Querying Factorized Probabilistic Triple Databases

Denis Krompaß[1], Maximilian Nickel[2] and Volker Tresp[1,3]
[1] Department of Computer Science. Ludwig Maximilian University,
[2] MIT, Cambridge and Istituto Italiano di Tecnologia
[3] Corporate Technology, Siemens AG

Outline

1. Introduction and Motivation
   - Knowledge Bases are Triple Stores
   - Exploiting Uncertainty in KBs

2. Constructing Probabilistic Knowledge Bases with the RESCAL Tensor-Factorization

3. Complex Querying of Factorized Probabilistic Knowledge Base Representations

Knowledge Bases are Triple Stores

- Represents facts of the world in a machine readable form

Knowledge Bases are Triple Stores

- Represents facts of the world in a machine readable form

SUBJECT **PREDICATE** **OBJECT**

+

- Machine Readable
- Utilize Background Knowledge in Applications
- Provide Additional Information (Search)

Knowledge Bases are Triple Stores

- Represents facts of the world in a machine readable form

SUBJECT **PREDICATE** **OBJECT**

+

- Machine Readable
- Utilize Background Knowledge in Applications
- Provide Additional Information (Search)

-

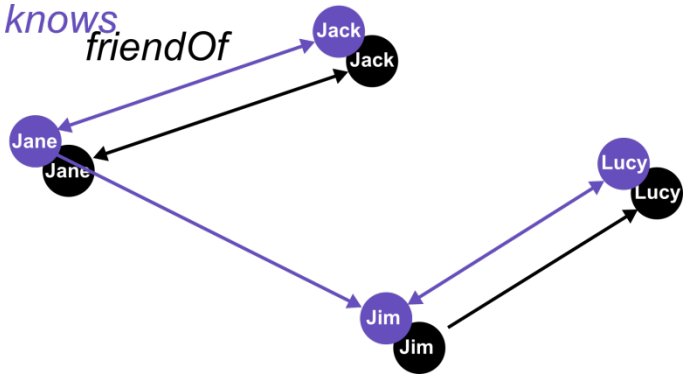
- Many False Facts
- Incomplete
- Only few attempts to represent triple-uncertainty

Benefits of Exploiting Uncertainty in Knowledge Bases

Deterministic Database

Subject	Object
Jack	Jane
Jane	Jim
Jim	Lucy

Subject	Object
Jack	Jane
Jim	Lucy

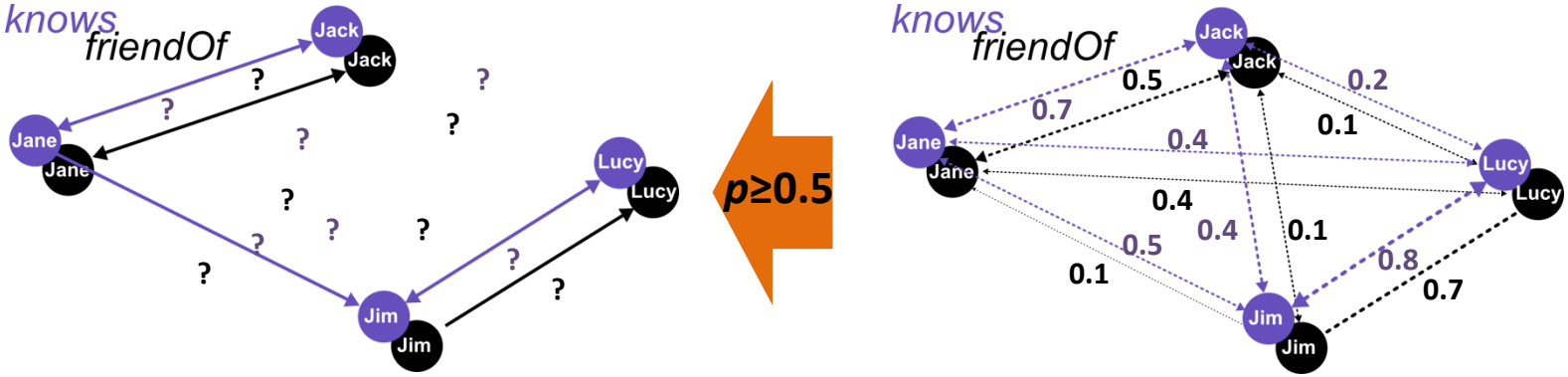


Benefits of Exploiting Uncertainty in Knowledge Bases

Deterministic Database

Subject	Object
Jack	Jane
Jane	Jim
Jim	Lucy

Subject	Object
Jack	Jane
Jim	Lucy



Benefits of Exploiting Uncertainty in Knowledge Bases

Deterministic Database

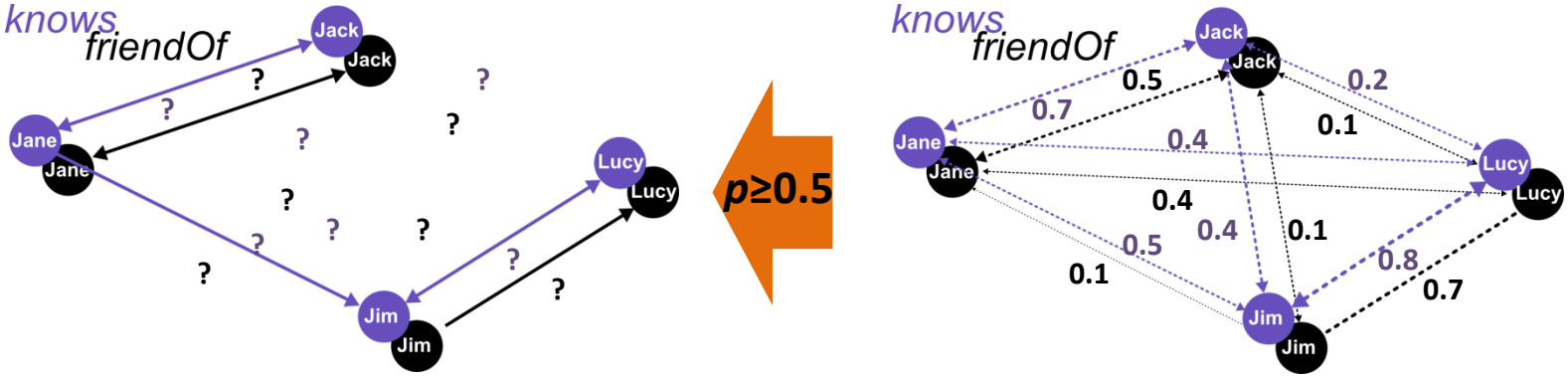
Subject	Object
Jack	Jane
Jane	Jim
Jim	Lucy

Subject	Object
Jack	Jane
Jim	Lucy

Probabilistic Tuple-Independent Database

Subject	Object	P	RndVar
Jack	Jane	0.7	Y_1
Jack	Lucy	0.2	Y_2
Jack	Jim	0.4	Y_3
Jane	Lucy	0.4	Y_4
…	…	…	…

Subject	Object	P	RndVar
Jack	Jane	0.5	X_1
Jack	Lucy	0.1	X_2
Jack	Jim	0.1	X_3
Jane	Lucy	0.4	X_4
…	…	…	…



Benefits of Exploiting Uncertainty in Knowledge Bases

Deterministic Database

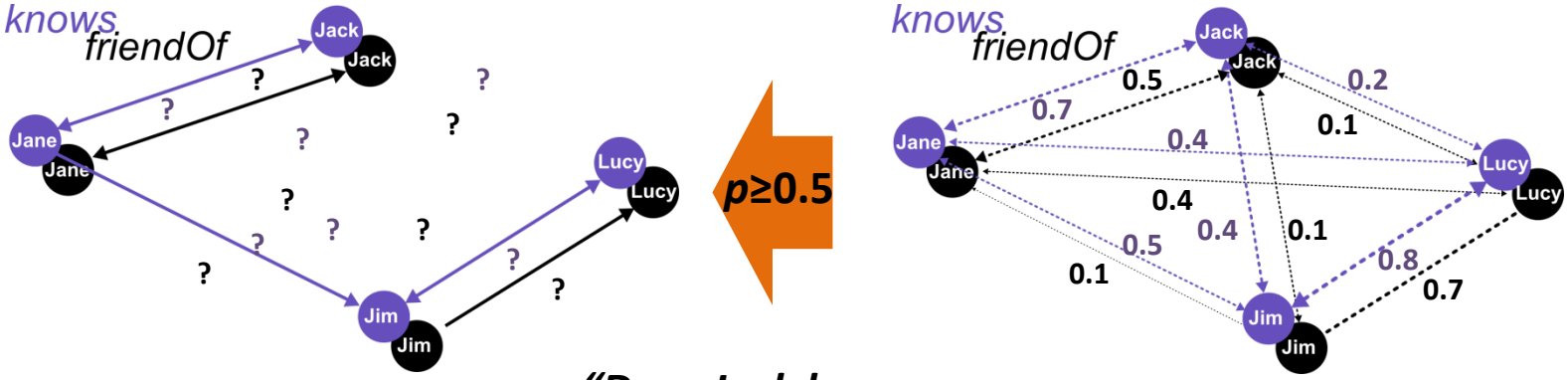
Subject	Object
Jack	Jane
Jane	Jim
Jim	Lucy

Subject	Object
Jack	Jane
Jim	Lucy

Probabilistic Tuple-Independent Database

Subject	Object	P	RndVar
Jack	Jane	0.7	Y_1
Jack	Lucy	0.2	Y_2
Jack	Jim	0.4	Y_3
Jane	Lucy	0.4	Y_4
…	…	…	…

Subject	Object	P	RndVar
Jack	Jane	0.5	X_1
Jack	Lucy	0.1	X_2
Jack	Jim	0.1	X_3
Jane	Lucy	0.4	X_4
…	…	…	…



"Does Jack know someone who is a friend of Lucy?"

$$Q :- \exists y.\, knows(\text{'Jack'}, y),\ friendOf(y, \text{'Lucy'})$$

Benefits of Exploiting Uncertainty in Knowledge Bases

Deterministic Database

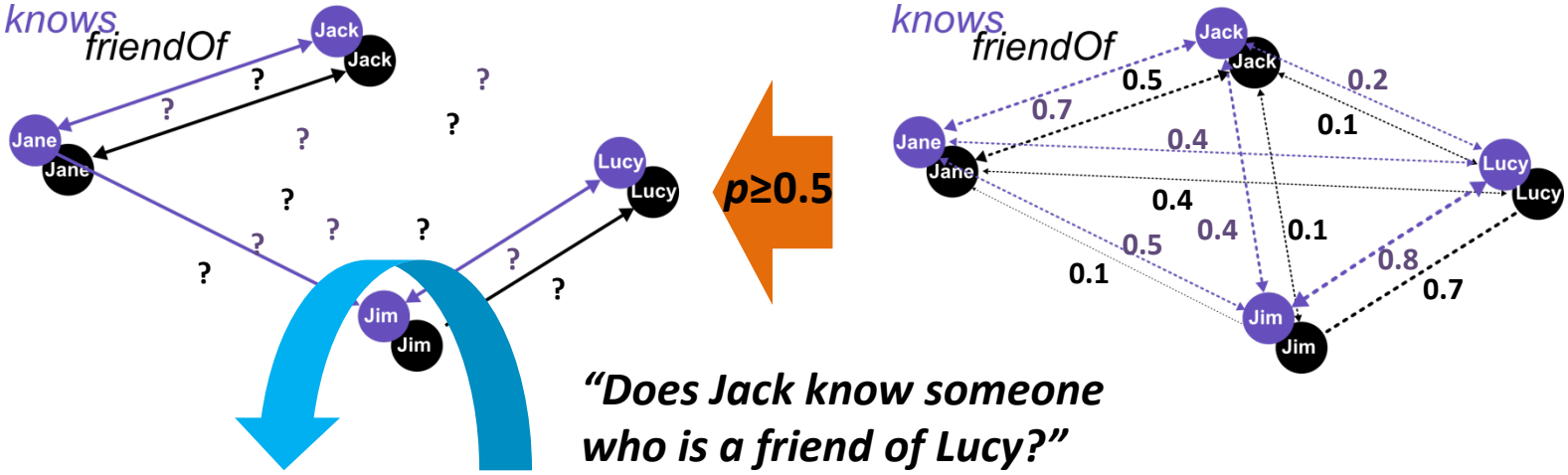
Subject	Object
Jack	Jane
Jane	Jim
Jim	Lucy

Subject	Object
Jack	Jane
Jim	Lucy

Probabilistic Tuple-Independent Database

Subject	Object	P	RndVar
Jack	Jane	0.7	Y_1
Jack	Lucy	0.2	Y_2
Jack	Jim	0.4	Y_3
Jane	Lucy	0.4	Y_4
…	…	…	…

Subject	Object	P	RndVar
Jack	Jane	0.5	X_1
Jack	Lucy	0.1	X_2
Jack	Jim	0.1	X_3
Jane	Lucy	0.4	X_4
…	…	…	…



"Does Jack know someone who is a friend of Lucy?"

$$Q :- \exists y.\ knows(\text{'Jack'}, y),\ friendOf(y, \text{'Lucy'})$$

"No, Jack does not know such a person."

Benefits of Exploiting Uncertainty in Knowledge Bases

Deterministic Database

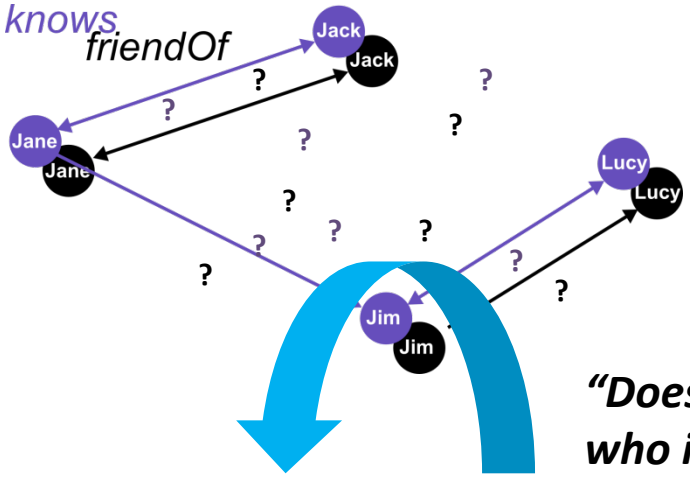
Subject	Object
Jack	Jane
Jane	Jim
Jim	Lucy

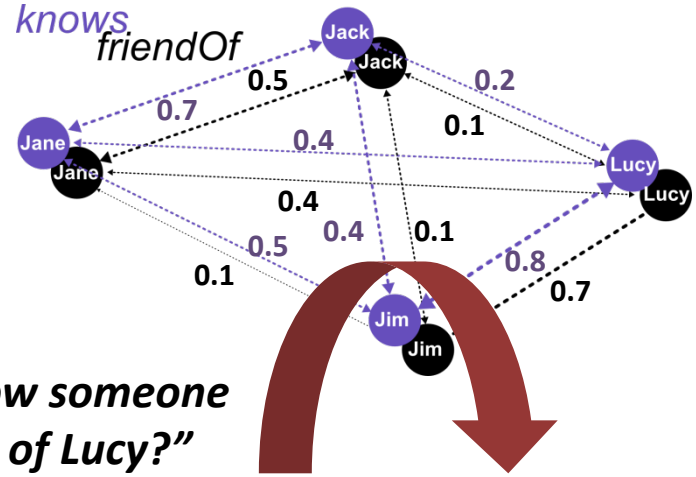
Subject	Object
Jack	Jane
Jim	Lucy

Probabilistic Tuple-Independent Database

Subject	Object	P	RndVar
Jack	Jane	0.7	Y_1
Jack	Lucy	0.2	Y_2
Jack	Jim	0.4	Y_3
Jane	Lucy	0.4	Y_4
…	…	…	…

Subject	Object	P	RndVar
Jack	Jane	0.5	X_1
Jack	Lucy	0.1	X_2
Jack	Jim	0.1	X_3
Jane	Lucy	0.4	X_4
…	…	…	…





"Does Jack know someone who is a friend of Lucy?"

"No, Jack does not know such a person."

$$Q :- \exists y.\, knows(\text{'Jack'}, y),\ friendOf(y, \text{'Lucy'})$$

*"Jack might know such a person with a probability of 63%."**

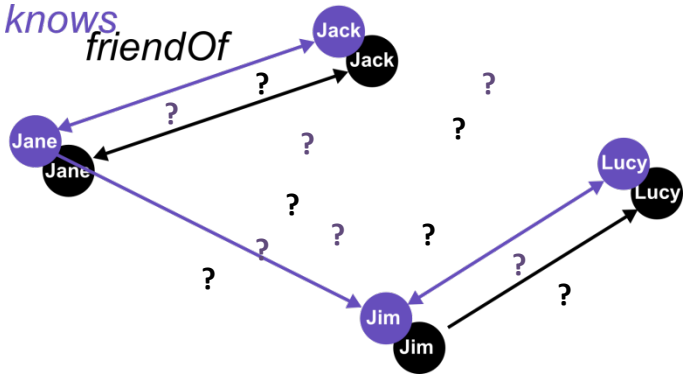
**Considering that a person knows and is a friend of itself*

Challenges when Exploiting Uncertainty in Knowledge Bases

Deterministic Database

Subject	Object
Jack	Jane
Jane	Jim
Jim	Lucy

Subject	Object
Jack	Jane
Jim	Lucy

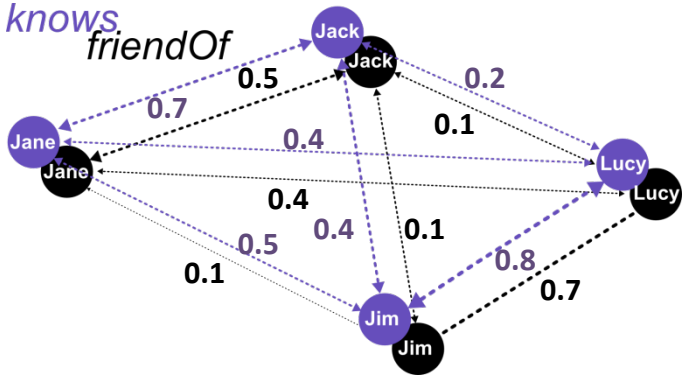


Challenges when Exploiting Uncertainty in Knowledge Bases

Probabilistic
Tuple-Independent Database

Subject	Object	P	RndVar
Jack	Jane	0.7	Y_1
Jack	Lucy	0.2	Y_2
Jack	Jim	0.4	Y_3
Jane	Lucy	0.4	Y_4
…	…	…	…

Subject	Object	P	RndVar
Jack	Jane	0.5	X_1
Jack	Lucy	0.1	X_2
Jack	Jim	0.1	X_3
Jane	Lucy	0.4	X_4
…	…	…	…



Reintroducing Uncertainty

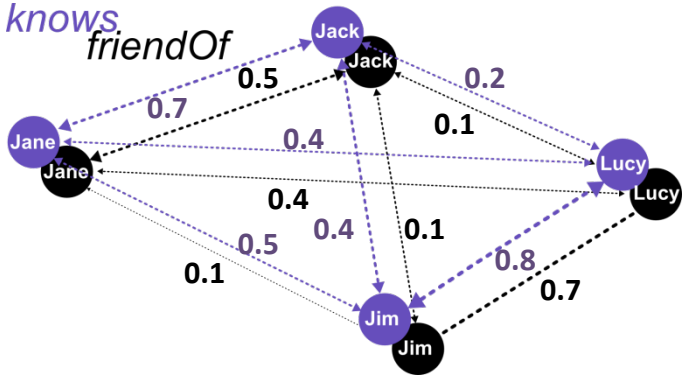
- Can be **intractable** for larger KBs
  - Millions of Entities
  - Thousands of Relation Types
  - Intractable many possible triples

Challenges when Exploiting Uncertainty in Knowledge Bases

Probabilistic Tuple-Independent Database

Subject	Object	P	RndVar
Jack	Jane	0.7	Y_1
Jack	Lucy	0.2	Y_2
Jack	Jim	0.4	Y_3
Jane	Lucy	0.4	Y_4
…	…	…	…

Subject	Object	P	RndVar
Jack	Jane	0.5	X_1
Jack	Lucy	0.1	X_2
Jack	Jim	0.1	X_3
Jane	Lucy	0.4	X_4
…	…	…	…



Reintroducing Uncertainty

- Can be **intractable** for larger KBs
  - Millions of Entities
  - Thousands of Relation Types
  - Intractable many possible triples

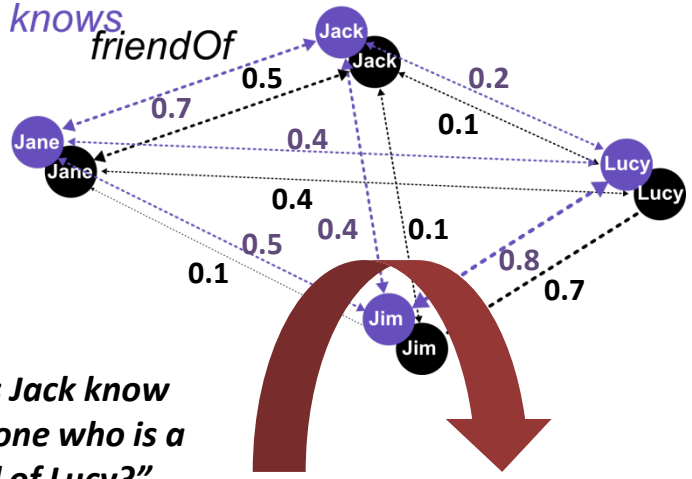
RESCAL

Challenges when Exploiting Uncertainty in Knowledge Bases

Probabilistic Tuple-Independent Database

Subject	Object	P	RndVar
Jack	Jane	0.7	Y_1
Jack	Lucy	0.2	Y_2
Jack	Jim	0.4	Y_3
Jane	Lucy	0.4	Y_4
…	…	…	…

Subject	Object	P	RndVar
Jack	Jane	0.5	X_1
Jack	Lucy	0.1	X_2
Jack	Jim	0.1	X_3
Jane	Lucy	0.4	X_4
…	…	…	…



"Does Jack know someone who is a friend of Lucy?"

*"Jack might know such a person with a probability of 63%."**

**Considering that a person knows and is a friend of itself*

Reintroducing Uncertainty

- Can be **intractable** for larger KBs
  - Millions of Entities
  - Thousands of Relation Types
  - Intractable many possible triples

RESCAL

Complex Querying

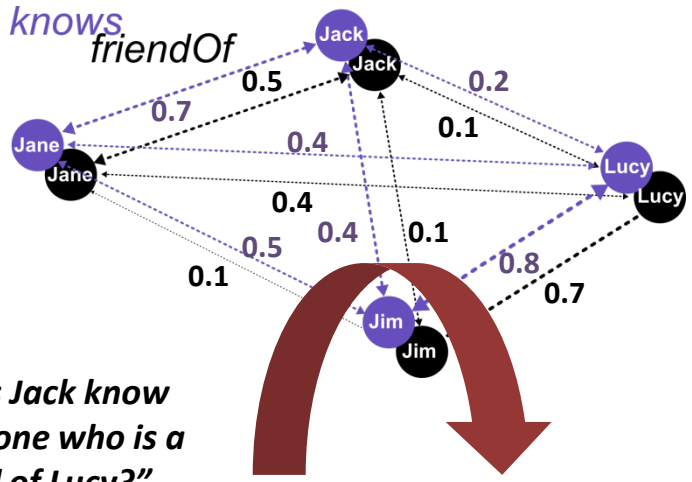
- ***Unsafe queries*** can be **intractable** due to exponential complexity **(#P)**
- ***Safe queries***
  - polynomial complexity
  - **can lead to long query processing times.**

Challenges when Exploiting Uncertainty in Knowledge Bases

Probabilistic Tuple-Independent Database

Subject	Object	P	RndVar
Jack	Jane	0.7	Y_1
Jack	Lucy	0.2	Y_2
Jack	Jim	0.4	Y_3
Jane	Lucy	0.4	Y_4
…	…	…	…

Subject	Object	P	RndVar
Jack	Jane	0.5	X_1
Jack	Lucy	0.1	X_2
Jack	Jim	0.1	X_3
Jane	Lucy	0.4	X_4
…	…	…	…



"Does Jack know someone who is a friend of Lucy?"

*"Jack might know such a person with a probability of 63%."**

**Considering that a person knows and is a friend of itself*

Reintroducing Uncertainty

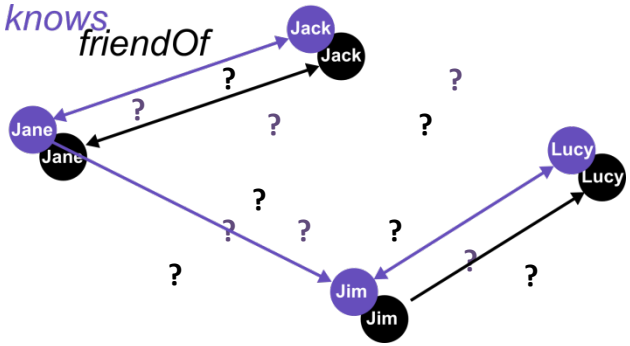
- Can be **intractable** for larger KBs
  - Millions of Entities
  - Thousands of Relation Types
  - Intractable many possible triples

RESCAL

Complex Querying

- ***Unsafe queries*** can be **intractable** due to exponential complexity **(#P)**
- ***Safe queries***
  - polynomial complexity
  - **can lead to long query processing times.**

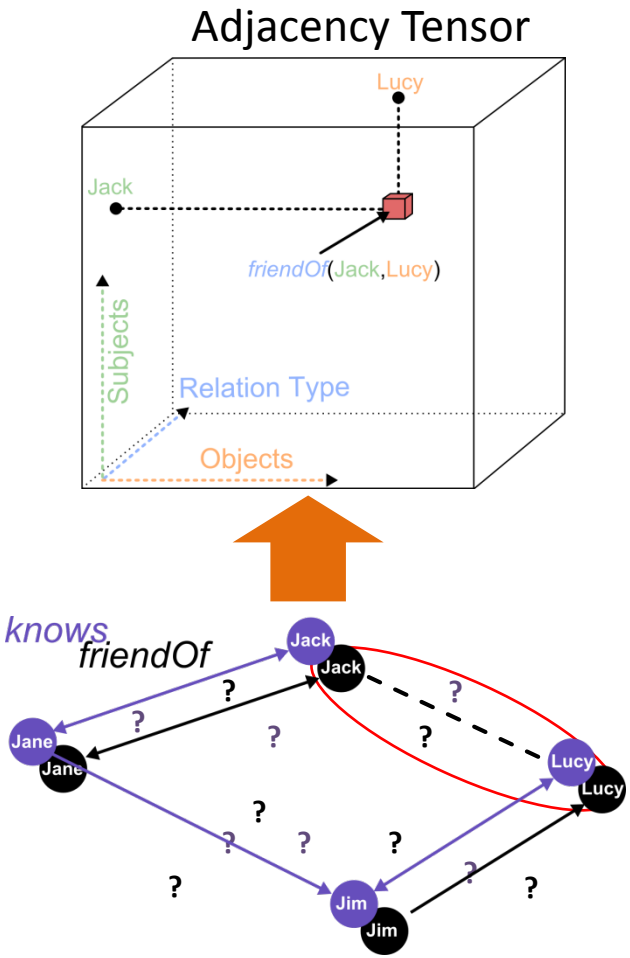
Probabilistic KB Construction with RESCAL



Subject	Object
Jack	Jane
Jane	Jim
Jim	Lucy

Subject	Object
Jack	Jane
Jim	Lucy

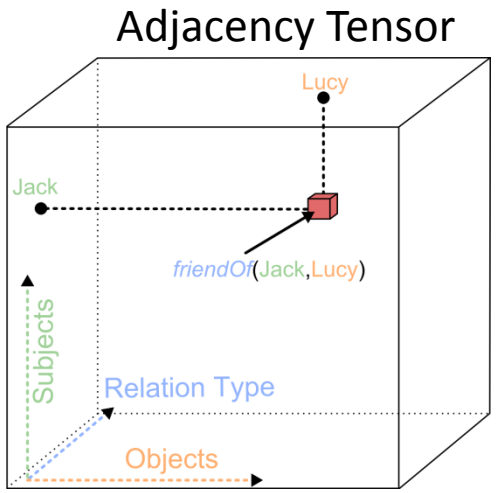
Probabilistic KB Construction with RESCAL

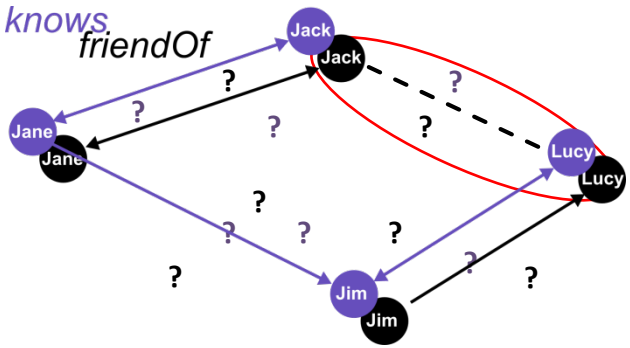


Subject	Object
Jack	Jane
Jane	Jim
Jim	Lucy

Subject	Object
Jack	Jane
Jim	Lucy

Probabilistic KB Construction with RESCAL

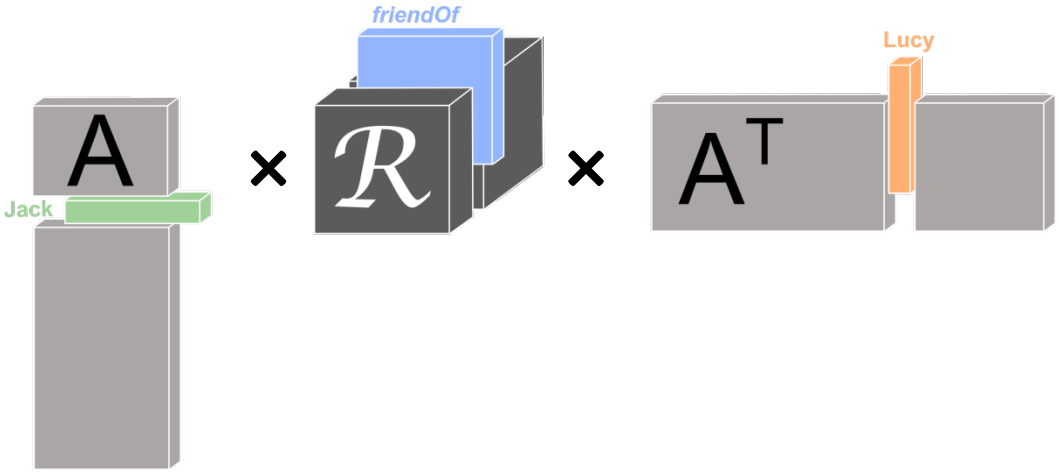




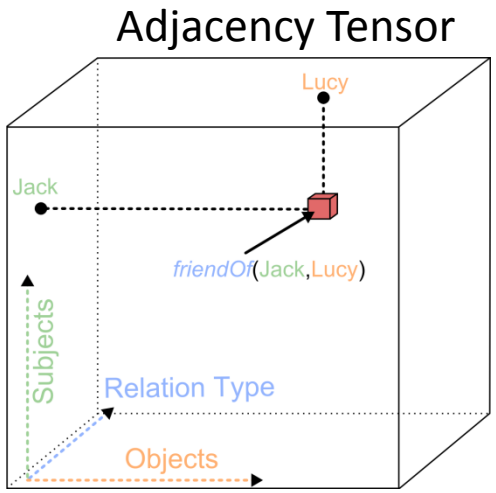
Subject	Object
Jack	Jane
Jane	Jim
Jim	Lucy

Subject	Object
Jack	Jane
Jim	Lucy

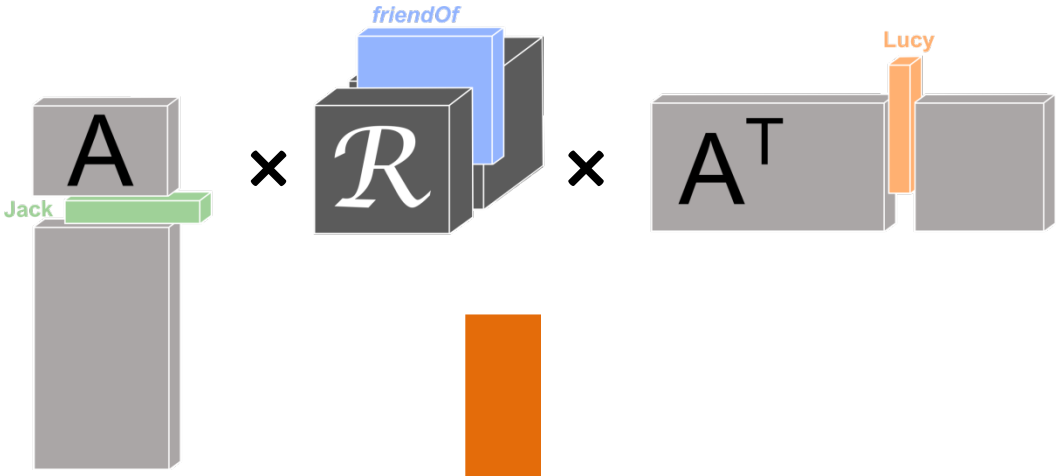
≈

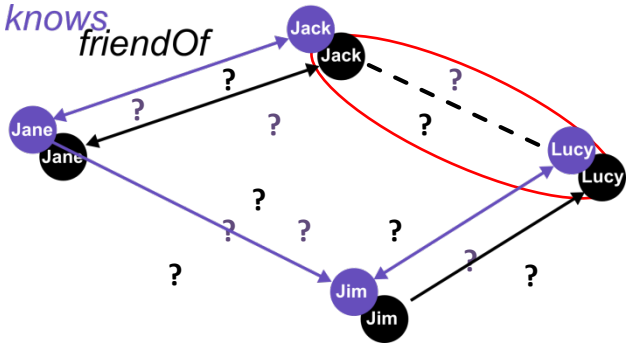


Probabilistic KB Construction with RESCAL



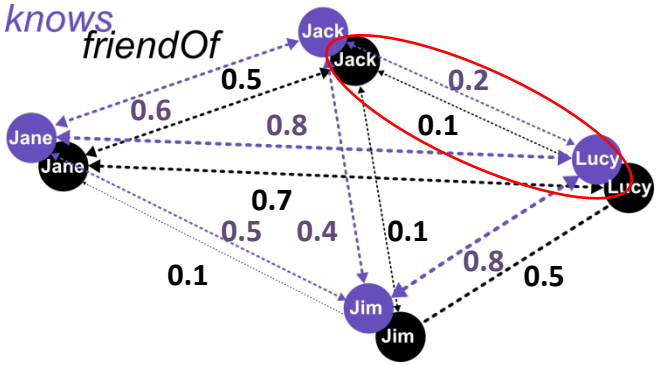
≈





Subject	Object
Jack	Jane
Jane	Jim
Jim	Lucy

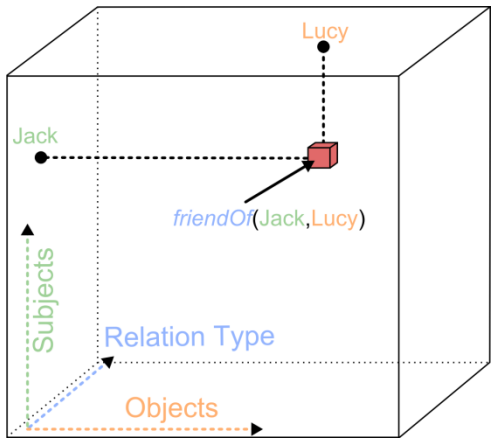
Subject	Object
Jack	Jane
Jim	Lucy



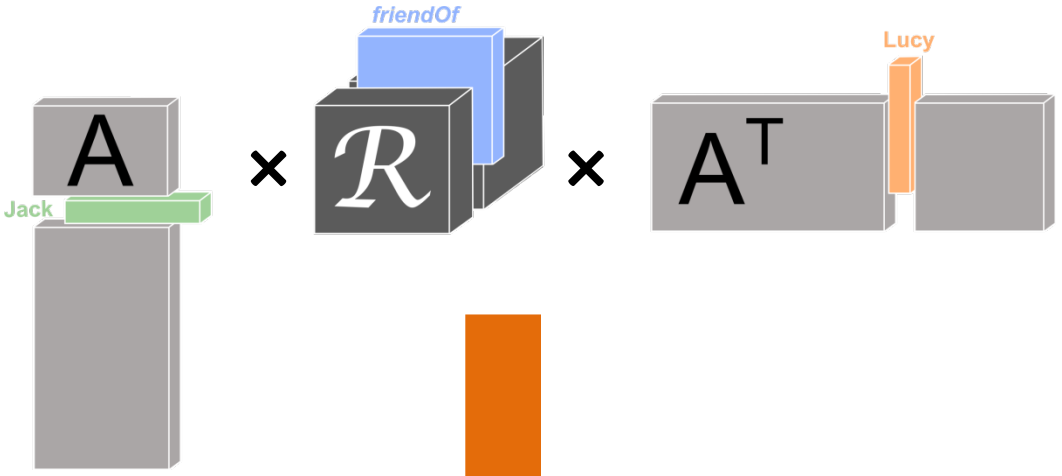
Subject	Object	P	RndVar
Jack	Jane	0.6	Y_1
Jack	Lucy	0.2	Y_2
Jack	Jim	0.4	Y_3
Jane	Lucy	0.8	Y_4
…	…	…	…

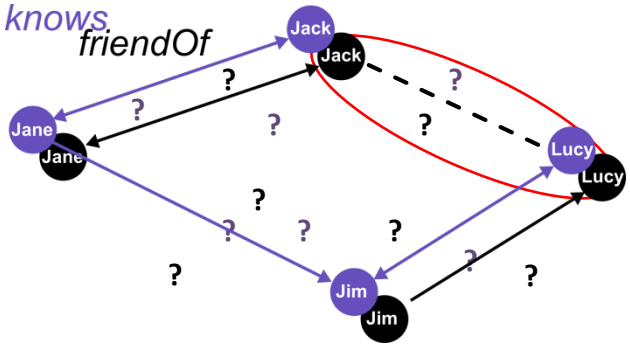
Subject	Object	P	RndVar
Jack	Jane	0.5	X_1
Jack	Lucy	0.1	X_2
Jack	Jim	0.1	X_3
Jane	Lucy	0.7	X_4
…	…	…	…

Probabilistic KB Construction with RESCAL



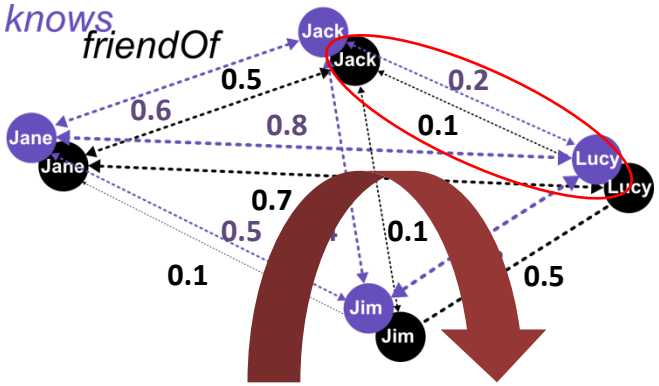
≈





Subject	Object
Jack	Jane
Jane	Jim
Jim	Lucy

Subject	Object
Jack	Jane
Jim	Lucy



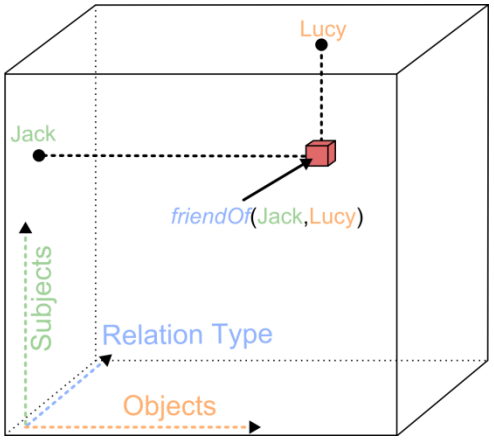
Subject	Object	P	RndVar
Jack	Jane	0.6	Y_1
Jack	Lucy	0.2	Y_2
Jack	Jim	0.4	Y_3
Jane	Lucy	0.8	Y_4
…	…	…	…

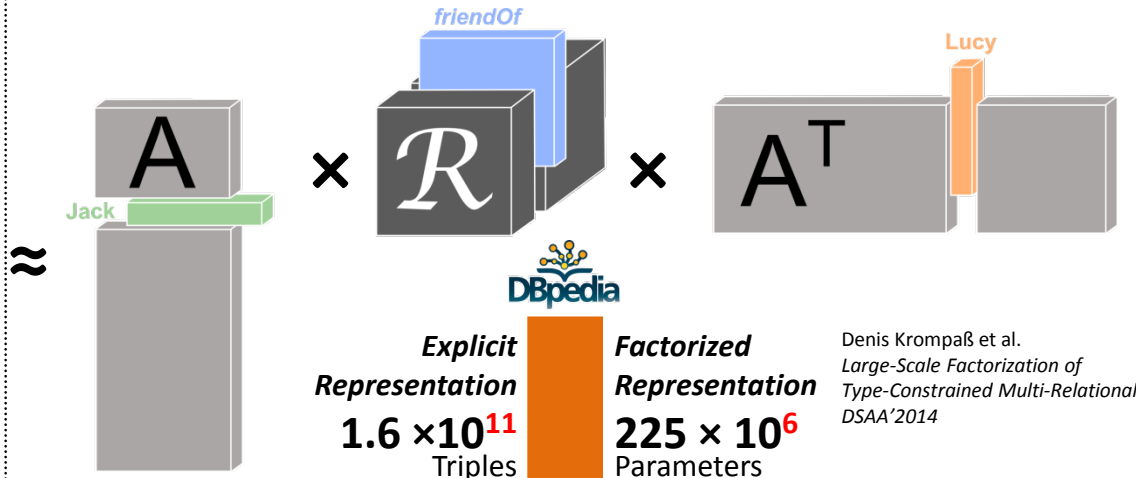
Subject	Object	P	RndVar
Jack	Jane	0.5	X_1
Jack	Lucy	0.1	X_2
Jack	Jim	0.1	X_3
Jane	Lucy	0.7	X_4
…	…	…	…

Query a triple:
„Is Jack a friend of Lucy?“

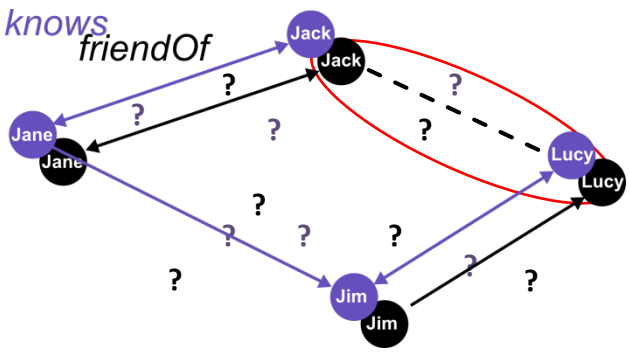
„Jack might be a friend of Lucy with a probability of 0.1 %“

Probabilistic KB Construction with RESCAL



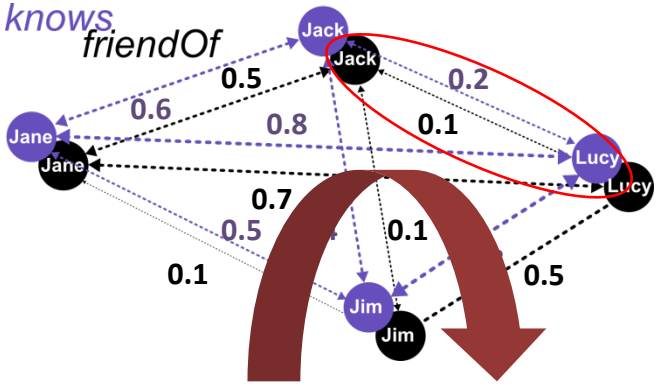


Denis Krompaß et al.
Large-Scale Factorization of Type-Constrained Multi-Relational Data.
DSAA'2014



Subject	Object
Jack	Jane
Jane	Jim
Jim	Lucy

Subject	Object
Jack	Jane
Jim	Lucy



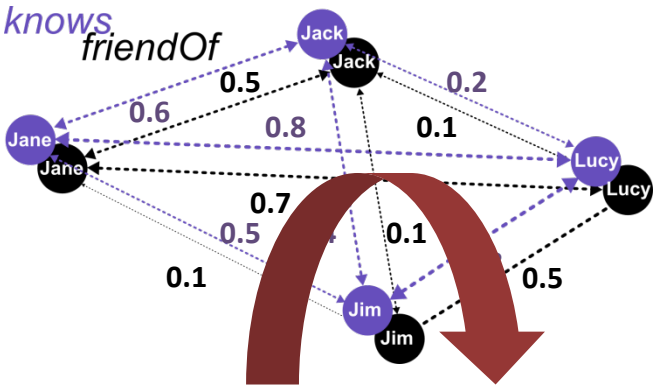
Subject	Object	P	RndVar
Jack	Jane	0.6	Y_1
Jack	Lucy	0.2	Y_2
Jack	Jim	0.4	Y_3
Jane	Lucy	0.8	Y_4
…	…	…	…

Subject	Object	P	RndVar
Jack	Jane	0.5	X_1
Jack	Lucy	0.1	X_2
Jack	Jim	0.1	X_3
Jane	Lucy	0.7	X_4
…	…	…	…

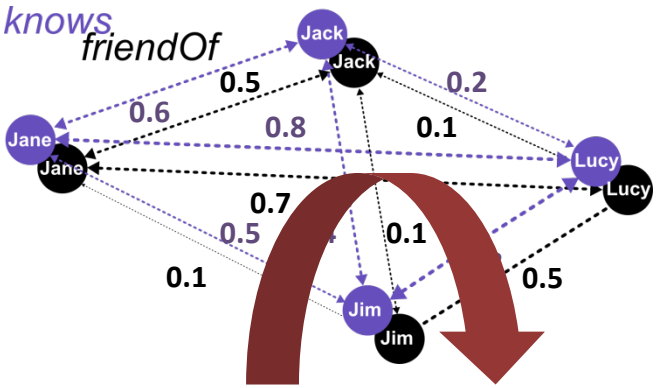
Query a triple:
„Is Jack a friend of Lucy?"

„Jack might be a friend of Lucy with a probability of 0.1 %"

Complex Querying: Pure Extensional Query Evaluation on Factorized PKBs



Complex Querying: Pure Extensional Query Evaluation on Factorized PKBs

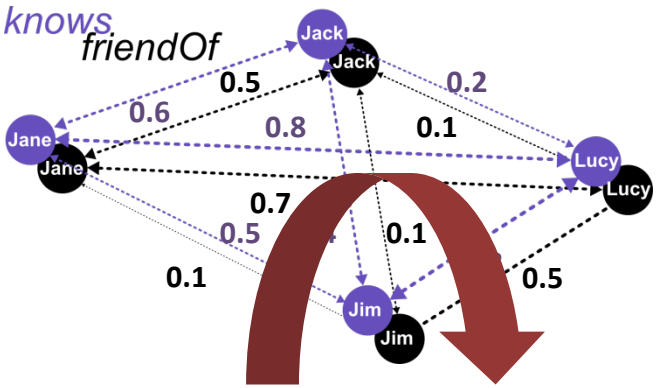


"Does Jack know someone who is a friend of Lucy?"

$Q :- \exists y.\, knows(\text{'Jack'}, y),\ friendOf\,(y, \text{'Lucy'})$

$$P(Q) = 1 - \prod_{a \in ADom} 1 - P(knows(\text{'Jack'}, a))P(\ friendOf\,(a, \text{'Lucy'}))$$

Complex Querying: Pure Extensional Query Evaluation on Factorized PKBs



"Does Jack know someone who is a friend of Lucy?"

$Q :- \exists y.\, knows(\text{'Jack'}, y),\ friendOf\,(y, \text{'Lucy'})$

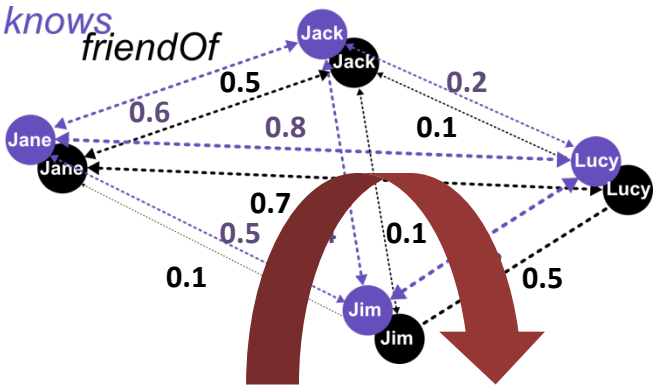
$$P(Q) = 1 - \prod_{a \in ADom} 1 - P(knows(\text{'Jack'}, a))P(\ friendOf\,(a, \text{'Lucy'}))$$

"Does Jack know a soccer player?"

$Q :- \exists y. \exists z. (knows(\text{'Jack'}, y), soccer(y, z))$

$$P(Q) = 1 - \prod_{a \in ADom} 1 - P(knows(\text{'Jack'}, a)) \left[1 - \prod_{b \in ADom} 1 - P(soccer(a, \text{b})) \right]$$

Complex Querying: Pure Extensional Query Evaluation on Factorized PKBs



- For each existential quantifier $\exists x.$ the ***independent-project*** rule has to be applied
  - ➢ **Nested loops**

"Does Jack know someone who is a friend of Lucy?"

$$Q :- \exists y.\, knows(\text{'Jack'}, y),\ friendOf\,(y, \text{'Lucy'})$$

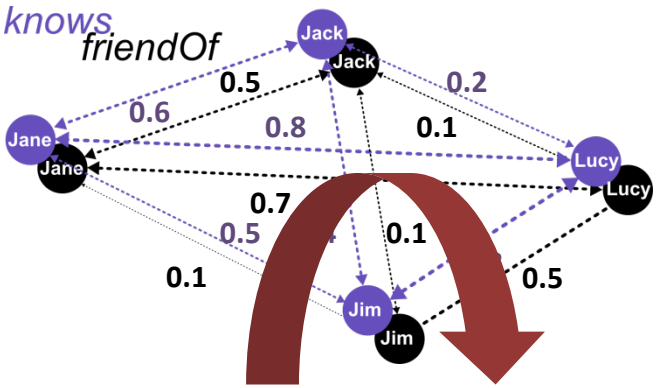
$$P(Q) = 1 - \prod_{a \in ADom} 1 - P(knows(\text{'Jack'}, a))P(\ friendOf\,(a, \text{'Lucy'}))$$

"Does Jack know a soccer player?"

$$Q :- \exists y. \exists z. (knows(\text{'Jack'}, y), soccer(y, z))$$

$$P(Q) = 1 - \prod_{a \in ADom} 1 - P(knows(\text{'Jack'}, a)) \left[1 - \prod_{b \in ADom} 1 - P(soccer(a, \text{b})) \right]$$

Complex Querying: Pure Extensional Query Evaluation on Factorized PKBs



- For each existential quantifier $\exists x.$ the ***independent-project*** rule has to be applied
  - ➢ **Nested loops**

"Does Jack know someone who is a friend of Lucy?"

$$Q :- \exists y.\, knows('\text{Jack}', y),\ friendOf\,(y, '\text{Lucy}')$$

$$P(Q) = 1 - \prod_{a \in ADom} 1 - P(knows('\text{Jack}', a))P(\,friendOf\,(a, '\text{Lucy}'))$$

Complexity scales already **cubic** in the size of the database if we ask the query **for all persons** in the database

"Does Jack know a soccer player?"

$$Q :- \exists y. \exists z. (knows('\text{Jack}', y), soccer(y, z))$$

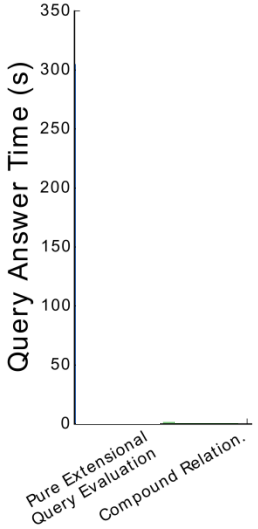
$$P(Q) = 1 - \prod_{a \in ADom} 1 - P(knows('\text{Jack}', a)) \left[1 - \prod_{b \in ADom} 1 - P(soccer(a, \text{b})) \right]$$

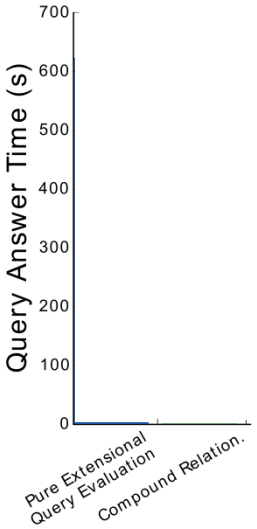
Querying DBpedia-Music

(44345 Entities, 7 Relations)

- ***„What songs or albums from the Pop-Rock genre are from musical artists that have/had a contract with Atlantic Records?“***

$$Q_1(x) :- \exists y.\, genre(x, \text{'Pop\_Rock'}),\; musicalArtist(x, y), recordLabel(y, \text{'Atlantic\_Records'})$$



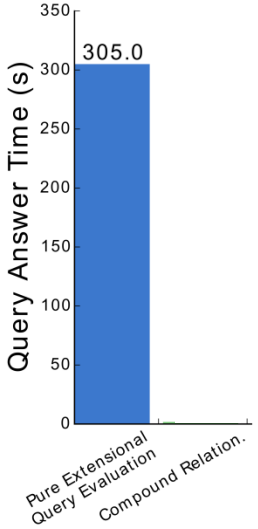


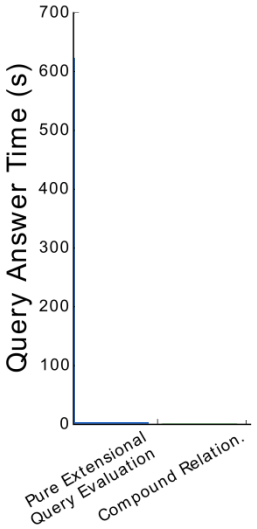
Querying DBpedia-Music

(44345 Entities, 7 Relations)

- ***„What songs or albums from the Pop-Rock genre are from musical artists that have/had a contract with Atlantic Records?“***

$$Q_1(x) :- \exists y.\, genre(x, \text{'Pop\_Rock'}),\; musicalArtist(x, y), recordLabel(y, \text{'Atlantic\_Records'})$$



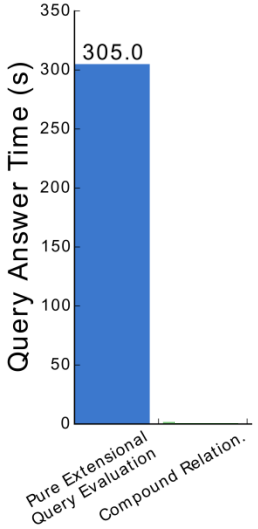


Querying DBpedia-Music

(44345 Entities, 7 Relations)

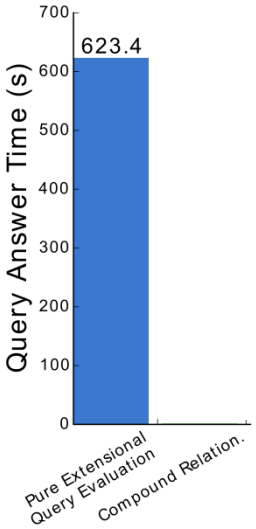
- ***„What songs or albums from the Pop-Rock genre are from musical artists that have/had a contract with Atlantic Records?“***

$$Q_1(x) :- \exists y.\, genre(x, \text{'Pop\_Rock'}),\; musicalArtist(x, y), recordLabel(y, \text{'Atlantic\_Records'})$$



- ***„Which musical artists from the Hip-Hop Music genre have/had a contract with Shady , Aftermath or Death Row Records?“***

$$Q_2(x) :- \exists y.\, genre(x, \text{'Hip\_Hop\_Music'}),\; musicalArtist(x, y), recordLabel(y, \{\text{'SD'}, \text{'AM'}, \text{'DR'}\})$$



Querying DBpedia-Music

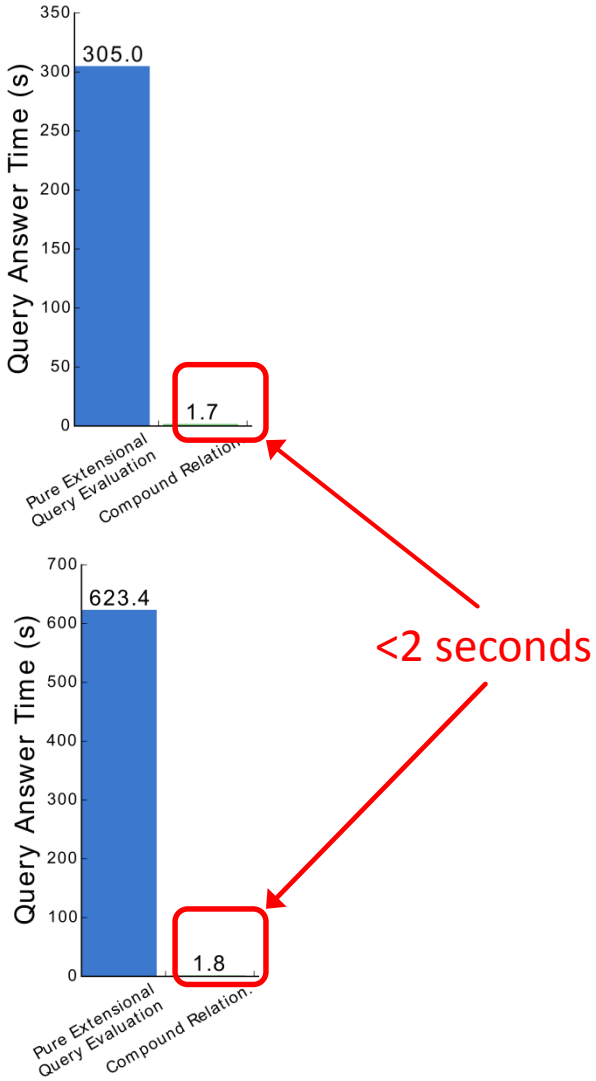
(44345 Entities, 7 Relations)

- ***„What songs or albums from the Pop-Rock genre are from musical artists that have/had a contract with Atlantic Records?“***

$$Q_1(x) :- \exists y.\, genre(x, \text{'Pop\_Rock'}),\; musicalArtist(x, y), recordLabel(y, \text{'Atlantic\_Records'})$$

- ***„Which musical artists from the Hip-Hop Music genre have/had a contract with Shady , Aftermath or Death Row Records?“***

$$Q_2(x) :- \exists y.\, genre(x, \text{'Hip\_Hop\_Music'}),\; musicalArtist(x, y), recordLabel(y, \{\text{'SD'}, \text{'AM'}, \text{'DR'}\})$$



Avoiding Independent-Project

"Does Jack know a soccer player?"

$Q :- \exists y.\exists z.(knows(\text{'Jack'}, y), soccer(y, z))$

$$P(Q) = 1 - \prod_{a \in ADom} 1 - P(knows(\text{'Jack'}, a)) \left[1 - \prod_{b \in ADom} 1 - P(soccer(a, \text{b})) \right]$$

Avoiding Independent-Project

"Does Jack know a soccer player?"

$Q :- \exists y.\exists z.(knows('\text{Jack}', y), soccer(y, z))$

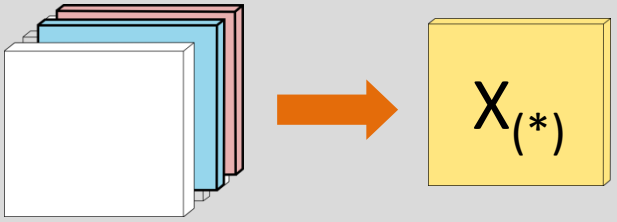
$$P(Q) = 1 - \prod_{a \in ADom} 1 - P(knows('\text{Jack}', a)) \left[1 - \prod_{b \in ADom} 1 - P(soccer(a, \text{b})) \right]$$

1. Construct deterministic compound relation $X_{(*)}$

Initial Deterministic KB

knows(), *soccer()*

knowsSoccerPlayerOf()



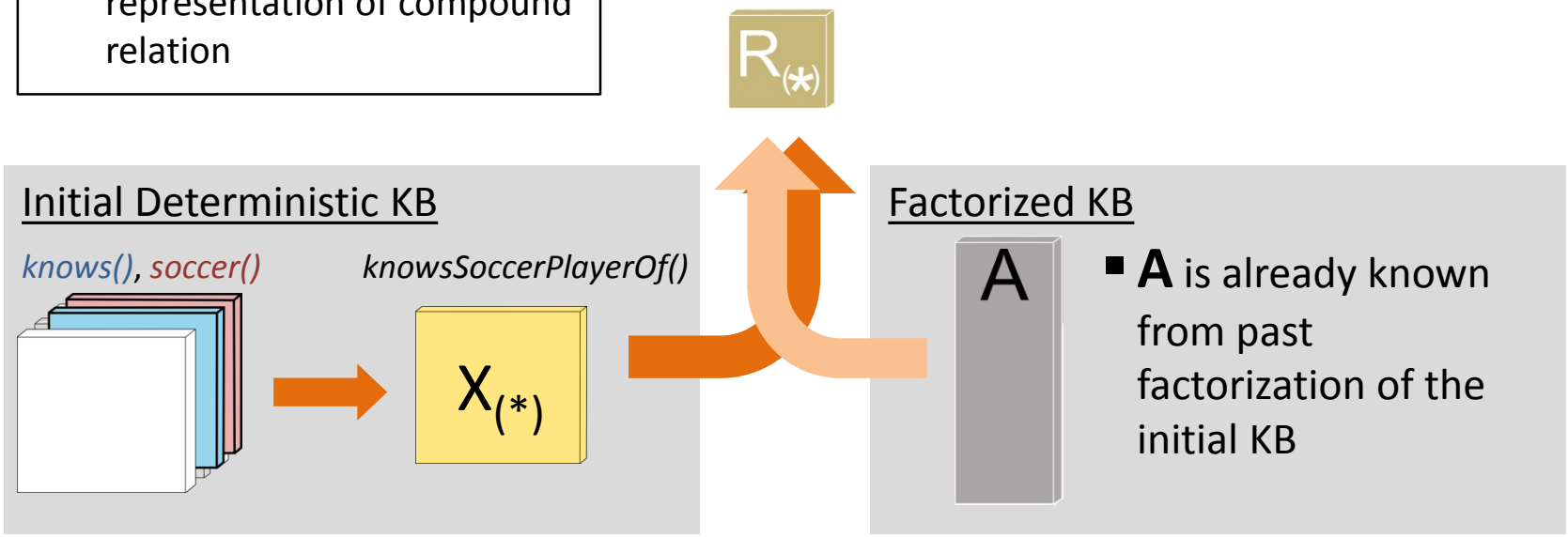
Avoiding Independent-Project

"Does Jack know a soccer player?"

$Q :- \exists y.\exists z.(knows('\text{Jack}', y), soccer(y, z))$

$$P(Q) = 1 - \prod_{a \in ADom} 1 - P(knows('\text{Jack}', a)) \left[1 - \prod_{b \in ADom} 1 - P(soccer(a, \text{b})) \right]$$

1. Construct deterministic compound relation $X_{(*)}$
2. Approximate latent representation of compound relation



Avoiding Independent-Project

"Does Jack know a soccer player?"

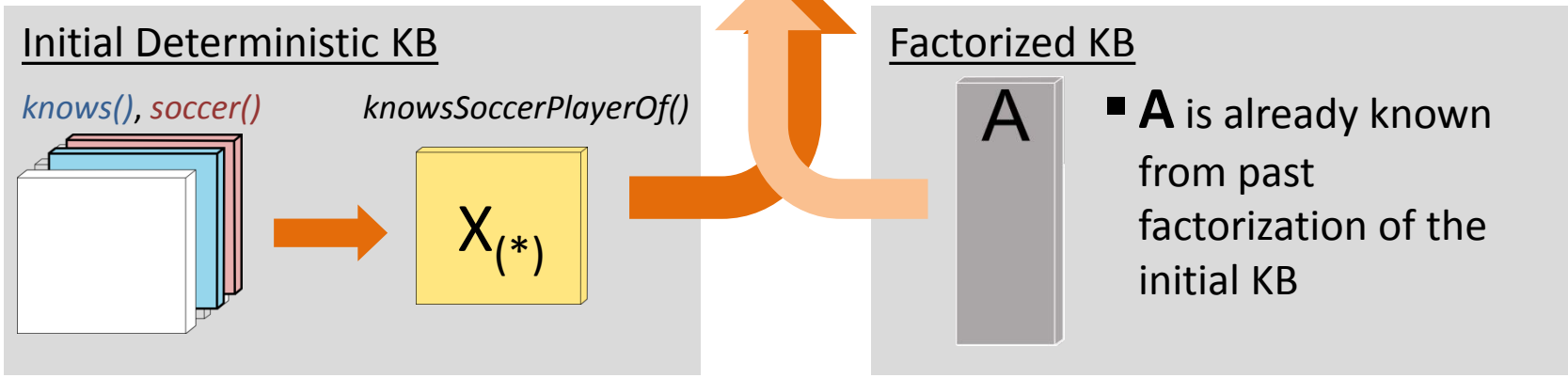
$Q :- \exists y.\exists z.(knows('\text{Jack}', y), soccer(y, z))$

$$P(Q) = 1 - \prod_{a \in ADom} 1 - P(knows('\text{Jack}', a)) \left[1 - \prod_{b \in ADom} 1 - P(soccer(a, \text{b})) \right]$$

1. Construct deterministic compound relation $X_{(*)}$
2. Approximate latent representation of compound relation

$$R_{(*)} = (A^T A \otimes A^T A + \lambda I)^{-1} (A \otimes A)^T X_{(*)}$$

$$R_{(*)} = V(S * U^T X_{(*)} U) V^T \quad | A = UDV^T$$



Avoiding Independent-Project

"Does Jack know a soccer player?"

$Q :- \exists y.\exists z.(knows('\text{Jack}', y), soccer(y, z))$

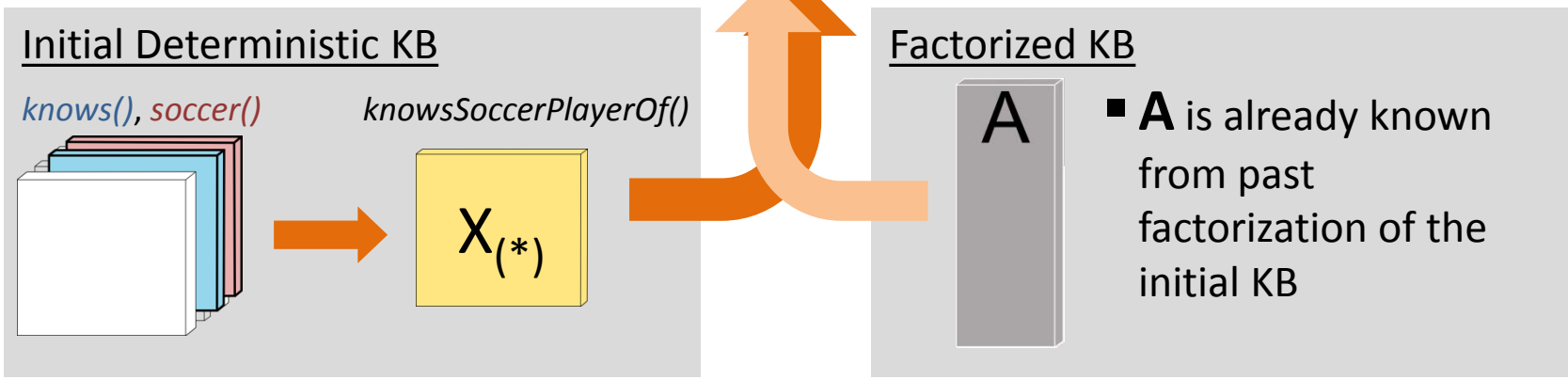
1. Construct deterministic compound relation $X_{(*)}$
2. Approximate latent representation of compound relation

$$P(Q) = 1 - \prod_{a \in ADom} 1 - P(knows('\text{Jack}', a)) \left[1 - \prod_{b \in ADom} 1 - P(soccer(a, b)) \right]$$

$$P(Q) \approx 1 - \prod_{b \in ADom} 1 - P(knowsSoccerPlayerOf('\text{Jack}', b))$$

$$R_{(*)} = (A^T A \otimes A^T A + \lambda I)^{-1} (A \otimes A)^T X_{(*)}$$

$$R_{(*)} = V(S * U^T X_{(*)} U) V^T \quad | A = UDV^T$$

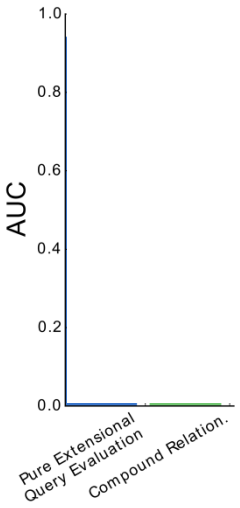


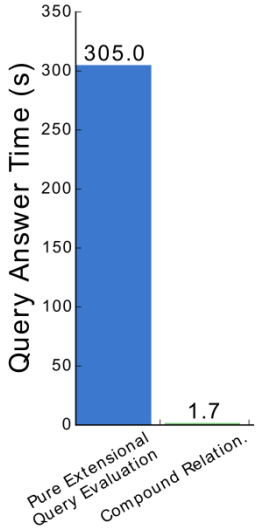
Querying DBpedia-Music

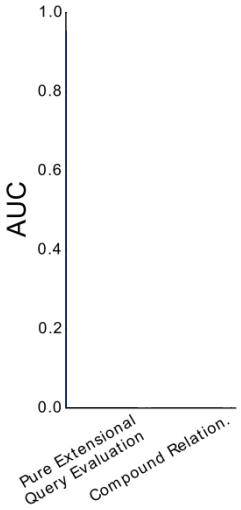
(44345 Entities, 7 Relations)

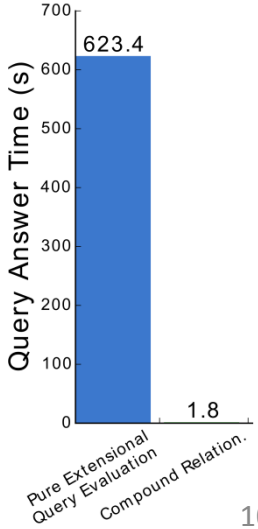
- ***„What songs or albums from the Pop-Rock genre are from musical artists that have/had a contract with Atlantic Records?“***

$$Q_1(x) :- \exists y.\, genre(x, \text{'Pop\_Rock'}),\; musicalArtist(x, y), recordLabel(y, \text{'Atlantic\_Records'})$$







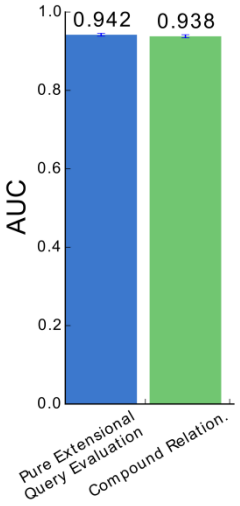


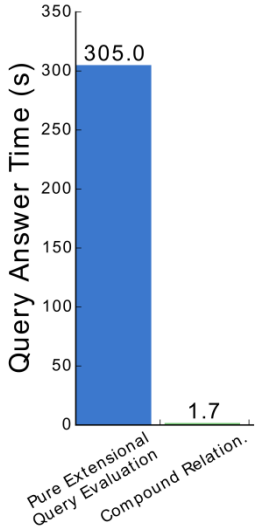
Querying DBpedia-Music

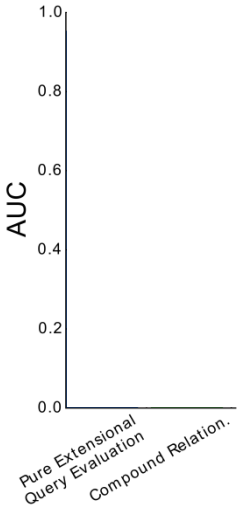
(44345 Entities, 7 Relations)

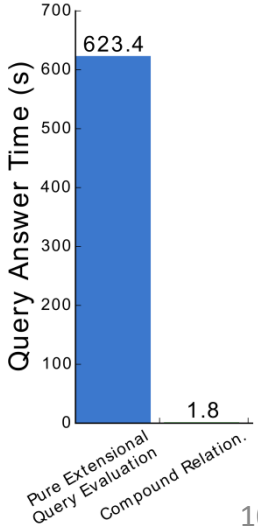
- ***„What songs or albums from the Pop-Rock genre are from musical artists that have/had a contract with Atlantic Records?“***

$$Q_1(x) :- \exists y.\, genre(x, \text{'Pop\_Rock'}),\; musicalArtist(x, y), recordLabel(y, \text{'Atlantic\_Records'})$$







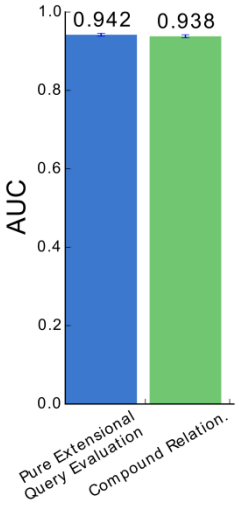


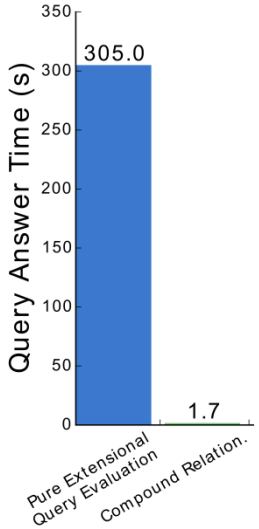
Querying DBpedia-Music

(44345 Entities, 7 Relations)

- ***„What songs or albums from the Pop-Rock genre are from musical artists that have/had a contract with Atlantic Records?“***

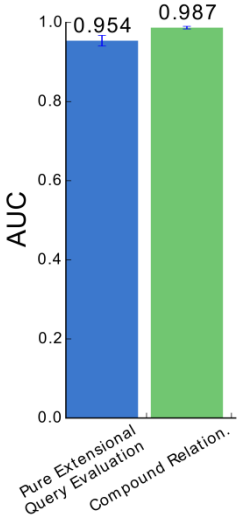
$$Q_1(x) :- \exists y.\, genre(x, \text{'Pop\_Rock'}),\; musicalArtist(x, y), recordLabel(y, \text{'Atlantic\_Records'})$$

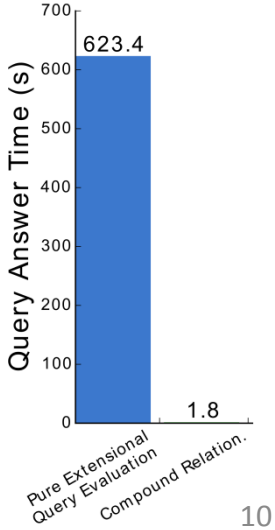




- ***„Which musical artists from the Hip-Hop Music genre have/had a contract with Shady , Aftermath or Death Row Records?“***

$$Q_2(x) :- \exists y.\, genre(x, \text{'Hip\_Hop\_Music'}),\; musicalArtist(x, y), recordLabel(y, \{\text{'SD'}, \text{'AM'}, \text{'DR'}\})$$





Querying DBpedia-Music

(44345 Entities, 7 Relations)

- ***„Which musical artists from the Hip-Hop music genre are associated with musical artists that have/had a contract with Interscope Records and are involved in an album whose first letter is a 'T'?"***

$$
\begin{aligned}
Q_3(x) :- \; & genre(x, \text{'Hip\_Hop\_Music'}), \\
& \exists y(associatedMusicalArtist(x, y), recordLabel(y, \text{'Interscope\_Records'})), \\
& \exists z.\exists t.(musicalArtist(z, x), album(z, \{\text{'AlbumT.*'}\}))
\end{aligned}
$$

Querying DBpedia-Music

(44345 Entities, 7 Relations)

- ***„Which musical artists from the Hip-Hop music genre are associated with musical artists that have/had a contract with Interscope Records and are involved in an album whose first letter is a 'T'?"***

$$Q_3(x) :- \; genre(x, \text{'Hip\_Hop\_Music'}),$$
$$\exists y(associatedMusicalArtist(x, y), recordLabel(y, \text{'Interscope\_Records'})),$$
$$\exists z.\exists t.(musicalArtist(z, x), album(z, \{\text{'AlbumT.*'}\}))$$

Pure Extensional Query Evaluation

Processing not finished after **6 hours**!

Querying DBpedia-Music

(44345 Entities, 7 Relations)

- ***„Which musical artists from the Hip-Hop music genre are associated with musical artists that have/had a contract with Interscope Records and are involved in an album whose first letter is a 'T'?"***

$$Q_3(x) :- \; genre(x, \text{'Hip\_Hop\_Music'}),$$
$$\exists y(associatedMusicalArtist(x, y), recordLabel(y, \text{'Interscope\_Records'})),$$
$$\exists z.\exists t.(musicalArtist(z, x), album(z, \{\text{'AlbumT.*'}\}))$$

Pure Extensional Query Evaluation	Exploiting Approximated Compound-Relations
Processing not finished after **6 hours**!	**3.6 seconds** AUC: 0.985

Summary

- Uncertainty can be reintroduced into deterministic representations of Knowledge Bases with RESCAL
- The factorized Representation can be exploited for complex querying
- Extensional query evaluation can be significantly accelerated by exploiting the RESCAL model (Compound Relations)

Questions ?

http://www.dbs.ifi.lmu.de/~krompass/

Denis.Krompass@campus.lmu.de