**Learning Arithmetic Circuits**

**Daniel Lowd**  
University of Washington  
<lowd@cs.washington.edu>

**Pedro Domingos**  
University of Washington  
<pedrod@cs.washington.edu>

**KEY IDEA:** Prefer models that allow for more efficient inference

**BACKGROUND:** From Bayesian networks to arithmetic circuits

**Algorithm:** Struct. learning + Circuit size penalty + Incremental compilation

**EXPERIMENTS:** Better accuracy, >10,000 times faster inference

---

**Dataset Comparison**

<table>
<thead>
<tr>
<th>Dataset</th>
<th>EachMovie</th>
<th>KDD Cup</th>
<th>MSWeb</th>
</tr>
</thead>
<tbody>
<tr>
<td>AC-Greedy</td>
<td>62ms</td>
<td>194ms</td>
<td>91ms</td>
</tr>
<tr>
<td>AC-Quick</td>
<td>162ms</td>
<td>196ms</td>
<td>115ms</td>
</tr>
<tr>
<td>Gibbs (1k steps)</td>
<td>7.22s</td>
<td>14.45s</td>
<td>1.88s</td>
</tr>
<tr>
<td>Gibbs (10k steps)</td>
<td>42.56s</td>
<td>113.4s</td>
<td>15.6s</td>
</tr>
<tr>
<td>Gibbs (100k steps)</td>
<td>452s</td>
<td>100s</td>
<td>154s</td>
</tr>
<tr>
<td>Gibbs (1M steps)</td>
<td>3912s</td>
<td>1124s</td>
<td>1556s</td>
</tr>
</tbody>
</table>

---

**Solution:**

- **Bayesian networks:** Compactly represent probability distribution over many variables  
  **Problem:** Number of parameters is exponential in the maximum number of parents  
  **Solution:** Context-specific independence

  \[ P(A,B,C,D) = P(A)P(B|A)P(C|A)P(D|B,C) \]

- **with decision-tree CPDs:**  
  **Problem:** Inference is exponential in tree-width  
  **Solution:** Compile to circuits

  \[ P(D|B,C) = \frac{1}{Z} \frac{1}{Z} P(B,C) \]

- **compiled to circuits:**  
  **Problem:** Inference is exponential in tree-width  
  **Solution:** Compile to arithmetic circuits

- **Details:** ACs for Inference

  - Bayesian network: \( P(A,B,C) = P(A)P(B|A)P(C|A,B) \)
  - Network polynomial:
    \[ 1A\cdot(B\cdot C + AB) + ... \]
  - Can compute arbitrary marginal queries by evaluating network polynomial.
  - Arithmetic circuits (ACs) offer efficient, factored representations of this polynomial.

  - ACs can take advantage of local structure such as context-specific independence.

---

**Algorithmic Details**

- **AC-Greedy:**
  - Create initial split list
  - For each split in list
    - Apply highest-scoring split to circuit
    - Reevaluate number of edges added only when another split may have affected it (AC-Greedy).

- **AC-Quick:**
  - Create initial product of ACs
  - Apply highest-scoring split to circuit
  - Add new child splits to list
  - Remove inconsistent splits from list

---

**EXPERIMENTS:**

- **EachMovie:**
  - Gibbs-1A
  - Gibbs-10k
  - Gibbs-100k
  - Gibbs-1M
  - AC-Greedy
  - AC-Quick

- **KDD Cup:**
  - Gibbs-1A
  - Gibbs-10k
  - Gibbs-100k
  - Gibbs-1M
  - AC-Greedy
  - AC-Quick

- **MSWeb:**
  - Gibbs-1A
  - Gibbs-10k
  - Gibbs-100k
  - Gibbs-1M
  - AC-Greedy
  - AC-Quick

---

**Summary:**

- **Bayesian network structure learning often selects models for which inference is intractable.**
- **Our new approach:**
  - Apply standard structure learning algorithm but penalize models with high inference cost.
  - Represent the distribution more compactly using arithmetic circuits and context-specific independence.
- **Now we can learn complex models that allow exact inference in milliseconds!**