At the Confluence of Logic and Learning

Guy Van den Broeck

Dagstuhl

September 3, 2019
Outline

1. The AI dilemma: logic vs. learning
2. Deep learning with symbolic knowledge
3. Efficient reasoning during learning
4. New machine learning formalisms
5. Statistical relational learning (tutorial)
Outline

1. The AI dilemma: logic vs. learning
2. Deep learning with symbolic knowledge
3. Efficient reasoning during learning
4. New machine learning formalisms
5. Statistical relational learning (tutorial)
The AI Dilemma

Pure Logic       Pure Learning
The AI Dilemma

- Slow thinking: deliberative, cognitive, model-based, extrapolation
- Amazing achievements until this day
- "Pure logic is brittle"
  noise, uncertainty, incomplete knowledge, …
The AI Dilemma

Pure Logic

- Fast thinking: instinctive, perceptive, model-free, interpolation
- Amazing achievements recently

“Pure learning is brittle”

bias, algorithmic fairness, interpretability, explainability, adversarial attacks, unknown unknowns, calibration, verification, missing features, missing labels, data efficiency, shift in distribution, general robustness and safety fails to incorporate a sensible model of the world

Pure Learning
The FALSE AI Dilemma

So all hope is lost?

Probabilistic World Models

- Joint distribution $P(X)$
- Wealth of representations: can be causal, relational, etc.
- Knowledge + data
- Reasoning + learning
Then why isn’t everything solved?

What did we gain?
What did we lose along the way?
A New Synthesis of Learning and Reasoning
Outline

1. The AI dilemma: logic vs. learning
2. Deep learning with symbolic knowledge
3. Efficient reasoning during learning
4. New machine learning formalisms
5. Statistical relational learning (tutorial)
6. Lifted probabilistic inference
Motivation: Vision

We also connect all pairs of identity nodes $y_{t,i}$ and $y_{t,j}$ if they appear in the same time $t$. We then introduce an edge potential that enforces mutual exclusion:

$$\psi_{\text{mutex}}(y_{t,i}, y_{t,j}) = \begin{cases} 1 & \text{if } y_{t,i} \neq y_{t,j} \\ 0 & \text{otherwise} \end{cases}$$

(5)

This potential specifies the constraint that a player can appear only once in a frame. For example, if the $i$-th detection $y_{t,i}$ has been assigned to Bryant, $y_{t,j}$ cannot have the same identity because Bryant is impossible to appear twice in a frame.

Motivation: Robotics

The method developed in this paper can be used in a broad variety of semantic mapping and object manipulation tasks, providing an efficient and effective way to incorporate collision constraints into a recursive state estimator, obtaining optimal or near-optimal solutions.

[Wong, L. L., Kaelbling, L. P., & Lozano-Perez, T., Collision-free state estimation. ICRA 2012]
Motivation: Language

• Non-local dependencies:
  “At least one verb in each sentence”

• Sentence compression
  “If a modifier is kept, its subject is also kept”

• NELL ontology and rules

… and much more!

[Chang, M., Ratinov, L., & Roth, D. (2008). Constraints as prior knowledge],
… and many many more!
Motivation: Deep Learning

Motivation: Deep Learning

DeepMind’s latest technique uses external memory to solve tasks that require logic and reasoning — a step toward more human-like AI.

... but ...

... optimal planner recalculating a shortest path to the end node. To ensure that the network always moved to a valid node, the output distribution was renormalized over the set of possible triples outgoing from the current node. The performance it also received input triples during the answer phase, indicating the actions chosen on the previous time-step.

Knowledge vs. Data

• Where did the world knowledge go?
  – Python scripts
    • Decode/encode cleverly
    • Fix inconsistent beliefs
  – Rule-based decision systems
  – Dataset design
  – “a big hack” (with author’s permission)

• In some sense we went backwards
Less principled, scientific, and intellectually satisfying ways of incorporating knowledge
1. Must take at least one of Probability (P) or Logic (L).
2. Probability (P) is a prerequisite for AI (A).
3. The prerequisites for KR (K) is either AI (A) or Logic (L).
Learning with Symbolic Knowledge

Today’s machine learning tools don’t take knowledge as input!
Deep Learning with Symbolic Knowledge

Data + Constraints

Learn

Deep Neural Network

Input

Neural Network

Output

Output is probability vector $\mathbf{p}$, not Boolean logic!
**Semantic Loss**

**Q:** How close is output \( p \) to satisfying constraint \( \alpha \)?

**Answer:** Semantic loss function \( L(\alpha, p) \)

- **Axioms, for example:**
  - If \( \alpha \) constrains to one label, \( L(\alpha, p) \) is cross-entropy
  - If \( \alpha \) implies \( \beta \) then \( L(\alpha, p) \geq L(\beta, p) \) \((\alpha \text{ more strict})\)

- **Implied Properties:**
  - If \( \alpha \) is equivalent to \( \beta \) then \( L(\alpha, p) = L(\beta, p) \) \(\text{SEMANTIC Loss!}\)
  - If \( p \) is Boolean and satisfies \( \alpha \) then \( L(\alpha, p) = 0 \)
Semantic Loss: Definition

**Theorem:** Axioms imply unique semantic loss:

\[
L^s(\alpha, p) \propto - \log \sum_{x \models \alpha} \prod_{i : x \models X_i} p_i \prod_{i : x \models \neg X_i} (1 - p_i)
\]

- **Probability of getting state** \(x\) **after flipping coins with probabilities** \(p\)
- **Probability of satisfying** \(\alpha\) **after flipping coins with probabilities** \(p\)
Simple Example: Exactly-One

• Data must have some label
  
  *We agree this must be one of the 10 digits:*

• Exactly-one constraint
  
  → For 3 classes:

  \[
  \begin{cases}
  x_1 \lor x_2 \lor x_3 \\
  \neg x_1 \lor \neg x_2 \\
  \neg x_2 \lor \neg x_3 \\
  \neg x_1 \lor \neg x_3
  \end{cases}
  \]

• Semantic loss:

  \[
  L^s(\text{exactly-one, } p) \propto - \log \sum_{i=1}^{n} p_i \prod_{j=1, j \neq i}^{n} (1 - p_j)
  \]

Only \(x_i = 1\) after flipping coins

Exactly one true \(x\) after flipping coins
Semi-Supervised Learning

• Intuition: Unlabeled data must have some label
  Cf. entropy minimization, manifold learning

• Minimize exactly-one semantic loss on unlabeled data

Train with

\[ \text{existing loss} + w \cdot \text{semantic loss} \]
Experimental Evaluation

### Competitive with state of the art in semi-supervised deep learning

<table>
<thead>
<tr>
<th>Accuracy % with # of used labels</th>
<th>100</th>
<th>1000</th>
<th>ALL</th>
</tr>
</thead>
<tbody>
<tr>
<td>AtlasRBF (Pitelis et al., 2014)</td>
<td>91.9 (±0.95)</td>
<td>96.32 (±0.12)</td>
<td>98.69</td>
</tr>
<tr>
<td>Deep Generative (Kingma et al., 2014)</td>
<td>96.67 (±0.14)</td>
<td>97.60 (±0.02)</td>
<td>99.04</td>
</tr>
<tr>
<td>Virtual Adversarial (Miyato et al., 2016)</td>
<td>97.67</td>
<td>98.64</td>
<td>99.36</td>
</tr>
<tr>
<td>Ladder Net (Rasmus et al., 2015)</td>
<td><strong>98.94 (±0.37)</strong></td>
<td><strong>99.16 (±0.08)</strong></td>
<td>99.43 (±0.02)</td>
</tr>
<tr>
<td>Baseline: MLP, Gaussian Noise</td>
<td>78.46 (±1.94)</td>
<td>94.26 (±0.31)</td>
<td>99.34 (±0.08)</td>
</tr>
<tr>
<td>Baseline: Self-Training</td>
<td>72.55 (±4.21)</td>
<td>87.43 (±3.07)</td>
<td></td>
</tr>
<tr>
<td>Baseline: MLP with Entropy Regularizer</td>
<td>96.27 (±0.64)</td>
<td>98.32 (±0.34)</td>
<td>99.37 (±0.12)</td>
</tr>
<tr>
<td>MLP with Semantic Loss</td>
<td>98.38 (±0.51)</td>
<td>98.78 (±0.17)</td>
<td>99.36 (±0.02)</td>
</tr>
</tbody>
</table>

### Outperforms SoA!

<table>
<thead>
<tr>
<th>Accuracy % with # of used labels</th>
<th>100</th>
<th>500</th>
<th>1000</th>
<th>ALL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ladder Net (Rasmus et al., 2015)</td>
<td>81.46 (±0.64)</td>
<td>85.18 (±0.27)</td>
<td>86.48 (±0.15)</td>
<td>90.46</td>
</tr>
<tr>
<td>Baseline: MLP, Gaussian Noise</td>
<td>69.45 (±2.03)</td>
<td>78.12 (±1.41)</td>
<td>80.94 (±0.84)</td>
<td>89.87</td>
</tr>
<tr>
<td>MLP with Semantic Loss</td>
<td><strong>86.74 (±0.71)</strong></td>
<td><strong>89.49 (±0.24)</strong></td>
<td><strong>89.67 (±0.09)</strong></td>
<td><strong>89.81</strong></td>
</tr>
</tbody>
</table>

Same conclusion on CIFAR10

<table>
<thead>
<tr>
<th>Accuracy % with # of used labels</th>
<th>4000</th>
<th>ALL</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNN Baseline in Ladder Net</td>
<td>76.67 (±0.61)</td>
<td>90.73</td>
</tr>
<tr>
<td>Ladder Net (Rasmus et al., 2015)</td>
<td>79.60 (±0.47)</td>
<td></td>
</tr>
<tr>
<td>Baseline: CNN, Whitening, Cropping</td>
<td>77.13</td>
<td>90.96</td>
</tr>
<tr>
<td>CNN with Semantic Loss</td>
<td><strong>81.79</strong></td>
<td>90.92</td>
</tr>
</tbody>
</table>
Outline

1. The AI dilemma: logic vs. learning
2. Deep learning with symbolic knowledge
3. Efficient reasoning during learning
4. New machine learning formalisms
5. Statistical relational learning (tutorial)
But what about *real* constraints?

- Path constraint

- Example: 4x4 grids
  \[2^{24} = 184 \text{ paths} + 16,777,032 \text{ non-paths}\]

- Easily encoded as logical constraints 😊

---

[Nishino et al., Choi et al.]
How to Compute Semantic Loss?

• In general: \#P-hard 😞

\[ L^s(\alpha, p) \propto -\log \sum_{x|\models \alpha} \prod_{i:x|\models X_i} p_i \prod_{i:x|\models \neg X_i} (1 - p_i) \]
Reasoning Tool: Logical Circuits

Representation of logical sentences:

\[(C \land \neg D) \lor (\neg C \land D)\]

C XOR D
Reasoning Tool: Logical Circuits

Representation of logical sentences:

Input:

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
<td>C</td>
<td>D</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
Tractable for Logical Inference

• Is there a solution? (SAT)
  – SAT(α ∨ β) iff SAT(α) or SAT(β)  (always)
  – SAT(α ∧ β) iff ???
Decomposable Circuits

B, C, D
Tractable for Logical Inference

• Is there a solution? (SAT) ✓
  – SAT(\(\alpha \lor \beta\)) iff SAT(\(\alpha\)) or SAT(\(\beta\)) (always)
  – SAT(\(\alpha \land \beta\)) iff SAT(\(\alpha\)) and SAT(\(\beta\)) (decomposable)

• How many solutions are there? (#SAT)

• Complexity linear in circuit size 😊
Deterministic Circuits

C XOR D

C ⇔ D
How many solutions are there? (**#SAT**)
Tractable for Logical Inference

• Is there a solution? (SAT) ✓
• How many solutions are there? (#SAT) ✓
• Conjoin, disjoin, equivalence checking, etc. ✓
• Complexity linear in circuit size 😊

• Compilation into circuit by
  – ↓ exhaustive SAT solver
  – ↑ conjoin/disjoin/negate

[Darwiche and Marquis, JAIR 2002]
How to Compute Semantic Loss?

- In general: #P-hard 😞
- With a logical circuit for $\alpha$: Linear 😊
- Example: exactly-one constraint:

$L(\alpha, p) = L(\alpha, p) = - \log(\ )$

Why? Decomposability and determinism!
Predict Shortest Paths

Add semantic loss for path constraint

<table>
<thead>
<tr>
<th>Test accuracy %</th>
<th>Coherent</th>
<th>Incoherent</th>
<th>Constraint</th>
</tr>
</thead>
<tbody>
<tr>
<td>5-layer MLP</td>
<td>5.62</td>
<td>85.91</td>
<td>6.99</td>
</tr>
<tr>
<td>Semantic loss</td>
<td>28.51</td>
<td>83.14</td>
<td>69.89</td>
</tr>
</tbody>
</table>

Is prediction the shortest path?  
This is the real task!

Are individual edge predictions correct?

Is output a path?

(same conclusion for predicting sushi preferences, see paper)
Conclusions 1

• Knowledge is (hidden) everywhere in ML
• Semantic loss makes logic differentiable
• Performs well semi-supervised
• Requires hard reasoning in general
  – Reasoning can be encapsulated in a circuit
  – No overhead during learning
• Performs well on structured prediction
• A little bit of reasoning goes a long way!
1. The AI dilemma: logic vs. learning
2. Deep learning with symbolic knowledge
3. Efficient reasoning during learning
4. **New machine learning formalisms**
5. Statistical relational learning (tutorial)
Another False Dilemma?

Classical AI Methods

- Hungry?
- $25?
- Sleep?
- Restaurant?

Sleep?

Clear Modeling Assumption
Well-understood

Neural Networks

“Black Box”
Empirical performance
Probabilistic Circuits

\[ \Pr(A, B, C, D) = 0.096 \]

Input:

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
Properties, Properties, Properties!

• Read conditional independencies from structure
• Interpretable parameters (XAI) (conditional probabilities of logical sentences)
• Closed-form parameter learning
• Efficient reasoning (linear 😊)
  – Computing conditional probabilities $\Pr(x \mid y)$
  – MAP inference: most-likely assignment to $x$ given $y$
  – Even much harder tasks: expectations, KLD, entropy, logical queries, decision making queries, etc.
# Probabilistic Circuits: Performance

**Density estimation benchmarks: tractable vs. intractable**

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Best Circuit</th>
<th>BN</th>
<th>MADE</th>
<th>VAE</th>
</tr>
</thead>
<tbody>
<tr>
<td>nltcs</td>
<td>-5.99</td>
<td>-6.02</td>
<td>-6.04</td>
<td>-5.99</td>
</tr>
<tr>
<td>msnbc</td>
<td>-6.04</td>
<td>-6.04</td>
<td>-6.06</td>
<td>-6.09</td>
</tr>
<tr>
<td>plants</td>
<td>-11.84</td>
<td>-12.65</td>
<td>12.32</td>
<td>-12.34</td>
</tr>
<tr>
<td>audio</td>
<td>-39.39</td>
<td>-40.50</td>
<td>-38.95</td>
<td>-38.67</td>
</tr>
<tr>
<td>jester</td>
<td>-51.29</td>
<td>-51.07</td>
<td>-52.23</td>
<td>-51.54</td>
</tr>
<tr>
<td>netflix</td>
<td>-55.71</td>
<td>-57.02</td>
<td>-55.16</td>
<td>-54.73</td>
</tr>
<tr>
<td>accidents</td>
<td>-26.89</td>
<td>-26.32</td>
<td>-26.42</td>
<td>-29.11</td>
</tr>
<tr>
<td>retail</td>
<td>-10.72</td>
<td>-10.87</td>
<td>-10.81</td>
<td>-10.83</td>
</tr>
<tr>
<td>pumbs*</td>
<td>-22.15</td>
<td>-21.72</td>
<td>-22.3</td>
<td>-25.16</td>
</tr>
<tr>
<td>dna</td>
<td>-79.88</td>
<td>-80.65</td>
<td>-82.77</td>
<td>-94.56</td>
</tr>
<tr>
<td>Kosarek</td>
<td>-10.52</td>
<td>-10.83</td>
<td>-</td>
<td>-10.64</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Best Circuit</th>
<th>BN</th>
<th>MADE</th>
<th>VAE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Book</td>
<td>-33.82</td>
<td>-36.41</td>
<td>-33.95</td>
<td>-33.19</td>
</tr>
<tr>
<td>movie</td>
<td>-50.34</td>
<td>-54.37</td>
<td>-48.7</td>
<td>-47.43</td>
</tr>
<tr>
<td>webkb</td>
<td>-149.20</td>
<td>-157.43</td>
<td>-149.59</td>
<td>-146.9</td>
</tr>
<tr>
<td>cr52</td>
<td>-81.87</td>
<td>-87.56</td>
<td>-82.80</td>
<td>-81.33</td>
</tr>
<tr>
<td>c20ng</td>
<td>-151.02</td>
<td>-158.95</td>
<td>-153.18</td>
<td>-146.90</td>
</tr>
<tr>
<td>bbc</td>
<td>-229.21</td>
<td>-257.86</td>
<td>-242.40</td>
<td>-240.94</td>
</tr>
<tr>
<td>ad</td>
<td>-14.00</td>
<td>-18.35</td>
<td>-13.65</td>
<td>-18.81</td>
</tr>
</tbody>
</table>

**Tractable Probabilistic Models**

- Antonio Vergari, University of California, Los Angeles
- Nicola Di Mauro, University of Bari
- Guy Van den Broeck, University of California, Los Angeles

Juli 22, 2019 - Conference on Uncertainty in Artificial Intelligence (UAI 2019) Tel Aviv
But what if I only want to classify?

Pr(Y|A, B, C, D)
Pr(Y, A, B, C, D)
Logistic Circuits

\[ Pr(Y = 1 \mid A, B, C, D) = \frac{1}{1 + e^{\exp(-1.9)}} = 0.869 \]

Input:

| A | B | C | D | Pr(Y | A, B, C, D) |
|---|---|---|---|----------------|
| 0 | 1 | 1 | 0 | ?              |
Learning Logistic Circuits

Parameter learning reduces to logistic regression:

$$\Pr(Y = 1 \mid x) = \frac{1}{1 + \exp(-x \cdot \theta)}$$

Features associated with each wire
“Global Circuit Flow” features

Learning parameters $\theta$ is convex optimization!

Greedy structure learning (cf. decision trees)
## Comparable Accuracy with Neural Nets

<table>
<thead>
<tr>
<th>Accuracy % on Dataset</th>
<th>MNIST</th>
<th>Fashion</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Baseline: Logistic Regression</strong></td>
<td>85.3</td>
<td>79.3</td>
</tr>
<tr>
<td><strong>Baseline: Kernel Logistic Regression</strong></td>
<td>97.7</td>
<td>88.3</td>
</tr>
<tr>
<td>Random Forest</td>
<td>97.3</td>
<td>81.6</td>
</tr>
<tr>
<td>3-layer MLP</td>
<td>97.5</td>
<td>84.8</td>
</tr>
<tr>
<td>RAT-SPN (Peharz et al. 2018)</td>
<td>98.1</td>
<td>89.5</td>
</tr>
<tr>
<td>SVM with RBF Kernel</td>
<td>98.5</td>
<td>87.8</td>
</tr>
<tr>
<td>5-layer MLP</td>
<td>99.3</td>
<td>89.8</td>
</tr>
<tr>
<td><strong>Logistic Circuit (binary)</strong></td>
<td>97.4</td>
<td>87.6</td>
</tr>
<tr>
<td><strong>Logistic Circuit (real-valued)</strong></td>
<td>99.4</td>
<td>91.3</td>
</tr>
<tr>
<td>CNN with 3 conv layers</td>
<td>99.1</td>
<td>90.7</td>
</tr>
<tr>
<td>ResNet (He et al. 2016)</td>
<td>99.5</td>
<td>93.6</td>
</tr>
</tbody>
</table>
Significantly Smaller in Size

<table>
<thead>
<tr>
<th>Number of Parameters</th>
<th>MNIST</th>
<th>Fashion</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Baseline: Logistic Regression</strong></td>
<td>&lt;1K</td>
<td>&lt;1K</td>
</tr>
<tr>
<td><strong>Baseline: Kernel Logistic Regression</strong></td>
<td>1,521K</td>
<td>3,930K</td>
</tr>
<tr>
<td><strong>Logistic Circuit (real-valued)</strong></td>
<td>182K</td>
<td>467K</td>
</tr>
<tr>
<td><strong>Logistic Circuit (binary)</strong></td>
<td>268K</td>
<td>614K</td>
</tr>
<tr>
<td>3-layer MLP</td>
<td>1,411K</td>
<td>1,411K</td>
</tr>
<tr>
<td>RAT-SPN (Peharz et al. 2018)</td>
<td>8,500K</td>
<td>650K</td>
</tr>
<tr>
<td>CNN with 3 Conv Layers</td>
<td>2,196K</td>
<td>2,196K</td>
</tr>
<tr>
<td>5-layer MLP</td>
<td>2,411K</td>
<td>2,411K</td>
</tr>
<tr>
<td>ResNet (He et al. 2016)</td>
<td>4,838K</td>
<td>4,838K</td>
</tr>
</tbody>
</table>
Better Data Efficiency

<table>
<thead>
<tr>
<th>Accuracy % with % of Training Data</th>
<th>MNIST</th>
<th>Fashion</th>
<th>Fashion</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>100%</td>
<td>10%</td>
<td>2%</td>
</tr>
<tr>
<td>5-layer MLP</td>
<td>99.3</td>
<td>98.2</td>
<td>94.3</td>
</tr>
<tr>
<td>CNN with 3 Conv Layers</td>
<td>99.1</td>
<td>98.1</td>
<td>95.3</td>
</tr>
<tr>
<td>Logistic Circuit (Binary)</td>
<td>97.4</td>
<td>96.9</td>
<td>94.1</td>
</tr>
<tr>
<td>Logistic Circuit (Real-Valued)</td>
<td><strong>99.4</strong></td>
<td>97.6</td>
<td><strong>96.1</strong></td>
</tr>
</tbody>
</table>
“Pure learning is brittle”

bias, algorithmic fairness, interpretability, explainability, adversarial attacks, unknown unknowns, calibration, verification, missing features, missing labels, data efficiency, shift in distribution, general robustness and safety

fails to incorporate a sensible model of the world

• Given a learned predictor $F(x)$
• Given a probabilistic world model $P(x)$
• How does the world act on learned predictors?

Can we solve these hard problems?
What to expect of classifiers?

• Missing features at prediction time
• What is expected prediction of $F(x)$ in $P(x)$?

$$E_{F,P}(y) = \mathbb{E}_{m \sim P(M|y)} [F(ym)]$$

$M$: Missing features
$y$: Observed Features
Explaining classifiers on the world

If the world looks like $P(x)$, then what part of the data is \textit{sufficient} for $F(x)$ to make the prediction it makes?
Outline

1. The AI dilemma: logic vs. learning
2. Deep learning with symbolic knowledge
3. Efficient reasoning during learning
4. New machine learning formalisms
5. Statistical relational learning (tutorial)
Pure Logic  Probabilistic World Models  Pure Learning

High-Level Probabilistic Representations, Reasoning, and Learning
# Graphical Model Learning [Pearl 1988]

## Medical Records

<table>
<thead>
<tr>
<th>Name</th>
<th>Cough</th>
<th>Asthma</th>
<th>Smokes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Bob</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Charlie</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Dave</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Eve</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

## Bayesian Network

- **Asthma**
- **Smokes**
- **Cough**

Rows are independent during learning and inference!
Statistical Relational Representations

Augment graphical model with relations between entities (rows).

**Intuition**

- Friends have similar smoking habits
- Asthma can be hereditary

**Markov Logic**

- 2.1 Asthma ⇒ Cough
- 3.5 Smokes ⇒ Cough
- 1.9 Smokes(x) ∧ Friends(x,y) ⇒ Smokes(y)
- 1.5 Asthma (x) ∧ Family(x,y) ⇒ Asthma (y)
Equivalent Graphical Model

- Statistical relational model (e.g., MLN)

\[ 1.9 \quad \text{Smokes}(x) \land \text{Friends}(x,y) \Rightarrow \text{Smokes}(y) \]

- Ground atom/tuple = random variable in \{true, false\} e.g., Smokes(Alice), Friends(Alice, Bob), etc.

- Ground formula = factor in propositional factor graph
Relational PGMs

• Markov logic
• Probabilistic soft logic (relaxation)
  – Random variables become continuous degrees of truth
  – Inference by convex optimization
  – Talk to Angelika
• Relational dependency networks
  – Learn local relational models that define a sampler
  – Talk to Sriraam
• Light on logic, heavy on PGMs
Probabilistic Logic Programming

- toss (biased) coin & draw ball from each urn
- win if (heads and a red ball) or (two balls of same color)

0.4 :: heads. **probabilistic fact**: heads is true with probability 0.4 (and false with 0.6)
• toss (biased) coin & draw ball from each urn

• win if (heads and a red ball) or (two balls of same color)

0.4 :: heads.

annotated disjunction: first ball is red with probability 0.3 and blue with 0.7

0.3 :: col(1,red); 0.7 :: col(1,blue) <- true.
Probabilistic Logic Programming

- toss (biased) coin & **draw ball from each urn**
- win if (heads and a red ball) or (two balls of same color)

0.4 :: heads.

**annotated disjunction**: first ball is red with probability 0.3 and blue with 0.7

0.3 :: col(1,red); 0.7 :: col(1,blue) <- true.

0.2 :: col(2,red); 0.3 :: col(2,green);
    0.5 :: col(2,blue) <- true.

**annotated disjunction**: second ball is red with probability 0.2, green with 0.3, and blue with 0.5
Probabilistic Logic Programming

- toss (biased) coin & draw ball from each urn
- **win if (heads and a red ball) or (two balls of same color)**

0.4 :: heads.

0.3 :: col(1,red); 0.7 :: col(1,blue) <- true.
0.2 :: col(2,red); 0.3 :: col(2,green);
    0.5 :: col(2,blue) <- true.

win :- heads, col(_,red).

**logical rule** encoding background knowledge
Probabilistic Logic Programming

- toss (biased) coin & draw ball from each urn
- **win if** (heads and a red ball) or (two balls of same color)

0.4 :: heads.

0.3 :: col(1,red); 0.7 :: col(1,blue) <- true.
0.2 :: col(2,red); 0.3 :: col(2,green);
0.5 :: col(2,blue) <- true.

win :- heads, col(_,red).
win :- col(1,C), col(2,C).

**logical rule** encoding background knowledge
Probabilistic Logic Programming

- toss (biased) coin & draw ball from each urn
- win if (heads and a red ball) or (two balls of same color)

```prolog
0.4 :: heads.
0.3 :: col(1,red); 0.7 :: col(1,blue) <- true.
0.2 :: col(2,red); 0.3 :: col(2,green);
    0.5 :: col(2,blue) <- true

win :- heads, col(_,red).
win :- col(1,C), col(2,C).
```
Possible Worlds

0.4 :: heads.

0.3 :: col(1,red); 0.7 :: col(1,blue) <- true.
0.2 :: col(2,red); 0.3 :: col(2,green); 0.5 :: col(2,blue) <- true.

win :- heads, col(_,red).
win :- col(1,C), col(2,C).

0.4 \times 0.3 \times 0.3
Possible Worlds

0.4 :: heads.

0.3 :: col(1,red); 0.7 :: col(1,blue) <- true.
0.2 :: col(2,red); 0.3 :: col(2,green); 0.5 :: col(2,blue) <- true.

win :- heads, col(_,red).
win :- col(1,C), col(2,C).

\[ 0.4 \times 0.3 \times 0.3 \quad \text{and} \quad (1-0.4) \times 0.3 \times 0.2 \]
Possible Worlds

0.4 :: heads.

0.3 :: col(1,red); 0.7 :: col(1,blue) \< true
0.2 :: col(2,red); 0.3 :: col(2,green); 0.5 :: col(2,blue) \< true.

win \< heads, col(_,red).
win \< col(1,C), col(2,C).

\[
\begin{align*}
0.4 \times 0.3 \times 0.3 & \quad (1-0.4) \times 0.3 \times 0.2 & \quad (1-0.4) \times 0.3 \times 0.3
\end{align*}
\]
\[ P(\text{win}) = \sum \text{Marginal Probability} = 0.562 \]
Probabilistic (Logic) Programming

Discrete probabilistic reachability program:

Logic Program (ProbLog)

```
path(X, Y) :- edge(X, Y).
path(X, Y) :- edge(X, Z),
            path(Z, Y).
edge(X, Y) :- ...random vars...
```

= Functional Program (Scala-like)

```
def path(start, end, visited=List())={
    if(start == end)
        return true
    if(visited.contains(start))
        return false
    return start.neighbors.exists{
        path(_, end, (visited+start))
    }
}
```

```
nodeA.neighbors = ...random vars...
nodeB.neighbors = ...random vars...
```
Probabilistic Programming Research

Programming Languages
- Symbolic Execution
- Abstract Interpretation
- Predicate Abstraction
- Model Checking
- Weakest Precondition

Artificial Intelligence
- Weighted Model Counting
- Bayesian Networks
- Independence
- Lifted Inference
- Knowledge Compilation

Probabilistic Predicate Abstraction
Symbolic Compilation
Probabilistic Databases

- Tuple-independent probabilistic database

<table>
<thead>
<tr>
<th>Scientist</th>
<th>x</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>Erdos</td>
<td>0.9</td>
<td></td>
</tr>
<tr>
<td>Einstein</td>
<td>0.8</td>
<td></td>
</tr>
<tr>
<td>Pauli</td>
<td>0.6</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Coauthor</th>
<th>x</th>
<th>y</th>
<th>P</th>
</tr>
</thead>
<tbody>
<tr>
<td>Erdos</td>
<td>Renyi</td>
<td>0.6</td>
<td></td>
</tr>
<tr>
<td>Einstein</td>
<td>Pauli</td>
<td>0.7</td>
<td></td>
</tr>
<tr>
<td>Obama</td>
<td>Erdos</td>
<td>0.1</td>
<td></td>
</tr>
</tbody>
</table>

- Learned from the web, large text corpora, ontologies, etc., using statistical machine learning.

[Suciu’11]
Pure Logic  Probabilistic World Models  Pure Learning

Probabilistic Logic Programming
Prolog meets probabilistic AI
Talk to Luc, Angelika, Vaishak, Kristian, etc.

Probabilistic Databases
Databases meets probabilistic AI
Talk to Dan, Dan, Ismail, Carsten, etc.

Weighted Model Integration
SAT modulo theories meets probabilistic AI
Talk to Vaishak
Approximate Lifted Probabilistic Inference

• Message passing symmetries
  – Identify which nodes will receive identical messages throughout algorithm
  – Fractional automorphisms
  – Found by color passing
  – *Talk to Kristian, Sriraam, Martin Grohe*

• Lifted MCMC
  – Compute exact automorphisms
  – Fun with group theory tools
  – Make MCMC samplers mix exponentially faster
Conclusions

Pure Logic  Probabilistic World Models  Pure Learning

Bring high-level representations, general knowledge, and efficient high-level reasoning to probabilistic models

Bring back models of the world, supporting new tasks, and reasoning about what we have learned, without compromising learning performance
Conclusions

• There is a lot of value in working on pure logic, pure learning
• But we can do more by finding a synthesis, a confluence

Let’s get rid of this false dilemma...
Thanks