What Logic Can Do For AI Today

Adnan Darwiche
UCLA

International Conference on Logic Programming (ICLP 2019)
Agenda

Logic for Computation
reducing ‘Beyond NP’ problems to logical reasoning

Logic for Background Knowledge
learning from a combination of data and knowledge

Logic for Meta Reasoning
reasoning about the behavior of machine learning systems
Agenda

Logic for Computation
reducing ‘Beyond NP’ problems to logical reasoning

Logic for Background Knowledge
learning from a combination of data and knowledge

Logic for Meta Reasoning
reasoning about the behavior of machine learning systems
Beyond NP
Bayesian Networks
Probabilistic Inference

Complete problems: hardest in their class

Input: Bayesian Network
Beyond NP

Prototypical problems

Boolean expressions:
(A or (not B) or C),
((not A) or D or (not E)),
....
Reduction Approaches

Class C

Prototypical problem (complete for Class C)

Problem X

Problem Y
Reduce

Solver for X

X-Answer

Y-Answer
1\textsuperscript{st} Line of Developments

Prototypical problems:
- Oztok et al, KR 2016
- Huang et al, AAAI 2006
- Darwiche, KR 2002
- Park, AAAI 2002

Boolean expressions:
\[(A \lor \neg B \lor C), \quad \neg A \lor D \lor \neg E),\]

- Prototypical problems
- SDP
- MAP
- Marginals
- MPE
- SAT
- MAJ-SAT
- E-MAJ-SAT
- MAJ-MAJ-SAT

Marginals
2nd Line of Developments

Boolean expressions:
(A or (not B) or C),
((not A) or D or (not E)),
...

Prototypical problems
(Systematic Approach
(Compile to Boolean Circuits)

MPE

Marginals

MAP

MAJ-MAJ-SAT

E-MAJ-SAT

MAJ-SAT

SAT

since ~2000

Park, AAAI 2002

Huang et al, AAAI 2006

Oztok et al, KR 2016
SAT: NP-complete

Boolean expression:
(A or B) and (not C)

SAT: Is there a satisfying instantiation?
Yes
MAJ-SAT: PP-complete

Boolean expression: 
(A or B) and (not C)

MAJ-SAT: Are the majority of instantiations satisfying? 
No
MAJ-SAT Variant

Model Counting

Boolean expression:
(A or B) and (not C)

#SAT: How many satisfying assignment?
3
MAJ-SAT Variant

Weighted Model Counting

Boolean expression:
(A or B) and (not C)

WMC: The added weight of satisfying assignments?
0.14 = 0.04 + 0.10 + 0.00

\[ w(A, \neg B, C) = w(A)w(\neg B)w(C) \]
**E-MAJ-SAT:**  
\(NP^{pp}\)-complete

**Boolean expression:**  
\((A \text{ or } B) \text{ and } (\text{not } C)\)

Split variables \(X=\{C\}, \ Y=\{A,B\}\)

**E-MAJ-SAT:** Is there an \(X\)-instantiation under which the majority of \(Y\)-instantiations satisfying?  
Yes
**E-MAJ-SAT:**

**NP}\textsuperscript{pp}-complete**

**Boolean expression:**

\[(A \text{ or } B) \text{ and } (\text{not } C)\]

Split variables \(X=\{C\}, Y=\{A,B\}\)

**E-MAJ-SAT:** Is there an \(X\)-instantiation under which the majority of \(Y\)-instantiations satisfying?  

Yes
**MAJ-MAJ-SAT:** $\text{PP}^\text{PP}$-complete

**Boolean expression:**

(A or B) and (not C)

Split variables $X=\{C\}$, $Y=\{A,B\}$

**MAJ-MAJ-SAT:** Is there a majority of $X$-instantiation under which the majority of $Y$-instantiations satisfying? No
MAJ-MAJ-SAT: $P^{PP}$-complete

Boolean expression:
$(A \lor B) \land \neg C$

Split variables $X=\{C\}$, $Y=\{A,B\}$

MAJ-MAJ-SAT: Is there a majority of $X$-instantiation under which the majority of $Y$-instantiations satisfying? No
Boolean expressions:

(\(A \lor (\neg B) \lor C\)),

((\neg A) \lor D \lor (\neg E)),
\[A \land C \leftrightarrow \theta_{C|A}\]
\[A \land \neg C \leftrightarrow \theta_{\neg C|A}\]
\[A \leftrightarrow \theta_A, \neg A \leftrightarrow \theta_{\neg A}\]
\[A \land B \leftrightarrow \theta_{B|A}\]
\[A \land \neg B \leftrightarrow \theta_{\neg B|A}\]
\[\vdots\]

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>Pr(.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>\theta_A \theta_B</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>\theta_A \theta_B</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
<td>\theta_A \theta_{\neg B</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
<td>\theta_A \theta_{\neg B</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
<td>\theta_{\neg A} \theta_B</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
<td>\theta_{\neg A} \theta_B</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
<td>\theta_{\neg A} \theta_{\neg B}</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>\theta_{\neg A} \theta_{\neg B}</td>
</tr>
</tbody>
</table>

\[A, B, \neg C, \theta_A, \theta_{B|A}, \theta_{\neg C|A}, \neg \theta_{\neg A}, \ldots, \neg \theta_{\neg C|A}\]

1 1 1 .3 .6 .8 1 1
Relational networks (251 networks)
(have logical constraints)
Knowledge Compilation

encoding

(A and (not B))
or(C and (not D))
or ((not C) and D)
...

Compiler

NNF Circuit

Answer in Linear Time

MAJ-MAJ-SAT
E-MAJ-SAT
MAJ-SAT
SAT

DNNF
d-DNNF
SDD
NNF Circuits

\[ P \lor L \]
\[ A \Rightarrow P \]
\[ K \Rightarrow (P \lor L) \]
Decomposability (DNNF)
Determinism (d-DNNF)

Darwiche, JANCL 2000

MAJ-SAT in linear time

Input: \(L, K, P, A\)
Model Counting
Weighted Model Counting

\[
P^*(\overline{b}) + P^*(b) = \sum_{\overline{c}} \lambda_{\overline{c}|a} \lambda_{\overline{c}} \theta_{\overline{c}|\overline{a}} \theta_{\overline{c}} \theta_{\overline{a}} \theta_{\overline{a}|\overline{a}}
\]

Arithmetic Circuit (AC)
Structured Decomposability

Pipatsrisawat & Darwiche, AAAI 2008
Partitioned Determinism (SDD Circuits)

Darwiche, IJCAI 2011

MAJ-MAJ-SAT in linear time using appropriate vtree
Oztok & Darwiche, KR 2016

Input: \(L, K, P, A\)
Beyond NP

Prototypical problems

Systematic Approach
(Compile to Boolean Circuits)

Boolean expressions:
(A or (not B) or C),
((not A) or D or (not E)),
....
Tractable Circuits: Knowledge Compilation

Negation Normal Form

Polytime Operations

Consistency (CO)
Validity (VA)
Clausal entailment (CE)
Sentential entailment (SE)
Implicant testing (IP)
Equivalence testing (EQ)
Model Counting (CT)
Model enumeration (ME)

Existential quantification
Conditioning
Conjoin, Disjoin, Negate

Succinctness

CNF, DNF, Prime Implicates / Implicants, OBDD, FBDD, DNNF, d-DNNF, SDD
Example compilers: c2d, sdd, and d4
Tractable Circuits: Knowledge Compilation

Polytime Operations
Consistency (CO)
Validity (VA)
Clausal entailment (CE)
Sentential entailment (SE)
Implicant testing (IP)
Equivalence testing (EQ)
Model Counting (CT)
Model enumeration (ME)
Existential quantification
Conditioning
Conjoin, Disjoin, Negate

Negation Normal Form

De Morgan’s laws:
- $\neg A \land \neg B \\
- A \land \neg B \\
- \neg A \land B \\
- \neg B \land A \\
- A \lor B \\
- \neg A \lor \neg B \\
- \neg A \lor B \\
- A \lor \neg B \\
- \neg C \\
- \neg D \\
- D \\
- \neg C \\

Conjoin, Disjoin, Negate

Example compilers: c2d, sdd, and d4

Darwiche & Marquis (JAIR 02)

SAT/SMT/AR Summer School 2019

Day 1 (July 3)

1. Keynote talk “Knowledge Compilation: Principles and Applications” by Adnan Darwiche (University of California, Los Angeles, USA)
c2d: compiled cnf to decomposable and deterministic circuits
mini-c2d: open source + modular design (sat box)
sdd library: SDD generalized OBDD (similar to CUDD)
Ace: Compiles Bayesian networks into circuits
Agenda

Logic for Computation
reducing ‘Beyond NP’ problems to logical reasoning

Logic for Background Knowledge
learning from a combination of data and knowledge

Logic for Meta Reasoning
reasoning about the behavior of machine learning systems
Learning with Background Knowledge

Logic (L)
Knowledge Representation (K)
Probability (P)
Artificial Intelligence (A)

Background Knowledge

Must take at least one of Probability or Logic.
Probability is a prerequisite for AI.
The prerequisites for KR is either AI or Logic.

\[ P \lor L \quad A \Rightarrow P \quad K \Rightarrow (P \lor L) \]
Learned PSDD

\[ P \lor L \]
\[ A \Rightarrow P \]
\[ K \Rightarrow (P \lor L) \]

closed-form parameter estimation under complete datasets

Kisa et al, KR 2014
induces a normalized distribution over satisfying assignments
(structured probability space)

<table>
<thead>
<tr>
<th>L</th>
<th>K</th>
<th>P</th>
<th>A</th>
<th>P(L,K,P,A)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>6.0%</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>54.0%</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0.5%</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0.6%</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>5.0%</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>10.0%</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4.4%</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1.0%</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1.0%</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.6%</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>17.6%</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>4.0%</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>2.4%</td>
</tr>
</tbody>
</table>

Pr(P,A)

<table>
<thead>
<tr>
<th>P</th>
<th>A</th>
<th>Pr(P,A)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>73.33%</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>16.67%</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>10.00%</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>10.00%</td>
</tr>
</tbody>
</table>
complete & canonical representation
tractable for MPE & marginals
closed-form parameter estimation under complete datasets

<table>
<thead>
<tr>
<th>L</th>
<th>K</th>
<th>P</th>
<th>A</th>
<th>P(L,K,P,A)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100%</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>83.33%</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>6.0%</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>54.0%</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>33.33%</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>40.0%</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>5.0%</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>10.0%</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4.4%</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1.0%</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>17.6%</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2.4%</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1.0%</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>2.0%</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>4.0%</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2.4%</td>
</tr>
</tbody>
</table>
Ignoring knowledge
Combinatorial Objects: Routes

Choi et al, AAAI 2016
Combinatorial Objects: Routes

Choi et al, AAAI 2016
Combinatorial Objects: Routes

Choi et al, AAAI 2016
Combining Knowledge & Data

Choi et al, NIPS 2017

Input: Knowledge (a map)

Input: Data (GPS routes)

Output: Probabilistic Model over Routes

- Estimate traffic
- Predict routes
- Predict the impact of an intervention
every query specifies a function: evidence \([0,1]^n\) to probability \([0,1]\)

Question: What’s the domain of the function \(\log(x+3)\)?

Probabilistic Graphical Models

dominated AI for about 2 decades

Does the student:

Recognize the function formula?

shift the domain properly?
Model Query is a Function

Evidence: A, C
Query: B
Model Query is a Function

Evidence: A, C
Query: B

\[ P^*(\overline{b}) + P^*(b) \]

Arithmetic Circuits (ACs)

\[ \theta_{\overline{b}|\overline{a}} \times \theta_b|\overline{a} \times \theta_{\overline{b}|a} \times \theta_b|a \]

\[ \lambda_{\overline{a}} \times \theta_{\overline{a}} \times \lambda_a \times \theta_a \]

\[ \theta_{\overline{c}|\overline{a}} \times \lambda_{\overline{c}} \times \theta_{\overline{c}|a} \times \lambda_c \times \theta_{c|a} \times \theta_c|a \]

Darwiche, JACM 2003
Model Query is a Function

Evidence: A, C
Query: B

Arithmetic Circuits (ACs)

Darwiche, JACM 2003
Model Query is a Function

Evidence: A, C
Query: B

Darwiche, JACM 2003

Arithmetic Circuits (ACs)
Model Query is a Function

Evidence: A, C
Query: B

Arithmetic Circuits (ACs)

P*(\(\tilde{b}\)) + P*(\(b\))

\[ a \quad \bar{a} \quad \lambda_{\bar{a}} \quad \theta_{\bar{a}} \quad \lambda_{a} \quad \theta_{a} \quad * \quad * \quad * \quad * \quad + \quad + \]

\[ \theta_{\bar{c}|\bar{a}} \quad \lambda_{\bar{c}} \quad \theta_{\bar{c}|\bar{a}} \quad \lambda_{c} \quad \theta_{c|a} \quad \theta_{c|a} \quad \theta_{c|a} \quad \theta_{c|a} \quad \theta_{c|a} \]

can be trained discriminatively from labeled data using gradient descent

can integrate domain knowledge (independence, known parameters)

Darwiche, JACM 2003
Model Query is a Function

Evidence: A, C
Query: B

can be trained discriminatively from labeled data using gradient descent

can integrate domain knowledge (independence, known parameters)

Darwiche, JACM 2003
Tensors: The New Reality

(illustrations from the web, not relevant to representing circuits)

2. Tensor = an object that is invariant under a change of coordinates, and... has components that change in a special, predictable way under a change of coordinates.

\[
\begin{align*}
\text{Coordinate Definition} & = 2 \rightarrow +1 \rightarrow +2 \\
& = 1 \rightarrow +3 \rightarrow +2
\end{align*}
\]
Tensors: The New Reality
(usage in machine learning, relevant to representing circuits)

• Tensors are multi-dimensional arrays. That’s it!

• Tensor operations are parallelized with support for GPUs.

• Implication: Orders of magnitude speedup.
Arithmetic Circuit (AC)

Evidence: A, C
Query: B

Darwiche, JACM 2003
Tensors: The New Reality

PyTAC

Constructing tensor graph:
  factor ops: size 7,521,292, count 1,117 (mulpro 288, mul 292, pro 147, norm 1, scale 72)
cpts: trained 0, fixed 173, replicas 0
evidence nodes 144
fixed zero-parameters 0
cache lookups 800, hits 0, rate 0.0%

Tensor graph size 10,863,814, ops 11,088
  reshape (c 866, s 3,110,712), transpose (c 1,302, s 3,200,400)
Compile Time: 6.985 sec

Evaluating AC: evidence size 5434, batch size 395, batch memory 15.986 GB
Evaluation Time: 39.484 sec (0.0073 sec per example)
PyTAC (Keras-like tool for compiling and training TACs, based on tensorflow)

```python
"simulate data from a tbn then learn back the tbn parameters using TAC"
def simulate_fit(tbn,e1,e2,q):
    size = 1024

    # simulate
    TAC = tac.TAC(tbn,(e1,e2),q,trainable=False)

    evidence, marginals = TAC.simulate(size,'grid')

    # visualize simulated data
    visualize.plot3D(evidence,marginals,e1,e2,q)

    # learn
    TAC = tac.TAC(tbn,(e1,e2),q,trainable=True)
    TAC.fit(evidence,marginals,loss_type='CE',metric_type='CE')
    predictions = TAC.evaluate(evidence)

    # visualize learned tac
    visualize.plot3D(evidence,predictions,e1,e2,q)
```
Agenda

Logic for Computation
reducing ‘Beyond NP’ problems to logical reasoning

Logic for Background Knowledge
learning from a combination of data and knowledge

Logic for Meta Reasoning
reasoning about the behavior of machine learning systems
Explaining ML Systems

- Labeled Data
- Learning Process
- ML System (Function) (BN, NN, RF)
- Instance
- Decision

Why did you make this decision? (e.g., decline loan application)
(e.g., decline loan application)
From Numbers to Decisions

+ Probabilistic Inference
From Numbers to Decisions

Test results: U, B, S

Decision Function

Yes, No

+ Probabilistic Inference
From Numbers to Decisions

Pregnant? (P)

<table>
<thead>
<tr>
<th>P</th>
<th>Pr(p)</th>
</tr>
</thead>
<tbody>
<tr>
<td>yes</td>
<td>0.87</td>
</tr>
<tr>
<td>no</td>
<td>0.13</td>
</tr>
</tbody>
</table>

Urine test (U)

| P  | U  | Pr(u|p) |
|----|----|--------|
| yes| -ve| 0.27   |
| no | +ve| 0.107  |

Blood test (B)

| P  | B  | Pr(b|p) |
|----|----|--------|
| yes| -ve| 0.36   |
| no | +ve| 0.106  |

Scanning test (S)

| P  | S  | Pr(s|p) |
|----|----|--------|
| yes| -ve| 0.10   |
| no | +ve| 0.01   |

Test results: U, B, S

Yes, No

Ordered Decision Graph (OBDD)

+ Probabilistic Inference
From Numbers to Decisions

Pregnant? (P)

Urine test (U)

Blood test (B)

Scanning test (S)

Test results: U, B, S

Yes, No

Instance: U=+ve, B=-ve, S=-ve

Ordered Decision Graph (OBDD)
From Numbers to Decisions

+ Probabilistic Inference

Test results: U, B, S

Yes, No

Instance: U=+ve, B=-ve, S=-ve

Ordered Decision Graph (OBDD)
Compiling BN Classifiers

Naïve Bayes
(Chan & Darwiche
UAI 03)

Latent Tree
(Shih, Choi & Darwiche
IJCAI 18)

General BN
(Shih, Choi & Darwiche
AAAI 19)

Why did the classifier predict “cat”?

Ordered Decision Diagram

MC  PI
Education Network
Printing Diagnosis Network
Cancer Network

[Diagram showing a network of cancer-related factors with nodes labeled such as Age, SkinThickening, NippleRetraction, and MassDensity.]
Cancer Decision Graph

156 nodes
(hundreds to millions for BNs with similar size)
# Size of Decision Diagrams

<table>
<thead>
<tr>
<th>Name</th>
<th># Vars</th>
<th>Root</th>
<th>Threshold</th>
<th>ODD Size</th>
<th>Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mammography</td>
<td>15</td>
<td>Reg-Outcome</td>
<td>0.02</td>
<td>156</td>
<td>7</td>
</tr>
<tr>
<td>Win95pts</td>
<td>16</td>
<td>Printer-Offline</td>
<td>0.50</td>
<td>291</td>
<td>21</td>
</tr>
<tr>
<td>Immex</td>
<td>17</td>
<td>Not-Understand</td>
<td>0.50</td>
<td>115</td>
<td>9</td>
</tr>
<tr>
<td>Adaptive-Testing</td>
<td>20</td>
<td>HV1</td>
<td>0.50</td>
<td>1164</td>
<td>40</td>
</tr>
<tr>
<td>Mooring</td>
<td>22</td>
<td>Environment</td>
<td>0.75</td>
<td>12840</td>
<td>938</td>
</tr>
<tr>
<td>Andes</td>
<td>24</td>
<td>Try-Kinematics</td>
<td>0.50</td>
<td>47</td>
<td>11708</td>
</tr>
<tr>
<td>Math-Skills</td>
<td>46</td>
<td>S6</td>
<td>0.50</td>
<td>3693629</td>
<td>61088</td>
</tr>
</tbody>
</table>
Explaining

Given a decision graph, we can explain the classifier’s decisions (algorithmically, not by visual inspection)
Explaining

Shih, Choi & Darwiche (IJCAI 18)

Given a decision graph, we can explain the classifier’s decisions (algorithmically, not by visual inspection)

• PI Explanations:
  “Which features make the other features irrelevant?”
Given a decision graph, we can explain the classifier’s decisions (algorithmically, not by visual inspection)

• PI Explanations:
  “Which features make the other features irrelevant?”

• MC Explanations (monotonic classifiers):
  “Which positive features are responsible for a *yes* decision?”
  “Which negative features are responsible for a *no* decision?”
Example Explanation

Susan tested positive for Scanning, Blood and Urine

Instance: \( U=+ve, B=+ve, S=+ve \)

Ordered Decision Graph
Example Explanation

Susan tested positive for **Scanning**, **Blood** and **Urine**

**Why** did you conclude that Susan is pregnant?

<table>
<thead>
<tr>
<th>Instance: U=+ve, B=+ve, S=+ve</th>
</tr>
</thead>
</table>

Ordered Decision Graph
Example Explanation

Susan tested positive for Scanning, Blood and Urine

**Why** did you conclude that Susan is pregnant?

**Because** the Scanning test came out positive
Example Explanation

Sally tested negative for Scanning, Blood and Urine
Example Explanation

Sally tested negative for Satin, Blood and Urine

Why did you conclude that Sally is not pregnant?
Example Explanation

Sally tested negative for Scanning, Blood and Urine

Why did you conclude that Sally is not pregnant?

Because the Scanning test, and one of the Blood and Urine tests came out negative (S=-ve and (B=-ve or U=-ve))
Example Explanation

Sally tested negative for Scanning, Blood and Urine

**Why** did you conclude that Sally is not pregnant?

**Because** the Scanning test, and one of the Blood and Urine tests came out negative (S=-ve and (B=-ve or U=-ve))

Explanations can be computed in linear time
Explaining in Linear Time

positive instance: U, ¬B, S
Explaining in Linear Time

positive instance: \( U, \neg B, S \)
1. Condition on $\neg B$

positive instance: $U, \neg B, S$
1. Condition on $\neg B$

positive instance: $U, \neg B, S$
2. Compute Minimum Cardinality

positive instance: $U, \neg B, S$
2. Compute Minimum Cardinality

positive instance: $U, \neg B, S$
2. Compute Minimum Cardinality

positive instance: $U, \neg B, S$

```

```

```
2. Compute Minimum Cardinality

positive instance: \( U, \neg B, S \)
2. Compute Minimum Cardinality

positive instance: \(U, \neg B, S\)
2. Compute Minimum Cardinality

positive instance: U, ¬B, S
3. Minimize

positive instance: \( \text{U, } \neg \text{B, S} \)
3. Minimize

positive instance: $U, \neg B, S$
3. Minimize

positive instance: \( U, \neg B, S \)
4. Enumerate

positive instance: U, ¬B, S
Explanation

positive instance: U, ¬B, S
Verification: Monotone Classifiers

Shih, Choi, Darwiche (PGM 18)
Verification: Monotone Classifiers

Positive instance remains positive even if we flip some features from – to +

Educational Testing:
Susan's correct answers include Jack's correct answers
Susan should pass if Jack passed

Credit Application:
Susan and Jack have the same characteristics, except that Susan has a higher income
Susan should be approved if Jack is approved
Verification: Monotone Classifiers

Positive instance remains positive even if we flip some features from $-$ to $+$

If $(+, -, -, +)$ is a positive instance, these instances are also positive

- $(+, +, -, +)$
- $(+, -, +, +)$
- $(+, +, +, +)$
- $(+, +, +, +)$
Verification: Monotone Classifiers

Positive instance remains positive even if we flip some features from – to +

If (+, -, -, +) is a positive instance, these instances are also positive
  • (+, +, -, +)
  • (+, -, +, +)
  • (+, +, +, +)
  • (+, +, +, +)

Educational Testing:
  Susan’s correct answers include Jack’s correct answers
  Susan should pass if Jack passed
Verification: Monotone Classifiers

Positive instance remains positive even if we flip some features from $-$ to $+$

If $(+, -, -, +)$ is a positive instance, these instances are also positive

- $(+, +, -, +)$
- $(+,-,+,+)$
- $(+,+,+,+)$

Educational Testing:

Susan’s correct answers include Jack’s correct answers
Susan should pass if Jack passed

Credit Application:

Susan and Jack have the same characteristics, except that Susan has a higher income
Susan should be approved if Jack is approved
Verification: Monotone Classifiers

- Quadratic complexity on ODDs

- Educational assessment classifier not monotone (threshold ½)
- Cancer classifier not monotone (threshold .02 based on BI-RADS assessment scale)

- Two patients, same mammography report except for personal history.
  - One with personal history à Benign
  - One with no personal history à Malignant

- Recent application to Health (predicting weight loss using random forests): Feedback into design of classifiers.
Verification: Monotone Classifiers

- Quadratic complexity on ODDs

- Educational assessment classifier not monotone (threshold $\frac{1}{2}$)
Verification: Monotone Classifiers

- Quadratic complexity on ODDs

- Educational assessment classifier not monotone (threshold $\frac{1}{2}$)

- Cancer classifier not monotone (threshold 0.02 based on BI-RADS assessment scale)
Verification: Monotone Classifiers

- Quadratic complexity on ODDs

- Educational assessment classifier not monotone (threshold ½)

- Cancer classifier not monotone (threshold .02 based on BI-RADS assessment scale)

- Two patients, same mammography report except for personal history.
  - One with personal history ➔ Benign
  - One with no personal history ➔ Malignant
Verification: Monotone Classifiers

- Quadratic complexity on ODDs

- Educational assessment classifier not monotone (threshold $\frac{1}{2}$)

- Cancer classifier not monotone (threshold $0.02$ based on BI-RADS assessment scale)

- Two patients, same mammography report except for personal history.
  - One with personal history $\rightarrow$ Benign
  - One with no personal history $\rightarrow$ Malignant

- Recent application to Health (predicting weight loss using random forests): Feedback into design of classifiers.
Reasoning about the Behavior of AI Systems

Expanded Scope:

Queries: Explanation, Robustness, Verification  
AI Systems: NN, Graphical Models, Random Forests  
Tractable Circuits: OBDDs, SDDs, DNNFs,...

How?

Compile ML system into a tractable circuit that makes identical decisions (identical input-output behavior)

A wealth of AI and CS tools become immediately relevant (e.g., knowledge compilation and formal verification)

Choi, Shi, Shih, Darwiche VNN 19
Reasoning about the Behavior of AI Systems

CNN (0-vs-1 CNN) with 98.74% accuracy

Why 0?

Because

I can fool you!

Choi, Shi, Shih, Darwiche VNN 19
Reasoning about the Behavior of AI Systems

two CNNs with almost same accuracy
one is significantly more robust
plots for $2^{256}$ instances!
compiled into SDD circuits
Reasoning about the Behavior of AI Systems

two CNNs with almost same accuracy
one is significantly more robust
plots for $2^{256}$ instances!

compiled into SDD circuits

• New role for symbolic AI & CS methods: Reason about what was learned

• Systems 1/2 (thinking fast and slow), reflection, meta-reasoning
Reasoning about the Behavior of AI Systems

two CNNs with almost same accuracy
one is significantly more robust
plots for $2^{256}$ instances!

• New role for symbolic AI & CS methods: Reason about what was learned
• Systems 1/2 (thinking fast and slow), reflection, meta-reasoning

VNN community: beyond compiling (still exact)
Reasoning about the Behavior of AI Systems

two CNNs with almost same accuracy
one is significantly more robust
plots for $2^{256}$ instances!
compiled into SDD circuits

- New role for symbolic AI & CS methods: Reason about what was learned
- Systems 1/2 (thinking fast and slow), reflection, meta-reasoning

VNN community: beyond compiling (still exact)
Much work on approximate reasoning
Compiling NNs into Tractable Circuits

Neural Network: Binary Inputs & Step Activations

Neural Network’s Boolean Circuit

Neuron’s Boolean Circuit (Chan & Darwiche 2003)
Random Forests

- 15 decision trees of height 6
- Trained by weka
- Total variables 10
- Total variables after splitting 54
- Dataset 40,000 records
- Size of compiled SDD: 6854
- Features:
  1. Revolving Utilization Of Unsecured Lines
  2. Age
  3. Number Of Time 30-59 Days Past Due Not Worse
  4. Debt Ratio
  5. Monthly Income
  6. Number Of Open Credit Lines And Loans
  7. Number Of Times 90 Days Late
  8. Number Real Estate Loans Or Lines
  9. Number Of Time 60-89 Days Past Due Not Worse
  10. Number Of Dependents

Tests (A>3) and (B <= 7.3) are Boolean. Random Forest can be converted into a Boolean circuit, which is then “compiled” into a tractable circuit.

Extracted Rules

If Utilization is less than 0.5, not even a single time 90 days late and not even a single time 60-89 days late, then the person is likely to pay back on time

...
Conclusion

Logic for Computation
reducing ‘Beyond NP’ problems to logical reasoning

Logic for Background Knowledge
learning from a combination of data and knowledge

Logic for Meta Reasoning
reasoning about the behavior of machine learning systems