Rule Induction and Reasoning in Knowledge Graphs

Daria Stepanova

Bosch Center for Artificial Intelligence, Renningen, Germany

ODSC 2019, 21.11.2019
Motivation

Preliminaries

Rule Learning

Exception-awareness

Incompleteness

Rules from Hybrid Sources
What is Knowledge?

Plato: “Knowledge is justified true belief”
What is Knowledge?

Plato: “Knowledge is justified true belief”

350 BC
Knowledge Graphs as Digital Knowledge

“Digital knowledge is semantically enriched machine processable data”
Semantic Web Search

winner of Australian Open 2018

Roger Federer
Tennis player
rogerfederer.com

Roger Federer is a Swiss professional tennis player who is currently ranked world No. 10 by the Association of Tennis Professionals. Many players and analysts have called him the greatest tennis player of all time. Wikipedia

Born: August 8, 1981 (age 35 years), Basel, Switzerland
Height: 1.85 m
Weight: 85 kg
Spouse: Mirka Federer (m. 2009)
Children: Lenny Federer, Myla Rose Federer, Charlene Riva Federer, Leo Federer
Semantic Web Search

Google

∃X winnerOf(X, AustralianOpen2018)
Semantic Web Search

winner of Australian Open 2018

Roger Federer

Tennis player

Born: August 8, 1981 (age 37 years), Basel, Switzerland
Height: 1.85 m
Weight: 85 kg
Spouse: Mirka Federer (m. 2001)
Children: Lenny Federer, Myla Rose Federer, Charlene Riva Federer, Leo Federer
living place of the winner of australian open 2018

About 1,220,000,000 results (1.10 seconds)

2018 Australian Open - Wikipedia
https://en.wikipedia.org/wiki/2018_Australian_Open
Roger Federer was the defending champion in the men’s singles event and successfully retained his title (his sixth), defeating Marin Čilić in the final, while Caroline Wozniacki won the women’s title, defeating Simona Halep in the final.

Venue: Melbourne Park
Location: Melbourne, Victoria, Australia
Prize money: A$55,000,000
Draw: 128S / 64D /

Missing: living | Must include: living
Mirka Federer
m. 2009

Miroslava "Mirka" Federer is a Slovak-born Swiss former professional tennis player. She reached her career-high WTA singles ranking of world No. 76 on 10 September 2001 and a doubles ranking of No. 215 on 24 August 1998. She is the wife of tennis player Roger Federer, having first met him at the 2000 Summer Olympics. Wikipedia
Semantic Web Search

living place of Mirka Federer

About 1.910.000 results (0,92 seconds)

Mirka Federer / Residence

Bottmingen, Switzerland
Human Reasoning

\[ \text{livesIn}(Y, Z) \leftarrow \text{marriedTo}(X, Y), \]
\[ \text{livesIn}(X, Z) \]

\[ \text{marriedTo}(\text{mirka}, \text{roger}) \]

\[ \text{livesIn}(\text{mirka}, \text{bottmingen}) \]

Married people live together

Mirka is married to Roger

Mirka lives in Bottmingen
Human Reasoning

\[ \text{livesIn}(Y, Z) \leftarrow \text{marriedTo}(X, Y), \]
\[ \text{livesIn}(X, Z) \]

\[ \text{marriedTo}((\text{mirka}, \text{roger}) \]

\[ \text{livesIn}((\text{mirka}, \text{bottmingen}) \]

\[ \text{livesIn}((\text{roger}, \text{bottmingen}) \]

*Married people live together*

*Mirka is married to Roger*

*Mirka lives in Bottmingen*

*Roger lives in Bottmingen*
**Human Reasoning**

\[
livesIn(Y, Z) \leftarrow marriedTo(X, Y),
\]
\[
livesIn(X, Z)
\]

\[
marriedTo(mirka, roger)
\]

\[
livesIn(mirka, bottmingen)
\]

\[
livesIn(roger, bottmingen)
\]

**Married people live together**

**Mirka is married to Roger**

**Mirka lives in Bottmingen**

**Roger lives in Bottmingen**

But where can a machine get such rules from?
Applications of Rule Learning

- Fact prediction
- Fact checking
- Data cleaning
- Domain description
- Finding trends in KGs
Motivation

Preliminaries

Rule Learning

Exception-awareness

Incompleteness

Rules from Hybrid Sources
Horn Rules

Rule: \[ a \leftarrow b_1, \ldots, b_m. \]

Informal semantics: If \( b_1, \ldots, b_m \) are true, then \( a \) must be true.

Logic program: Set of rules

Example: ground rule

% If Mirka is married to Roger and lives in B., then Roger lives there too
\[ \text{livesIn(roger, bottmingen)} \leftarrow \text{isMarried(mirka, roger)}, \text{livesIn(mirka, bottmingen)} \]
Horn Rules

Rule: \[ a \leftarrow b_1, \ldots, b_m. \]

Informal semantics: If \( b_1, \ldots, b_m \) are true, then \( a \) must be true.

Logic program: Set of rules

Example: non-ground rule

% Married people live together
\( \text{livesIn}(Y, Z) \leftarrow \text{isMarried}(X, Y), \text{livesIn}(X, Z) \)
Nonmonotonic Rules

Rule:  \( a \leftarrow b_1, \ldots, b_m, \text{not } b_{m+1}, \ldots, \text{not } b_n. \)

Informal semantics: If \( b_1, \ldots, b_m \) are true and none of \( b_{m+1}, \ldots, b_n \) is known, then \( a \) must be true.

Closed World Assumption (CWA): facts not known to be true are false

Example: nonmonotonic rule

\% Two married live together unless one is a researcher
\[
livesIn(Y, Z) \leftarrow isMarried(X, Y), livesIn(X, Z), \text{not researcher}(Y)
\]
Nonmonotonic Rules

Rule: \[ a \leftarrow b_1, \ldots, b_m, \text{not } b_{m+1}, \ldots, \text{not } b_n. \]

Informal semantics: If \( b_1, \ldots, b_m \) are true and none of \( b_{m+1}, \ldots, b_n \) is known, then \( a \) must be true.

Closed World Assumption (CWA): facts not known to be true are false

\( \text{not} \) is different from \( \neg \).

% At a rail road crossing cross the road if no train is known to approach
walk \( \leftarrow \) at(L), crossing(L), \text{not train approaches}(L)

% At a rail road crossing cross the road if no train approaches
walk \( \leftarrow \) at(L), crossing(L), \( \neg \text{train approaches}(L) \)
Answer Set Programs

Evaluation of ASP programs is model-based

Answer set program (ASP) is a set of nonmonotonic rules

(1) \text{isMarriedTo}(\text{mary}, \text{john})
(2) \text{livesIn}(\text{mary}, \text{ulm})
(3) \text{livesIn}(Y, Z) \leftarrow \text{isMarriedTo}(X, Y), \text{livesIn}(X, Z), 
\text{not \ researcher}(Y)
Answer Set Programs

Evaluation of ASP programs is model-based

1. **Grounding**: substitute all variables with constants in all possible ways

---

**Answer set program (ASP)** is a set of nonmonotonic rules

(1) `isMarriedTo(mary, john)`
(2) `livesIn(mary, ulm)`
(3) `livesIn(Y, Z) ← isMarriedTo(X, Y), livesIn(X, Z), not researcher(Y)`
Answer Set Programs

Evaluation of ASP programs is model-based
1. Grounding: substitute all variables with constants in all possible ways

Answer set program (ASP) is a set of nonmonotonic rules

1. `isMarriedTo(mary, john)`
2. `livesIn(mary, ulm)`
3. `livesIn(john, ulm) ← isMarriedTo(mary, john), livesIn(mary, ulm), not researcher(john)`
Answer Set Programs

**Evaluation** of ASP programs is model-based
1. **Grounding:** substitute all variables with constants in all possible ways
2. **Solving:** compute a minimal model (answer set) \( I \) satisfying all rules

**Answer set program (ASP) is a set of nonmonotonic rules**

(1) \( \text{isMarriedTo}(\text{mary}, \text{john}) \)  
(2) \( \text{livesIn}(\text{mary}, \text{ulm}) \)  
(3) \( \text{livesIn}(\text{john}, \text{ulm}) \leftarrow \text{isMarriedTo}(\text{mary}, \text{john}), \text{livesIn}(\text{mary}, \text{ulm}), \text{not researcher}(\text{john}) \)

\( I=\{\text{isMarriedTo}(\text{mary}, \text{john}), \text{livesIn}(\text{mary}, \text{ulm}), \text{livesIn}(\text{john}, \text{ulm})\} \)

**CWA:** \( \text{researcher}(\text{john}) \) can not be derived, thus it is false
Answer Set Programs

Evaluation of ASP programs is model-based
1. Grounding: substitute all variables with constants in all possible ways
2. Solving: compute a minimal model (answer set) satisfying all rules

Answer set program (ASP) is a set of nonmonotonic rules

(1) isMarriedTo(mary, john) (2) livesIn(mary, ulm)
(3) livesIn(john, ulm) ← isMarriedTo(mary, john), livesIn(mary, ulm), not researcher(john)
(4) researcher(john)

\[ I = \{ \text{isMarriedTo(mary, john), livesIn(mary, ulm), livesIn(john, ulm)} \} \]
Answer Set Programs

Evaluation of ASP programs is model-based
1. Grounding: substitute all variables with constants in all possible ways
2. Solving: compute a minimal model (answer set) I satisfying all rules

Answer set program (ASP) is a set of nonmonotonic rules

1. `isMarriedTo(mary, john)`
2. `livesIn(mary, ulm)`
3. `livesIn(john, ulm) ← isMarriedTo(mary, john), livesIn(mary, ulm), not researcher(john)`
4. `researcher(john)`

$I = \{isMarriedTo(mary, john), livesIn(mary, ulm), livesIn(john, ulm)\}$

Particularly suited for reasoning under incompleteness!
Motivation

Preliminaries

Rule Learning

Exception-awareness

Incompleteness

Rules from Hybrid Sources
Reasoning with Incomplete Information

**Default Reasoning**

Assume normal state of affairs, unless there is evidence to the contrary

*By default married people live together.*
Reasoning with Incomplete Information

**Default Reasoning**

Assume normal state of affairs, unless there is evidence to the contrary

*By default married people live together.*

**Abduction**

Choose between several explanations that explain an observation

*John and Mary live together. They must be married.*
### Reasoning with Incomplete Information

<table>
<thead>
<tr>
<th>Default Reasoning</th>
<th>Abduction</th>
<th>Induction</th>
</tr>
</thead>
</table>
| Assume normal state of affairs, unless there is evidence to the contrary.  
*By default married people live together.* | Choose between several explanations that explain an observation.  
*John and Mary live together. They must be married.* | Generalize a number of similar observations into a hypothesis.  
*Given many examples of spouses living together generalize this knowledge.*
Reasoning with Incomplete Information

**Default Reasoning**
Assume normal state of affairs, unless there is evidence to the contrary

*By default married people live together.*

**Abduction**
Choose between several explanations that explain an observation

*John and Mary live together. They must be married.*

**Induction**
Generalize a number of similar observations into a hypothesis

*Given many examples of spouses living together generalize this knowledge.*
History of Inductive Learning

- **AI & Machine Learning 1960s-70s:**
  Banerji, Plotkin, Vere, Michalski, ...

- **AI & Machine Learning 1980s:**
  Shapiro, Sammut, Muggleton, ...

- **Inductive Logic Programming (ILP) 1990s:**
  Muggleton, Quinlan, De Raedt, ...

- **Statistical Relational Learning 2000s:**
  Getoor, Koller, Domingos, Sato, ...
Inductive Learning from Examples [Muggleton, 1991]

Given:

- \( E^+ = \{fatherOf(john, mary), fatherOf(david, steve)\} \)
- \( E^- = \{fatherOf(kathy, ellen), fatherOf(john, steve)\} \)
- \( T = \{parentOf(john, mary), male(john),
    parentOf(david, steeve), male(david),
    parentOf(kathy, ellen), female(kathy)\} \)

- Language bias: Horn rules with 2 body atoms
Learning from Examples

Inductive Learning from Examples [Muggleton, 1991]

Given:

- \( E^+ = \{ \text{fatherOf}(john, mary), \text{fatherOf}(david, steve) \} \)
- \( E^- = \{ \text{fatherOf}(kathy, ellen), \text{fatherOf}(john, steve) \} \)
- \( T = \{ \text{parentOf}(john, mary), \text{male}(john), \)
  \text{parentOf}(david, steve), \text{male}(david), \)
  \text{parentOf}(kathy, ellen), \text{female}(kathy) \} \)
- Language bias: Horn rules with 2 body atoms

Possible hypothesis:

- \( Hyp : \text{fatherOf}(X, Y) \leftarrow \text{parentOf}(X, Y), \text{male}(X) \)
Inductive Learning from Interpretations [Raedt and Dzeroski, 1994]

Given:

- $I = \{\text{isMarriedTo(mirka, roger)}, \text{livesIn(mirka, b)}, \text{livesIn(roger, b)}, \text{bornIn(mirka, b)}\}$
- $T = \{\text{isMarriedTo(mirka, roger)}; \text{bornIn(mirka, b)}; \text{livesIn(X, Y) ← bornIn(X, Y)}\}$
- Language bias: Horn rules with 2 body atoms
Learning from Interpretations

Inductive Learning from Interpretations [Raedt and Dzeroski, 1994]

Given:

- \( I = \{ \text{isMarriedTo(mirka, roger)}, \text{livesIn(mirka, b)}, \text{livesIn(roger, b)}, \text{bornIn(mirka, b)} \} \)
- \( T = \{ \text{isMarriedTo(mirka, roger)}; \text{bornIn(mirka, b)}; \text{livesIn(X, Y) \leftarrow bornIn(X, Y)} \} \)
- Language bias: Horn rules with 2 body atoms

Possible Hypothesis:

- \( Hyp : \text{livesIn(Y, Z) \leftarrow isMarriedTo(X, Y), bornIn(X, Z)} \)
Common Techniques in ILP

- **Generality (⊇):** essential component of symbolic learning systems
- **Generalization as θ-subsumption**
  - **Atoms:** $a ⊇ b$ iff a substitution $θ$ exists such that $aθ = b$
Common Techniques in ILP

- **Generality ($\preceq$):** essential component of symbolic learning systems
- **Generalization as $\theta$-subsumption**
  - **Atoms:** $a \preceq b$ iff a substitution $\theta$ exists such that $a\theta = b$
    
    \[ \text{person}(X) \preceq \text{person}(\text{roger}), \quad \theta = \{X/\text{roger}\} \]
Common Techniques in ILP

- **Generality (⪰)**: essential component of symbolic learning systems
- **Generalization as $\theta$-subsumption**
  - **Atoms**: $a \succeq b$ iff a substitution $\theta$ exists such that $a\theta = b$
    - $person(X) \succeq person(roger), \theta = \{X/roger\}$
  - **Clause**: $C \succeq D$ iff $\theta$ exists, s.t. $C\theta \subseteq D$
Common Techniques in ILP

- **Generality ($\succeq$):** essential component of symbolic learning systems
- **Generalization as $\theta$-subsumption**
  - **Atoms:** $a \succeq b$ iff a substitution $\theta$ exists such that $a\theta = b$
    \[
    \text{person}(X) \succeq \text{person}(\text{roger}), \quad \theta = \{X/\text{roger}\}
    \]
  - **Clause:** $C \succeq D$ iff $\theta$ exists, s.t. $C\theta \subseteq D$
    \[
    \{\text{worksAt}(X,Y)\} \succeq \{\text{worksAt}(Z,\text{bosch}), \text{researcher}(Z)\},
    \]
Common Techniques in ILP

- **Generality (⪰):** essential component of symbolic learning systems

- **Generalization as θ-subsumption**
  - **Atoms:** $a \triangleright b$ iff a substitution $\theta$ exists such that $a\theta = b$
    
    \[
    \text{person}(X) \triangleright \text{person}(\text{roger}), \quad \theta = \{ X/\text{roger} \}
    \]
  - **Clause:** $C \triangleright D$ iff $\theta$ exists, s.t. $C\theta \subseteq D$
    
    \[
    \{ \text{worksAt}(X, Y) \} \triangleright \{ \text{worksAt}(Z, \text{bosch}), \text{researcher}(Z) \}, \quad \theta = \{ X/Z, Y/\text{bosch} \}
    \]
Common Techniques in ILP

- **Generality ($\supseteq$)**: essential component of symbolic learning systems

- **Generalization as $\theta$-subsumption**
  - Atoms: $a \supseteq b$ iff a substitution $\theta$ exists such that $a\theta = b$
    - $\text{person}(X) \supseteq \text{person}(\text{roger})$, $\theta = \{X/\text{roger}\}$
  - Clause: $C \supseteq D$ iff $\theta$ exists, s.t. $C\theta \subseteq D$
    - $\{\text{worksAt}(X, Y)\} \supseteq \{\text{worksAt}(Z, \text{bosch}), \text{researcher}(Z)\}$
      - $\theta = \{X/Z, Y/\text{bosch}\}$

- **Generalization as entailment**
  - Logic program: $Hyp1 \supseteq Hyp2$ iff $Hyp1 \models Hyp2$
Common Techniques in ILP

- **Generality (⊆):** essential component of symbolic learning systems
- **Generalization as θ-subsumption**
  - **Atoms:** $a \supseteq b$ iff a substitution $\theta$ exists such that $a\theta = b$
    
    \[
    \text{person}(X) \supseteq \text{person}(\text{roger}), \quad \theta = \{X/\text{roger}\}
    \]
  - **Clause:** $C \supseteq D$ iff $\theta$ exists, s.t. $C\theta \subseteq D$
    
    \[
    \{\text{worksAt}(X, Y)\} \supseteq \{\text{worksAt}(Z, \text{bosch}), \text{researcher}(Z)\},
    \theta = \{X/Z, Y/bosch\}
    \]
- **Generalization as entailment**
  - **Logic program:** $\text{Hyp1} \supseteq \text{Hyp2}$ iff $\text{Hyp1} \models \text{Hyp2}$
    
    \[
    \begin{align*}
    \text{person}(X) & \leftarrow \text{researcher}(X) & \text{person}(\text{mat}) & \leftarrow \text{researcher}(\text{mat}) \\
    & \text{Hyp1} & & \text{Hyp2}
    \end{align*}
    \]
Common Techniques in ILP

- **Generality ($\supseteq$):** essential component of symbolic learning systems

- **Generalization as $\theta$-subsumption**
  - **Atoms:** $a \supseteq b$ iff a substitution $\theta$ exists such that $a\theta = b$
    
    $\text{person}(X) \supseteq \text{person}(\text{roger}), \quad \theta = \{X/\text{roger}\}$

  - **Clause:** $C \supseteq D$ iff $\theta$ exists, s.t. $C\theta \subseteq D$
    
    \[
    \{\text{worksAt}(X, Y)\} \supseteq \{\text{worksAt}(Z, \text{bosch}), \text{researcher}(Z)\},
    \quad \theta = \{X/Z, Y/\text{bosch}\}
    \]

- **Generalization as entailment**
  - **Logic program:** $\text{Hyp1} \supseteq \text{Hyp2}$ iff $\text{Hyp1} \models \text{Hyp2}$
    
    $\underbrace{\text{person}(X) \leftarrow \text{researcher}(X)}_{\text{Hyp1}} \underbrace{\text{person(mat)} \leftarrow \text{researcher(mat)}}_{\text{Hyp2}}$

    $\text{Hyp1} \supseteq \text{Hyp2}$
Common Techniques in ILP

- Generality ($\succeq$): essential component of symbolic learning systems

- Generalization as $\theta$-subsumption
  - Atoms: $a \succeq b$ iff a substitution $\theta$ exists such that $a\theta = b$
    
    $\text{person}(X) \succeq \text{person}(\text{roger})$, $\theta = \{X/\text{roger}\}$
  
  - Clause: $C \succeq D$ iff $\theta$ exists, s.t. $C\theta \subseteq D$
    
    $\{\text{worksAt}(X, Y)\} \succeq \{\text{worksAt}(Z, \text{bosch}), \text{researcher}(Z)\}$,
    $\theta = \{X/Z, Y/bosch\}$

- Generalization as entailment
  
  - Logic program: $\text{Hyp1} \succeq \text{Hyp2}$ iff $\text{Hyp1} \models \text{Hyp2}$
  
    $\text{person}(X) \leftarrow \text{researcher}(X)$ $\text{person}(X) \leftarrow \text{researcher}(X), \text{alive}(X)$

  $\text{Hyp1}$ $\text{Hyp2}$
Common Techniques in ILP

- **Generality (≥)**: essential component of symbolic learning systems
- **Generalization as θ-subsumption**
  - **Atoms**: \(a \geq b\) iff a substitution \(\theta\) exists such that \(a\theta = b\)
    
    \[\text{person}(X) \geq \text{person}(\text{roger}), \quad \theta = \{X/\text{roger}\}\]
  - **Clause**: \(C \geq D\) iff \(\theta\) exists, s.t. \(C\theta \subseteq D\)
    
    \[\{\text{worksAt}(X, Y)\} \geq \{\text{worksAt}(Z, \text{bosch}), \text{researcher}(Z)\},\]
    \(\theta = \{X/Z, Y/\text{bosch}\}\)
- **Generalization as entailment**
  - **Logic program**: \(\text{Hyp1} \geq \text{Hyp2}\) iff \(\text{Hyp1} \models \text{Hyp2}\)
    
    \[
    \begin{align*}
    \text{person}(X) &\leftarrow \text{researcher}(X) & \text{Hyp1} \\
    \text{person}(X) &\leftarrow \text{researcher}(X), \text{alive}(X) & \text{Hyp2}
    \end{align*}
    \]
    
    \(\text{Hyp1} \geq \text{Hyp2}\)
Common Techniques in ILP

- **Generality ($\succeq$)**: essential component of symbolic learning systems
- **Generalization as $\theta$-subsumption**
  - **Atoms**: $a \succeq b$ iff a substitution $\theta$ exists such that $a\theta = b$
    
    $$\text{person}(X) \succeq \text{person}(\text{roger}), \quad \theta = \{X/\text{roger}\}$$
  - **Clause**: $C \succeq D$ iff $\theta$ exists, s.t. $C\theta \subseteq D$
    
    $$\{\text{worksAt}(X, Y)\} \succeq \{\text{worksAt}(Z, \text{bosch}), \text{researcher}(Z)\},$$
    $$\theta = \{X/Z, Y/\text{bosch}\}$$

- **Generalization as entailment**
  - **Logic program**: $Hyp1 \succeq Hyp2$ iff $Hyp1 \models Hyp2$
    
    $$\underbrace{\text{person}(X) \leftarrow \text{researcher}(X)}_{Hyp1} \quad \underbrace{\text{person}(X) \leftarrow \text{researcher}(X), \text{alive}(X)}_{Hyp2}$$
    $$Hyp1 \succeq Hyp2$$

  - **Relative entailment**: $Hyp1 \succeq Hyp2$ wrt $T$ iff $Hyp1 \cup T \models Hyp2$
Common Techniques in ILP

- **Generality (⪰):** essential component of symbolic learning systems
- **Generalization as \( \theta \)-subsumption
  - **Atoms:** \( a \⪰ b \) iff a substitution \( \theta \) exists such that \( a\theta = b \)
    \[ \text{person}(X) \⪰ \text{person}(\text{roger}), \quad \theta = \{X/\text{roger}\} \]
  - **Clause:** \( C \⪰ D \) iff \( \theta \) exists, s.t. \( C\theta \subseteq D \)
    \[ \{\text{worksAt}(X, Y)\} \⪰ \{\text{worksAt}(Z, \text{bosch}), \text{researcher}(Z)\}, \quad \theta = \{X/Z, Y/bosch\} \]
- **Generalization as entailment
  - **Logic program:** \( \text{Hyp1} \⪰ \text{Hyp2} \) iff \( \text{Hyp1} \models \text{Hyp2} \)
    \[
    \begin{align*}
    \text{person}(X) & \leftarrow \text{researcher}(X) \\
    \text{person}(X) & \leftarrow \text{researcher}(X), \text{alive}(X)
    \end{align*}
    \]
    \[ \text{Hyp1} \⪰ \text{Hyp2} \]
  - **Relative entailment:** \( \text{Hyp1} \⪰ \text{Hyp2} \) wrt \( T \) iff \( \text{Hyp1} \cup T \models \text{Hyp2} \)
    \[ \text{livesIn(\text{roger}, \text{bottmingen})} \ \text{?} \ \text{livesIn(\text{roger}, \text{switzerland})} \]
Common Techniques in ILP

- Generality ($\succeq$): essential component of symbolic learning systems
- Generalization as $\theta$-subsumption
  - Atoms: $a \succeq b$ iff a substitution $\theta$ exists such that $a\theta = b$
    
    $\text{person}(X) \succeq \text{person}(\text{roger}), \quad \theta = \{X/\text{roger}\}$
  - Clause: $C \succeq D$ iff $\theta$ exists, s.t. $C\theta \subseteq D$
    
    \[
    \{\text{worksAt}(X, Y)\} \succeq \{\text{worksAt}(Z, \text{bosch}), \text{researcher}(Z)\},
    \theta = \{X/Z, Y/\text{bosch}\}
    \]
- Generalization as entailment
  - Logic program: $\text{Hyp1} \succeq \text{Hyp2}$ iff $\text{Hyp1} \models \text{Hyp2}$
    
    \[
    \begin{align*}
    \text{person}(X) & \leftarrow \text{researcher}(X) \\
    \text{person}(X) & \leftarrow \text{researcher}(X), \text{alive}(X)
    \end{align*}
    
    \text{Hyp1} \succeq \text{Hyp2}
  - Relative entailment: $\text{Hyp1} \succeq \text{Hyp2}$ wrt $T$ iff $\text{Hyp1} \cup T \models \text{Hyp2}$
    
    \[
    \text{livesIn(}\text{roger, bottmingen}) \models \text{livesIn(}\text{roger, switzerland})
    \]
    
    $T : \text{livesIn}(X, \text{switzerland}) \leftarrow \text{livesIn}(X, \text{bottmingen})$
Common Techniques in ILP

- **Generality (⪰):** essential component of symbolic learning systems
- **Generalization as \( \theta \)-subsumption
  - **Atoms:** \( a \succeq b \) iff a substitution \( \theta \) exists such that \( a\theta = b \)
    
    \[
    \text{person}(X) \succeq \text{person}(roger), \quad \theta = \{X / roger\}
    \]
  - **Clause:** \( C \succeq D \) iff \( \theta \) exists, s.t. \( C\theta \subseteq D \)
    
    \[
    \{\text{worksAt}(X, Y)\} \succeq \{\text{worksAt}(Z, bosch), \text{researcher}(Z)\},
    \theta = \{X / Z, Y / bosch\}
    \]
- **Generalization as entailment**
  - **Logic program:** \( \text{Hyp1} \succeq \text{Hyp2} \) iff \( \text{Hyp1} \models \text{Hyp2} \)
    
    \[
    \begin{align*}
    &\underbrace{\text{person}(X) \leftarrow \text{researcher}(X)}_{\text{Hyp1}} \quad \underbrace{\text{person}(X) \leftarrow \text{researcher}(X), \text{alive}(X)}_{\text{Hyp2}} \\
    \end{align*}
    \]
    
    \( \text{Hyp1} \succeq \text{Hyp2} \)
  - **Relative entailment:** \( \text{Hyp1} \succeq \text{Hyp2} \) wrt \( T \) iff \( \text{Hyp1} \cup T \models \text{Hyp2} \)
    
    \[
    \begin{align*}
    &\text{livesIn}(roger, bottmingen) \succeq \text{livesIn}(roger, switzerland) \\
    &T : \text{livesIn}(X, switzerland) \leftarrow \text{livesIn}(X, bottmingen)
    \end{align*}
    \]
Common Techniques in ILP

- **Clause refinement** [Shapiro, 1991]: e.g., MIS, FOIL, etc.
  - Explore clause search space from general to specific or vice versa to find a hypothesis that covers all examples.

```
livesIn(X, Y) ←
  add atom
  livesIn(U, V)
  unify variables
  livesIn(bob, Y) ←
  unify variable to constant
  livesIn(X, X) ←
```
Common Techniques in ILP

- **Clause refinement** [Shapiro, 1991]: e.g., MIS, FOIL, etc.
  - Explore clause search space from general to specific or vice versa to find a hypothesis that covers all examples.

  \[
  \text{livesIn}(X, Y) \leftarrow \text{add atom} \quad \text{livesIn}(X, Y) \leftarrow \text{livesIn}(U, V) \leftarrow \text{unify variables} \quad \text{livesIn}(bob, Y) \leftarrow \text{unify variable to constant} \quad \text{livesIn}(X, X) \leftarrow
  \]

- **Inverse entailment** [Muggleton, 1995]: e.g., Progol, etc.
  - Properties of deduction to make hypothesis search space finite
Zoo of Other ILP Tasks

ILP tasks can be classified along several dimensions:

- **type of the data source**, e.g., positive/negative examples, interpretations, answer sets [Law et al., 2015]
- **type of the output knowledge**, e.g., rules, DL ontologies [Lehmann, 2009]
- **the way the data is given as input**, e.g., all at once, incrementally [Katzouris et al., 2015]
- **availability of an oracle**, e.g., human in the loop
- **quality of the data source**, e.g., noisy [Evans and Grefenstette, 2018]
- **data (in)completeness**, e.g., OWA vs CWA...
- **background knowledge**, e.g., DL ontology [d’Amato et al., 2016], hybrid theories [Lisi, 2010]
Classical ILP for KGs

ILP Goal

"The goal of ILP is to develop a correct (and complete) algorithm which efficiently computes hypotheses." [Sakama, 2005]

Knowledge Graphs

But the world knowledge is complex, and this might not always be possible in the context of KGs due to several issues...
Specialities of KGs

**Open World Assumption**: negative facts cannot be easily derived

*Maybe Roger Federer is a researcher and Albert Einstein was a ballet dancer?*
Specialities of KGs

**Open World Assumption**: negative facts cannot be easily derived

*Maybe Roger Federer is a researcher and Albert Einstein was a ballet dancer?*

> We dance for laughter, we dance for tears, we dance for madness, we dance for fears, we dance for hopes, we dance for screams, we are the dancers, we create the dreams.

— Albert Einstein
Challenges of Rule Induction from KGs

**Data bias**: KGs are extracted from text, which typically mentions only popular entities and interesting facts about them.

“*Man bites dog phenomenon*”¹

¹[https://en.wikipedia.org/wiki/Man_bites_dog_(journalism)]
Challenges of Rule Induction from KGs

**Huge size**: Modern KGs contain billions of facts

*E.g., Google KG stores 70 billion facts*
Challenges of Rule Induction from KGs

*World knowledge is complex*, none of its “models” is perfect
Question:
How can we still learn rules from KGs, which do not perfectly fit the data, but still reflect interesting correlations that can predict sufficiently many correct facts?

Answer:
Relational association rule mining! Roots in classical datamining.
Association Rules

- **Classical data mining task:** Given a transaction database, find out products (called itemsets) that are frequently bought together and form recommendation rules.

| Transaction 1 | 🍎 🍻 🥦 🍗 |
| Transaction 2 | 🍎 🍻 🥦 |
| Transaction 3 | 🍎 🍻 |
| Transaction 4 | 🍎 🍇 |
| Transaction 5 | 🥦 🍻 🍗 |
| Transaction 6 | 🥦 🍻 |
| Transaction 7 | 🥦 🍻 |
| Transaction 8 | 🥦 🍇 |

Out of 4 people who bought apples, 3 also bought beer.
Some Rule Measures

Support, confidence, lift

Support \([\text{🍎}]\) = 4

<table>
<thead>
<tr>
<th>Transaction 1</th>
<th>Transaction 2</th>
<th>Transaction 3</th>
<th>Transaction 4</th>
<th>Transaction 5</th>
<th>Transaction 6</th>
<th>Transaction 7</th>
<th>Transaction 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>🍎 🍊 🍏 🍗</td>
<td>🍎 🍊 🍏 🍗</td>
<td>🍎 🍊 🍏</td>
<td>🍎 🍊 🍏 🍗</td>
<td>🍎 🍊 🍏 🍗</td>
<td>🍎 🍊 🍏</td>
<td>🍎 🍊 🍏 🍗</td>
<td>🍎 🍊 🍏 🍗</td>
</tr>
</tbody>
</table>
Some Rule Measures

Support, confidence, lift

Support \{\text{Apple}\} = 4

Confidence \{\text{Apple} \rightarrow \text{Beer}\} = \frac{\text{Support} \{\text{Apple}, \text{Beer}\}}{\text{Support} \{\text{Apple}\}}

<table>
<thead>
<tr>
<th>Transaction 1</th>
<th>Transaction 2</th>
<th>Transaction 3</th>
<th>Transaction 4</th>
<th>Transaction 5</th>
<th>Transaction 6</th>
<th>Transaction 7</th>
<th>Transaction 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>🍎 🍊 🍯 🍗</td>
<td>🍎 🍊 🍯 🍗</td>
<td>🍎 🍊 🍯 🍗</td>
<td>🍎 🍊 🍯 🍗</td>
<td>🍎 🍊 🍯 🍗</td>
<td>🍎 🍊 🍯 🍗</td>
<td>🍎 🍊 🍯 🍗</td>
<td>🍎 🍊 🍯 🍗</td>
</tr>
</tbody>
</table>
Some Rule Measures

Support, confidence, lift

Support \( [\text{🍎}] = 4 \)

Confidence \( \{\text{🍎} \rightarrow \text{🍺} \} = \frac{\text{Support} \{\text{🍎,釀} \}}{\text{Support} \{\text{🍎} \}} \)

Lift \( \{\text{🍎} \rightarrow \text{釀} \} = \frac{\text{Support} \{\text{🍎,釀} \}}{\text{Support} \{\text{🍎} \} \times \text{Support} \{\text{釀} \}} \)

<table>
<thead>
<tr>
<th>Transaction 1</th>
<th>Transaction 2</th>
<th>Transaction 3</th>
<th>Transaction 4</th>
<th>Transaction 5</th>
<th>Transaction 6</th>
<th>Transaction 7</th>
<th>Transaction 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>🍎 🍺 🍏 🍗</td>
<td>🍎 🍺 🍏 🍗</td>
<td>🍎 🍺 🍏 🍗</td>
<td>🍎 🍺 🍏 🍗</td>
<td>🍎 🍺 🍏 🍗</td>
<td>🍎 🍺 🍏 🍗</td>
<td>🍎 🍺 🍏 🍗</td>
<td>🍎 🍺 🍏 🍗</td>
</tr>
</tbody>
</table>
Frequent Itemset Mining

- A=apple, B=beer... Frequent patterns are in green.
- Monotonicity: any superset of an infrequent pattern is infrequent
  At the heart of Apriori algorithm
Relational Association Rule Learning

- **WARMER** [Goethals and den Bussche, 2002]
- Upgrade frequent itemsets to frequent conjunctive queries

CQ: return all people with their spouses and living places

\[ q_1(X, Y, Z) : \neg \text{isMarriedTo}(X, Y) \land \text{livesIn}(X, Z) \]

Output: 6 tuples, i.e., \( \text{supp}(q_1) = 6 \)

CQ: return all people with their spouses and living places

\[ q_2(X, Y, Z) : \neg \text{isMarriedTo}(X, Y) \land \text{livesIn}(X, Z) \land \text{livesIn}(Y, Z) \]

Output: 3 tuples, i.e., \( \text{supp}(q_2) = 3 \)
Relational Association Rule Learning

- **WARMER** [Goethals and den Bussche, 2002]
- Upgrade frequent itemsets to frequent conjunctive queries
  - traverse the lattice
  - get frequent CQs based on user-specified value
  - split into body and head
  - rank based on a rule measure, e.g., confidence
Horn Rule Learning from KGs

**WARMER**: confidence

**CWA**: Whatever is not known is false.

```
conf(r) = | 1 + | r: livesIn(X, Z) ← isMarriedTo(Y, X), livesIn(Y, Z) |
```

- Brad isMarriedTo Ann
  - livesIn Berlin
    - livesIn Amsterdam
- Bob isMarriedTo Alice
  - livesIn Amsterdam
- Ann hasBrother John
  - livesIn Berlin
  - livesIn Chicago
  - IsA Researcher
- John isMarriedTo Kate
  - livesIn Berlin
  - livesIn Chicago
- Dave isMarriedTo Clara
  - livesIn Chicago
  - IsA Researcher
Horn Rule Learning from KGs

**WARMER**: confidence

**CWA**: Whatever is not known is false.

$$conf(r) = \frac{|\triangledown|}{|\triangledown| + |\triangleright|} = \frac{2}{4}$$

$$r : \text{livesIn}(X, Z) \iff \text{isMarriedTo}(Y, X), \text{livesIn}(Y, Z)$$
Horn Rule Learning from KGs

**WARMER**: confidence

**CWA**: Whatever is not known is false.

\[
\text{conf}(r) = \frac{|\Delta|}{|\Delta| + |\Theta|} = \frac{2}{4}
\]

\[r : \text{livesIn}(X, Z) \leftarrow \text{isMarriedTo}(Y, X), \text{livesIn}(Y, Z)\]
AMIE [Galarraga et al., 2015]: PCA confidence

PCA: If at least 1 living place of Alice is known, then all are known.

\[ conf_{PCA}(r) = \frac{|\vartriangle|}{|\vartriangle| + |\heartsuit|} = \frac{2}{3} \]

\( r : \text{livesIn}(X, Z) \leftrightarrow \text{isMarriedTo}(Y, X), \text{livesIn}(Y, Z) \)
AMIE Refinement Operators

- 1. livesIn(X, Y) ← add dangling atom
   - livesIn(X, Y) ← marriedTo(X, Z)
   - livesIn(X, Y) ← marriedTo(X, Z), livesIn(Z, Y)

- 2. livesIn(X, Y) ← add instantiated atom
   - livesIn(X, Y) ← isA(X, researcher)

Motivation

Preliminaries

Rule Learning

Exception-awareness

Incompleteness

Rules from Hybrid Sources
Nonmonotonic Rule Learning

Nonmonotonic rule mining from KGs: OWA is a challenge!

\[ r : \text{livesIn}(X, Z) \leftarrow \text{isMarriedTo}(Y, X), \text{livesIn}(Y, Z), \neg \text{researcher}(X) \]
Quality-based Horn Theory Revision

Given:

- Available KG

![Diagram showing the relationship between ideal KG and available KG]
Horn Theory Revision

Quality-based Horn Theory Revision

Given:

- Available KG
- Horn rule set
Horn Theory Revision

Quality-based Horn Theory Revision

Given:
- Available KG
- Horn rule set

Find:
- Nonmonotonic revision of Horn rule set
Horn Theory Revision

Quality-based Horn Theory Revision

Given:
- Available KG
- Horn rule set

Find:
- Nonmonotonic revision of Horn rule set with better predictive quality

Ideal KG (unknown)

Maximize

Minimize

Available KG

Revised rule predictions

Horn rule predictions
Avoid Data Overfitting

How to distinguish exceptions from noise?

\[ r1 : \text{livesIn}(X, Z) \leftarrow \text{isMarriedTo}(Y, X), \text{livesIn}(Y, Z), \text{not researcher}(X) \]
Avoid Data Overfitting

How to distinguish exceptions from noise?

\[
\begin{align*}
\text{r1: } & \text{livesIn}(X, Z) \leftarrow \text{isMarriedTo}(Y, X), \text{livesIn}(Y, Z), \text{not researcher}(X) \\
& \text{not livesIn}(X, Z) \leftarrow \text{isMarriedTo}(Y, X), \text{livesIn}(Y, Z), \text{researcher}(X)
\end{align*}
\]
Avoid Data Overfitting

How to distinguish exceptions from noise?

\begin{align*}
\text{r1 : } & \text{livesIn}(X, Z) \leftarrow \text{isMarriedTo}(Y, X), \text{livesIn}(Y, Z), \text{not researcher}(X) \\
& \text{not livesIn}(X, Z) \leftarrow \text{isMarriedTo}(Y, X), \text{livesIn}(Y, Z), \text{researcher}(X) \\
\text{r2 : } & \text{livesIn}(X, Z) \leftarrow \text{bornIn}(X, Z), \text{not moved}(X) \\
& \text{not livesIn}(X, Z) \leftarrow \text{bornIn}(X, Z), \text{moved}(X)
\end{align*}
Avoid Data Overfitting

How to distinguish exceptions from noise?

\[ r1 : \text{livesIn}(X, Z) \leftarrow \text{isMarriedTo}(Y, X), \text{livesIn}(Y, Z), \text{not researcher}(X) \]
\[ \text{not livesIn}(X, Z) \leftarrow \text{isMarriedTo}(Y, X), \text{livesIn}(Y, Z), \text{researcher}(X) \]

\[ r2 : \text{livesIn}(X, Z) \leftarrow \text{bornIn}(X, Z), \text{not moved}(X) \]
\[ \text{not livesIn}(X, Z) \leftarrow \text{bornIn}(X, Z), \text{moved}(X) \]

\{\text{livesIn}(c, d), \text{not livesIn}(c, d)\} \text{ are conflicting predictions} 

**Intuition:** Rules with good exceptions should make few conflicting predictions
Horn Theory Revision

Quality-based Horn Theory Revision

Given:
- Available KG
- Horn rule set

Find:
- Nonmonotonic revision of Horn rules, such that
  - number of conflicting predictions is minimal
  - average conviction is maximal

D. Tran, D. Stepanova, M. Gad-Elrab, F. Lisi, G. Weikum. Towards Nonmonotonic Relational Learning from KGs. ILP2016
Exception Candidates

\[ r: \text{livesIn}(X, Z) \leftarrow \text{isMarriedTo}(Y, X), \text{livesIn}(Y, Z) \]

\{ \text{not researcher}(X) \}
\{ \text{not artist}(Y) \}
Experiments

- **Approximated ideal KG**: original KG
- **Available KG**: for every relation randomly remove 20% of facts from approximated ideal KG
- **Horn rules**: \( h(X, Y) \leftarrow p(X, Z), q(Z, Y) \)
- **Exceptions**: \( e_1(X), e_2(Y), e_3(X, Y) \)
- **Predictions** are computed using answer set solver DLV

https://github.com/htran010589/nonmonotonic-rule-mining.git
Experiments

- **Approximated ideal KG**: original KG
- **Available KG**: for every relation randomly remove 20% of facts from approximated ideal KG
- **Horn rules**: \( h(X, Y) \leftarrow p(X, Z), q(Z, Y) \)
- **Exceptions**: \( e_1(X), e_2(Y), e_3(X, Y) \)
- **Predictions** are computed using answer set solver DLV

Examples of revised rules:

Plots of films in a sequel are written by the same writer, unless a film is American

\[ r_1 : \text{writtenBy}(X, Z) \leftarrow \text{hasPredecessor}(X, Y), \text{writtenBy}(Y, Z), \text{not american_film}(X) \]

Spouses of film directors appear on the cast, unless they are silent film actors

\[ r_2 : \text{actedIn}(X, Z) \leftarrow \text{isMarriedTo}(X, Y), \text{directed}(Y, Z), \text{not silent_film_actor}(X) \]

https://github.com/htran010589/nonmonotonic-rule-mining.git
Motivation

Preliminaries

Rule Learning

Exception-awareness

Incompleteness

Rules from Hybrid Sources
Reasonable Rules

People with the same parents are likely siblings.
Reasonable Rules

- People with the same parents are likely siblings
  - \[ \text{conf}(r_1) = |\{ \text{hasSibling}(X, Z) \in G \}| = 2 \]
  - \[ \text{conf}_{PCA}(r_1) = |\{ \text{hasParent}(X, Y), \text{hasChild}(Y, Z) \} \} = 2 \]

Diagram:
- Pete hasSibling Mary
- John hasParent hasChild Mathew hasSibling hasChild Alice
- Ben hasParent hasChild Anna
- Bob hasChild
Reasonable Rules

 ✓ People with the same parents are likely siblings

\[ r_1 : \text{hasSibling}(X, Z) \leftarrow \text{hasParent}(X, Y), \text{hasChild}(Y, Z) \]
Reasonable Rules

People with the same parents are likely siblings

\[ r_1 : \text{hasSibling}(X, Z) \leftarrow \text{hasParent}(X, Y), \text{hasChild}(Y, Z) \]
Reasonable Rules

 ✓ People with the same parents are likely siblings

\[ conf(r_1) = \frac{|\uparrow|}{|\uparrow| + |\bigcirc|} = \frac{2}{4} \]

\[ r_1 : \text{hasSibling}(X, Z) \leftarrow \text{hasParent}(X, Y), \text{hasChild}(Y, Z) \]
Reasonable Rules

✓ People with the same parents are likely siblings

$$r_1 : \text{hasSibling}(X, Z) \leftarrow \text{hasParent}(X, Y), \text{hasChild}(Y, Z)$$

$$\text{conf}(r_1) = \frac{|\triangledown|}{|\triangledown| + |\bigcirc|} = \frac{2}{4}$$

$$\text{conf}_{\text{pca}}(r_1) = \frac{|\triangledown|}{|\{\bigcirc|\text{hasSibling}(X,_)\in G\}|} = \frac{2}{2}$$
Erroneous Rules due to Data Bias

- Pete: hasChild Mary
- Mary: worksAt, educatedAt TUWien
- TUWien: worksAt, educatedAt Mathew, Alice
- Mathew: hasChild Alice
- Alice: worksAt, educatedAt Bob
- Bob: educatedAt MPI
- Anna: worksAt, educatedAt MPI
Erroneous Rules due to Data Bias
Erroneous Rules due to Data Bias

If one is studying in a university where you teach, he/she is your child

\[
\begin{align*}
  r_2 : & \quad \text{hasChild}(X, Z) \leftarrow \text{worksAt}(X, Y), \text{educatedAt}(Z, Y)
\end{align*}
\]
Erroneous Rules due to Data Bias

× If one is studying in a university where you teach, he/she is your child

\[ r_2 : \text{hasChild}(X, Z) \leftarrow \text{worksAt}(X, Y), \text{educatedAt}(Z, Y) \]
Erroneous Rules due to Data Bias

✔ If one is studying in a university where you teach, he/she is your child

\[ \text{conf}(r_2) = \frac{|\text{hasChild}|}{|\text{hasChild}| + |\text{other}|} = \frac{2}{4} \]

\[ \text{conf}_{pca}(r_2) = \frac{|\text{hasChild}|}{|\{\text{hasChild} \in G\}|} = \frac{2}{2} \]

\[ r_2 : \text{hasChild}(X, Z) \leftarrow \text{worksAt}(X, Y), \text{educatedAt}(Z, Y) \]
Exploiting Meta-data in Rule Learning

**Goal:** make use of cardinality constraints on edges of the KG to improve rule learning.

Cardinality Statements

- $\text{num}(p, s)$: Number of outgoing $p$-edges from $s$ in the ideal KG
- $\text{miss}(p, s)$: Number of missing $p$-edges from $s$ in the available KG
- If $\text{miss}(p, s) = 0$, then $\text{complete}(p, s)$, otherwise $\text{incomplete}(p, s)$

In the example:

- $\text{num}(\text{hasChild}, \text{john}) = 3$
- $\text{miss}(\text{hasChild}, \text{john}) = 1$
- $\text{incomplete}(\text{hasChild}, \text{john})$
Cardinality Constraints on Edges

- Mining cardinality assertions from the Web [Mirza et al., 2016]
  - “... John has 2 children ...”

- Estimating recall of KGs by crowd sourcing [Razniewski et al., 2016]
  - 20 % of Nobel laureates in physics are missing

- Predicting completeness in KGs [Galárraga et al., 2017]
  - Add \textit{complete}(john, hasChild) to KG and mine rules
    \textit{complete}(X, hasChild) \leftarrow \textit{child}(X)
Completeness Confidence

$conf_{comp}$: do not penalize rules that predict new facts in incomplete areas

$$conf_{comp}(r) = \frac{|\triangle|}{|\triangle| + |\bigcirc| - npi(r)}$$

- $npi(r)$: number of facts added to incomplete areas by $r$
- Generalizes standard confidence ($miss(r) = 0$)
- Generalizes PCA confidence ($miss(r) \in \{0, +\infty\}$)
Other Completeness-aware Measures

**precision\textsubscript{comp}**: penalize $r$ that predict facts in complete areas

$$\text{precision}\textsubscript{comp}(r) = 1 - \frac{npc(r)}{|\bigtriangleup| + |\bigtriangleup\triangledown|}$$

**recall\textsubscript{comp}**: ratio of missing facts filled by $r$

$$\text{recall}\textsubscript{comp}(r) = \frac{npi(r)}{\sum_s \text{miss}(h, s)}$$

**dir\textunderscore metric**: proportion of predictions in complete and incomplete parts

$$\text{dir\textunderscore metric}(r) = \frac{npi(r) -npc(r)}{2 \cdot (npi(r) +npc(r))} + 0.5$$

**wdm**: weighted combination of confidence and directional metric

$$\text{wdm}(r) = \beta \cdot \text{conf}(r) + (1 - \beta) \cdot \text{dir\textunderscore metric}(r)$$

https://github.com/Tpt/CARL.git
Motivation

Preliminaries

Rule Learning

Exception-awareness

Incompleteness

Rules from Hybrid Sources
Ideal KG

\[ \mu(r, G^i) \]: measure quality of the rule \( r \) on \( G^i \)
\[ \mu(r, G^i) : \text{measure quality of the rule } r \text{ on } G^i, \text{ but } G^i \text{ is unknown} \]
Probabilistic Reconstruction of Ideal KG

$$\mu(r, \mathcal{G}_p^i): \text{measure quality of } r \text{ on } \mathcal{G}_p^i$$
Hybrid Rule Measure

$$\mu(r, G^i_p) = (1 - \lambda) \times \mu_1(r, G) + \lambda \times \mu_2(r, G^i_p)$$
Hybrid Rule Measure

\[ \mu(r, G^i_p) = (1 - \lambda) \times \mu_1(r, G) + \lambda \times \mu_2(r, G^i_p) \]

- \( \lambda \in [0..1] \): weighting factor

- \( \mu_1 \): descriptive quality of rule \( r \) over the available KG \( G \)
  - confidence
  - PCA confidence

- \( \mu_2 \): predictive quality of \( r \) relying on \( G^i_p \) (probabilistic reconstruction of the ideal KG \( G^i \))
**Intuition:** For $\langle s, p, o \rangle$ in KG, find $s, p, o$ such that $s + p \approx o$.

The “error of translation” of a true KG fact should be smaller by a certain margin than the “error of translation” of an out-of-KG one.
**Intuition:** For \( \langle s, p, o \rangle \) in KG, find \( s, p, o \) such that \( s + p \approx o \)

The “error of translation” of a true KG fact should be smaller by a certain margin than the “error of translation” of an out-of-KG one
**Intuition:** For \( \langle s, p, o \rangle \) in KG, find \( s, p, o \) such that \( s + p \approx o \)

The “error of translation” of a true KG fact should be smaller by a certain margin than the “error of translation” of an out-of-KG one.
**Intuition:** For \( \langle s, p, o \rangle \) in KG, find \( s, p, o \) such that \( s + p \approx o \)

The “error of translation” of a true KG fact should be smaller by a certain margin than the “error of translation” of an out-of-KG one.
Embedding-based Rule Learning

Prune rule search space relying on

- novel hybrid embedding-based rule measure
Evaluation Setup

- **Datasets:**
  - FB15K: 592K facts, 15K entities and 1345 relations
  - Wiki44K: 250K facts, 44K entities and 100 relations

- **Training graph** $G$: remove 20% from the available KG

- **Embedding models** $G_p^i$:
  - TransE [Bordes et al., 2013], HoIE [Nickel et al., 2016]
  - With text: SSP [Xiao et al., 2017]

- **Goals:**
  - Evaluate effectiveness of our hybrid rule measure
    \[
    \mu(r, G^i) = (1 - \lambda) \times \mu_1(r, G) + \lambda \times \mu_2(r, G^i)
    \]
  - Compare against state-of-the-art rule learning systems

https://github.com/hovinhthinh/RuLES.git
Evaluation of Hybrid Rule Measure

- **top_5**
- **top_10**
- **top_20**
- **top_50**
- **top_100**
- **top_200**

(a) Conf-HoIE
(b) Conf-SSP
(c) PCA-SSP

• Positive impact of embeddings in all cases for \(\lambda = 0.3\).

• Note: in (c) comparison to AMIE [Galarraga et al., 2015] (\(\lambda = 0\)).
Evaluation of Hybrid Rule Measure

- Positive impact of embeddings in all cases for $\lambda = 0.3$
- **Note:** in (c) comparison to AMIE [Galarraga et al., 2015] ($\lambda = 0$)

![Graphs showing evaluation of hybrid rule measure](image-url)
Example Rules

Examples of rules learned from Wikidata

Script writers stay the same throughout a sequel, but not for TV series

\[ r_1 : \text{scriptwriterOf}(X, Y) \leftarrow \text{precededBy}(Y, Z), \text{scriptwriterOf}(X, Z), \text{not isA}(Z, \text{tvSeries}) \]

Nobles are typically married to nobles, but not in the case of Chinese dynasties

\[ r_2 : \text{nobleFamily}(X, Y) \leftarrow \text{spouse}(X, Z), \text{nobleFamily}(Z, Y), \text{not isA}(Y, \text{chineseDynasty}) \]
Rule-based Fact Checking

Query: Hawking influencedBy Dirac?

Inference Rules:
- influencedBy(X,Y) ← hasAcademicAdvisor(X,Y);
- influencedBy(X,Y) ← scientist(X), wrote(X,Z), praises(Z,Y);

ExFaKT

Explanations

KG:
- wrote(Hawking, "Dirac: The Man and His Work, 1998");
- scientist(Hawking);

Text:
- "Dirac has done more than anyone this century, with the exception of Einstein..."


Rule-based Fact Checking

Motivation
Preliminaries
Rule Learning
Exception-awareness
Incompleteness
Rules from Hybrid Sources

Summary

• Classical rule learning methods from ILP
• Rule learning from Knowledge Graphs
• Exploiting embeddings to guide rule learning
• Rule-based fact checking
Summary

• Classical rule learning methods from ILP
• Rule learning from Knowledge Graphs
• Exploiting embeddings to guide rule learning
• Rule-based fact checking

Interested in PhD/internship?
At BCAI we are hiring!

jobs.smartrecruiters.com/BoschGroup/743999698936956-phd-combined-reasoning-and-learning-approaches
Huge Thanks!

- For collaborations on the presented work:
  - Mohamed Gad-elrab, Thinh Vinh Ho, Hai Dang Tran, Thomas Pellissier-Tanon, Gerhard Weikum, Jacopo Urbani, Evgeny Kharlamov, Francesca A. Lisi, Simon Razniewski, Paramita Mirza

- For fruitful discussions and/or making slides available online:
  - Thomas Eiter, Stephen Muggleton, Luc De Raedt, Fabian Suchanek


References II

Jens Lehmann.
DL-Learner: Learning concepts in description logics.

Francesca A. Lisi.
Inductive Logic Programming in Databases: From Datalog to DL+log.

Paramita Mirza, Simon Razniewski, and Werner Nutt.
Expanding wikidata’s parenthood information by 178%, or how to mine relation cardinality information.
In ISWC 2016 Posters & Demos, 2016.

Stephen Muggleton.
Inductive logic programming.

Stephen Muggleton.
Inverse entailment and progol.

Maximilian Nickel, Lorenzo Rosasco, and Tomaso A. Poggio.
Holographic embeddings of knowledge graphs.
In AAAI, 2016.

Luc De Raedt and Saso Dzeroski.
First-order jk-clausal theories are pac-learnable.

Simon Razniewski, Fabian M. Suchanek, and Werner Nutt.
But what do we actually know?
Chiaki Sakama.  
Induction from answer sets in nonmonotonic logic programs.  

Ehud Y. Shapiro.  
Inductive inference of theories from facts.  

Han Xiao, Minlie Huang, Lian Meng, and Xiaoyan Zhu.  
SSP: semantic space projection for knowledge graph embedding with text descriptions.  