Learning Probabilistic Sentential Decision Diagrams Under Logic Constraints by Sampling and Averaging

Renato L. Geh
renatolg@ime.usp.br

Denis D. Mauá
ddm@ime.usp.br

Department of Computer Science
Institute of Mathematics and Statistics, University of São Paulo

UAI 2021
Probabilistic Sentential Decision Diagrams (PSDDs):

- Structured Decomposable probabilistic circuits
- Encode certain knowledge as logic constraints
- Encode uncertain knowledge as probabilities
- Interpretable syntax
- Many inferences are **exact** and **tractable**:
  - Evidence
  - Marginals
  - MLE Parameter Learning
  - Most Probable Explanation
  - Expectations
  - KL-divergence

PSDD circuit represents recursive decomposition of formula:

\[
\bigvee_{i=1}^{k} (p_i \land s_i), \text{ where each prime } p_i \text{ and sub } s_i \text{ are logical formulae}
\]

Darwiche [2011], Kisa et al. [2014]
**Probabilistic Sentential Decision Diagrams**

**Existing PSDD learners:**

**LEARNPSDD (Liang et al. [2017]):**
- ✗ Requires initial PSDD encoding the support...
- ✗ Scales poorly to complex formulae and/or high dimension...
- ✗ Costly whole circuit evaluation at every iteration...
- ✔ Very good performance!

**STRUDEL (Dang et al. [2020]):**
- ✔ Constructs an initial PSDD structure (from a CLT)!
- ✗ But does not encode constraints...
- ✔ Scales to high dimension!
- ✗ As long as the circuit doesn’t get too big...

**SAMPLEPSDD (this work):**
- ✔ Scales to high dimension and complex formulae!
- ✔ Constructs a structure consistent with constraints!
- ✗ But does so by relaxing the formula...
- ✗ Performance varies on set bounds and vtree structure...
Common assumption: primes $p_i$ are conjunctions of literals.

$$\phi(A, B, C, D) = (A \land \neg B \land \neg D) \lor (B \land \neg C \land D)$$

Problem: size of circuit is exponential in the size of $p_i$
**SamplePSDD**

**Solution:** randomly sample a bounded number ($k$) of $p_i$

**Example, $k = 3$:**

$$s_i = \phi|_{p_i}$$

**But:** this violates structure decomposability

$\neg C \land D$ contains $C$, and $C \not\in S$

$\neg B \land \neg C \land D$ contains $B$ and $C$, and $B, C \not\in S$
New solution: relax logical constraints $\phi$

Now all $s_i$ respect $S$
Experiments

**Evaluation:** we sample 30 PSDDs and use 5 ensemble strategies:

- Likelihood weighting (LLW),
- Uniform weights,
- Expectation Maximization (EM),
- Stacking,
- Bayesian Model Combination (BMC);

comparing against **Strudel, LearnPSDD** and **LearnSPN**.

**Datasets:** we evaluate with 5 data + knowledge as logic constraints:

<table>
<thead>
<tr>
<th>Dataset</th>
<th>#vars</th>
<th>#train</th>
<th>φ’s size</th>
</tr>
</thead>
<tbody>
<tr>
<td>⇒ LED</td>
<td>14</td>
<td>5000</td>
<td>23</td>
</tr>
<tr>
<td>⇒ LED + Images</td>
<td>157</td>
<td>700</td>
<td>39899</td>
</tr>
<tr>
<td>Sushi Ranking</td>
<td>100</td>
<td>3500</td>
<td>17413</td>
</tr>
<tr>
<td>Sushi Top 5</td>
<td>10</td>
<td>3500</td>
<td>37</td>
</tr>
<tr>
<td>Dota 2 Games</td>
<td>227</td>
<td>92650</td>
<td>1308</td>
</tr>
</tbody>
</table>

Our approach **fares better with fewer data**, yet remains **competitive under lots of data**.

Mattei et al. [2020], Kamishima [2003], Shen et al. [2017], Choi et al. [2015], Gens and Domingos [2013], Dang et al. [2020]
## Experiments

**Evaluation:** we sample 30 PSDDs and use 5 ensemble strategies:

- Likelihood weighting (LLW),
- Uniform weights,
- Expectation Maximization (EM),
- Stacking,
- Bayesian Model Combination (BMC);

comparing against STRUDEL, LEARNPSDD and LEARNSPN.

**Datasets:** we evaluate with 5 data + knowledge as logic constraints:

<table>
<thead>
<tr>
<th>Dataset</th>
<th>#vars</th>
<th>#train</th>
<th>$\phi$’s size</th>
</tr>
</thead>
<tbody>
<tr>
<td>LED</td>
<td>14</td>
<td>5000</td>
<td>23</td>
</tr>
<tr>
<td>LED + IMAGES</td>
<td>157</td>
<td>700</td>
<td>39899</td>
</tr>
<tr>
<td>$\Rightarrow$ SUSHI RANKING</td>
<td>100</td>
<td>3500</td>
<td>17413</td>
</tr>
<tr>
<td>$\Rightarrow$ SUSHI TOP 5</td>
<td>10</td>
<td>3500</td>
<td>37</td>
</tr>
<tr>
<td>DOTA 2 GAMES</td>
<td>227</td>
<td>92650</td>
<td>1308</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Train data percentage</th>
<th>Test log-likelihood</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5% (17)</td>
<td>-7.0×10⁴</td>
</tr>
<tr>
<td>1% (35)</td>
<td>-6.0×10⁴</td>
</tr>
<tr>
<td>2.5% (87)</td>
<td>-5.0×10⁴</td>
</tr>
<tr>
<td>5% (175)</td>
<td>-4.0×10⁴</td>
</tr>
<tr>
<td>10% (350)</td>
<td>-3.0×10⁴</td>
</tr>
<tr>
<td>15% (525)</td>
<td>-2.0×10⁴</td>
</tr>
<tr>
<td>20% (700)</td>
<td>-11000</td>
</tr>
<tr>
<td>25% (875)</td>
<td>-10000</td>
</tr>
<tr>
<td>50% (1750)</td>
<td>-9000</td>
</tr>
<tr>
<td>75% (2625)</td>
<td>-8000</td>
</tr>
<tr>
<td>100% (3500)</td>
<td>-7000</td>
</tr>
</tbody>
</table>

Our approach **fares better with fewer data**, yet remains **competitive under lots of data**.

Mattei et al. [2020], Kamishima [2003], Shen et al. [2017], Choi et al. [2015], Gens and Domingos [2013], Dang et al. [2020]
Experiments

**Evaluation:** we sample 30 PSDDs and use 5 ensemble strategies:

- ![Likelihood weighting (LLW)](#)
- ![Uniform weights](#)
- ![Expectation Maximization (EM)](#)
- ![Stacking](#)
- ![Bayesian Model Combination (BMC)](#)

comparing against **STRUDEL, LEARNPSDD** and **LEARNSPN**.

**Datasets:** we evaluate with 5 data + knowledge as logic constraints:

<table>
<thead>
<tr>
<th>Dataset</th>
<th>#vars</th>
<th>#train</th>
<th>$\phi$’s size</th>
</tr>
</thead>
<tbody>
<tr>
<td>LED</td>
<td>14</td>
<td>5000</td>
<td>23</td>
</tr>
<tr>
<td>LED + IMAGES</td>
<td>157</td>
<td>700</td>
<td>39899</td>
</tr>
<tr>
<td>SUSHI RANKING</td>
<td>100</td>
<td>3500</td>
<td>17413</td>
</tr>
<tr>
<td>SUSHI TOP 5</td>
<td>10</td>
<td>3500</td>
<td>37</td>
</tr>
<tr>
<td>⇒ DOTA 2 GAMES</td>
<td>227</td>
<td>92650</td>
<td>1308</td>
</tr>
</tbody>
</table>

Our approach **fares better with fewer data**, yet remains **competitive under lots of data**.

Mattei et al. [2020], Kamishima [2003], Shen et al. [2017], Choi et al. [2015], Gens and Domingos [2013], Dang et al. [2020]
Experiments

What is the impact of higher $k$'s and right-leaning vtrees in log-likelihood and consistency?

Samples perform better with higher $k$'s and right-leaning vtrees...

...but at a cost to complexity.