1. Motivations:

Studying relations can be useful for two reasons:

1. In Multi-Target Tracking: taking into account relations between different objects can improve tracking, e.g. if two objects are moving together they will behave similarly, if two players are in the same team they will coordinate their actions, ...

2. For Activity Recognition tasks: studying relations between different objects can be helpful to recognize particular actions that involve multiple actors.

Under Markov assumption: Bayesian Filter algorithm:

\[
\text{Belief: } b_i(s) = p(s_i|x_{i:t}) = kp(z_i|x_i) / p(s_i|x_i) ds_i
\]

Relations in the State result in correlating the State of different instantiations between them.

1st assumption:

Sensor model: part of the state relative to relations, \( s' \), not directly observable

\[
p(z_i|s_i) = p(z_i|s_i) = p(z_i|s_i)
\]

2nd assumption:

Transition model:

\[
p(s_i|s_{i-1}) = p(s_i, s_{i-1}, s_{i-1}, s_i)
\]

But \( s_i \) independent by \( s_{i-1} \) given \( s_{i-1} \) and \( s_i \)

\[
p(s_i, s_{i-1}, s_{i-1}) = p(s_i|s_{i-1}, s_{i-1}) p(s_{i-1}|s_{i-1}, s_i)
\]

6. Preliminary experiments and results:

Crossroads:

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1/6</td>
<td>1/16</td>
<td>1/16</td>
<td>1/5</td>
<td>1/6</td>
<td>1/6</td>
</tr>
<tr>
<td>1/6</td>
<td>1/2</td>
<td>1/1</td>
<td>1/6</td>
<td>1/6</td>
<td>1/6</td>
</tr>
</tbody>
</table>

Comparing a not-relational Particle filter to our Relational Particle Filter: average error for each of the 15 tracks over 10 iterations of the filtering algorithms, in red, blue and green the error relative to three particular groups of objects in relation:

The worst (time step 24) and the best (time step 12) ROC curve for the relation recognition task.

2. Basic Definitions:

Relational Domain:
the set of object-types and relations and/or predicates between them.

Relational State:
the states of a Relational Domain is the set of instantiations of the objects in the Domain, their attributes and their relations.

Relational State = State of Instantiations (s') + State of Relations (s').

State evolves with time.

3. Relational Dynamic Bayesian Networks:

RDBNs:
couple of RBNs (intra-slice distribution, sensor model; inter-slice distribution, transition model)

RBNs:
set of nodes (one for each predicate/relation/attribute) whose causality is encoded as a DAG.

4. Transition and Sensor Models:

Under Markov assumption: RBNs:

\[
\begin{align*}
\P(x_i|s_i,x_{i-1}&,s_{i-1}) = p(x_i|x_{i-1},s_{i-1},s_i) \\
\end{align*}
\]

More formally:

\[
\P(x(x_i_k)|s(x_{i-1},s_{i-1}),s_i) = p(x(x_i_k)|s(x_{i-1},s_{i-1}),s_i)
\]

5. Relational Particle Filter:

Algorithm 1: Pseudo Code for the Relational Particle Filter algorithm

\[
\begin{align*}
&\text{for all } m = 1 : M \\
&\text{sample } x_i^{(m)} \sim p(x_i|x_{i-1},s_{i-1}) \\
&\text{sample } x_i^{(m)} \sim p(x_i|x_{i-1},s_{i-1}) \\
&\text{compute weights for the state of instantiations:} \\
&\text{compute weights for the state of relations:} \\
&\text{weights normalization:} \\
&\text{Resample } b_i(s) \text{ from } \{x_i^{(m_{1:2})},x_i^{(m_{1:2})}\} \text{ according to weights:} \\
&\text{with repetition.}
\end{align*}
\]

Data: 15 simulated objects, moving on the crossroads for 30 time steps. From each cell, an object can jump to one of the next n cells, where n depends by the cell. Some objects are "traveling" together. If traveling together (i.e. if the relation is true), two (or more) objects will always be in cells from which it is possible for one to reach the other or vice-versa. If moving together, two objects will behave similarly (i.e. if one turns left, the other will follow).