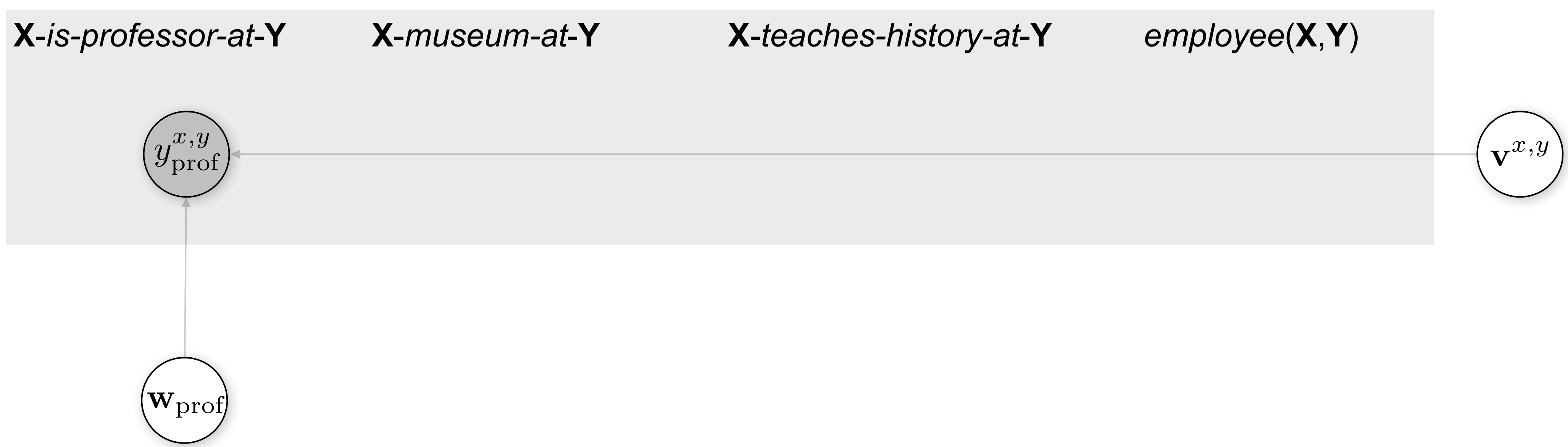
observed
training data
parameter

$$p(y^{x,y}_{\text{prof}} = 1 | \mathbf{f}^{x,y}_{\text{prof}}, \mathbf{w}_{\text{prof}}) \propto \exp[< \mathbf{f}^{x,y}_{\text{prof}}, \mathbf{w}_{\text{prof}} >]$$

Model F: Latent Feature (Factorization)

[Collins et al, 2001]

Model the probability of a pair (x,y) being in relation “prof”



training data

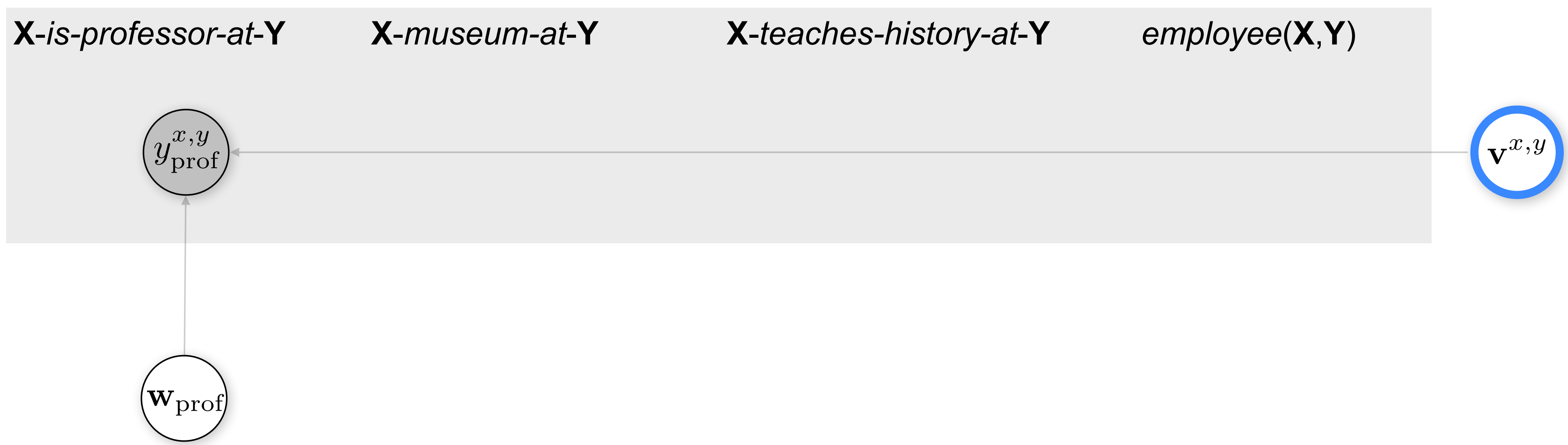
parameter

$$p(y^{x,y}_{\text{prof}} = 1 | \mathbf{v}^{x,y}, \mathbf{w}_{\text{prof}}) \propto \exp[< \mathbf{v}^{x,y}, \mathbf{w}_{\text{prof}} >]$$

Model F: Latent Feature (Factorization)

[Collins et al, 2001]

Per tuple **latent feature** vector



training data

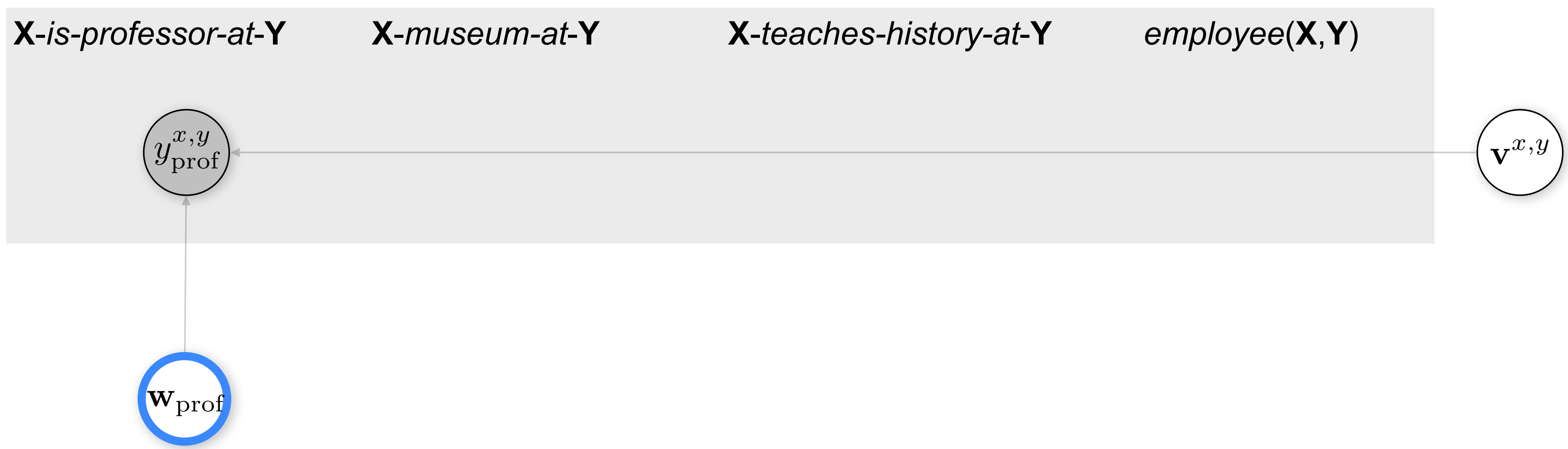
parameter

$$p(y^{x,y}_{\text{prof}} = 1 | \underline{\mathbf{v}^{x,y}}, \mathbf{w}_{\text{prof}}) \propto \exp[< \underline{\mathbf{v}^{x,y}}, \mathbf{w}_{\text{prof}} >]$$

Model F: Latent Feature (Factorization)

[Collins et al, 2001]

Per tuple **latent feature** vector



training data

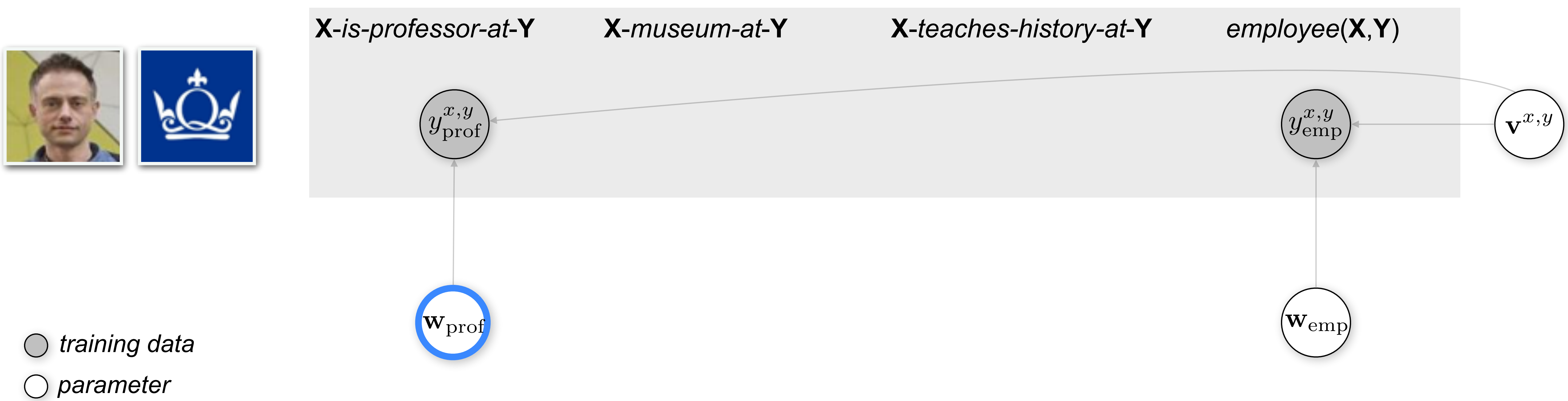
parameter

$$p(y^{x,y}_{\text{prof}} = 1 | \mathbf{v}^{x,y}, \underline{\mathbf{w}_{\text{prof}}}) \propto \exp[< \mathbf{v}^{x,y}, \underline{\mathbf{w}_{\text{prof}}} >]$$

Model F: Latent Feature (Factorization)

[Collins et al, 2001]

Per tuple **latent feature** vector

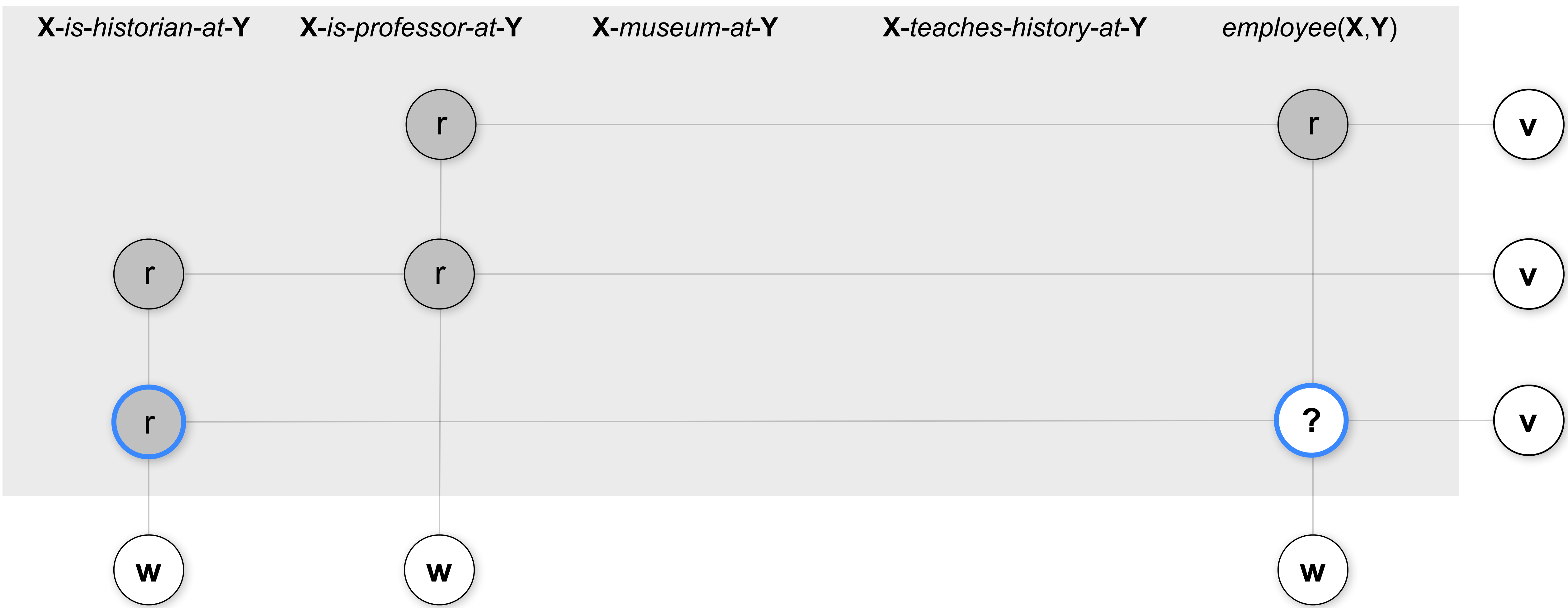


$$p(y^{x,y}_{\text{prof}} = 1 | \mathbf{v}^{x,y}, \underline{\mathbf{w}_{\text{prof}}}) \propto \exp[< \mathbf{v}^{x,y}, \underline{\mathbf{w}_{\text{prof}}} >]$$

$$= sigmoid\,(< \mathbf{v}^{x,y}, \mathbf{w}_{\text{prof}} >)$$

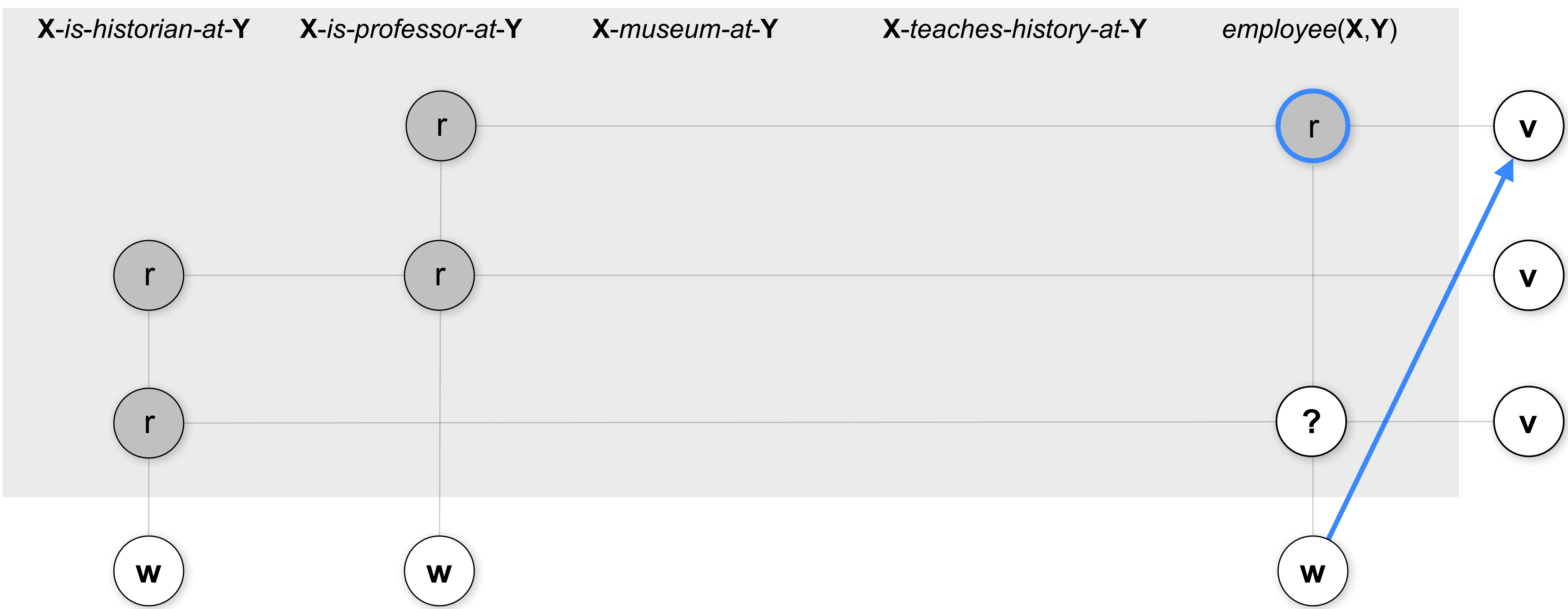
Model F: Latent Feature (Factorization)

Transitive Reasoning



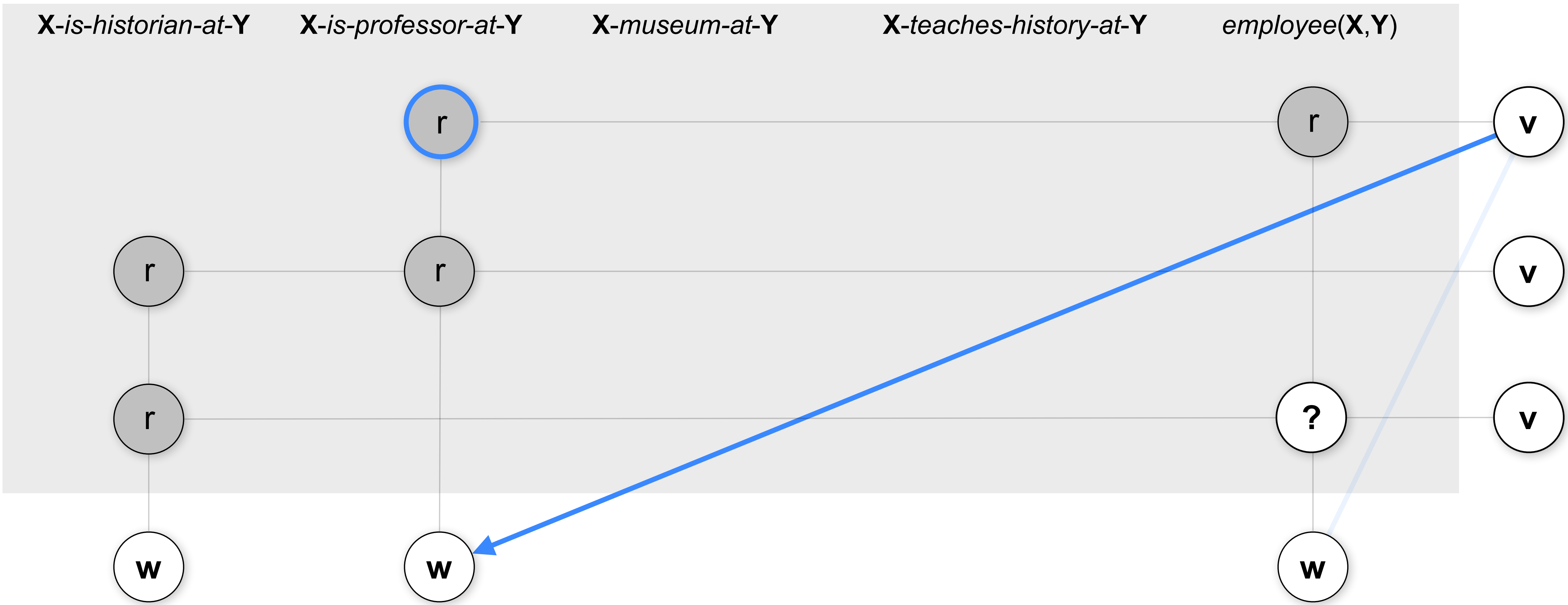
Model F: Latent Feature (Factorization)

Transitive Reasoning



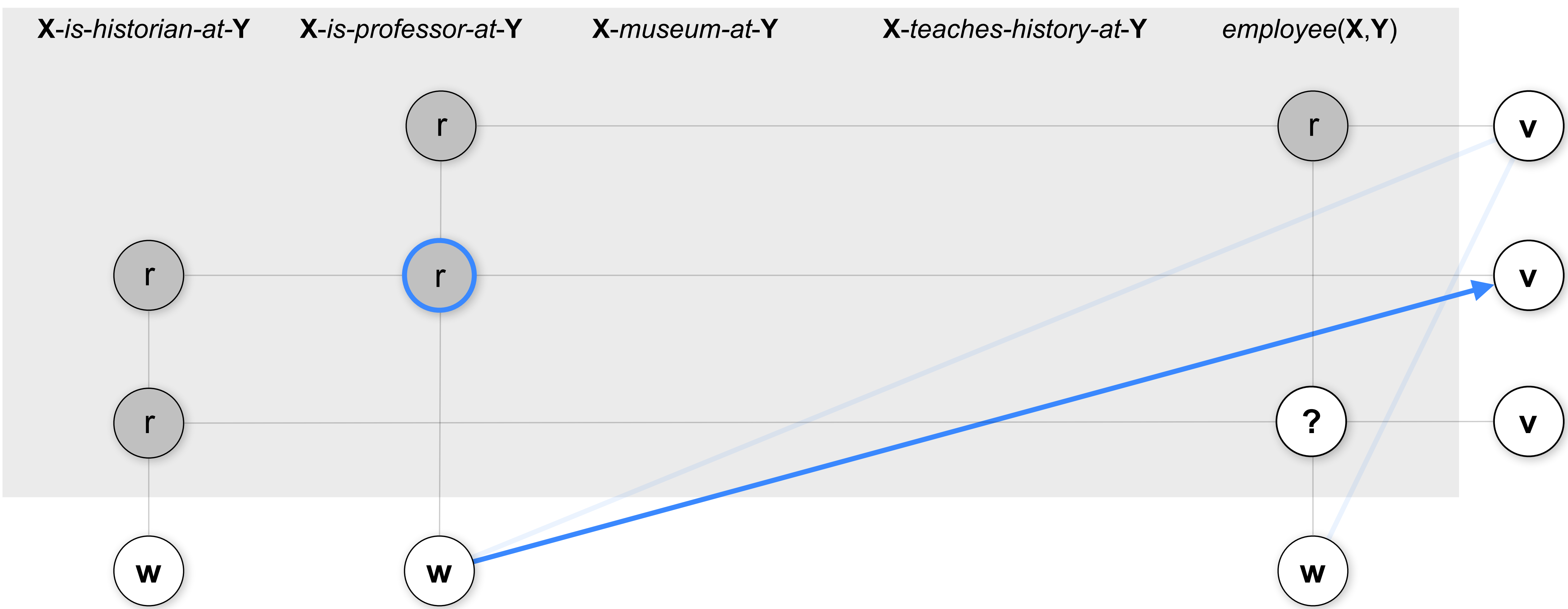
Model F: Latent Feature (Factorization)

Transitive Reasoning



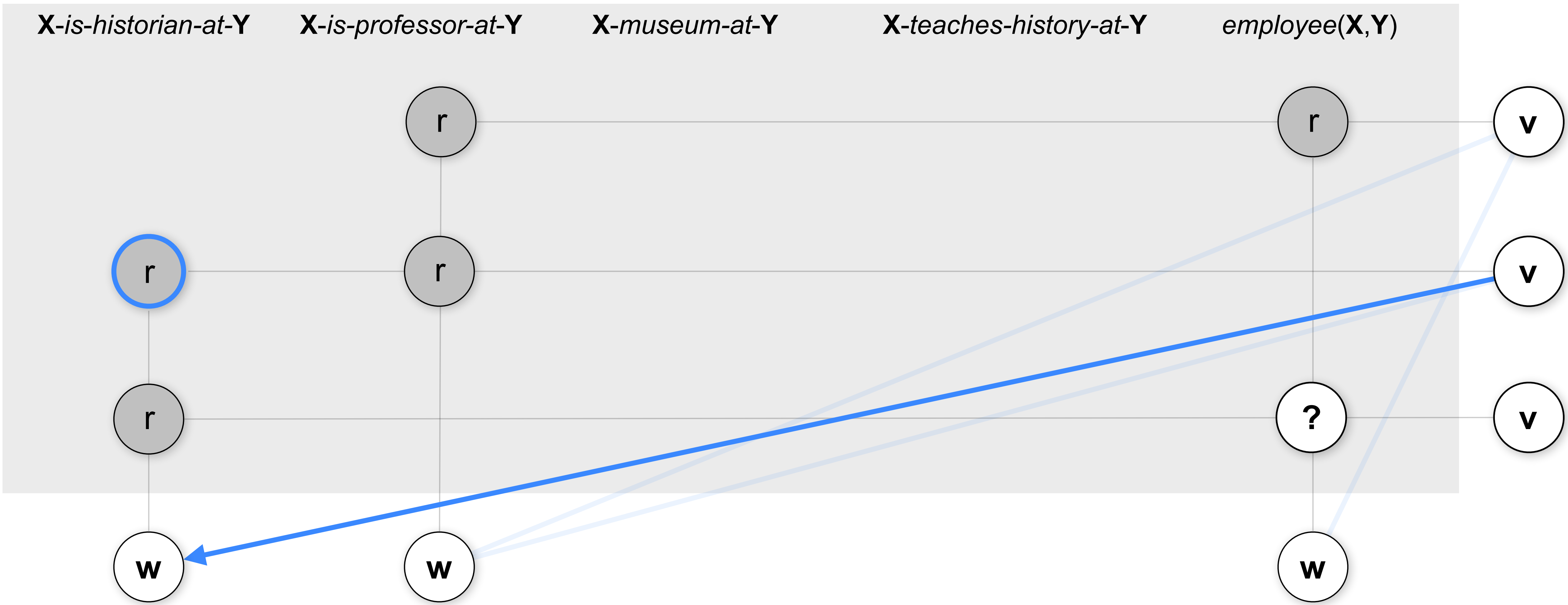
Model F: Latent Feature (Factorization)

Transitive Reasoning



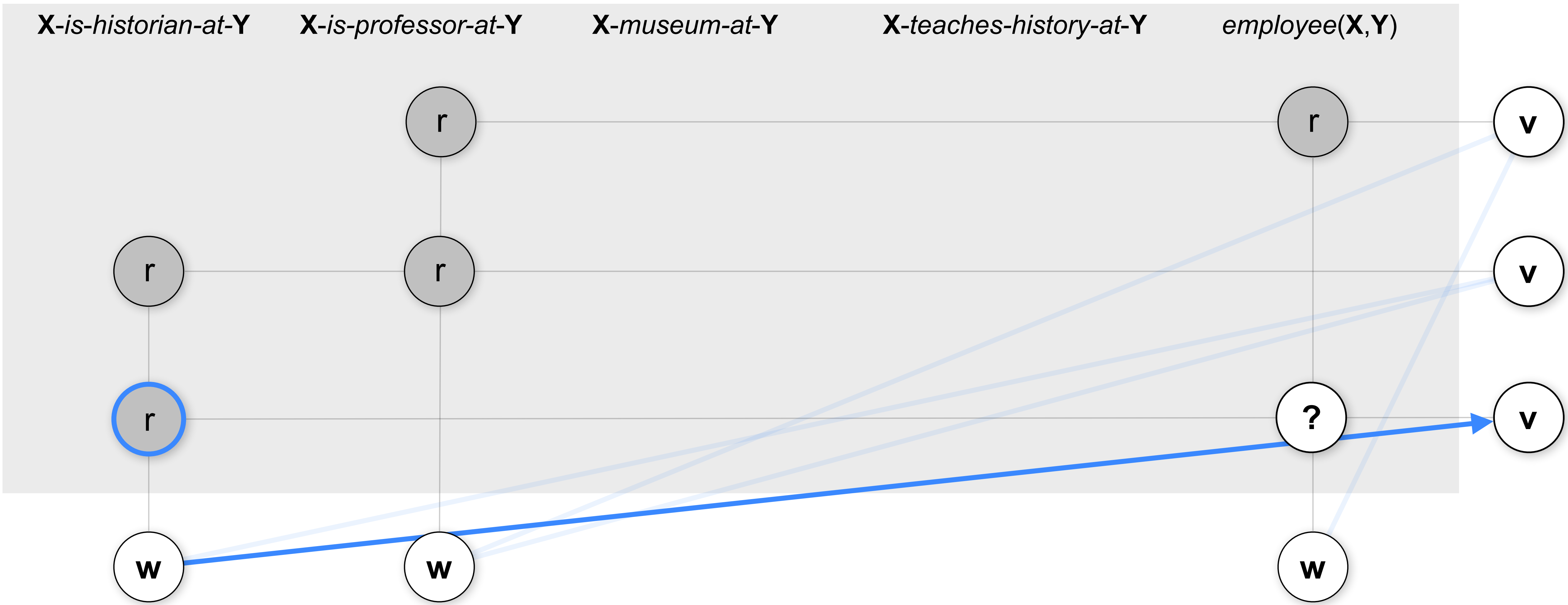
Model F: Latent Feature (Factorization)

Transitive Reasoning



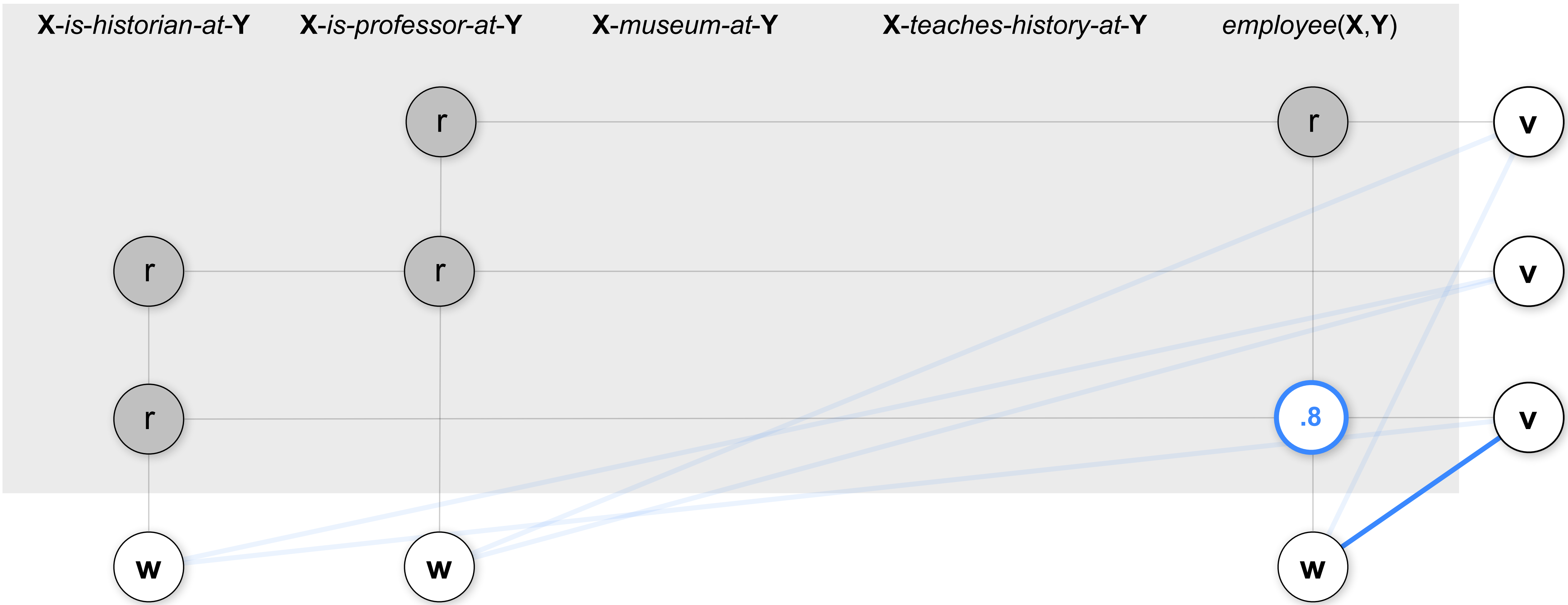
Model F: Latent Feature (Factorization)

Transitive Reasoning



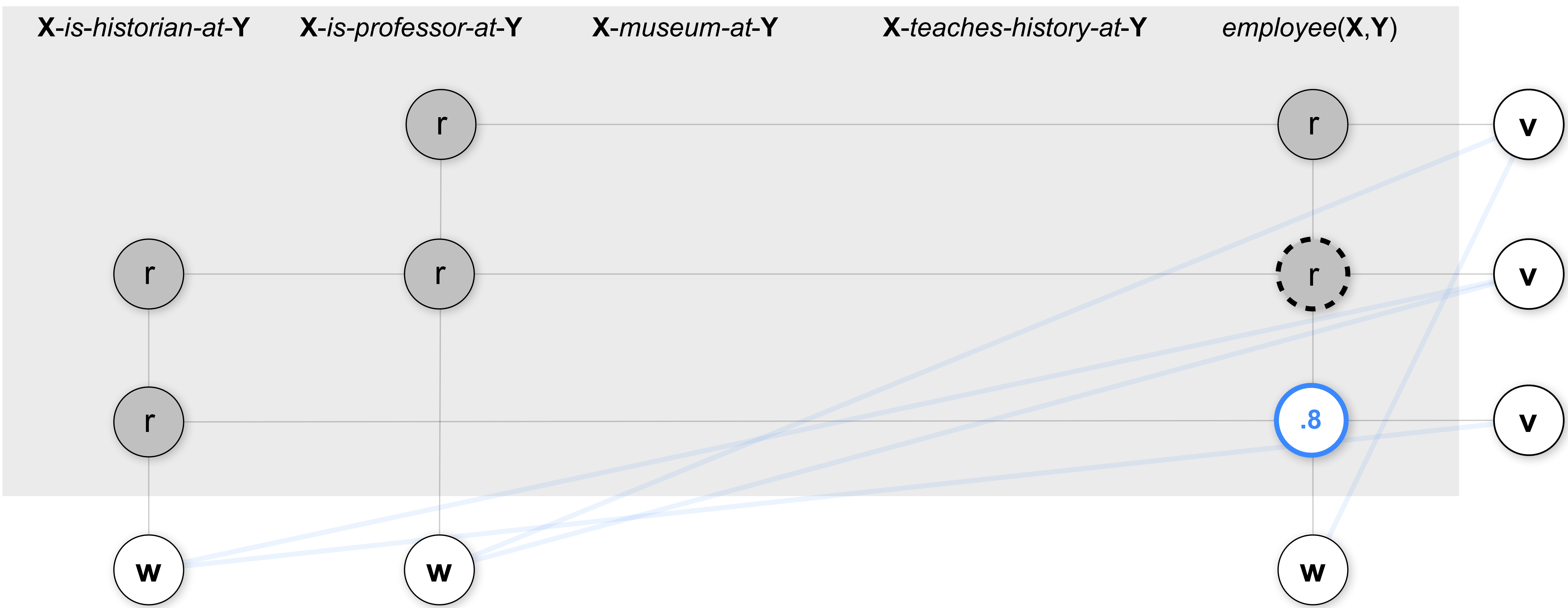
Model F: Latent Feature (Factorization)

Transitive Reasoning



Model F: Latent Feature (Factorization)

Bootstrapping without fantasy



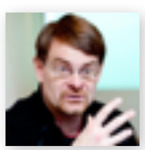
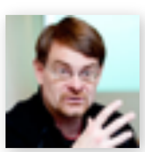
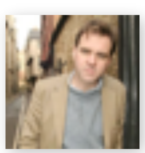
Model E: Selectional Preferences

Relations have **entity type restriction**

	X-*is-professor-at*-**Y**	**X**-*museum-at*-**Y**	**X**-*teaches-history-at*-**Y**

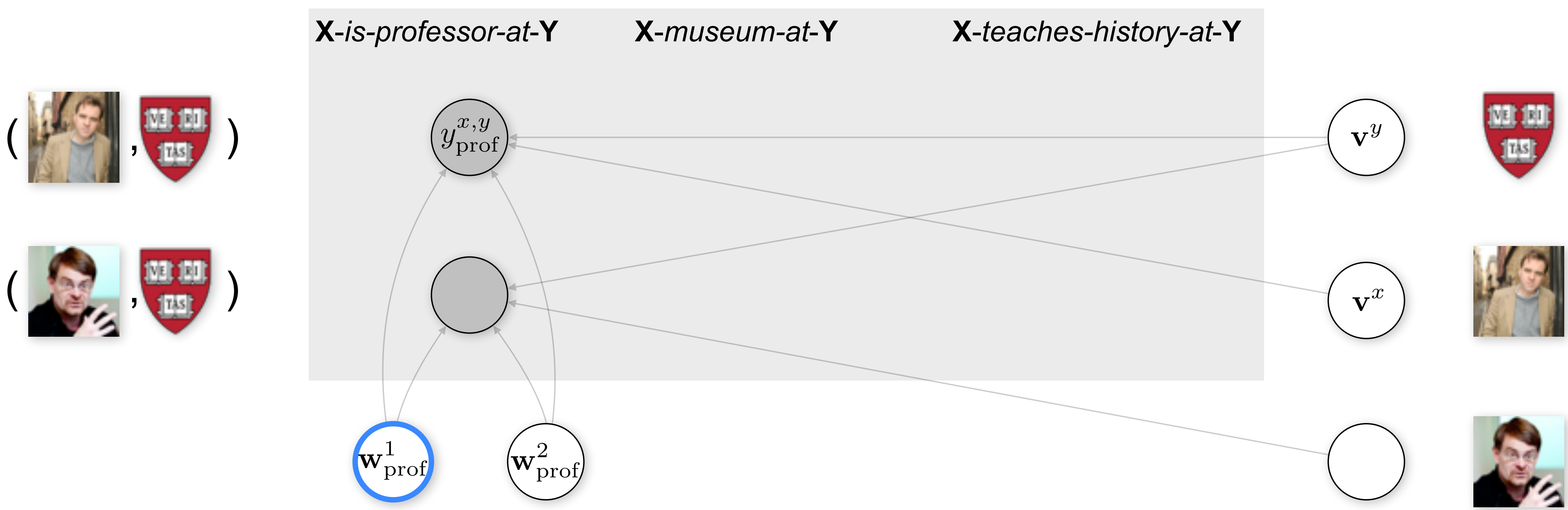
Model E: Selectional Preferences

Relations have **entity type restriction**

	X-*is-professor-at*-**Y**	**X**-*museum-at*-**Y**	**X**-*teaches-history-at*-**Y**
( ,)			
( ,)			
( , )	0	0	0

Model E: Selectional Preferences

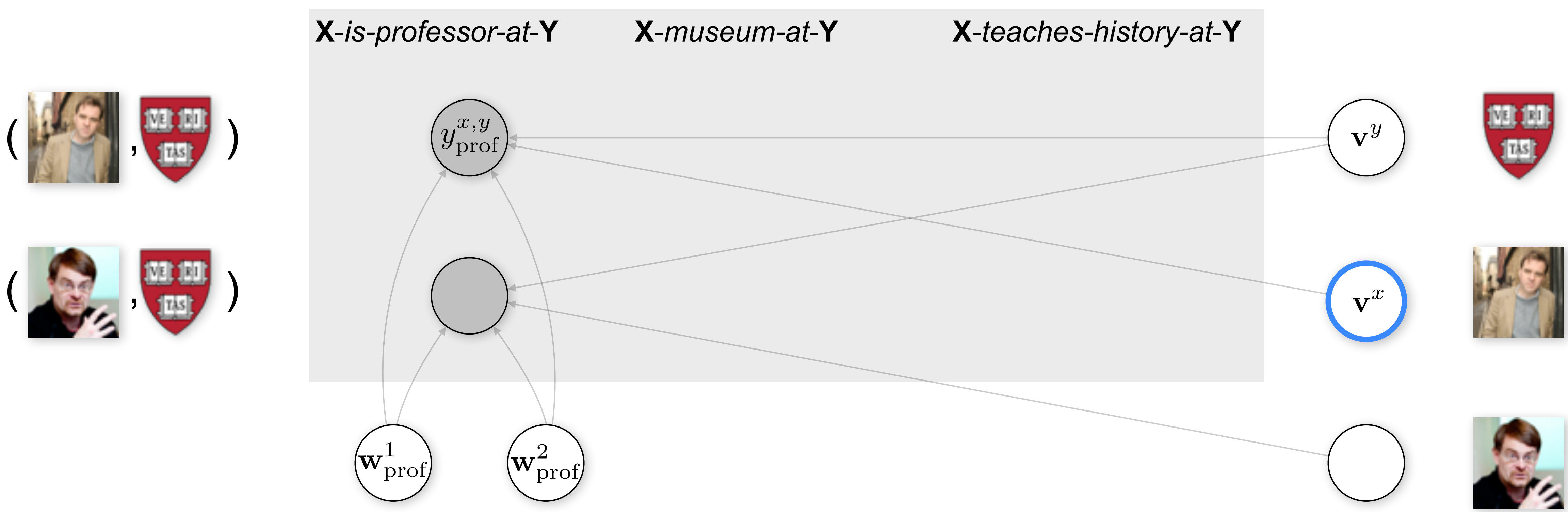
Argument Slot 1 weight vector ...



$$p(y^{x,y}_{\text{prof}} = 1 | \ldots) \propto \exp[< \mathbf{v}^x, \underline{\mathbf{w}}^1_{\text{prof}} > + < \mathbf{v}^y, \mathbf{w}^2_{\text{prof}} >]$$

Model E: Selectional Preferences

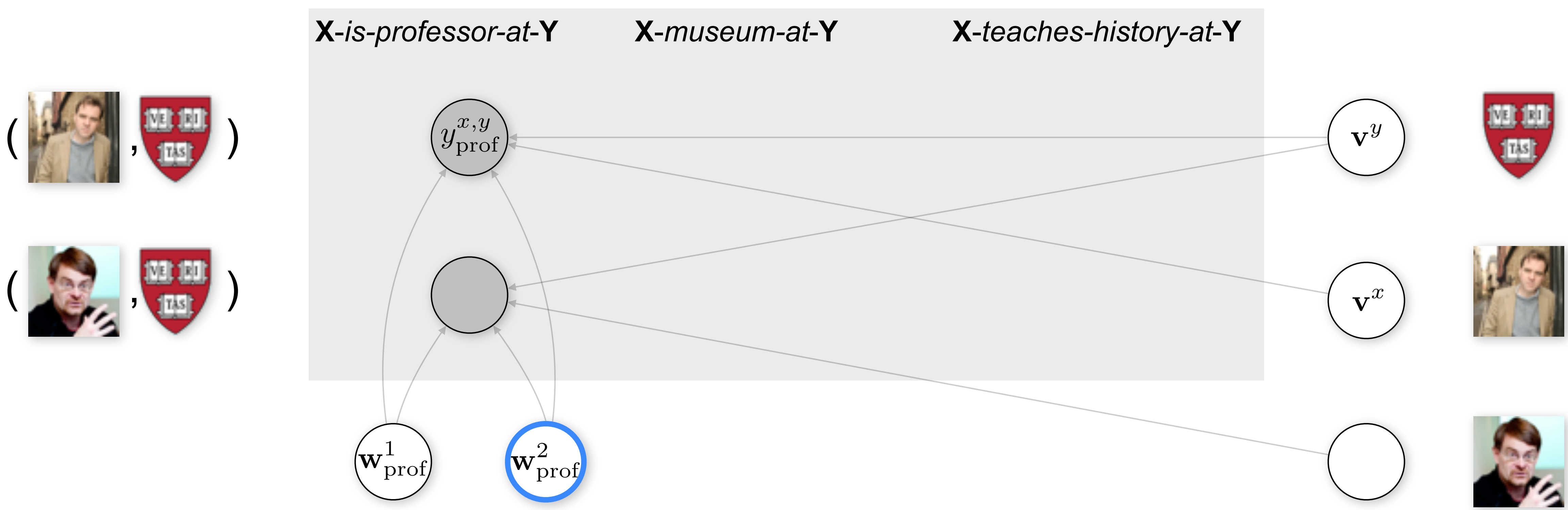
... dot-product with feature vector of **entity 1**



$$p(y^{x,y}_{\text{prof}} = 1 | \ldots) \propto \exp[< \mathbf{v}^x, \mathbf{w}^1_{\text{prof}} > + < \mathbf{v}^y, \mathbf{w}^2_{\text{prof}} >]$$

Model E: Selectional Preferences

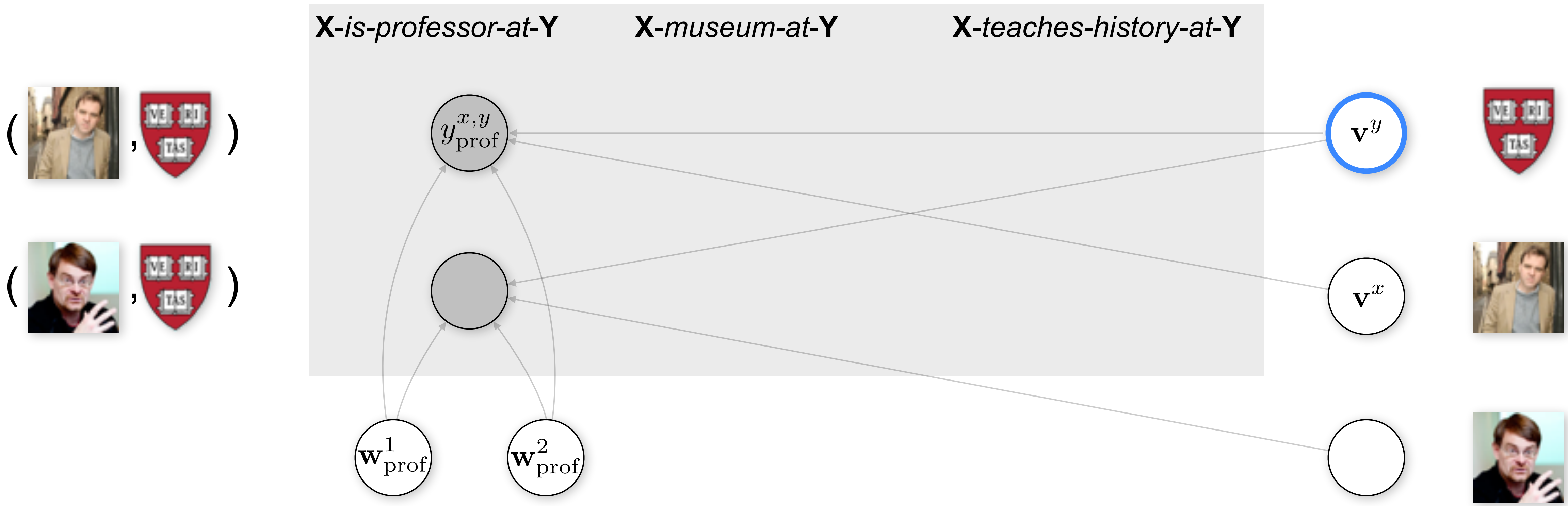
Argument Slot 2 weight vector ...



$$p(y^{x,y}_{\text{prof}} = 1 | \ldots) \propto \exp[< \mathbf{v}^x, \mathbf{w}^1_{\text{prof}} > + < \mathbf{v}^y, \underline{\mathbf{w}}^2_{\text{prof}} >]$$

Model E: Selectional Preferences

... dot-product with feature vector of **entity 2**



$$p(y^{x,y}_{\text{prof}} = 1 | \ldots) \propto \exp[< \mathbf{v}^x, \mathbf{w}^1_{\text{prof}} > + < \underline{\mathbf{v}^y}, \mathbf{w}^2_{\text{prof}} >]$$

Combinations

models capture different aspects of the data, combine them (e.g., NF)

$$p(y_{\mathrm{emp}}^{x,y} = 1 | \ldots) \propto \exp[< \mathbf{f}_{\mathrm{emp}}^{x,y}, \mathbf{w}_{\mathrm{emp}}^{\mathrm{N}} > + < \mathbf{v}^{x,y}, \mathbf{w}_{\mathrm{emp}}^{\mathrm{F}} >]$$

Evaluation (Structured)

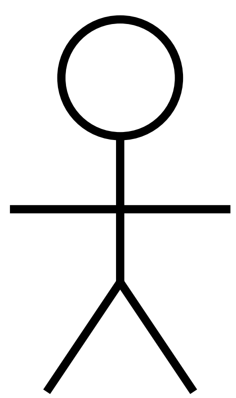
[Mintz 09; Yao 11; Surdenau 12]

Evaluate **average precision** per **Freebase relation**.

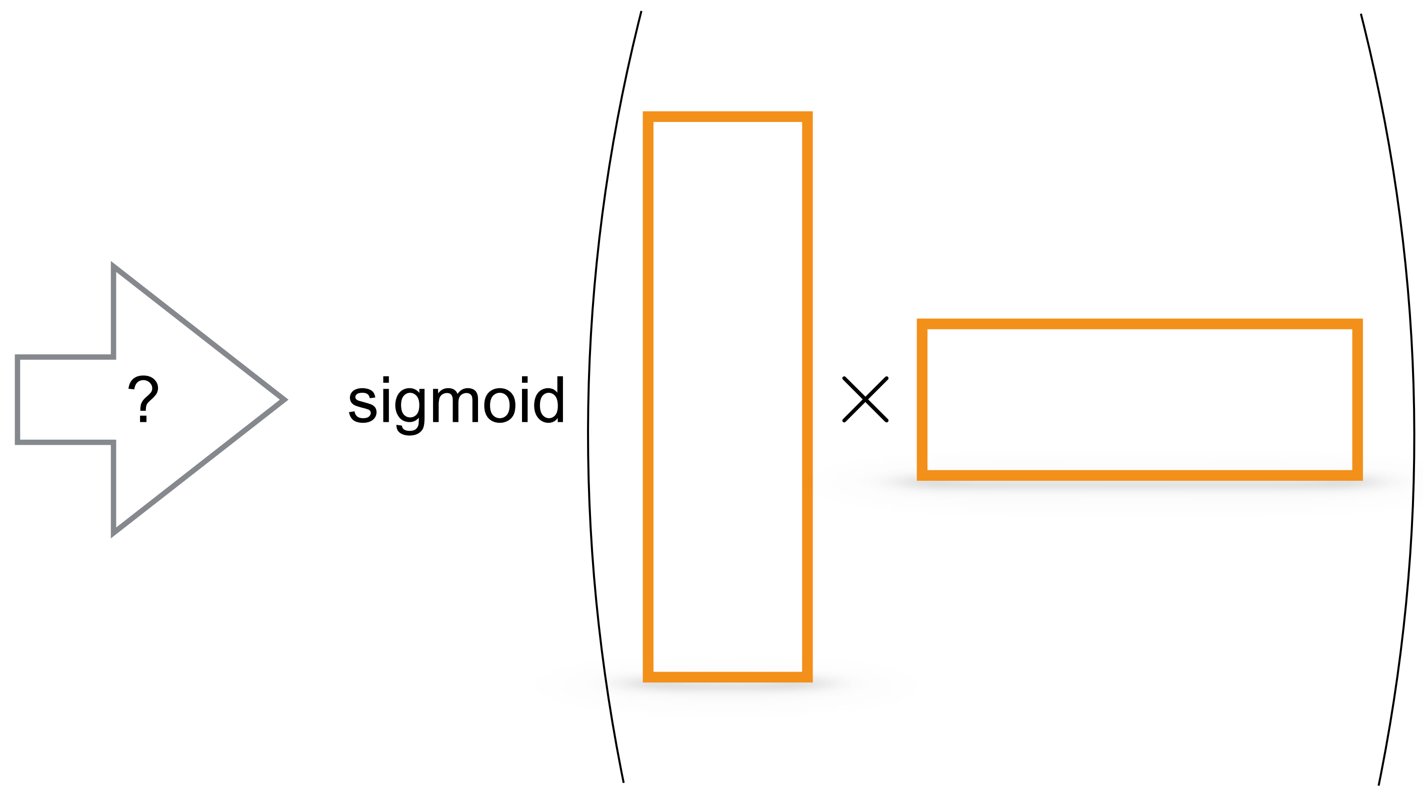
Relation	MI09	YA11	SU12	N+F+E
employee	0.67	0.64	0.7	**0.79**
containedby	0.48	0.51	0.54	**0.69**
parents	0.			0.39
...	.			...
Weighted MAP	0.48	0.52	0.57	**0.69**
MAP	0.32	0.42	0.56	**0.63**

~45 minutes to train our models on 4000 relations, ~50k entity pairs

Injecting Knowledge

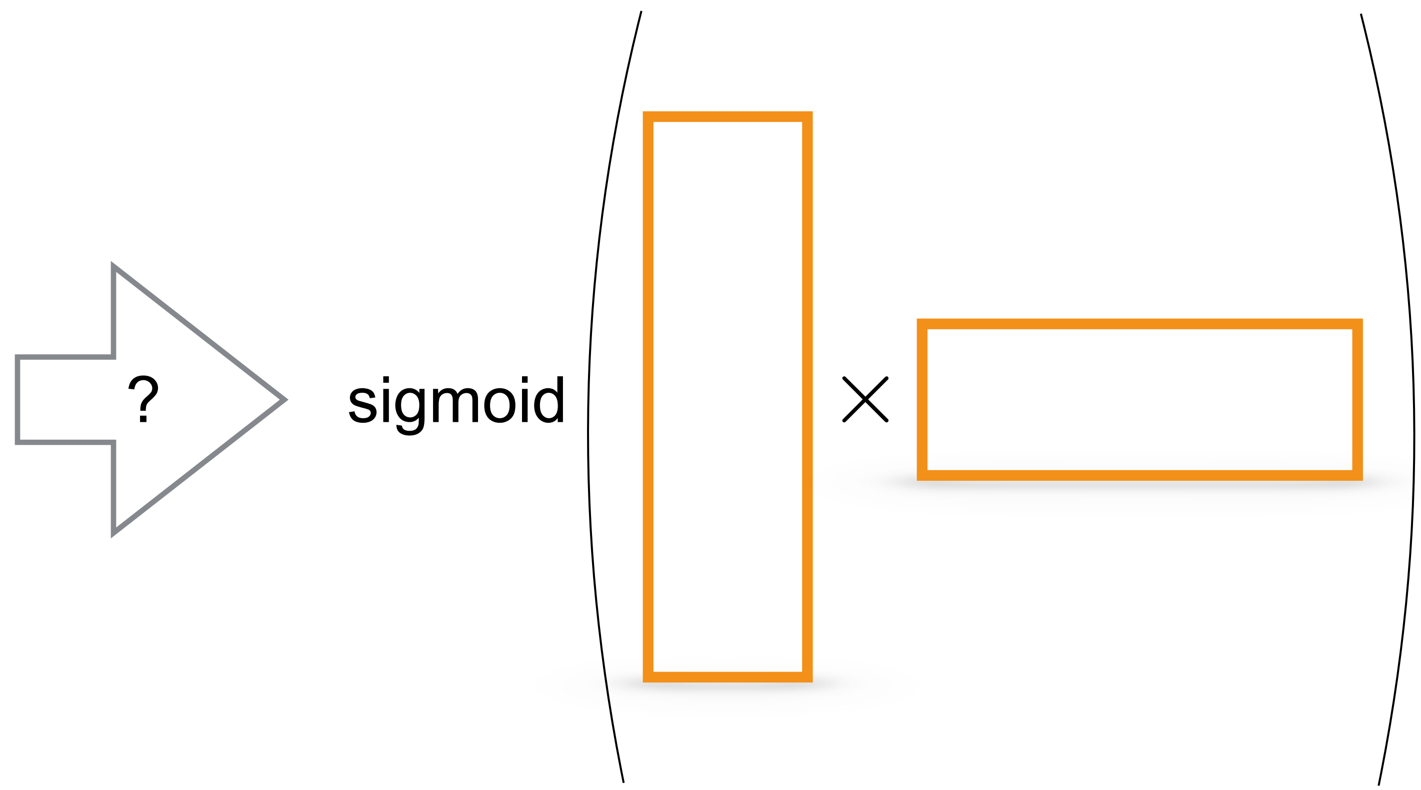


"lecturers are employees!"



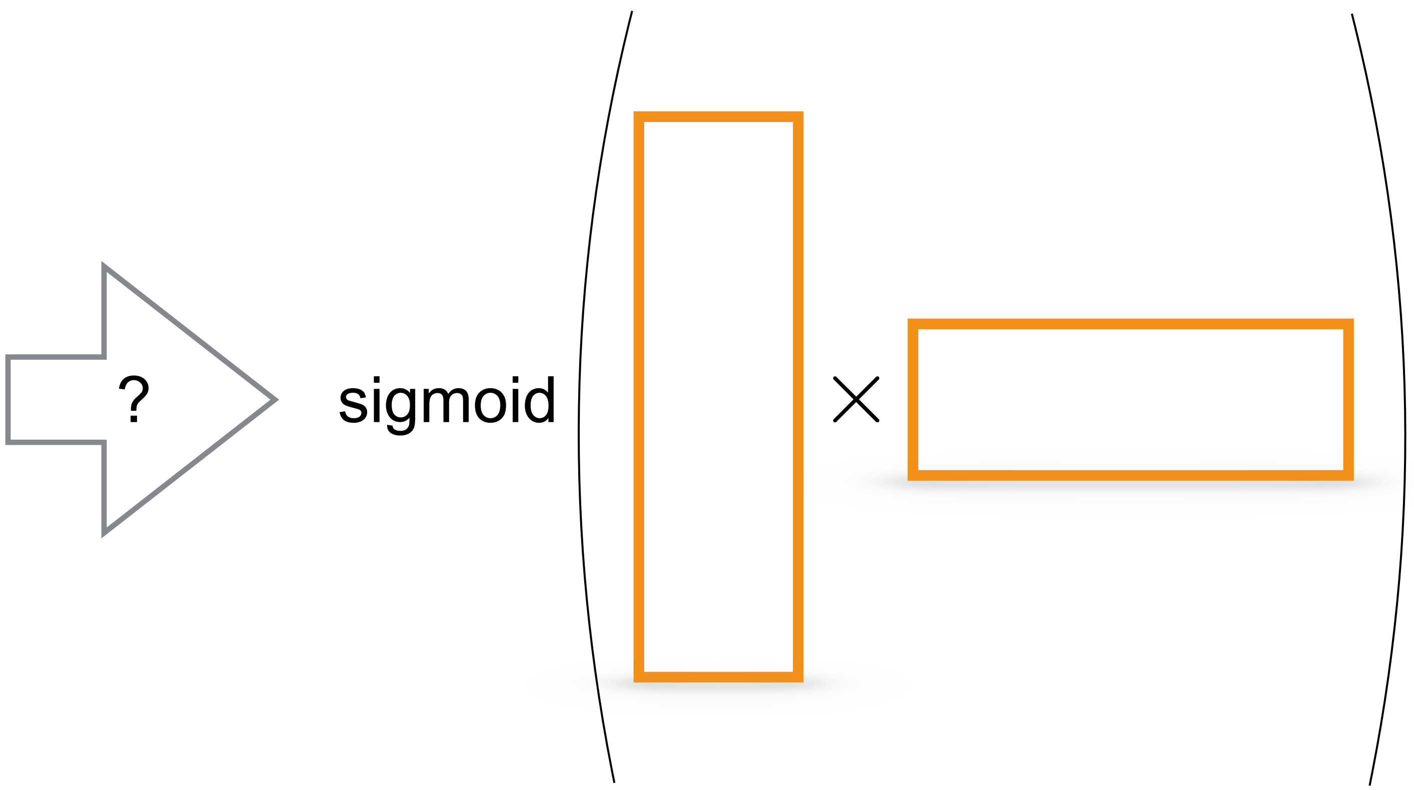
Injecting Knowledge

"a liquid turns into a solid when its temperature is lowered below its freezing point



Injecting Knowledge: Rules

$\forall x,y$: birthplace(x,y)
$\Rightarrow$ bornIn(x,y)



Goal: Predict Unseen Cells …

native-of	*'s birthplace*	*bornIn*	*livesIn*
		1	1
1	1		
	1	?	
1			

… By Using Rules and Data

native-of	*'s birthplace*	*bornIn*	*livesIn*
		1	1
1	1		
	1	1	.8
1		1	.8

$\forall x,y$: birthplace$(x,y) \Rightarrow$ bornIn(x,y)

Baselines

native-of	's birthplace	bornIn	livesIn
		1	1
1	1	1	
	1	1	.8
1	.8	.8	.6

- Rules only
- Rules after learning
- Rules before learning

$\forall x,y$: birthplace(x,y) $\Rightarrow$ bornIn(x,y)

Pre-Injection may not add data at all

native-of	*’s birthplace*	*bornIn*	*livesIn*
1	1	1	1
1			
	1		
1			

- Rules before learning

$\forall x,y$: birthplace$(x,y) \Rightarrow$ bornIn(x,y)

Pre-Injection may not add data at all

native-of	*'s birthplace*	*bornIn*	*livesIn*
1			1
1			
	1		
1			

- Rules before learning

$\forall x,y$: birthplace$(x,y) \Rightarrow$ bornIn(x,y)

Idea: Iterate

native-of	*'s birthplace*	*bornIn*	*livesIn*
1	.8	1	1
1	.8	1	
	1		
1			

- Inference with model
- Apply rules

… and learn again

$\forall x,y$: birthplace(x,y) $\Rightarrow$ bornIn(x,y)

Our approach

[Rocktaeschel et al 15]

- **Directly** optimise to **fulfil formulae in expectation**
- formulae have **compositional expectations**

$$E_{\mathbf{v},\mathbf{w}}[birthplace(Seb,HH)] = E_{\mathbf{v},\mathbf{w}}[y_{birthplace}^{Seb,HH}] = sigm(<\mathbf{v}^{Seb,HH},\mathbf{w}_{birthplace}>)$$

$$E_{\mathbf{v},\mathbf{w}}[r(X_1,X_2)] = sigm(<\mathbf{v}^{X_1,X_2},\mathbf{w}_r>)$$

$$E_{\mathbf{v},\mathbf{w}}[A \wedge B] = E_{\mathbf{v},\mathbf{w}}[A] \times E_{\mathbf{v},\mathbf{w}}[B]$$

$$E_{\mathbf{v},\mathbf{w}}[\neg A] = 1 - E_{\mathbf{v},\mathbf{w}}[A]$$

$$E_{\mathbf{v},\mathbf{w}}[A \Rightarrow B] = 1 - (E_{\mathbf{v},\mathbf{w}}[A] \times (1 - E_{\mathbf{v},\mathbf{w}}[B]))$$

Our approach

[Rocktaeschel et al 15]

- **Directly** optimise to **fulfil formulae in expectation**
- formulae have **compositional expectations**
- **quantification** through **grounding**

$$
\begin{aligned}
E_{\mathbf{v},\mathbf{w}}[\forall x.f(x)] &= E_{\mathbf{v},\mathbf{w}}[f(X_1)\wedge\ldots\wedge f(X_n)] \\
&= E_{\mathbf{v},\mathbf{w}}[f(X_1)]\times\ldots\times E_{\mathbf{v},\mathbf{w}}[f(X_n)]
\end{aligned}
$$

General Framework

[Rocktaeschel et al 15]

- Find embeddings **v** and **w** that…
- Maximize **log expectation** of a set of formulae f

$$\arg\max_{\mathbf{v},\mathbf{w}} \sum_f \log\left(E_{\mathbf{v},\mathbf{w}}[f]\right)$$

- **Generalizes** regular (binary) matrix factorization with logistic loss
- Get gradients by **back-propagation** through log(E[.]) tree
- Optimize via **SGD** / Adagrad etc.

Experiments

[Rocktaeschel et al 15]

- “Zero-shot” learning
  - Given: a lot of surface form data, but **no Freebase relations**
  - Goal: given few (36) Freebase rules, learn to Freebase relations

Experiments: Zero-Shot Learning

Remove Freebase data from training set …

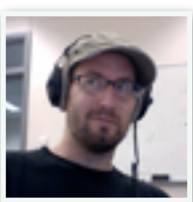
Freebase

X-*is-professor-at*-**Y**	**X**-*museum-at*-**Y**	**X**-*teaches-history-at*-**Y**	*employee*(**X**,**Y**)
1	1		1
1			
1		1	

Experiments: Zero-Shot Learning

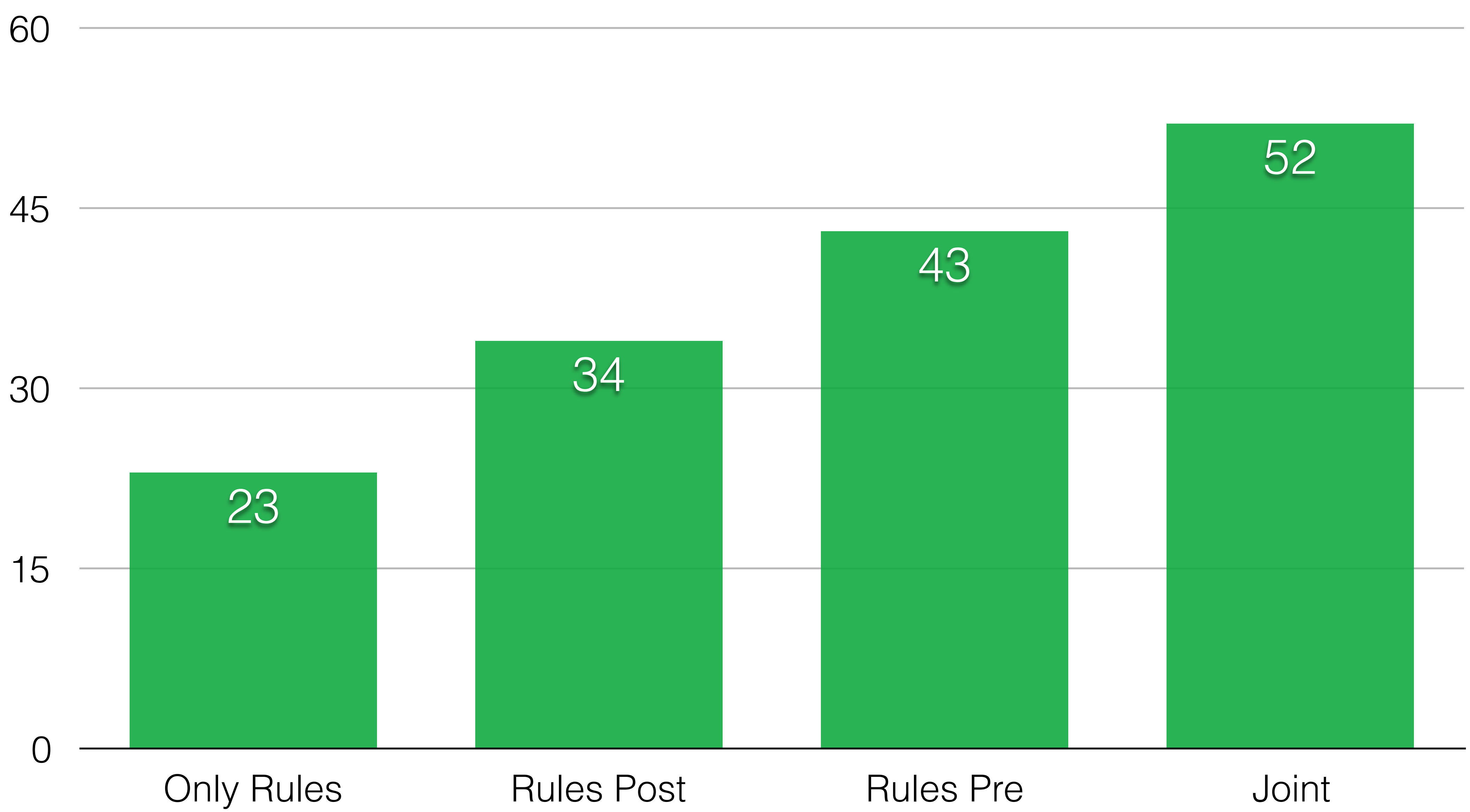
and learn only from surface form relations, and rules

Freebase

	X-*is-professor-at*-**Y**	**X**-*museum-at*-**Y**	**X**-*teaches-history-at*-**Y**	*employee*(**X**,**Y**)
	1	1		
	1			
	1		1	
				

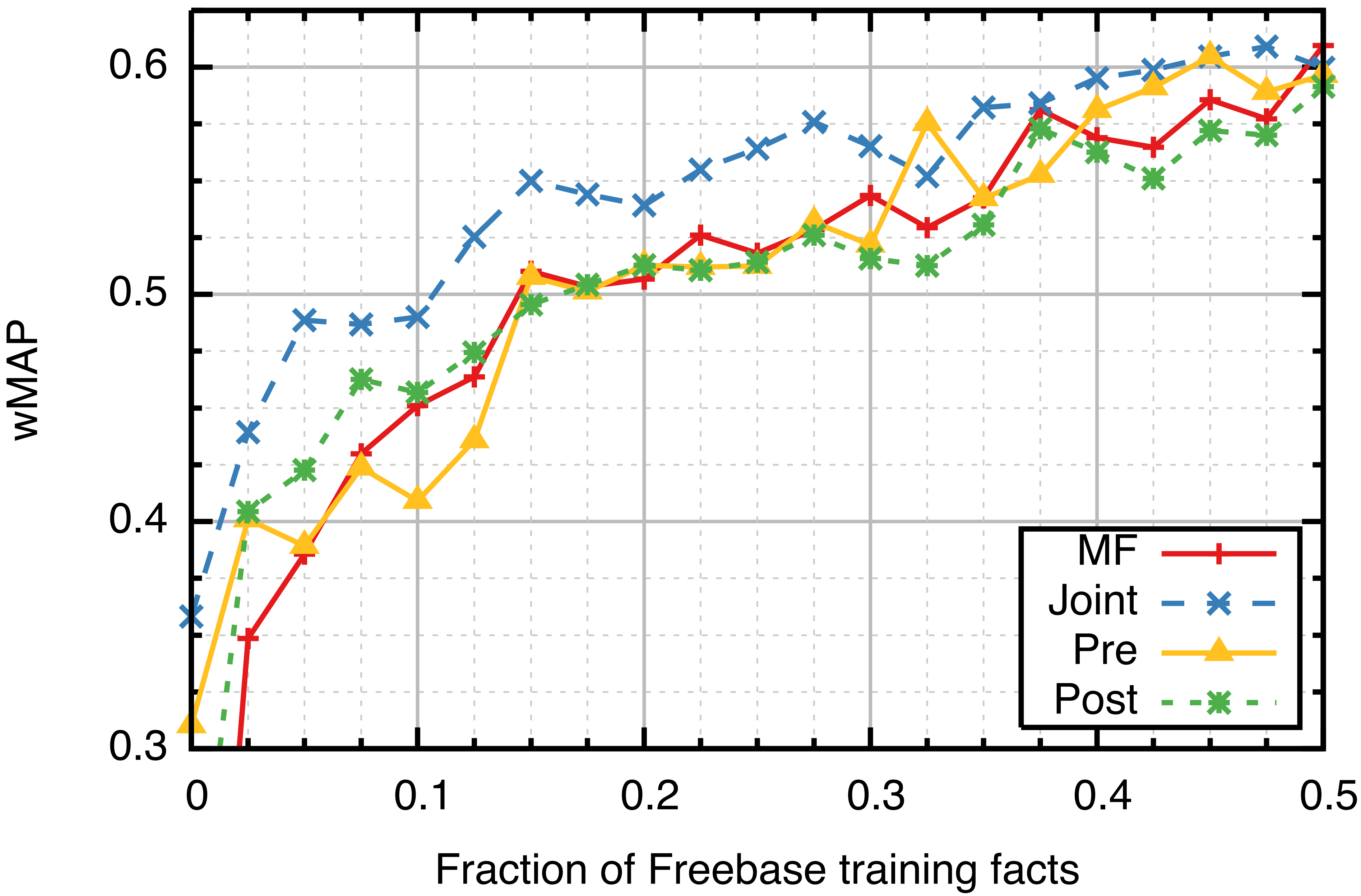
Zero-Shot Learning Results (MAP)

[Rocktaeschel et al 15]

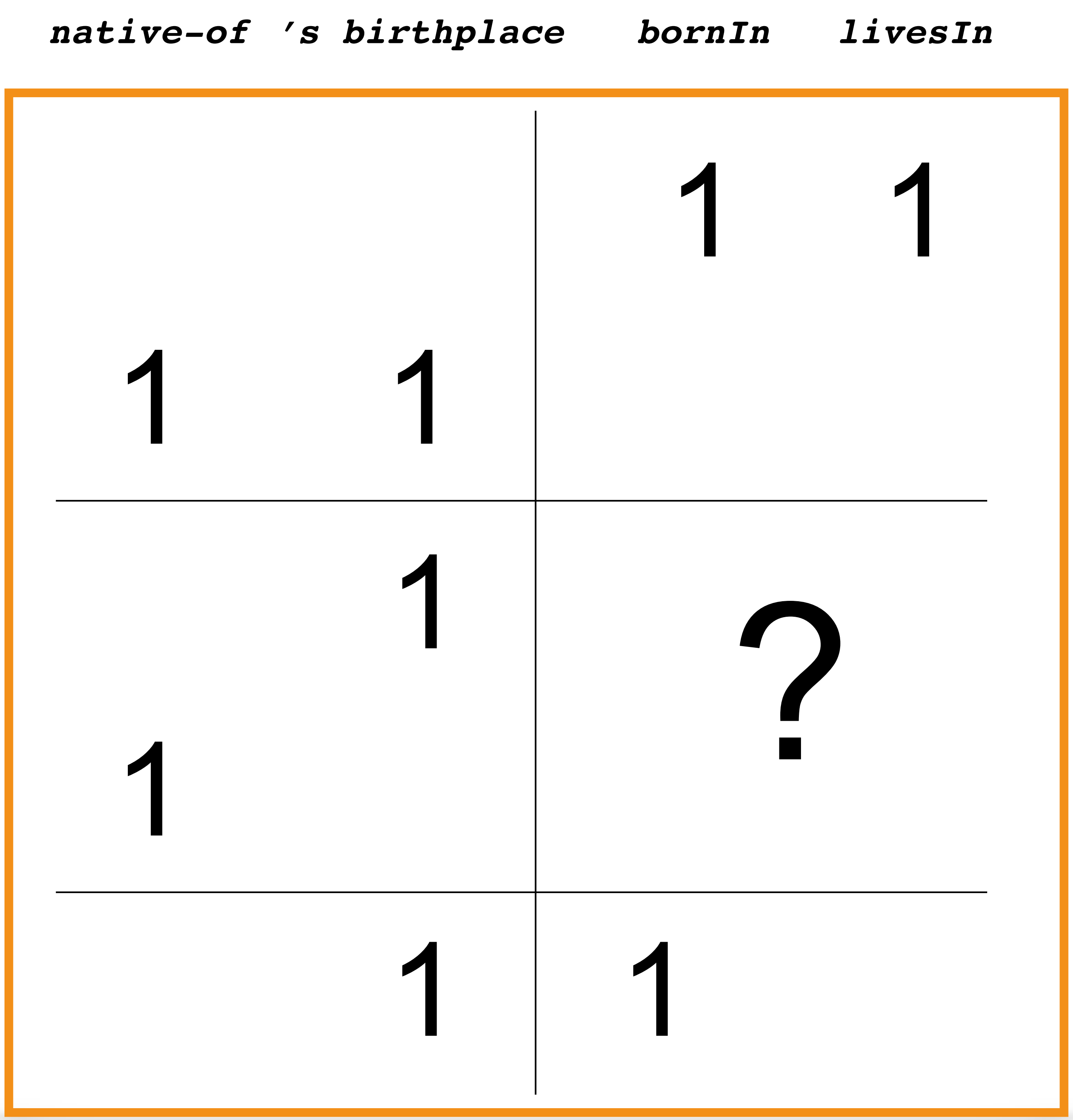


Learning Curve

[Rocktaeschel et al 15]



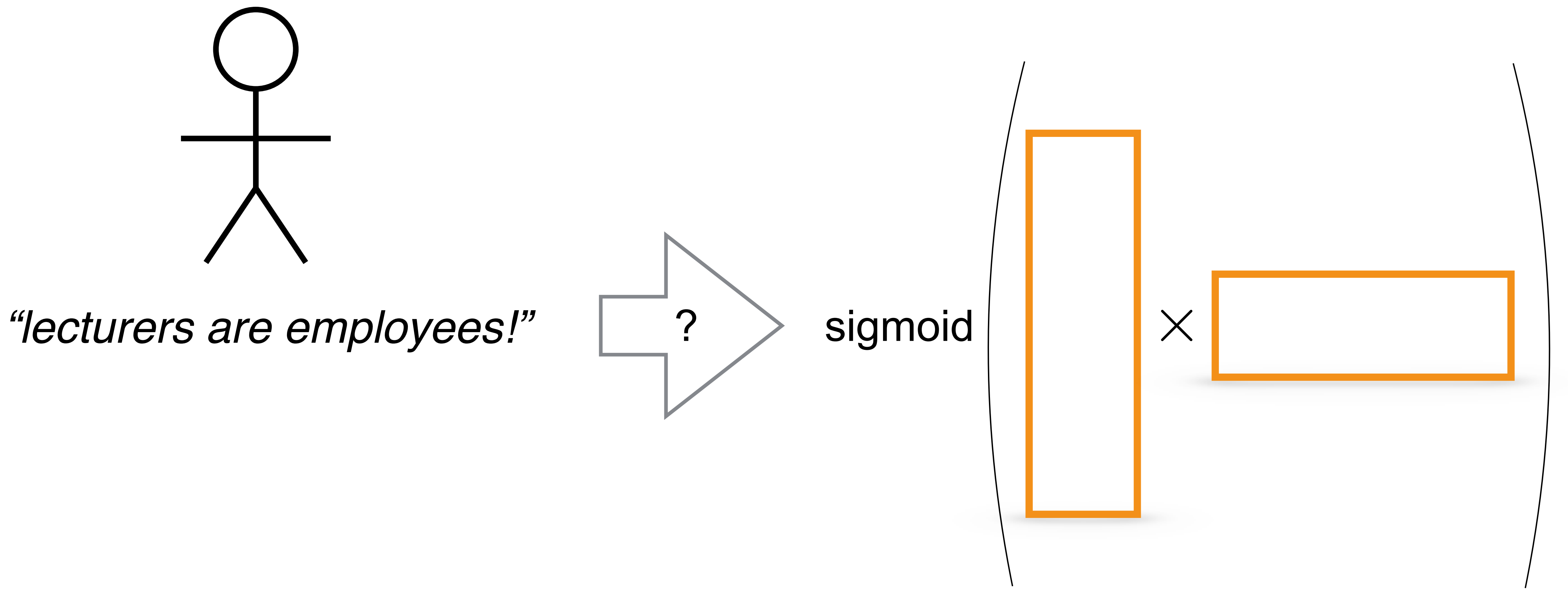
Generating Data?



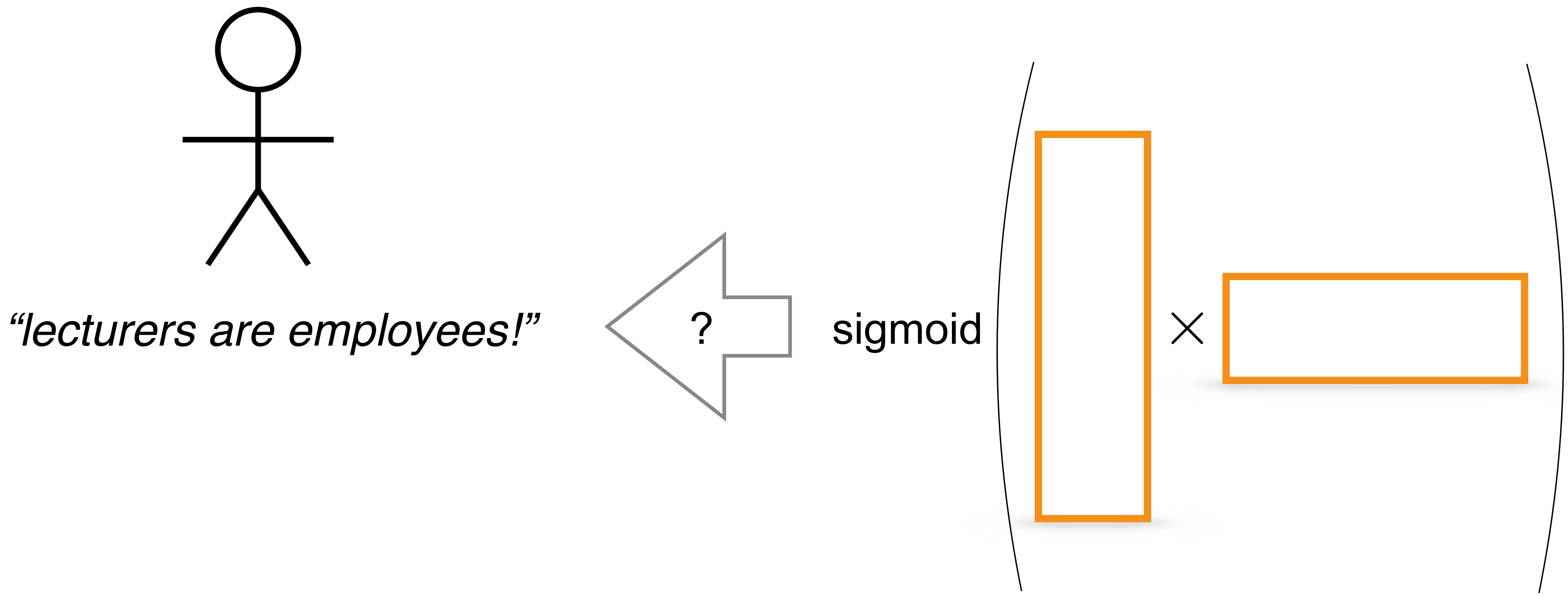
Hasn’t worked yet

- **Row embeddings overtrain**
- **At test time premise appears with other relations**

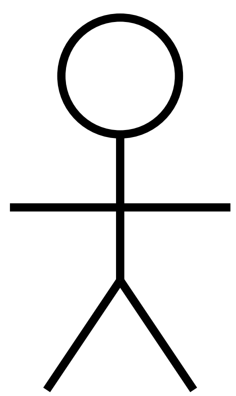
Challenge 1: Injecting Symbolic Rules



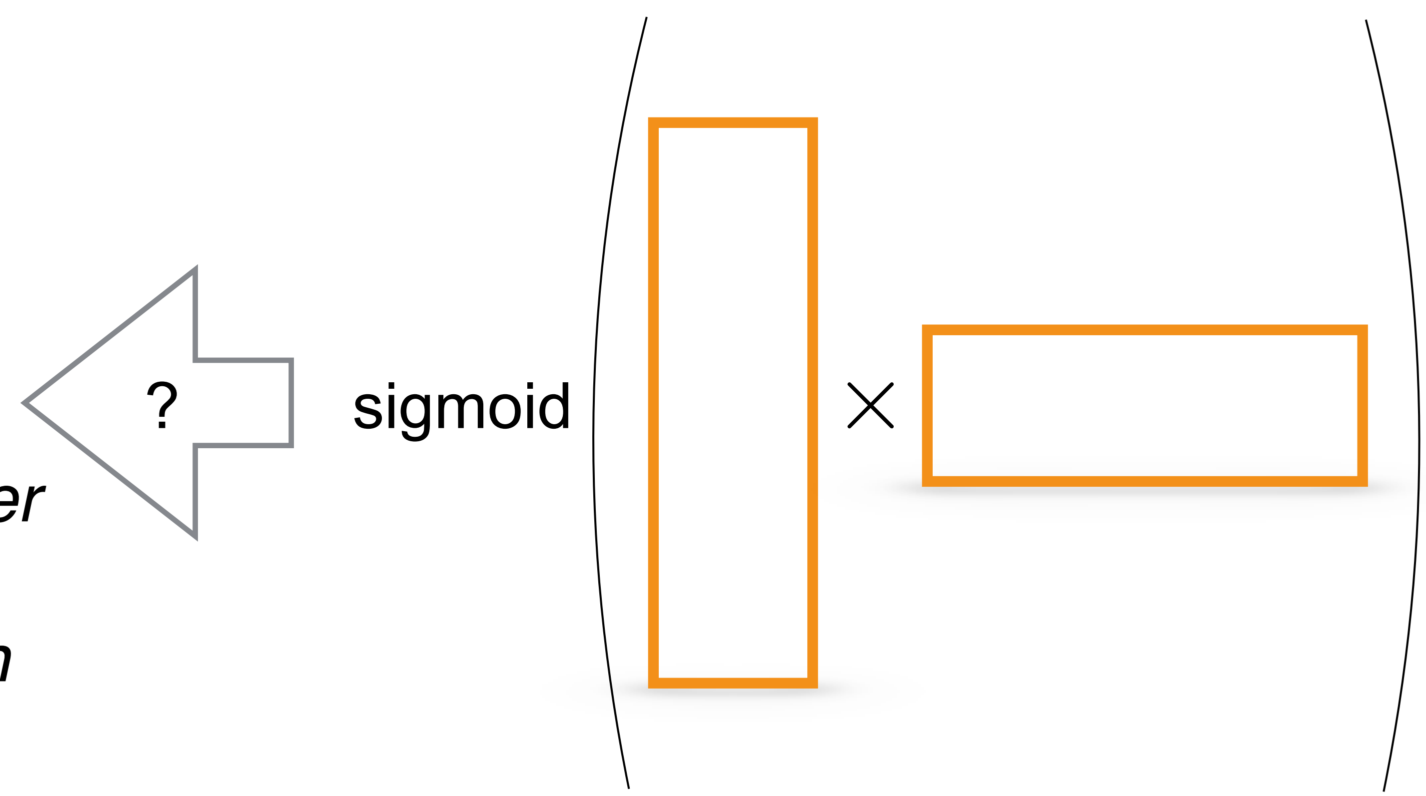
Challenge 2: Extracting Explanations



Challenge 2: Extracting Explanations



"I returned Sebastian because we know he is a lecturer at UCL, which is in London, so he most likely lives in London …



[Thrun 1995, NIPS, Craven 1996, NIPS]

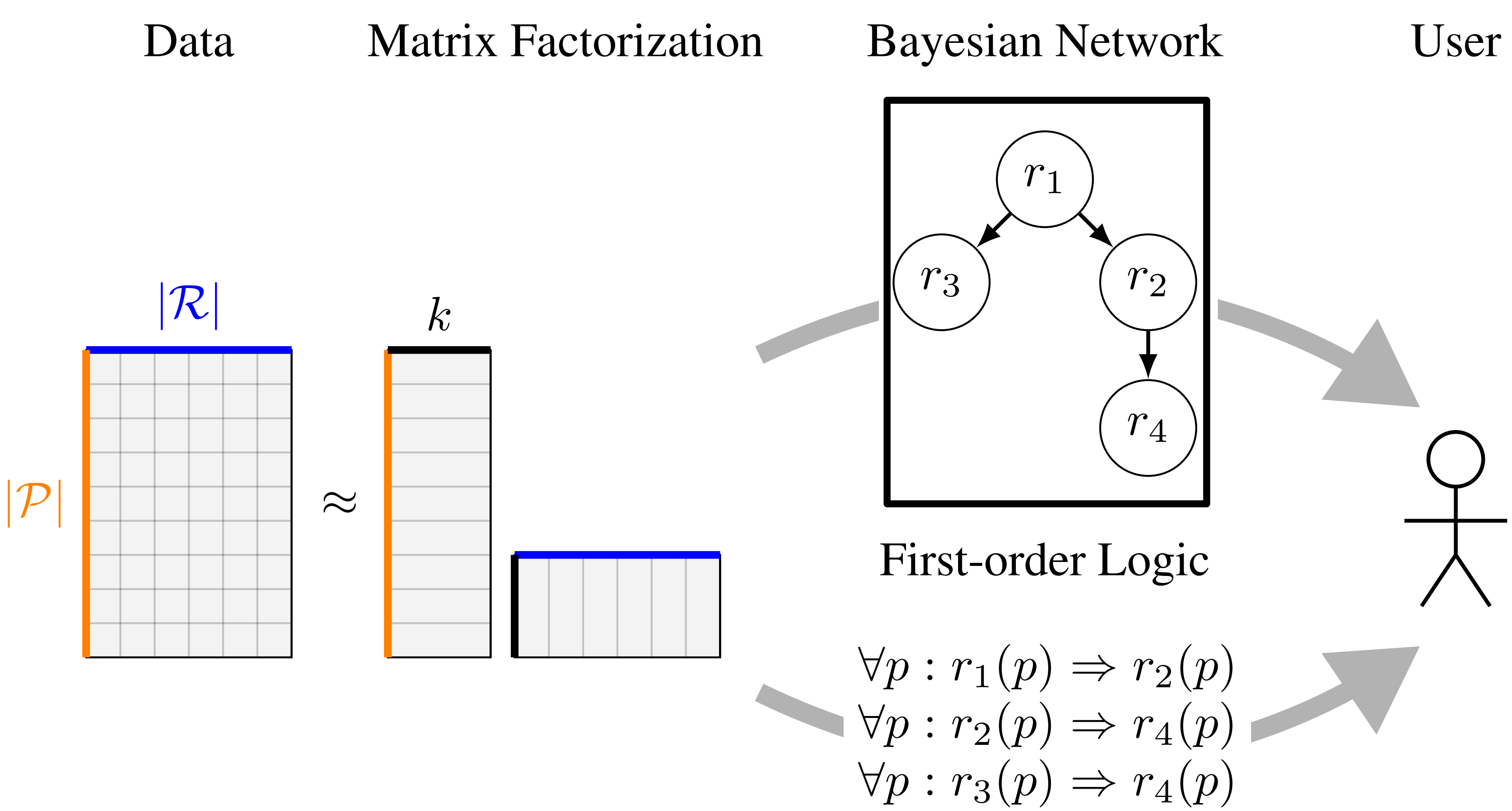
“Knowledge Extraction”

- Learn a more interpretable model from distributed representations (for interpretation, not for use)
  - Neural Networks => **if-then rules** (Thrun, 95)
  - Neural Networks => **Decision Trees** (Craven, 96)
- Open Questions
  - Go **beyond classification**: joint models
  - Use to provide **proofs** of complex predictions
  - Integrate into a **dialog** between human and machine

[Sanchez et al. 2015, KRR]

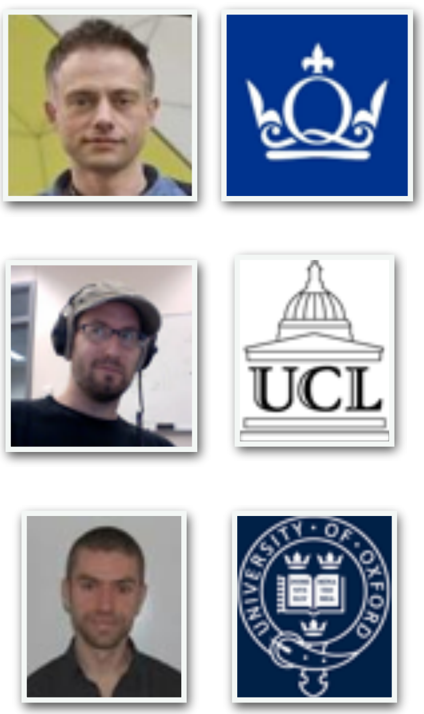
Explaining Matrix Factorization

[Sanchez et al. 2015, KRR]



Extracting Bayesian Networks

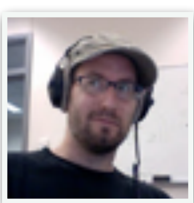
Learn Embeddings from Data



	X-is-professor-at-Y	*X-museum-at-Y*	*X-teaches-history-at-Y*	*employee(X,Y)*
	1	1		1
	1			
	1		1	

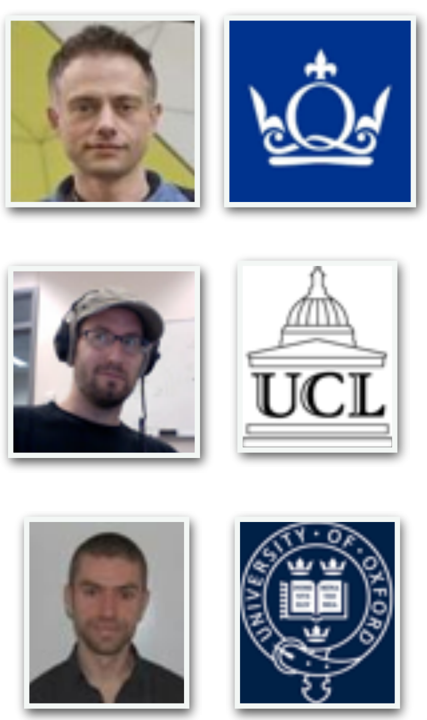
Extracting Bayesian Networks

Generate data from embeddings (threshold or sample)

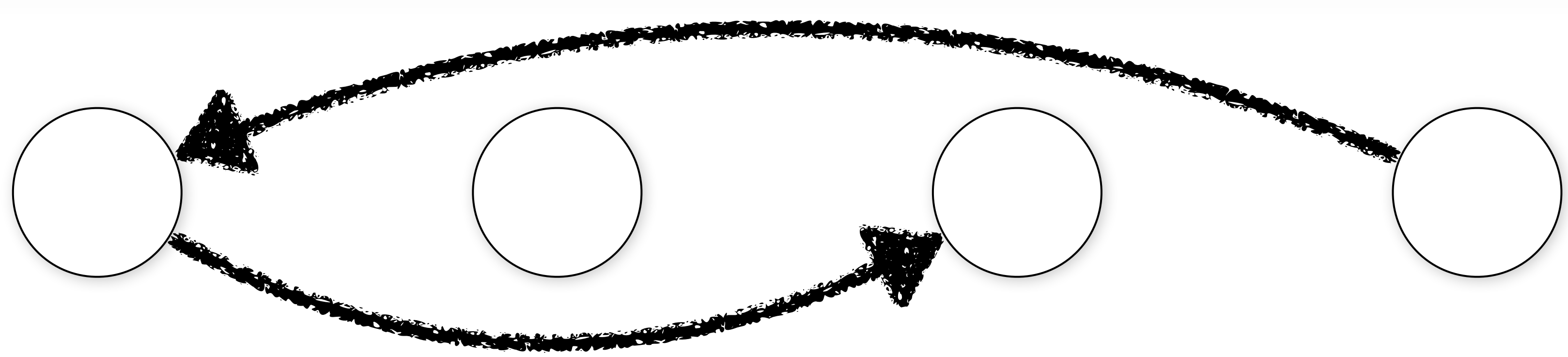
	X-*is-professor-at*-**Y**	**X**-*museum-at*-**Y**	**X**-*teaches-history-at*-**Y**	*employee*(**X**,**Y**)
	1	1	0	1
	1	0	0	1
	1	0	1	1

Extracting Bayesian Networks

Learning a tree shaped Bayesian Network



X-*is-professor-at*-**Y**	**X**-*museum-at*-**Y**	**X**-*teaches-history-at*-**Y**	*employee*(**X**,**Y**)
1	1	0	1
1	0	0	1
1	0	1	1

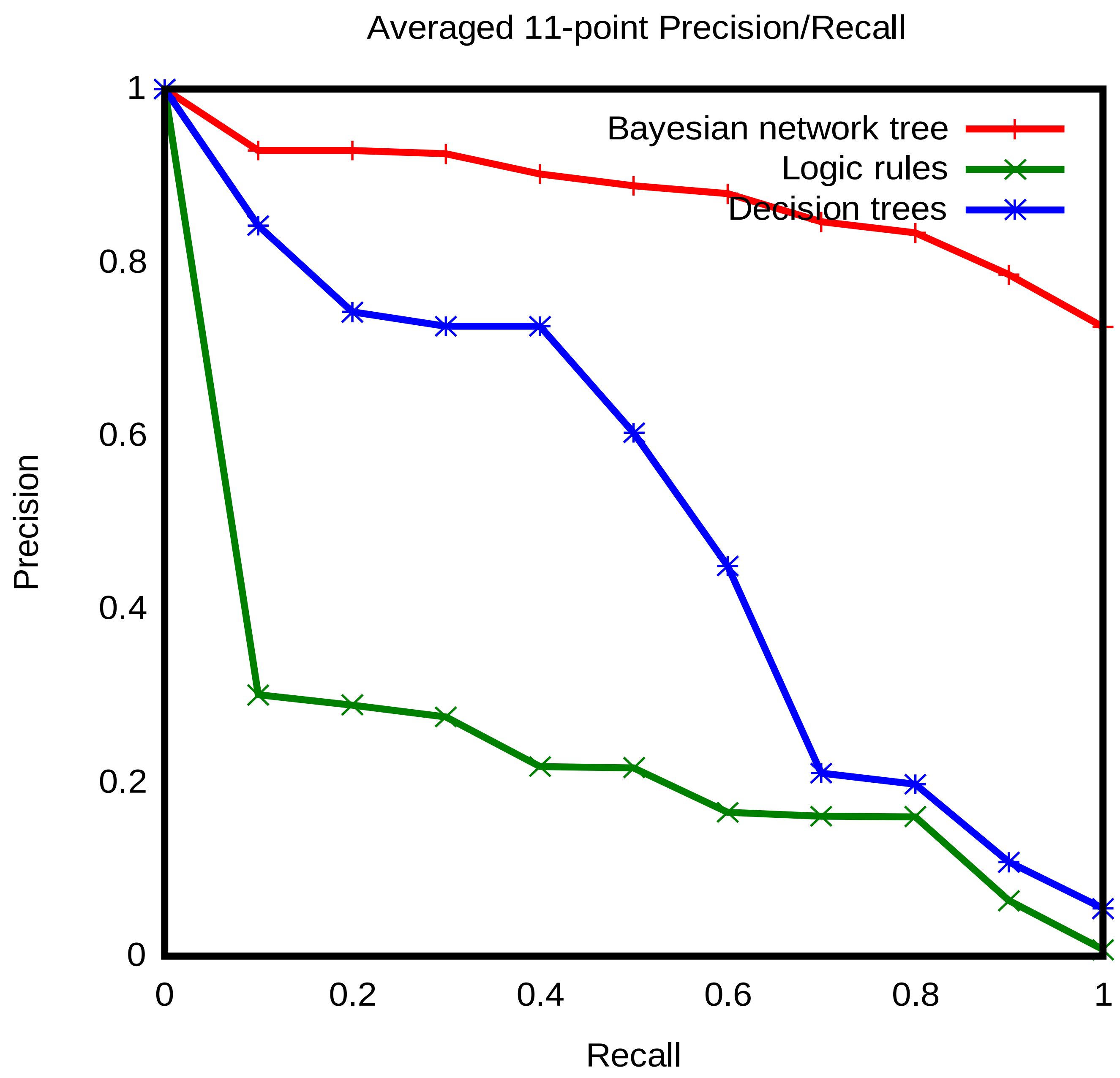


Benefits of Bayesian Network Trees

[Sanchez et al. 2015, KRR]

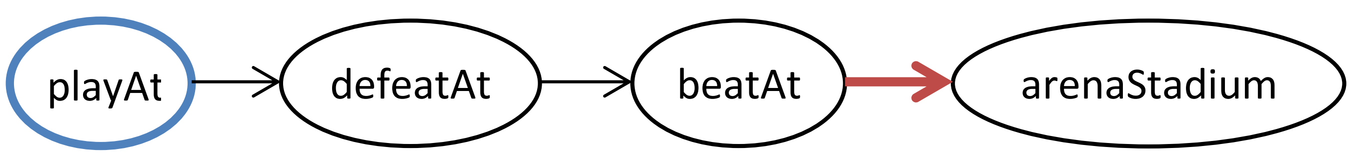
- Provide a **joint model** over all relations
  - more **compact** (than one decision tree per relation)
  - more **faithful** to the **joint** MF model
- **Probabilistic** interpretation, captures probabilistic nature of MF
- Very **scalable**
  - Learning: **Prim Algorithm** to find Maximum Spaning Tree over mutual information
  - Inference: Belief Propagation in **non-loopy** graph

Faithfulness



A "Proof"

- Model <u>observed</u>
  playAt(**Eagles**, **Canton**)
- Model wrongly <u>predicted</u>
  arenaStadium(**Eagles**, **Canton**)
- The Bayesian Network can provide this "proof"



- Todo: evaluate this in a downstream "debugging" task

Summary

- Do semantics in a **probabilistic relational reasoner**
- Reasoner: **matrix/tensor factorization** (or other LV models)
- Models itself don't need to be interpretable if we know …
- … how to **Interact with uninterpretable models**
  - **inject** explanations and logical rules
    - Approach: **optimize** embeddings to fulfil formulae
  - **extract** explanations
    - for example: by using an interpretable BN **proxy** model

Thanks

Training

Negative Data

Usually **unavailable** or **sparse**, so...

X-*is-historian-at*-**Y**	**X**-*is-professor-at*-**Y**	**X**-*museum-at*-**Y**	**X**-*teaches-history-at*-**Y**	*employee*(**X**,**Y**)
	1	1		1
1	1			
	1		1	
1				

Negative Data

…**subsample**, which **can** work...

X-*is-historian-at*-**Y**	**X**-*is-professor-at*-**Y**	**X**-*museum-at*-**Y**	**X**-*teaches-history-at*-**Y**	*employee*(**X**,**Y**)
	1	**1**		**1**
1	**1**	**0**		
	1		**1**	
1				

Negative Data

but often **does not**

X-*is-historian-at*-**Y**	**X**-*is-professor-at*-**Y**	**X**-*museum-at*-**Y**	**X**-*teaches-history-at*-**Y**	*employee*(**X**,**Y**)
	1	1		1
1	1			
0	1		1	
1				

Negative Data

and you need to sample a lot (wasting resources)

X-*is-historian-at*-**Y**	**X**-*is-professor-at*-**Y**	**X**-*museum-at*-**Y**	**X**-*teaches-history-at*-**Y**	*employee*(**X**,**Y**)
0	1	1	0	1
1	1	0		0
0	1	0	1	
1	0		0	

Implicit Feedback

Often users only **click/view/buy** items, or not, but **no rating**

User 1	User 2	User 3	User 4	User 5	
	1	**1**		**1**	Item 1
1	**1**				Item 2
	1		**1**		Item 3
1					Item 4

Ranking

[Rendle et al.,09]

for all (**o**bserved,**n**ot observed) pairs in column: prob(o) > prob(n)

X-is-historian-at-Y	*X-is-professor-at-Y*	*X-museum-at-Y*	*X-teaches-history-at-Y*	*employee(X,Y)*
	1	1		1
0.9	1			
0.95	1		1	
1				

Ranking

[Rendle et al.,09]

for all (**o**bserved,**n**ot observed) pairs in a column: prob(o) > prob(n)

X-*is-historian-at*-**Y**	**X**-*is-professor-at*-**Y**	**X**-*museum-at*-**Y**	**X**-*teaches-history-at*-**Y**	*employee*(**X**,**Y**)
	1	**1**		**1**
0.9 $\vee$	**1**			
0.85	**1**		**1**	
1				

Training: Stochastic Gradient Descent

[Rendle et al.,09]

Sample observed fact...

X-*is-historian-at*-**Y**	**X**-*is-professor-at*-**Y**	**X**-*museum-at*-**Y**	**X**-*teaches-history-at*-**Y**	*employee*(**X**,**Y**)
	1	**1**		**1**
1	**1**			
	1		**1**	
1				

Training: Stochastic Gradient Descent

[Rendle et al.,09]

Sample unobserved cell for same relation

X-*is-historian-at*-**Y**	**X**-*is-professor-at*-**Y**	**X**-*museum-at*-**Y**	**X**-*teaches-history-at*-**Y**	*employee*(**X**,**Y**)
	1	1		1
1	1			
	1		1	
1				

Training: Stochastic Gradient Descent

[Rendle et al.,09]

Estimate current beliefs and gradient, **update** parameters accordingly

X-*is-historian-at*-**Y**	**X**-*is-professor-at*-**Y**	**X**-*museum-at*-**Y**	**X**-*teaches-history-at*-**Y**	*employee*(**X**,**Y**)
	0.8	1		1
1	1			
	1		1	
1	0.9			

Training: Stochastic Gradient Descent

[Rendle et al.,09]

Estimate current beliefs and gradient, **update** parameters accordingly

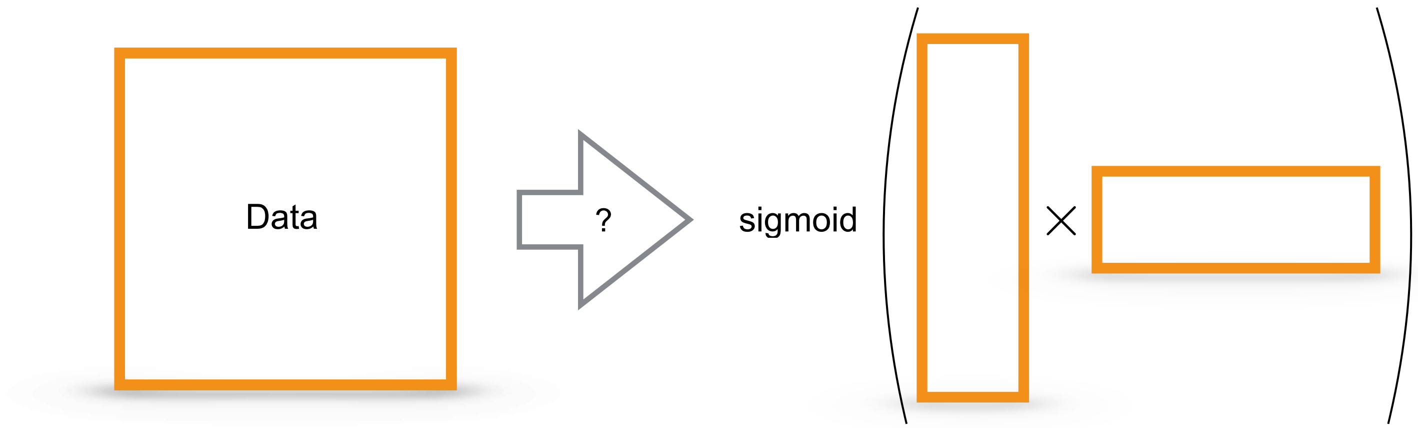
X-*is-historian-at*-**Y**	**X**-*is-professor-at*-**Y**	**X**-*museum-at*-**Y**	**X**-*teaches-history-at*-**Y**	*employee*(**X**,**Y**)
	0.85	1		1
1	1			
	1		1	
1	0.8			

How can we do this?

<table>
<tr><th>native-of</th><th>'s birthplace</th><th>bornIn</th><th>livesIn</th></tr>
<tr><td></td><td></td><td>1</td><td>1</td></tr>
<tr><td>1</td><td>1</td><td></td><td></td></tr>
<tr><td></td><td>1</td><td rowspan="2" colspan="2">?</td></tr>
<tr><td>1</td><td></td></tr>
</table>

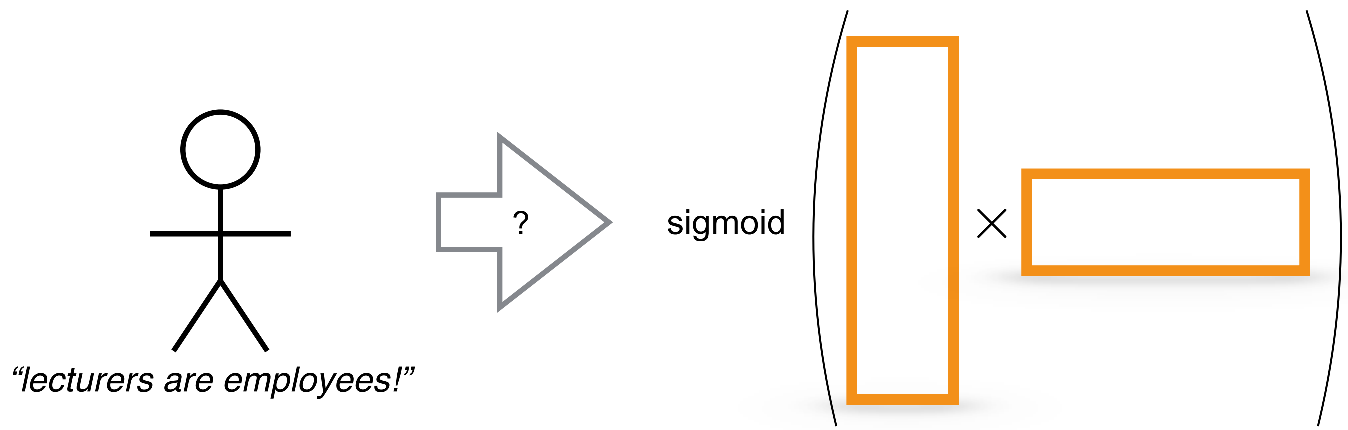
birthplace(x,y) => bornIn(x,y)

Overview: Embeddings and …

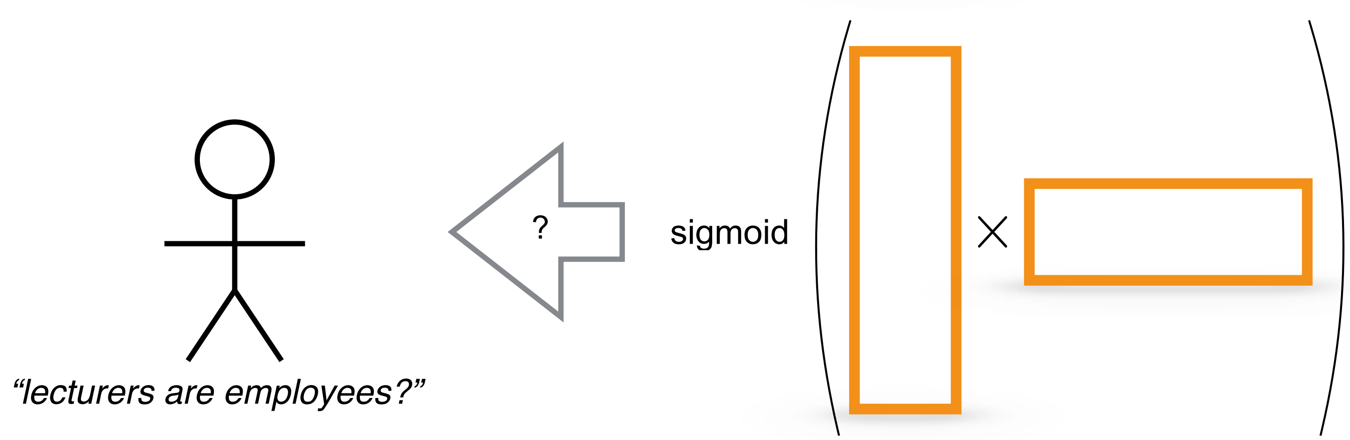


- **Learning** from Data
  [NAACL 2013]

“Talking to Uninterpretable Models”



- **Injecting** Knowledge
  [SP 2014, NAACL 2015]



- **Extracting** Knowledge
  [KRR 2015]