Soft Logic
in Computer Science

Seminar 236801

Computer Science Department
Technion – Israel Institute of Technology
Seminar Goals

• Get familiar with soft/probabilistic variations of logic
  ▪ What does it mean for a logical formula to be “probably true”?  
  ▪ Logic in machine learning (SRL – Statistical Relational Learning)  
  ▪ Statistical interpretations of database queries (logic)  
    ▪ Logic based solution ≠ strict rules
• Experience reading a scientific paper of a top international venue
• Presentation skills
  ▪ Build the story  
  ▪ Prepare slides  
  ▪ Frontal presentation
Scientific Background
General Setting

• **Signature** consists of predicates (relation names)
  - A predicate has a fixed sequence of attributes, sometimes from specialized domains
  - Extensional database (EDB): predicates known in advance
    - Person(id), Friends(id1,id2,), Address(id,adrs)
  - Intentional database (IDB): inferred predicates
    - HasFriendsInAddress(id,adrs)

• **Rules**: logic (1st order) over the predicates
  - \( \forall x,y \ [\text{Friends}(x,y) \rightarrow \text{Friends}(y,x)] \)

• **Instance / database / knowledge base** over the signature: finite table for each EDB predicate
  - Person(54), Person(31), Friends(54,31), Address(54,“14 Hassan Shuqri”)
Example 1: Burglary (Judea Pearl)

Scenario: We track house alarms going off; wish to make a good guess of whether a house is robbed or there is an earthquake.

Inference rules: Burglary causes an alarm, earthquake causes an alarm, some alarms may be broken and not go off.

<table>
<thead>
<tr>
<th>Predicates</th>
<th>Rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Alarm(x)</td>
<td>• Burglary(x) → Alarm(x)</td>
</tr>
<tr>
<td>• BrokenAlarm(x)</td>
<td>• Earthquake() → Alarm(x)</td>
</tr>
<tr>
<td>• Burglary(x)</td>
<td>• BrokenAlarm(x) → ¬Alarm(x)</td>
</tr>
<tr>
<td>• Earthquake()</td>
<td></td>
</tr>
</tbody>
</table>
Example 2: Entity Resolution

Scenario: Detect duplicate accounts in a social network

Inference rules: Similarity in name and addresses, equivalence (reflexivity, symmetry, transitivity)

<table>
<thead>
<tr>
<th>Predicates</th>
<th>Rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>Person(id,fn,ln,adrs)</td>
<td>• Person(i1,f1,a1) ∧ Person(i2,f1,a2) → Same(i1,i2)</td>
</tr>
<tr>
<td>Similar(str1,str2)</td>
<td>• Person(i1,f1,l1,a1) ∧ Person(i2,f2,l2,a2) ∧ Similar(f1,f2) ∧ Similar(l1,l2) ∧ Close(a1,s2) → Same(i1,i2)</td>
</tr>
<tr>
<td>SameCity(adrs1,adrs2)</td>
<td>• Person(i1,f1,l1,a1) ∧ Person(i2,f2,l2,a2) ∧ ¬SameCountry(a1,a2) → ¬Same(i1,i2)</td>
</tr>
<tr>
<td>SameCountry(adrs1,adrs2)</td>
<td>• Person(i,f,l,a) → Same(i,i)</td>
</tr>
<tr>
<td>Same(id1,id2)</td>
<td>• Same(i1,i2) → Same(i2,i1)</td>
</tr>
<tr>
<td></td>
<td>• Same(i1,i2) ∧ Same(i2,i3) → Same(i1,i3)</td>
</tr>
</tbody>
</table>
Example 3: Smoking

Scenario: We wish to determine who is in high risk of cancer, given some information about smokers and friendships

Inference rules: Smoking causes cancer; friends have similar smoking habits

<table>
<thead>
<tr>
<th>Predicates</th>
<th>Rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>Friends(x,y)</td>
<td>• Smokes(x) → Cancer(x)</td>
</tr>
<tr>
<td>Smokes(x)</td>
<td>• Friends(x,y) → (Smokes(x) ↔ Smokes(y))</td>
</tr>
<tr>
<td>Cancer(x)</td>
<td></td>
</tr>
</tbody>
</table>
3 Semantics of Probabilistic Logic

1. Probabilistic Databases (a.k.a. Distributional Semantics)
2. Probabilistic Programming
3. Template Grounding
Probabilistic Databases

Probabilistic Programming

Template Grounding
Probabilistic Databases

Probabilistic Programming

Template Grounding
Semantics 1: Probabilistic Databases

- Idea: Database (EDB) facts are probabilistic
  - Example: each tuple exists with some probability, independently of others
  - Hence, we get a probability space over possible databases
- The rules are ordinary logic, usual semantics
- But the outcome is probabilistic, since (conceptually) each random input database gives rise to a different outcome
Probabilistic Databases: Example

Inference tasks:
- Marginal probability
  \[ \Pr(\text{Earthquake} \mid \text{observed alarms}) = ? \]
- Maximum likelihood
  \( \text{Max-probability random DB, given observed alarms?} \)

- ShouldAlarm\((x) \leftarrow \text{Earthquake}() \)
- ShouldAlarm\((x) \leftarrow \text{Burglary}(x) \)
- Alarm\((x) \leftarrow \neg \text{BrokenAlarm}(x), \text{ShouldAlarm}(x) \)

Prolog / Datalog
Probabilistic Databases

Rules

Probabilistic Databases

DB

Program w/ randomness

Probabilistic Programming

DB

Rules

Statistical Model

Probabilistic Programming

Probabilistic Databases

Template Grounding
Semantics 2: Probabilistic Programming

• Idea: Impettrative (procedural) program over the EDB facts
  ▪ Constructs (IDB) logical facts
  ▪ Allows to make random choices / generate random numbers
• Similar to the traditional concept of randomized algorithms
Probabilistic Programming: Example

1. House(h1), House(h2), House(h3)
2. EarthQuake $\sim$ Flip[0.001]
3. Burglary[$x$] $\sim$ Flip[0.01] ← House(x)
4. BrokenAlarm[$x$] $\sim$ Flip[0.05] ← House(x)
5. Alarm[$x$] $\sim$
    
    if ( (BrokenAlarm[$x$]=0 $\land$ Burglary[$x$]=1) $\lor$ EarthQuake )
    then Flip[0.9]
    else Flip[0.01]
Probabilistic Programming for Advancing Machine Learning (PPAML)

Dr. Suresh Jagannathan

Probabilistic Databases

Probabilistic Programming

Template Grounding

Probabilistic Databases

Probabilistic Programming

Probabilistic Programming
Semantics 3: Template Grounding

• A (ground) fact is a possible tuple (values from proper domains)
  ▪ Each attribute has a finite domain of possible values
• A fact is viewed as a Boolean (true/false) random variable
• Possible world = assignment of true/false to every possible fact
• Rules have free (unbound) variables
• In a possible world, a ground rule can be either true or false
• Result: statistical model, every ground rule $g$ contributes a component/factor, depending on the satisfaction of $g$
Example: Markov Logic Network [Domingos+]

\[ \Pr(\text{world}) = \exp( \sum_{\text{rules } r} \sum_{\text{groundings of } r \text{ true in } \text{world}} \text{weight}(r)) / Z \]

<table>
<thead>
<tr>
<th>Predicates</th>
<th>Rules</th>
<th>Domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>Friends(x,y)</td>
<td>2.0: Smokes(x) → Cancer(x)</td>
<td>Aria, Anna, Alla, Bob, Barney, Bill</td>
</tr>
<tr>
<td>Smokes(x)</td>
<td>1.6: Friends(x,y) → (Smokes(x)) ↔ Smokes(y))</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Predicates</th>
<th>Rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>Person(id,fn,ln,adrs)</td>
<td>3.0: Person(i1,f,l,a1) ∧ Person(i2,f,l,a2) → Same(i1,i2)</td>
</tr>
<tr>
<td>Similar(str1,str2)</td>
<td>0.5: Person(i1,f1,l1,a1) ∧ Person(i2,f2,l2,a2) ∧ Similar(f1,f2) ∧ Similar(l1,l2) ∧ \ Close(a1,s2) → Same(i1,i2)</td>
</tr>
<tr>
<td>SameCity(adrs1,adrs2)</td>
<td>4.5: Person(i1,f1,l1,a1) ∧ Person(i2,f2,l2,a2) ∧ ¬SameCountry(a1,a2) → ¬Same(i1,i2)</td>
</tr>
<tr>
<td>SameCountry(adrs1,adrs2)</td>
<td>hard: Person(i1,f,l,a1) → Same(i1,i1)</td>
</tr>
<tr>
<td>Same(id1,id2)</td>
<td>hard: Same(i1,i2) → Same(i2,i1)</td>
</tr>
<tr>
<td></td>
<td>hard: Same(i1,i2) ∧ Same(i2,i3) → Same(i1,i3)</td>
</tr>
</tbody>
</table>
Fuzzy Logic

• **Boolean logic**: Variable/facts take values in \( \{0,1\} \)
  - Consequently, formulas take values in \( \{0,1\} \)
  - Example: the soup is either hot (1) or cold (0)

• **Fuzzy logic**: Variable/facts take values in the interval \([0,1]\)
  - Consequently, formulas take values in the interval \([0,1]\)
  - Example: the soup has a hotness degree (temperature) in \([0,1]\)
  - Special case of *many-valued logic*

• Various forms of fuzzy logic exist, and differ on the interpretation of the logical operators
Example: Łukasiewicz Logic

<table>
<thead>
<tr>
<th>Operator</th>
<th>Notation</th>
<th>Truth Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weak conjunction</td>
<td>$x \wedge y$</td>
<td>$\min(x, y)$</td>
</tr>
<tr>
<td>Strong conjunction</td>
<td>$x &amp; y$</td>
<td>$\max(0, x+y-1)$</td>
</tr>
<tr>
<td>Weak disjunction</td>
<td>$x \lor y$</td>
<td>$\max(x, y)$</td>
</tr>
<tr>
<td>Strong disjunction</td>
<td>$x \mid y$</td>
<td>$\min(1, x+y)$</td>
</tr>
<tr>
<td>Negation</td>
<td>$\neg x$</td>
<td>$1-x$</td>
</tr>
<tr>
<td>Implication</td>
<td>$x \rightarrow y$</td>
<td>$\min(1, 1-x+y)$</td>
</tr>
</tbody>
</table>

Łukasiewicz Logic in Probabilistic Soft Logic (PSL) [Getoor+]:

$$\Pr(\text{world}) = \exp\left( \sum_{\text{rules } r} \sum_{\text{groundings } g \text{ of } r} \text{weight}(r) \times \text{truth}_{\text{world}(g)} \right) / Z$$

Normalization

Assignment of truth values in [0,1] to ground facts
Research Challenges (All Paradigms)

- Models: syntax & semantics
- Inference algorithms, heuristics, complexity
- Applications to pursue specific tasks
Requirements
Seminar Requirements

• Paper presentation
• Preparation meeting with the lecturer before presentation
• Mandatory attendance
• Weekly preparation
  ▪ Answer background questions on the presentations of next lecture (later)

• Grading:
  ▪ Presentation: %80
  ▪ Weekly preparation: %20
Selecting a Paper for This Seminar

• List of papers on the course website
• You can select a paper on your own; has to be:
  ▪ Related to seminar topic
  ▪ From a top conference/journal
  ▪ Approved by the lecturer
• Email to me a ranked list of 3 papers by next week
• I’ll assign the paper and date
  ▪ You may get a paper outside your list
  ▪ Distinct paper per student
Presentation

• 45 minutes + 5 minutes QA
• Individual task
• Computer slides mandatory
  ▪ Use standard presentation software: MS PowerPoint, Apple Keynote, Latex Beamer, Apache OpenOffice, Google Slides
  ▪ Will be published on the course website, publically available
  ▪ English
• At least 7 days before the presentation, we meet one-on-one
  ▪ Deliver the presentation
  ▪ Get feedback
  ▪ Improve before the actual lecture
Lecture’s Grading Components

- Story, Slides, Delivery
- Lecture quality, interest, flow
- Quality of presentation slides
- Speaker’s understanding of the paper
  - In particular, ability to provide intuitive explanations beyond the details
- Depth & volume of covered material
- Problems that arise in preparation meeting have no impact
External Resources

• Related content from referenced papers: highly regarded
• Material/ideas from external presentations: allowed
• You should disclose all material used
  ▪ I should completely understand what is yours and what is taken from external resources
  ▪ Do not overdo; the vast majority of content should be yours
• Resources: Google, Google Scholar, DBLP
• Useful search tricks: quotes ; filetype:pdf ; filetype:ppt
• (Online Illustration)
Background Question (Per Presentation)

a) Which of the 3 semantics of prob. logic is adopted in the paper?
   1. Probabilistic Databases
   2. Probabilistic Programming
   3. Template Grounding
   4. None / NA

b) What type of a paper is it?
   1. Proposing a novel framework / tool / language
   2. Solving a problem in an existing framework

c) Describe in 180-220 words:
   - 1st type: The framework’s goal and main novelty (claimed)
   - 2nd type: The attacked problem and main novel results (claimed)
## Preliminary Plan

<table>
<thead>
<tr>
<th>Week</th>
<th>Date</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>30/10</td>
<td>Intro</td>
</tr>
<tr>
<td>2</td>
<td>06/11</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>13/11</td>
<td>Distributional + Prob DB</td>
</tr>
<tr>
<td>4</td>
<td>20/11</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>27/11</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>04/12</td>
<td>Probabilistic Programming</td>
</tr>
<tr>
<td>7</td>
<td>11/12</td>
<td>Template Grounding</td>
</tr>
<tr>
<td>8</td>
<td>18/12</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>No Lecture (Hanukkah)</td>
</tr>
<tr>
<td>9</td>
<td>01/01</td>
<td>Template Grounding</td>
</tr>
<tr>
<td>10</td>
<td>08/01</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>15/01</td>
<td>Techniques / Applications / Others</td>
</tr>
<tr>
<td>12</td>
<td>22/01</td>
<td></td>
</tr>
</tbody>
</table>

Semester Ends 26/01
Good Luck!