Computers and Thought

Guy Van den Broeck

IJCAI

August 16, 2019
Outline

1. What would 2011 junior PhD student Guy think? …*please help me make sense of this field*…

2. What do I work on and why?
   - High-level probabilistic reasoning
   - A new synthesis of learning and reasoning

3. Personal thank you messages
The AI Dilemma of 2019

**Deep learning** approaches the problem of designing intelligent machines by postulating a large number of very simple information processing elements, arranged in a [...] network, and certain processes for facilitating or inhibiting their activity.

**Knowledge representation and reasoning** take a much more macroscopic approach [...] They believe that intelligent performance by a machine is an end difficult enough to achieve without “starting from scratch”, and so they build into their systems as much complexity of information processing as they are able to understand and communicate to a computer.

Edward Feigenbaum and Julian Feldman
Neural cybernetics approaches the problem of designing intelligent machines by postulating a large number of very simple information processing elements, arranged in a [.] network, and certain processes for facilitating or inhibiting their activity.

Cognitive model builders take a much more macroscopic approach [.]. They believe that intelligent performance by a machine is an end difficult enough to achieve without “starting from scratch”, and so they build into their systems as much complexity of information processing as they are able to understand and communicate to a computer.
The AI Dilemma

Pure Logic

Pure Learning
The AI Dilemma

Pure Logic

• Slow thinking: deliberative, cognitive, model-based, extrapolation
• Amazing achievements until this day

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

• Fast thinking: instinctive, perceptive, model-free, interpolation
• Amazing achievements recently
The AI Dilemma

- 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
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
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?

Pure Logic  Probabilistic World Models  Pure Learning

What did we gain?

What did we lose along the way?
Pure Logic  Probabilistic World Models  Pure Learning

High-Level Probabilistic Reasoning
Simple Reasoning Problem

Probability that first card is Hearts? 1/4
Automated Reasoning

Let us automate this:

1. Probabilistic graphical model (e.g., factor graph)

2. Probabilistic inference algorithm (e.g., variable elimination or junction tree)
Let us automate this:

1. Probabilistic graphical model (e.g., factor graph) is fully connected!

2. Probabilistic inference algorithm (e.g., variable elimination or junction tree) builds a table with $52^{52}$ rows
Tractable High-Level Reasoning

What's going on here?
Which property makes reasoning tractable?

- High-level (first-order) reasoning
- Symmetry
- Exchangeability

⇒ Lifted Inference
Model distribution at first-order level:

\[ \forall p, \exists c, \text{Card}(p,c) \]
\[ \forall c, \exists p, \text{Card}(p,c) \]
\[ \forall p, \forall c, \forall c', \text{Card}(p,c) \land \text{Card}(p,c') \Rightarrow c = c' \]

Can we now be efficient in the size of our domain?
How does this relate to learning?

i.i.d. assumption
independent and identically distributed
“Smokers are more likely to be friends with other smokers.”
“Colleagues of the same age are more likely to be friends.”
“People are either family or friends, but never both.”
“If X is family of Y, then Y is also family of X.”
“Universities in California are more likely to be rivals.”
Lifted Inference Example: Counting Possible Worlds

\[ \forall x, y \in \text{People}: \text{Smokes}(x) \land \text{Friends}(x, y) \Rightarrow \text{Smokes}(y) \]

- If we know \( D \) precisely: who smokes, and there are \( k \) smokers?

  **Database:**
  - Smokes(Alice) = 1
  - Smokes(Bob) = 0
  - Smokes(Charlie) = 0
  - Smokes(Dave) = 1
  - Smokes(Eve) = 0
  ...

  \[ \Rightarrow 2^{n^2 - k(n-k)} \text{ worlds} \]

- If we know that there are \( k \) smokers?

  \[ \Rightarrow \binom{n}{k} 2^{n^2 - k(n-k)} \text{ worlds} \]

- In total...

  \[ \Rightarrow \sum_{k=0}^{n} \binom{n}{k} 2^{n^2 - k(n-k)} \text{ worlds} \]
Theorem: Model counting for FO$^2$ in polynomial time in the number of constants/nodes/entities/people/cards.

Corollary: Partition functions efficient to compute in 2-variable Markov logic, relational factor graphs, etc.
“Smokers are more likely to be friends with other smokers.”
“Colleagues of the same age are more likely to be friends.”
“People are either family or friends, but never both.”
“If X is family of Y, then Y is also family of X.”
“Universities in California are more likely to be rivals.”
Can Everything Be Lifted?

**Theorem:** There exists an FO$^3$ model $\Theta_1$ for which just counting possible worlds is $\#P_1$-complete in the domain size.

What about learning?

- Learn better models faster
- Tractability is a great inductive bias!
Pure Logic  Probabilistic World Models  Pure Learning

“A confluence of ideas, a meeting place of two streams of thought”

Probabilistic Logic Programming
Prolog meets probabilistic AI

Probabilistic Databases
Databases meets probabilistic AI

Weighted Model Integration
SAT modulo theories meets probabilistic AI
A New Synthesis of Learning and Reasoning
Another False Dilemma?

Classical AI Methods

Hungry? → Sleep? → Restaurant?

$25? → Have I 25$?

Yes

No

Go to restaurant

Buy a hamburger

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>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>0</td>
</tr>
</tbody>
</table>

SPNs, ACs
PSDDs, CNs

\[ (0.1 \times 1) + (0.9 \times 0) \]
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>
<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>
<td>Book</td>
<td>-33.82</td>
<td>-36.41</td>
<td>-33.95</td>
<td>-33.19</td>
</tr>
<tr>
<td>msnbc</td>
<td>-6.04</td>
<td>-6.04</td>
<td>-6.06</td>
<td>-6.09</td>
<td>movie</td>
<td>-50.34</td>
<td>-54.37</td>
<td>-48.7</td>
<td>-47.43</td>
</tr>
<tr>
<td>kdd2000</td>
<td>-2.12</td>
<td>-2.19</td>
<td><strong>-2.07</strong></td>
<td>-2.12</td>
<td>webkb</td>
<td>-149.20</td>
<td>-157.43</td>
<td>-149.59</td>
<td><strong>-146.9</strong></td>
</tr>
<tr>
<td>plants</td>
<td>-11.84</td>
<td>-12.65</td>
<td>12.32</td>
<td>-12.34</td>
<td>cr52</td>
<td>-81.87</td>
<td>-87.56</td>
<td>-82.80</td>
<td><strong>-81.33</strong></td>
</tr>
<tr>
<td>audio</td>
<td>-39.39</td>
<td>-40.50</td>
<td>-38.95</td>
<td>-38.67</td>
<td>c20ng</td>
<td>-151.02</td>
<td>-158.95</td>
<td>-153.18</td>
<td><strong>-146.90</strong></td>
</tr>
<tr>
<td>jester</td>
<td>-51.29</td>
<td><strong>-51.07</strong></td>
<td>-52.23</td>
<td>-51.54</td>
<td>bbc</td>
<td><strong>-229.21</strong></td>
<td>-257.86</td>
<td>-242.40</td>
<td>-240.94</td>
</tr>
<tr>
<td>netflix</td>
<td>-55.71</td>
<td>-57.02</td>
<td>-55.16</td>
<td>-54.73</td>
<td>ad</td>
<td>-14.00</td>
<td>-18.35</td>
<td><strong>-13.65</strong></td>
<td>-18.81</td>
</tr>
<tr>
<td>accidents</td>
<td>-26.89</td>
<td><strong>-26.32</strong></td>
<td>-26.42</td>
<td>-29.11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>retail</td>
<td>-10.72</td>
<td>-10.87</td>
<td>-10.81</td>
<td>-10.83</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>pumbs*</td>
<td>-22.15</td>
<td><strong>-21.72</strong></td>
<td>-22.3</td>
<td>-25.16</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>dna</td>
<td>-79.88</td>
<td>-80.65</td>
<td>-82.77</td>
<td>-94.56</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Kosarek</td>
<td>-10.52</td>
<td>-10.83</td>
<td>-</td>
<td>-10.64</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
But what if I only want to classify?

Pr(\(Y|A, B, C, D\))

\(\not{\text{Pr}(Y, A, B, C, D)}\)

Logistic Circuits
## 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>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>100%</td>
<td>10%</td>
</tr>
<tr>
<td>5-layer MLP</td>
<td>99.3</td>
<td><strong>98.2</strong></td>
</tr>
<tr>
<td>CNN with 3 Conv Layers</td>
<td>99.1</td>
<td>98.1</td>
</tr>
<tr>
<td>Logistic Circuit (Binary)</td>
<td>97.4</td>
<td>96.9</td>
</tr>
<tr>
<td>Logistic Circuit (Real-Valued)</td>
<td><strong>99.4</strong></td>
<td>97.6</td>
</tr>
</tbody>
</table>
Probabilistic & Logistic Circuits

- Statistical ML "Probability"
- Symbolic AI "Logic"
- Connectionism "Deep"
Reasoning about World Model + Classifier

“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 $P(x)$?

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

**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 *sufficient* for $F(x)$ to make the prediction it makes?
Conclusions

**Pure Logic**

Bring high-level representations, general knowledge, and efficient high-level reasoning to the world of probability

**Probabilistic World Models**

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

**Pure Learning**
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

• In another 56 years:
  
  Let’s get rid of this false dilemma…
References

- **Computers and thought**

- **Cards example of high-level reasoning**

- **Exchangeability as a source of tractability**
References

• **FO² liftability theorem**

• **Intractability of FO³**

• **Lifted learning**
References

• **Confluences of ideas**

• **Probabilistic logic programming**

• **Probabilistic databases**

• **Weighted model integration**
References

- **Probabilistic circuits**

- **Logistic circuits**

- **What to expect of classifiers?**

& unpublished work in progress
Thank You Messages

• IJCAI and the awards committee
• Luc De Raedt
• Stuart Russell, Dan Suciu, Kristian Kersting, David Poole, Dan Roth, George Varghese, Rina Dechter, Lise Getoor, Dan Olteanu
• Kurt Driessens, Wannes Meert, Arthur Choi
• The generosity of my many collaborators
  Mathias Niepert, Jesse Davis, Siegfried Nijssen, etc.
• Irma and my family and friends
Thank You Messages

• My amazing students & StarAI lab @ UCLA

• My “communities”: SRL, StarAI, Probabilistic (Logic) Programming, TPM, KC, …

• Funding
Thank You All