StarAI
Exact Symmetries and Changing Domains in Static PRMs
Tutorial ECAI 2020

Tanya Braun, Marcel Gehrke, Ralf Möller
Universität zu Lübeck
Agenda

• Probabilistic relational models (PRMs) [Ralf]
• Exact symmetries and changing domains in static PRMs [Tanya]
  • Exact symmetries
  • Changing domains
  • Unknown domains
• Stable inference over time in dynamic PRMs [Marcel]
• Summary [Tanya]

Goal: Overview of central ideas
Semantics of a PRM

• Joint probability distribution $P_G$
  by grounding

$$P_G = \frac{1}{Z} \prod_{f \in gr(G)} f$$

$$Z = \sum_{\nu \in r(v(gr(G)))} \prod_{f \in gr(G)} f_i(\pi_{rv(f_i)}(\nu))$$

$\pi_{variables}(\nu) =$ projection of $\nu$ onto variables
Grounded Model

- Given domains
  - $D(X) = \{alice, eve, bob\}$
  - $D(M) = \{m_1, m_2\}$
  - $D(D) = \{flood, fire\}$
  - $D(W) = \{virus, war\}$
Grounding Semantics

- Equivalence of lifted and grounded calculations
- Never worse than propositional inference

\[
P(\text{Epid}) \propto \sum_{s \in \mathfrak{r}(\text{Sick}(x_1))} f(\text{Epid}, \text{Sick}(x_1) = s) \\
\cdot \sum_{s \in \mathfrak{r}(\text{Sick}(x_2))} f(\text{Epid}, \text{Sick}(x_2) = s) \\
\cdot \ldots \\
\cdot \sum_{s \in \mathfrak{r}(\text{Sick}(x_n))} f(\text{Epid}, \text{Sick}(x_n) = s)
\]

\[
= f' \cdot f' \cdot \ldots f' = (f')^n
\]

\[
P(\text{Epid}) \propto \left( \sum_{s \in \mathfrak{r}(\text{Sick}(X))} f(\text{Epid}, \text{Sick}(X) = s) \right)^n = (f')^n
\]
Complexity

- **Query answering problem**
  - Given a model, ask for probability distribution of a grounded PRV

- Given a model that allows for lifted calculations
  - I.e., no groundings during solving an instance of the problem

- Solving an instance of the problem is possible in time *polynomial in domain sizes*
  - No longer exponential in domain sizes

→ The query answering algorithm is *domain-lifted*

Van den Broeck (2011)
Completeness

• No groundings in all possible models given some characteristic
  • Algorithm is *domain-lifted* in each possible model

• Model characteristics
  • Two logical variables per parfactor
    $g(A(X,Y), B(X,Y))$
    $g(A(X,Y), C(X), C(Y)), X \neq Y$
    $g(A(X,Y), D(X), E(Y))$
  • One logical variable per PRV (arbitrarily many logical variables per parfactor)
    $g(A(X), B(Y), C(Z))$
  • Holds for various domain-lifted algorithms, e.g.,
    • Lifted variable elimination Taghipour et al. (2013)
    • Lifted junction tree algorithm B (2020)
    • First-order knowledge compilation Van den Broeck (2011)

• Class of such models called *liftable*
Completeness

• Models with other constellations may be computed without groundings but not all possible models
  • E.g., for lifted variable elimination, models with three logical variables

\[ g(A(X,Y,Z),B(Y),C(Z)) \rightarrow \text{liftable} \]

\[ g(A(X,Y),A(Y,Z),A(X,Z)) \rightarrow \text{not liftable} \]

→ Not complete for three logical variables per parfactor
Tractability

• An query answering problem is tractable
  • when it is solved by an efficient algorithm, running in
time polynomial in the number of random variables

• Assume that the number of random variables is
characterised by domain sizes
  • Then, solving a query answering problem is tractable
under domain-liftability
  • Runtime might still be exponential in other terms

• More general results by
  Niepert and Van den Broeck (2014)
  • Tractability through Exchangeability
From a Ground Model to a Lifted Model

Using exact symmetries in the ground model
Compression

Goal

Input

Nat(D) \rightarrow Man(W)
\text{Epid}
\text{Travel}(X) \rightarrow \text{Treat}(X, M)
\text{Sick}(X)

\text{Man} \cdot \text{war}
\text{Man} \cdot \text{virus}
\text{Nat} \cdot \text{flood}
\text{Nat} \cdot \text{fire}

\text{Epid}
\text{Travel} (X)
\text{Treat} (X, M)
\text{Sick} (X)

\text{Input}

\text{Travel} \cdot \text{alice}
\text{Sick} \cdot \text{alice}
\text{Treat} \cdot \text{alice} \cdot m_1
\text{Treat} \cdot \text{alice} \cdot m_2
\text{Travel} \cdot \text{eve}
\text{Sick} \cdot \text{eve}
\text{Treat} \cdot \text{eve} \cdot m_1
\text{Treat} \cdot \text{eve} \cdot m_2
\text{Treat} \cdot \text{bob} \cdot m_1
\text{Treat} \cdot \text{bob} \cdot m_2
\text{Travel} \cdot \text{bob}
\text{Sick} \cdot \text{bob}
A Bit of History...

- Pearl’s Belief propagation
  - Messages on Bayes net
  - Exact for polytrees (no cycles in undirected graph!)
  - Precursor of junction tree alg. (cycles go into clusters)

Pearl (1982), Lauritzen and Spiegelhalter (1988)
Loopy Belief Propagation

- Pass messages on graph
  - If no cycles: exact
  - Else: approximate

- Lifted (loopy) belief propagation
  - Exploit computational symmetries
  - Compress graph whenever nodes would send identical messages
  - Send messages on compressed graph

→ Colour passing algorithm for compression
Compression: Pass the colours around*

- **Colour nodes according to the evidence you have**
  - No evidence, say **red**
  - State „one“, say **brown**
  - State „two“, say **orange**
  - ...

- **Colour factors distinctively according to their equivalences**
  For instance, assuming $f_1$ and $f_2$ to be identical and $B$ appears at the second position within both, say **blue**

*can also be done at the „lifted“, i.e., relational level

Singla and Domingos (2008), Kersting et al. (2009), Ahmadi et al. (2013)
Compression

1. Colour nodes and factors
   - 1 colour for the nodes: ●
   - 3 colours for the factors: ■ □ ▲
Compression

2. Factors collecting colours from nodes, signing their own colours to the collected ones
Compression

3. Nodes collecting colours from factors
4. Recolour nodes based on collected signatures
   • 5 colours for the nodes: ● ● ● ● ●
   • Factors as before
Compression

5. If no new colour created, stop. Otherwise, pass colours again.

- Before: ●
- After: ● ● ● ● ●

- Trav: alice
- Sick: alice
- Treat: lice.m
- Trav: lice.m
- Trav: eve
- Sick: eve
- Treat: eve.m
- Treat: bob
- Trav: bob
- Sick: bob
- Treat: bob.m
- Treat: bob.m
- Treat: bob.m
Compression

2. Factors collecting colours from nodes, signing their own colours to the collected ones
Compression

3. Nodes collecting colours from factors
4. Recolour nodes based on collected signatures
   • 5 colours for the nodes: ● ● ● ● ●
   • Factors as before
   • No new colour!
5. If no new colour created, stop. Otherwise, pass colours again.
   • Before: •••••
   • After: •••••
   • No new colour!
Compression

• Compressed graph:
Colour Passing Compression

• Algorithm:
  1. Each factor collects the colours of its neighbouring nodes
  2. Each factor „signs“ its colour signature with its own colour
  3. Each node collects the signatures of its neighbouring factors
  4. Nodes are recoloured according to the collected signatures
  5. If no new colour is created stop, otherwise go back to 1

• Compress a model (lifted or grounded) based on semantics
  • Uses exact symmetries in factors
    • Same colour if factors considered equivalent
  • Ignores syntax
    • E.g., names of randvars

Singla and Domingos (2008), Kersting et al. (2009), Ahmadi et al. (2013)
Exact Symmetries

• Symmetries in (propositional) model allow for compact representation using parameters
  • PRVs for sets of indistinguishable randvars

• If randvars are indistinguishable,
  • what about yielding similar or even indistinguishable observations?

→ Next part!

Have a nice break!

We see each other again in 15 minutes.
Symmetric Models & Symmetric Evidence

There are only so many values one can observe
Symmetric Evidence

• Observations for specific randvars of a PRV can be
  • One of the range values
  • Not available

• Example: \( \text{sick}(X), r(\text{sick}(X)) = \{\text{true}, \text{false}\} \)
  • \( \text{sick}(x_1) = \text{sick}(x_2) = \cdots = \text{sick}(x_{10}) = \text{true} \)
  • \( \text{sick}(x_{11}) = \text{sick}(x_{12}) = \cdots = \text{sick}(x_{20}) = \text{false} \)

<table>
<thead>
<tr>
<th>\text{sick}(X^T)</th>
<th>g_e^T</th>
</tr>
</thead>
<tbody>
<tr>
<td>false</td>
<td>0</td>
</tr>
<tr>
<td>true</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>\text{sick}(X^F)</th>
<th>g_e^F</th>
</tr>
</thead>
<tbody>
<tr>
<td>false</td>
<td>1</td>
</tr>
<tr>
<td>true</td>
<td>0</td>
</tr>
</tbody>
</table>

• \( D(X^T) = \{x_1, \ldots, x_{10}\}, D(X^F) = \{x_{11}, \ldots, x_{20}\} \)
• Observations for \( \text{sick}(x_{21}) \ldots \text{sick}(x_n) \) not available
Symmetric Evidence

• Evidence: $g^T_e, g^F_e$
  • $\mathcal{D}(X^T) = \{x_1, \ldots, x_{10}\}$
  • $\mathcal{D}(X^F) = \{x_{11}, \ldots, x_{20}\}$
  • $\mathcal{D}(X') = \{x_{21}, \ldots, x_n\}$

• Shattering based on evidence

Taghipour et al. (2013a)
Evidence Absorption

• Absorb evidence:
  • Set values to 0 where range value ≠ observation
    • Equivalent to multiplying $g$ with $g_e$
  • Possibly eliminate variable
    • Drop lines with values set to 0
    • Drop column of evidence PRV
  • Example
    • $Sick(X^T) = true$
    • $Sick(X^F) = false$

<table>
<thead>
<tr>
<th>Epid</th>
<th>Sick($X^F$)</th>
<th>$g^F$</th>
</tr>
</thead>
<tbody>
<tr>
<td>false</td>
<td>false</td>
<td>5</td>
</tr>
<tr>
<td>false</td>
<td>true</td>
<td>4</td>
</tr>
<tr>
<td>true</td>
<td>false</td>
<td>4</td>
</tr>
<tr>
<td>true</td>
<td>true</td>
<td>6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Epid</th>
<th>Sick($X^T$)</th>
<th>$g^T$</th>
</tr>
</thead>
<tbody>
<tr>
<td>false</td>
<td>false</td>
<td>5 0</td>
</tr>
<tr>
<td>false</td>
<td>true</td>
<td>1</td>
</tr>
<tr>
<td>true</td>
<td>false</td>
<td>4 0</td>
</tr>
<tr>
<td>true</td>
<td>true</td>
<td>6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Epid</th>
<th>$g^F$</th>
</tr>
</thead>
<tbody>
<tr>
<td>false</td>
<td>5</td>
</tr>
<tr>
<td>true</td>
<td>4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Epid</th>
<th>$g^T$</th>
</tr>
</thead>
<tbody>
<tr>
<td>false</td>
<td>1</td>
</tr>
<tr>
<td>true</td>
<td>6</td>
</tr>
</tbody>
</table>
Symmetric Evidence

• Shattering based on evidence

• After absorption
Lifted Evidence & Completeness

• Evidence is liftable if observations for
  • Propositional randvars
  • PRVs with one logical variable
    • One set of constants per variable
    • E.g., observations for, e.g., Travel(X), Sick(X)

• Evidence for PRVs with two logical variables no longer liftable
  • Liftable cases possible but no guarantee for all possible constellations
  • More by Van den Broeck and Darwiche (2013) on special classes
Symmetries in Queries

Indistinguishable query terms

Also: a highlight paper here at ECAI 2020!
Indistinguishable Query Terms

- Indistinguishable randvars in query:

  \[ P(\text{Sick}(alice), \text{Sick}(eve), \text{Sick}(bob)) \]

- Standard LVE:
  - Shattering
    - Leads to groundings w.r.t. constants in query

B and Möller (2018)
... And Their Effect

• Query: \( P(\text{Sick}(\text{alice}), \text{Sick}(\text{eve}), \text{Sick}(\text{bob})) \)

• After shattering, eliminate all non-query terms
  • Identical computations during elimination
... And Their Effect

- Query: $P(\text{Sick(alice)}, \text{Sick(eve)}, \text{Sick(bob)})$
- After shattering, eliminate all non-query terms
  - Identical computations during elimination
  - Large intermediate results

![Graph Diagram]

- $Epid$
- $g$
- $\text{Sick(alice)}$
- $\text{Sick(eve)}$
- $\text{Sick(bob)}$
... And Their Effect

• Query: $P(Sick(\text{alice}), Sick(\text{eve}), Sick(\text{bob}))$
• After shattering, eliminate all non-query terms
• Symmetries in result

```
<table>
<thead>
<tr>
<th></th>
<th>Sick(\text{alice})</th>
<th>Sick(\text{eve})</th>
<th>Sick(\text{bob})</th>
<th>g'</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>false</td>
<td>false</td>
<td>false</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>false</td>
<td>false</td>
<td>true</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>false</td>
<td>true</td>
<td>false</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>false</td>
<td>true</td>
<td>true</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>true</td>
<td>false</td>
<td>false</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>true</td>
<td>false</td>
<td>true</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>true</td>
<td>true</td>
<td>false</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>true</td>
<td>true</td>
<td>true</td>
<td>4</td>
</tr>
</tbody>
</table>
```

#$_x[Sick(X)]$ | $g$ |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>[0,3]</td>
<td>1</td>
</tr>
<tr>
<td>[1,2]</td>
<td>2</td>
</tr>
<tr>
<td>[2,1]</td>
<td>3</td>
</tr>
<tr>
<td>[3,0]</td>
<td>4</td>
</tr>
</tbody>
</table>
Lifted Queries

- Parameterised query: $P(Sick(X))$

- Standard LVE:
  - Shattering
    - If $X$ references a subdomain, then two groups
  - Elimination
    - Using standard LVE
  - Encode symmetries using so-called counting random variables, which have histograms as range values
    - Using LVE operator called count-conversion
    - If not already a by-product of elimination

B and Möller (2018)
Lifted Queries & Completeness

• Given a liftable model and liftable evidence

• Complexity
  • The complexity of LVE for liftable queries is polynomial in domain sizes.

• Completeness
  • Parameterised query terms with only one parameter per term and one set of constants per domain are liftable.
    • Otherwise, groundings may be unavoidable, e.g.,
      Query $P(B(X,Y))$ in model $g(A(X), B(X,Y), C(Y))$

• Corollary
  • Counting random variables compactly represent the result of liftable queries.
Known Domains

• Grounding semantics is only defined given specific domains for logical variables
  • Evidence for known constants
  • Queries reference known constants
• Also, models usually learned on a specific domain

• What if...
  • domains change?
  • domains are unknown?
Leaving a specific domain behind…

What happens if domains change?
Changing Domains

• Keep semantics as before
  • Assume that parfactors accurately describe world

• Posterior probabilities change depending on domain sizes
  • Example by Poole (2003)
... Without Effects

• **(Conditional) Independence**
  PRVs, containing logical variables $X$, that are (conditionally) independent from query terms $\rightarrow$ domains of $X$ have no influence on query results

• E.g., given $Epid = e$,
  • $D(D)$ and $D(W)$ do not matter for queries regarding $Travel$, $Sick$, and $Treat$
  • $D(X)$ and $D(M)$ do not matter for queries regarding $Nat$ and $Man$

$\rightarrow$ Partly invariant under increasing domain sizes
... Without Effects

• A simple case of so-called **projectivity**
  After shattering, query terms are independent of model parts containing logical variables \( X \rightarrow \) domains of \( X \) have no influence on query results
  • Depends on model structure
  • More by Jaeger and Schulte (2018)

• E.g., \( P(\text{Sick}(x_1)) \)
  • \( \mathcal{D}(X) = \{x_1, \ldots, x_n\} \)
  • After shattering:
    • \( \mathcal{D}(X) = \{x_2, \ldots, x_n\} \)
    • Upper part independent from lower part; \( \mathcal{D}(X) \) irrelevant

→ Partly invariant under increasing domain sizes
Growing Domain Sizes

- Let domain size $n$ grow
  - With grounding semantics, posteriors change
    - Can lead to extreme behaviour in the posteriors
- Example: $Epid$ gets more and more neighbours with $n$ rising

$$P(Epid) \propto \left( \sum_{s \in r(Sick(x))} g(Epid, Sick(x) = s) \right)^n$$

$$= (g'(Epid))^n = g''(Epid) = g^\alpha(Epid)$$

\[
\begin{array}{c|c|c}
Epid & g' & g'' \\
\hline
false & a & a^n \\
true & b & b^n \\
\end{array}
\]

\[
\begin{array}{c|c|c}
Epid & g^\alpha \\
\hline
false & \frac{a^n}{a^n + b^n} \\
true & \frac{b^n}{a^n + b^n} \\
\end{array}
\]

$$1 \over 1 + (b/a)^n$$

\(n \to \infty\)

$$b > a \quad 0$$

$$a > b \quad 1$$

Sigmoid function
Growing Domain Sizes

• How to avoid extreme behaviour?

→ Adapt values in model dependent on domain size

• Approach for MLNs: **Domain-size aware MLNs**
  
  • Assume predicates $P_1, ..., P_m$ occur in a first-order formula $F$  
    • Count number of connections $c_j$ for each predicate $P_j$ given new domains  
    • Build a connection vector $[c_1, ..., c_m]$  
    • Choose $\max_{c_i} [c_1, ..., c_m]$ as scaling-down factor  
      • Instead of max, other functions possible  
      • Works best if the values in $[c_1, ..., c_m]$ do not vary that much  
  
  • Given an MLN with a set of formulas $F_i$ with weights $w_i$  
    • Rescale each $w_i$ with scaling-down factor $s_i$ computed for $F_i$ as $\frac{w_i}{s_i}$  

• Analogous approach possible for parfactors

*Mittal et al. (2019)*
Leaving a specific domain behind...

What happens if a domain is unknown?
Known Domains

- General domains of logical variables, e.g.,
  - $\mathcal{D}(X) = \{alice, eve, bob\}$ or
  - $\mathcal{D}(X) = \{x_1, \ldots, x_n\}$
- Constraint $C_i$ in each parfactor $g_i$, e.g.,
  - $C_3 = ((X, M), \mathcal{D}(X) \times \mathcal{D}(M))$
- Based on constraints, grounding semantics apply
  - Lifted algorithms work
Unknown Domains

• General domains of logical variables, e.g.,
  • $\mathcal{D}(X) = \{alice, eve, bob\}$-or
  • $\mathcal{D}(X) = \{x_1, \ldots, x_n\}$

• Constraint $C_i$ in each parfactor $g_i$, e.g.,
  • $C_3 = ((X, M), \mathcal{D}(X) \times \mathcal{D}(M))$

• Based on constraints, grounding semantics apply
  • Lifted algorithms work
Template Model + Constraint Program

- **Template model**: Parfactors with empty constraints, e.g.,
  - \( C_3 = ((X, M), \bot) \)

- **Constraint program**: Generate specific constraints for a template model given a domain, e.g., using probabilistic Datalog:

  ```prolog
  element_of_C3(X,Y1) :- linked(X,Y1,Y2).
  element_of_C3(X,Y2) :- linked(X,Y1,Y2).
  linked(X,Y1,Y2) :- instance_of_X(X) & pair(Y1,Y2).
  0.7 pair(t1,t2).
  0.2 pair(t2,t3).
  0.1 pair(t1,t3)
  ```

  Yields 3 constraint sets per domain
Domain Worlds

- Specify or generate possible domains
- Encode assumptions, e.g.,
  - Small domains more likely than large domains
  - Only rough
- For $X$, e.g.,
  - Beta-binomial distribution ($\alpha = 6, \beta = 15$)

Yields 20 domains with probability > 0
Groundings-based Semantics

Inputs
- Template model
  - Empty constraints
- Constraint program
  - Fill empty constraints given a domain world
  - Can generate a probability distribution over models
- Domain worlds
  - Generate possible worlds as input to constraint program
  - Can be a probability distribution over domains

Approach
- Generate a set of possible models
  - Can be a probability distribution over possible models
  - Within model: grounding semantics apply
    - Lifted algorithms work again
- Reasoning over possible models
  - New query types
Interworkings

• Distribution over domains

Together, they yield 20 · 3 constraint sets, each with probability > 0

• ... as input to probabilistic constraint program

```
element_of_C3(X,Y1) :- linked(X,Y1,Y2).
element_of_C3(X,Y2) :- linked(X,Y1,Y2).
linked(X,Y1,Y2) :- instance_of_X(X) & pair(Y1,Y2).
0.7 pair(t1,t2).
0.2 pair(t2,t3).
0.1 pair(t1,t3)
```
Filtering

• Together, they yield $20 \cdot 3$ constraint sets, each with probability $> 0$
  • Some probabilities very low

• Filtering based on probabilities; e.g.,
  • Threshold $t$
  • Keep only those models whose probabilities make up, e.g., 95% of the distribution around its mean or maximum value

• Cascading filtering
  1. Filter domain worlds
  2. Filter constraint sets resulting from remaining domain worlds
New Queries Emerging

- Exploration
  - Model and query probabilities w.r.t.
    - Domain sizes (as in changing domains + grounding semantics)
    - Skyline query

- Model checking
  - E.g., does the probability of
    - an individual being sick decrease with larger domains?
    - an epidemic rise if more people travel?

B and Möller (2019)
Wrap-up Symmetries and Domains

• Exact symmetries in PRMs
  • Grounding semantics
  • Tractability of query answering problem
  • Colour passing for exact compression of models
  • Symmetric evidence for lifted evidence handling
  • Lifted queries for lifted query answering

• Changing domains
  • Models that are invariant under increasing domain sizes
  • Adapting weights to avoid extreme behaviour

• Unknown domains
  • Set of or distribution over universes

Next: Stable inference over time in dynamic PRMs
Bibliography

Alphabetically sorted
Bibliography

• Ahmadi et al. (2013)

• B (2020)

• B and Möller (2018)

• B and Möller (2019)
Bibliography

• Jaeger and Schulte (2018)

• Kersting et al. (2009)

• Lauritzen and Spiegelhalter (1988)

• Mittal et al. (2019)
• **Niepert and Van den Broeck (2014)**

• **Pearl (1982)**

• **Poole (2003)**

• **Poole et al. (2014)**
Bibliography

• Singla and Domingos (2008)

• Taghipour et al. (2013)

• Taghipour et al. (2013a)

• Van den Broeck (2011)
Bibliography

• **Van den Broeck and Darwiche (2013)**

• **Van den Broeck and Davis (2012)**
The End

*PRMs are a true backbone of AI, and this tutorial emphasized only some central topics. We definitely did not cite all publications relevant to the whole field of PRMs here. We would like to thank all our colleagues for making their slides available (see some of the references to papers for respective credits). Slides or parts of it are almost always modified.