Multi-Entity Bayesian Networks Learning in Predictive Situation Awareness

Cheol Young Park [STUDENT]
Dr. Kathryn Blackmond Laskey
Dr. Paulo Costa
Shou Matsumoto [STUDENT]

Index

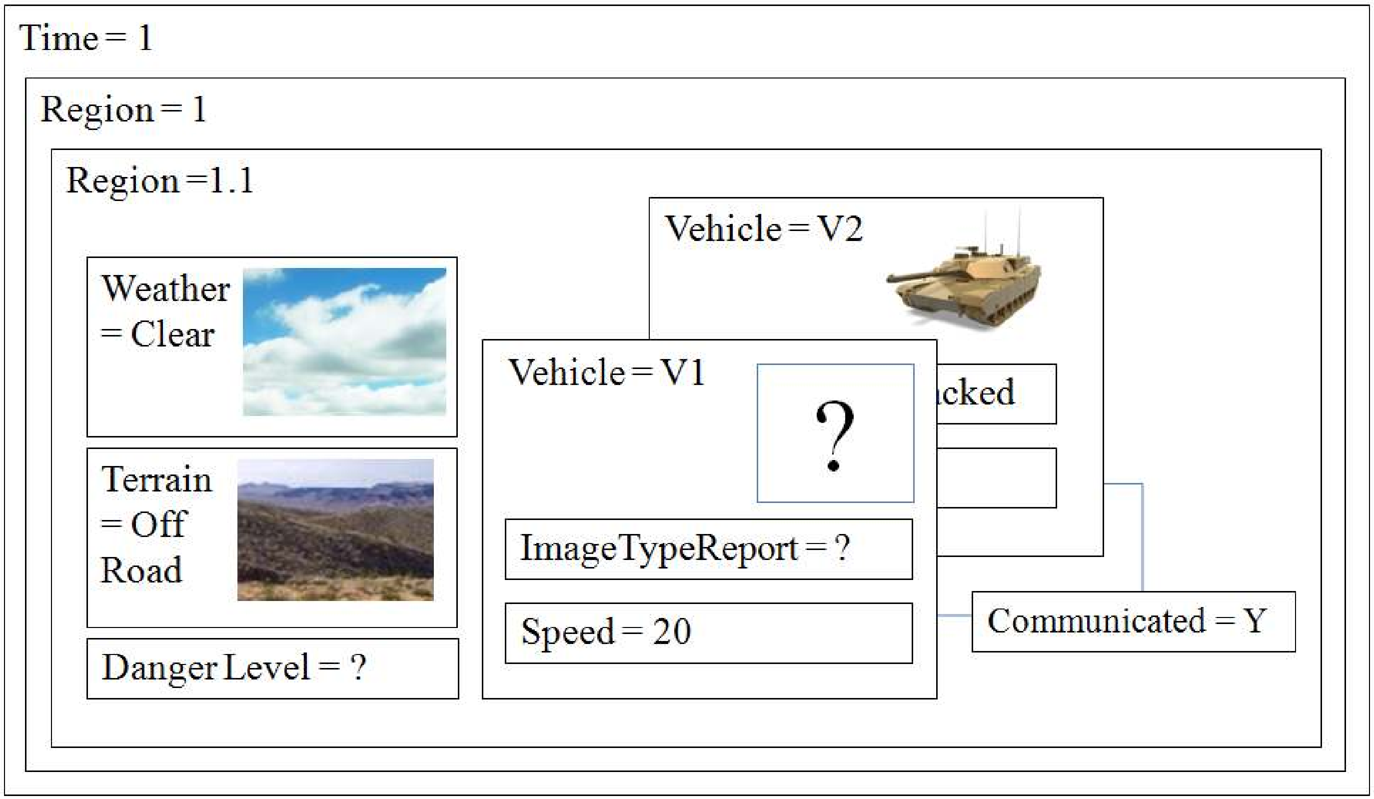
1. Introduction

Data fusion-SAW-C2

- Data Fusion
  - Integration Process of multiple data and knowledge
- Situation Awareness (SAW)
  - Perception
  - Comprehension
  - Projection
- Predictive Situation Awareness (PSAW)
  - Estimation and prediction of an evolving situation over time

1. Introduction

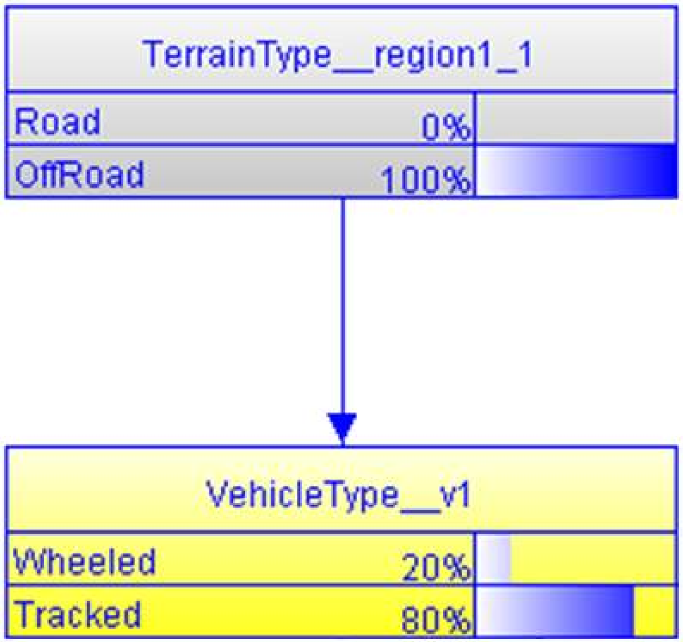
An example of PSAW situation



What is the type of the V1 given the observations ?
What is the danger level of the region 1.1 given the observations ?

1. Introduction

Bayesian Networks for the example



Directed Acyclic Graph (DAG)

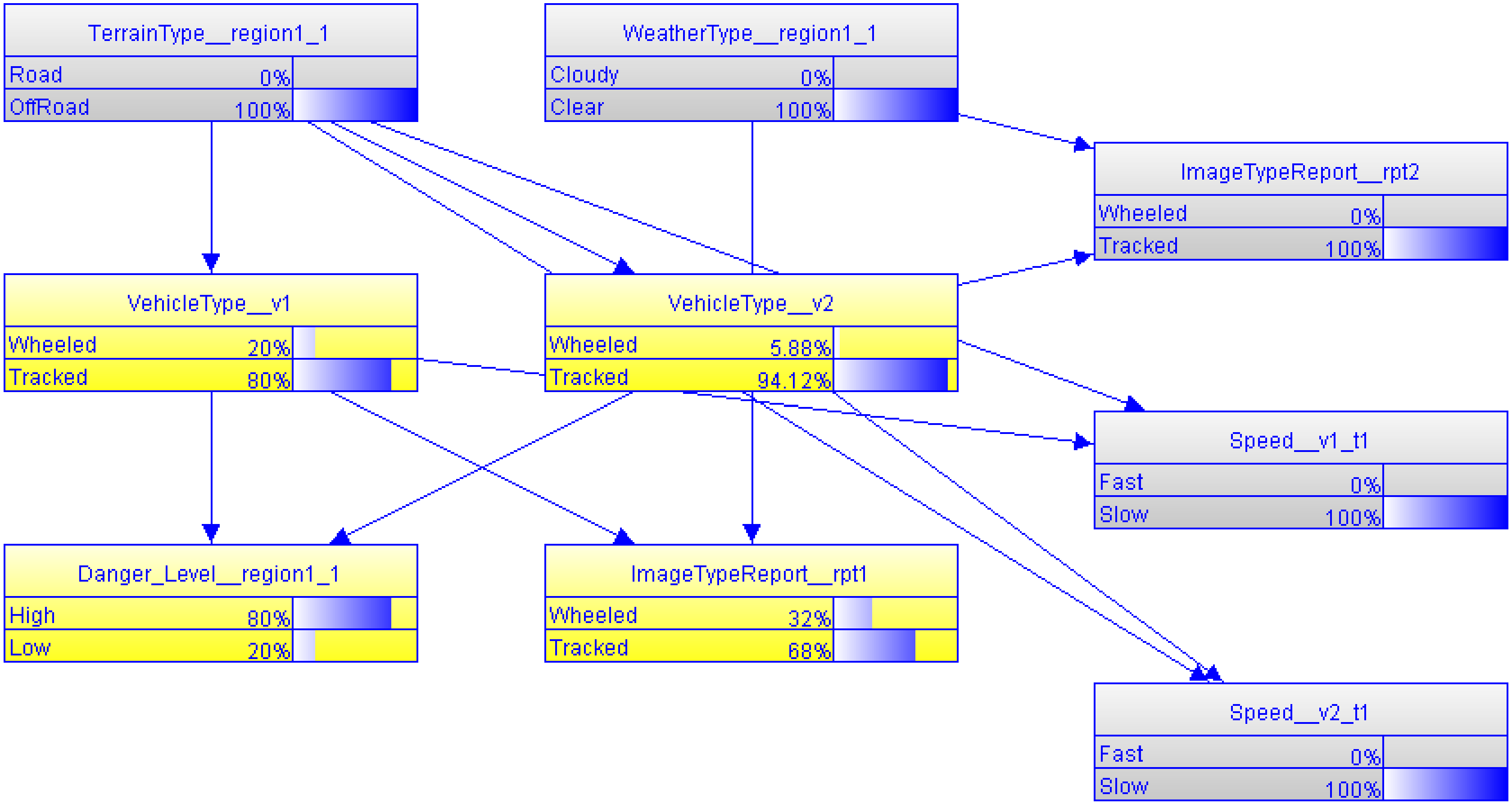
TerrainType__region1_1	Road	OffRoad
Wheeled	0.8	0.2
Tracked	0.2	0.8

Conditional Probability Distribution (CPD)

Observations: Terrain Type of region 1.1
Queries: Vehicle Type of V1

1. Introduction

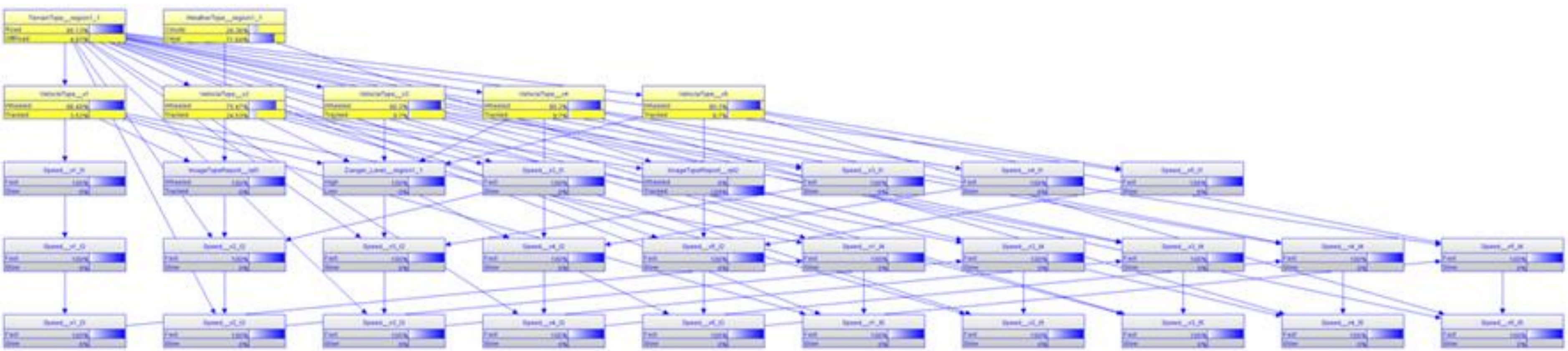
Bayesian Networks for the example



Observations: Terrain Type, Weather, Image of V2, Speed of V1 and V2

Queries: Vehicle Type of V1, Danger level of region 1.1

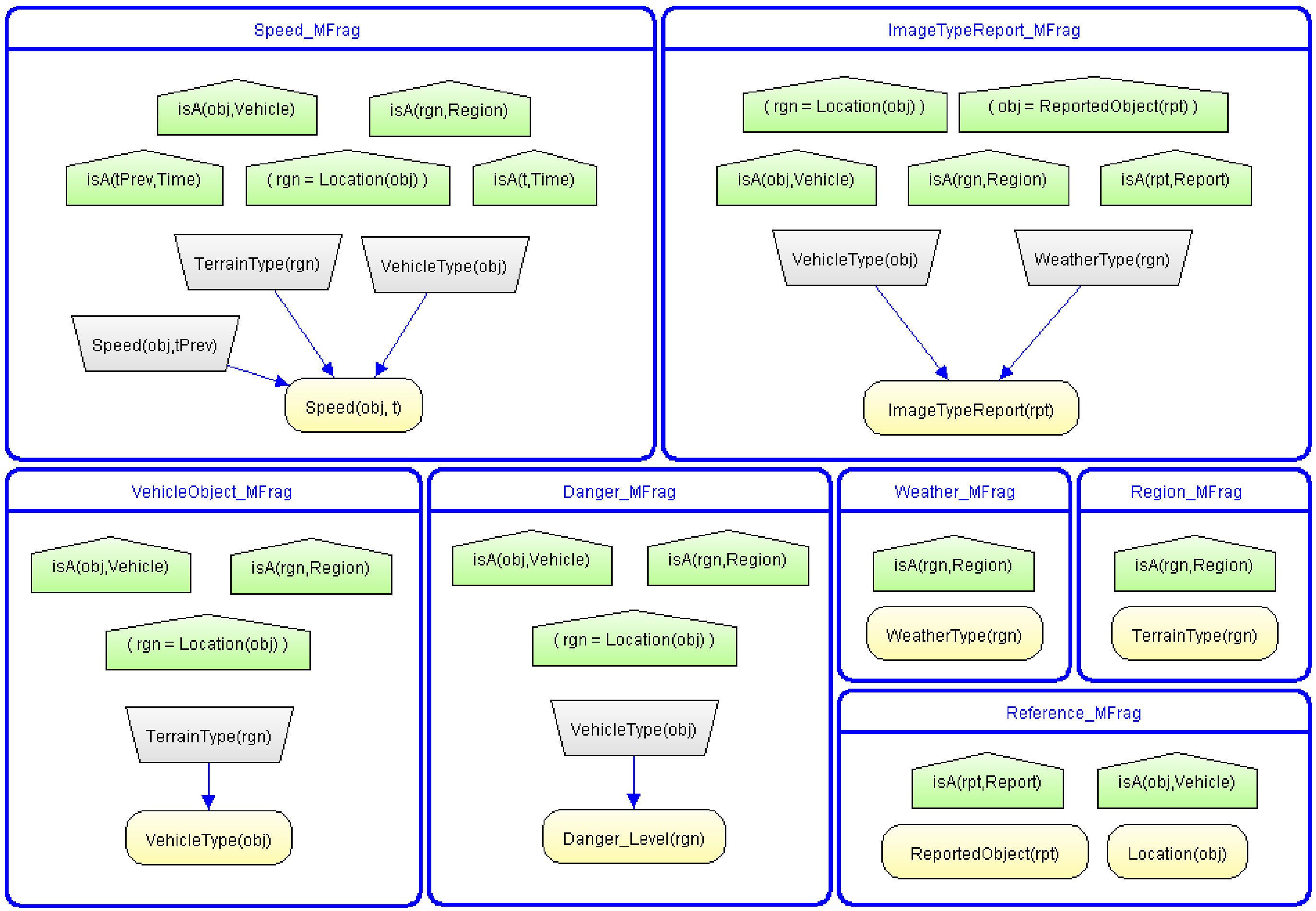
Bayesian Networks for the example



Observations: Image type and speed for V1 ~ V5 in time 1 ~ 5

Queries: Vehicle Type of V1, V2, V3, V4, and V5

MEBN Model(MTheory) from the example



MFrag

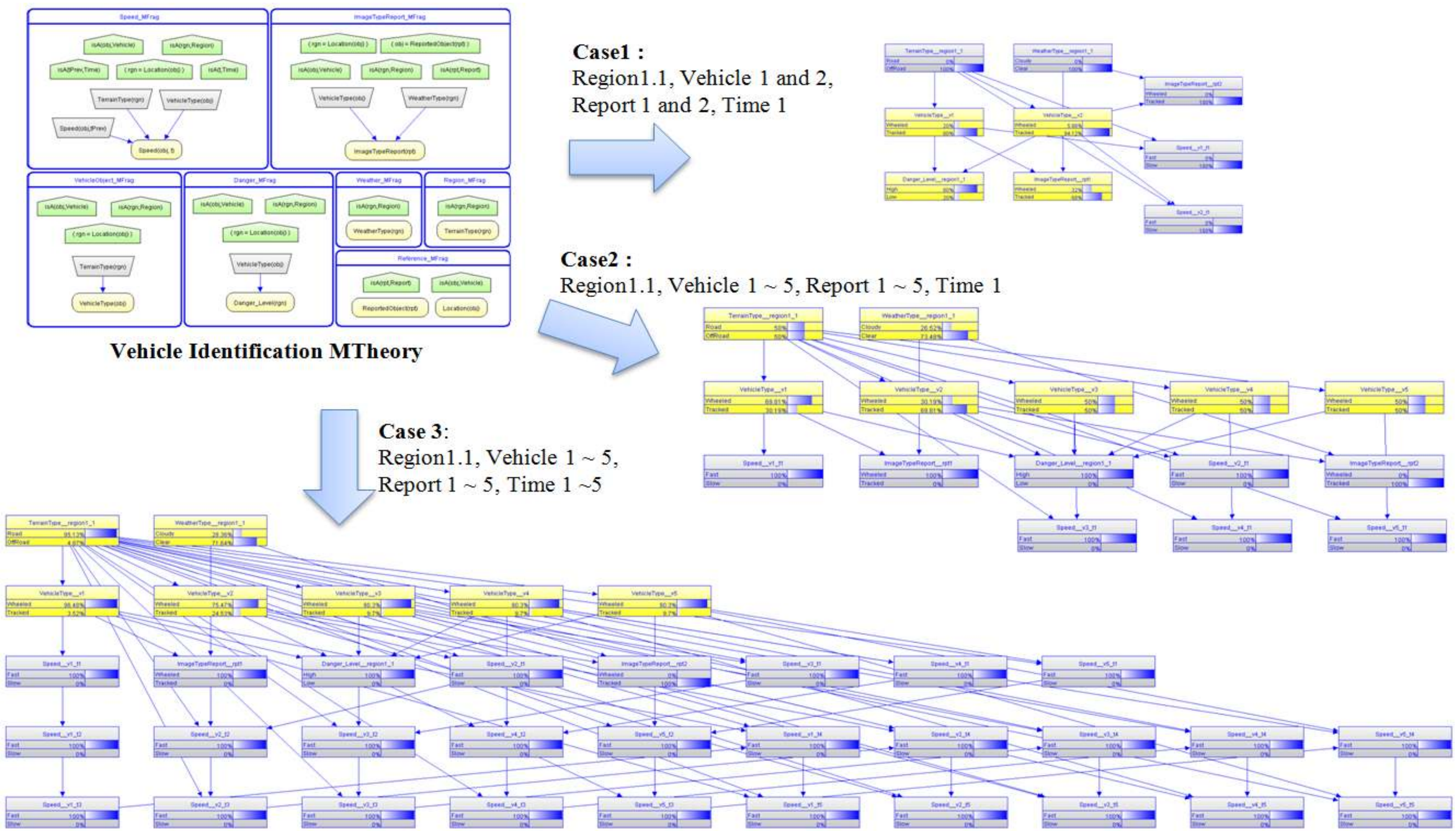
isA(tPrev,Time) Context Node

VehicleType(obj) Input Node

Danger_Level(rgn) Resident Node

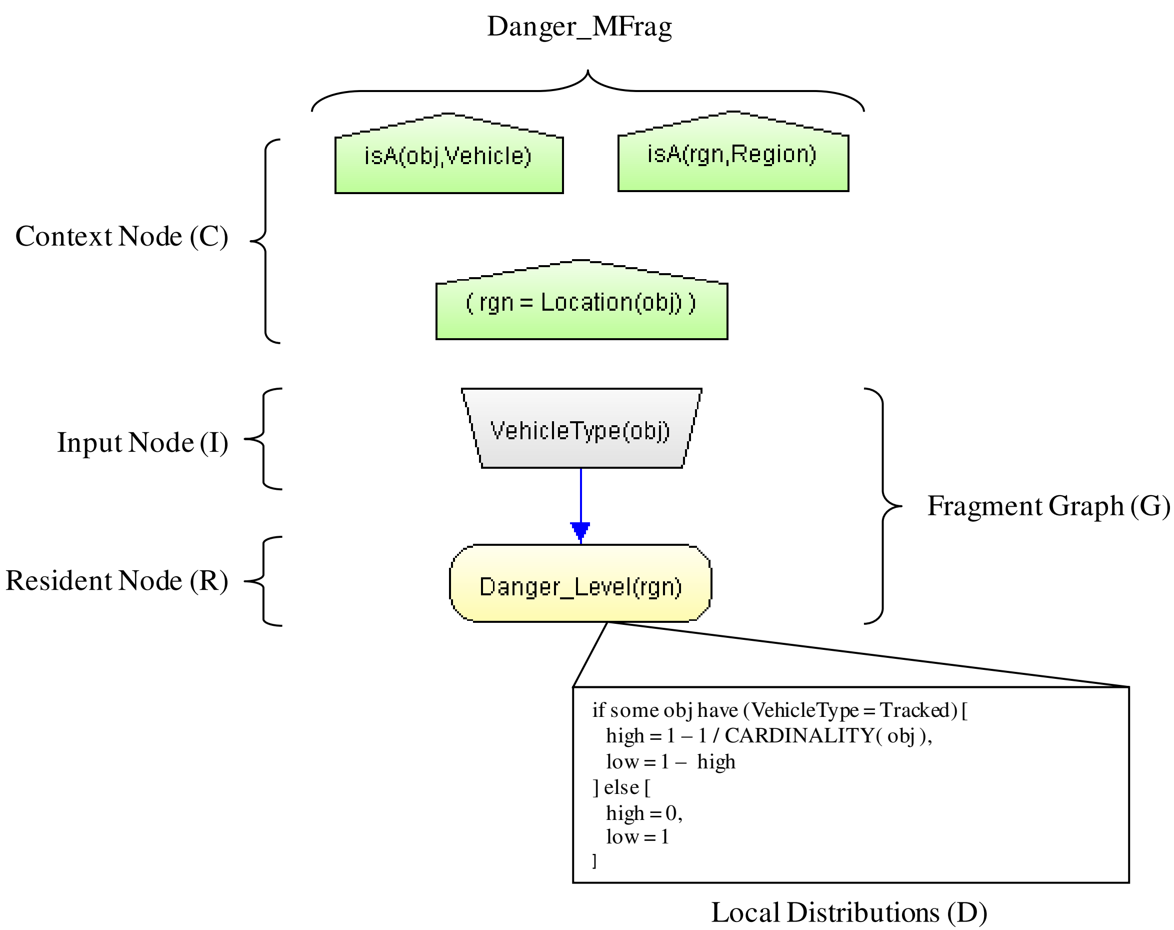
1. Introduction

SSBN generation



Given entities, the MTheory can generate many different BNs

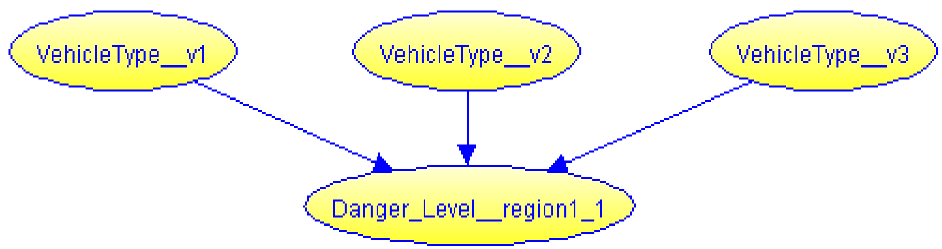
A Danger MFrag



Context node represents conditions under which the distribution defined in the MFrag is valid
Resident node is a random variable containing a term of First Order Logic
Input node is an imported resident node from other MFrag

1. Introduction

Generated SSBN from the Danger MFrag



VehicleType__v3	Wheeled				Tracked			
VehicleType__v2	Wheeled		Tracked		Wheeled		Tracked	
VehicleType__v1	Wheeled	Tracked	Wheeled	Tracked	Wheeled	Tracked	Wheeled	Tracked
High	0	0	0	0.5	0	0.5	0.5	0.66
Low	1	1	1	0.5	1	0.5	0.5	0.34

Given entities, V1, V2, V3, and Region1.1, the above situation-specific Bayesian Networks (SSBN) is derived from the Danger MFrag with the conditional probability table (CPT)

2. Problem Statement

- Old approach
  - Manual MEBN modeling
- Problem of Manual MEBN modeling
  - labor-intensive
  - insufficiently agile process

3. Basic MEBN Learning

- MEBN-RM(Relational Model) Model
- Basic MEBN Parameter Learning
- Basic MEBN Structure Learning

3. Basic MEBN Learning

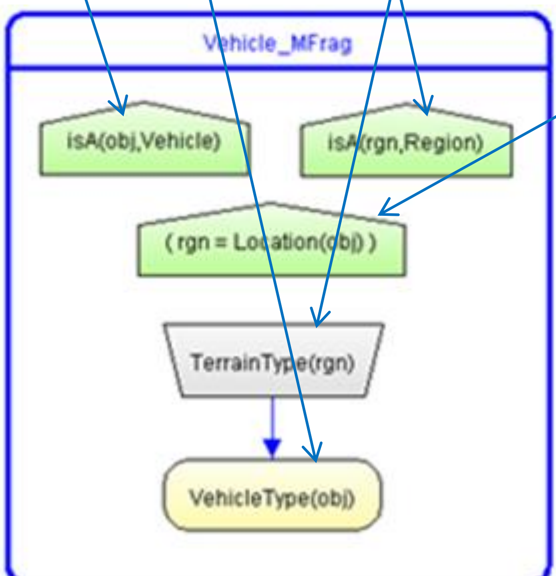
MEBN-RM Model

Vehicle	
obj	VehicleType
v1	Wheeled
v2	Tracked
v3	Tracked
v4	Tracked
v5	Wheeled
v6	Tracked

Region		
rgn	TerrainType	UpperRegion
r1	OffRoad	null
r1_1	Road	r1
r1_2	OffRoad	r1
r2	OffRoad	null
r2_1	OffRoad	r2
r2_1_1	Road	r2_1

Report		
rpt	ImageTypeReort	ReportedObject
rpt1	Wheeled	v1
rpt2	Wheeled	v1
rpt3	Tracked	v1
rpt4	Tracked	v2
rpt5	Wheeled	v2
rpt6	Tracked	v2

Location		
obj	t	rgn
v1	t1	r1
v1	t2	r1
v1	t3	r1
v2	t1	r2_1
v2	t2	r2_1
v2	t3	r2_1



Type	Name	Example
1	Isa	Isa(obj, VehicleObject), Isa(rgn, Region), Isa(t, Time), Isa(rpt, Report)
2	Value-Constraint	VehicleType(obj) = Wheeled
3	Slot-Filler	obj = Reported Object(rpt)
4	Entity-Constraint	Communication(obj1,obj2)

Table 1. Context Node Types on MEBN-RM Model

RM	Resident Node
Attribute	Function/ Predicate
Key	Arguments
Cell of Attribute	Output

Table 2. Function of MEBN-RM Model

Basic MEBN Parameter Learning

$$\hat{\theta} = arg\,max_{\theta \in \Theta}\, p(\,\theta \mid D, M\,)$$

Optimal parameter

MTheory

Relational Dataset

A set of parameters in
Local Probability Distribution

To estimate the parameter, Maximum Likelihood Estimation (MLE) is used

Basic MEBN Structure Learning

$$\hat{M} = arg\ max_{M \in \mathcal{M}}\ p(\ M \mid D\ )$$

Optimal MTheory

Relational Dataset

A set of possible MTheories

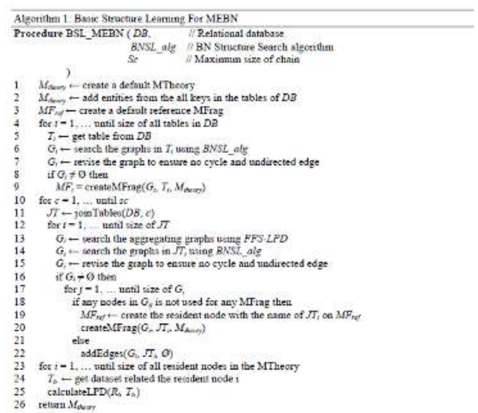
To search a local BN, GES(Greedy equivalence search) algorithm is used

Basic MEBN Structure Learning Algorithm

RM
Dataset

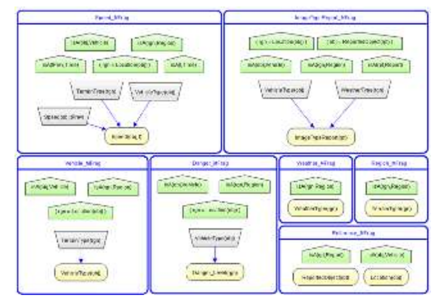
Any Bayesian Networks Structure Algorithm

Basic MEBN Structure Learning Algorithm



```
Algorithm 1: Basic Structure Learning For MEBN
Procedure BSL_MEBN ( DB,          // Relational database
                     BNSL_alg     // BN Structure Search algorithm
                     Sc           // Maximum size of chain
           )
M_theory ← create a default MTheory
M_theory ← add entities from the all keys in the tables of DB
MF_ref ← create a default reference MFrag
for i = 1, … until size of all tables in DB
  T_i ← get table from DB
  G_i ← search the graphs in T_i using BNSL_alg
  G_i ← revise the graph to ensure no cycle and undirected edge
  if G_i ≠ Ø then
    MF_i = createMFrag(G_i, T_i, M_theory)
for c = 1, … until sc
  JT ← joinTables(DB, c)
  for i = 1, … until size of JT
    G_i ← search the aggregating graphs using FFS-LPD
    G_i ← search the graphs in JT_i using BNSL_alg
    G_i ← revise the graph to ensure no cycle and undirected edge
    if G_i ≠ Ø then
      for j = 1, … until size of G_i
        if any nodes in G_ij is not used for any MFrag then
          MF_ref ← create the resident node with the name of JT_i on MF_ref
          createMFrag(G_ij, JT_i, M_theory)
        else
          addEdges(G_ij, JT_i, Ø)
for i = 1, … until size of all resident nodes in the MTheory
  T_ri ← get dataset related the resident node i
  calculateLPD(R_i, T_ri)
return M_theory
```

MTheory

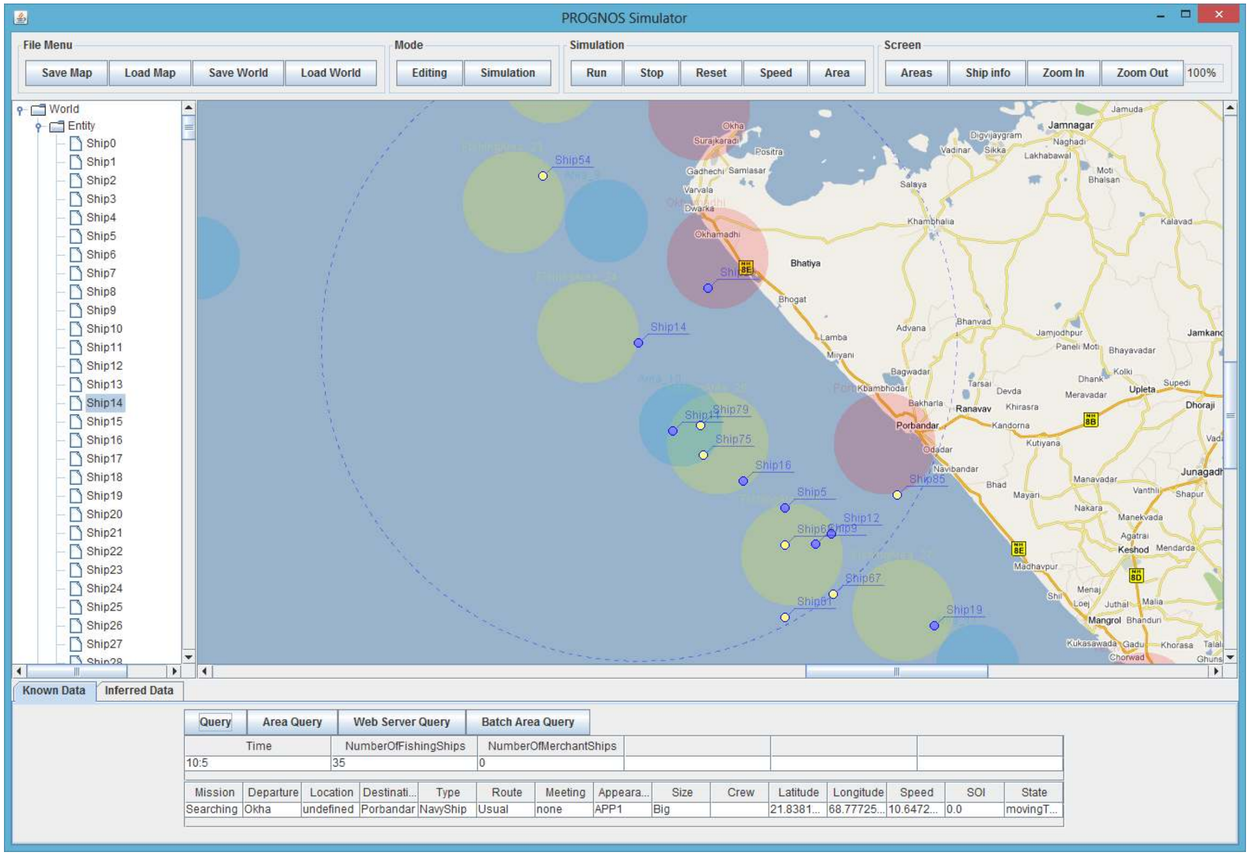


The initial ingredients of the algorithm are the dataset expressed in RM and any Bayesian Network Structure searching algorithm

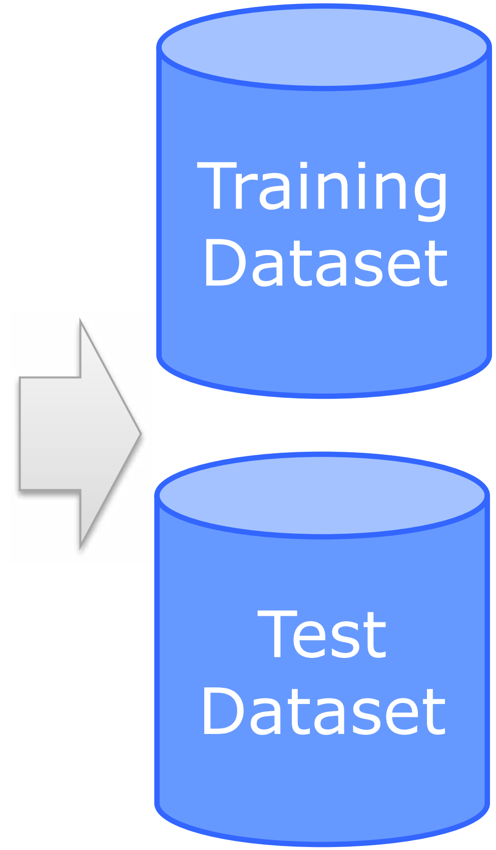
4. Case Study

- Generating Training and Test data
- Evaluating MTheory
- Learned MTheory
- Accuracy of P(SOI(Ship Of Interest) | Evidences)

Generating Training and Test data

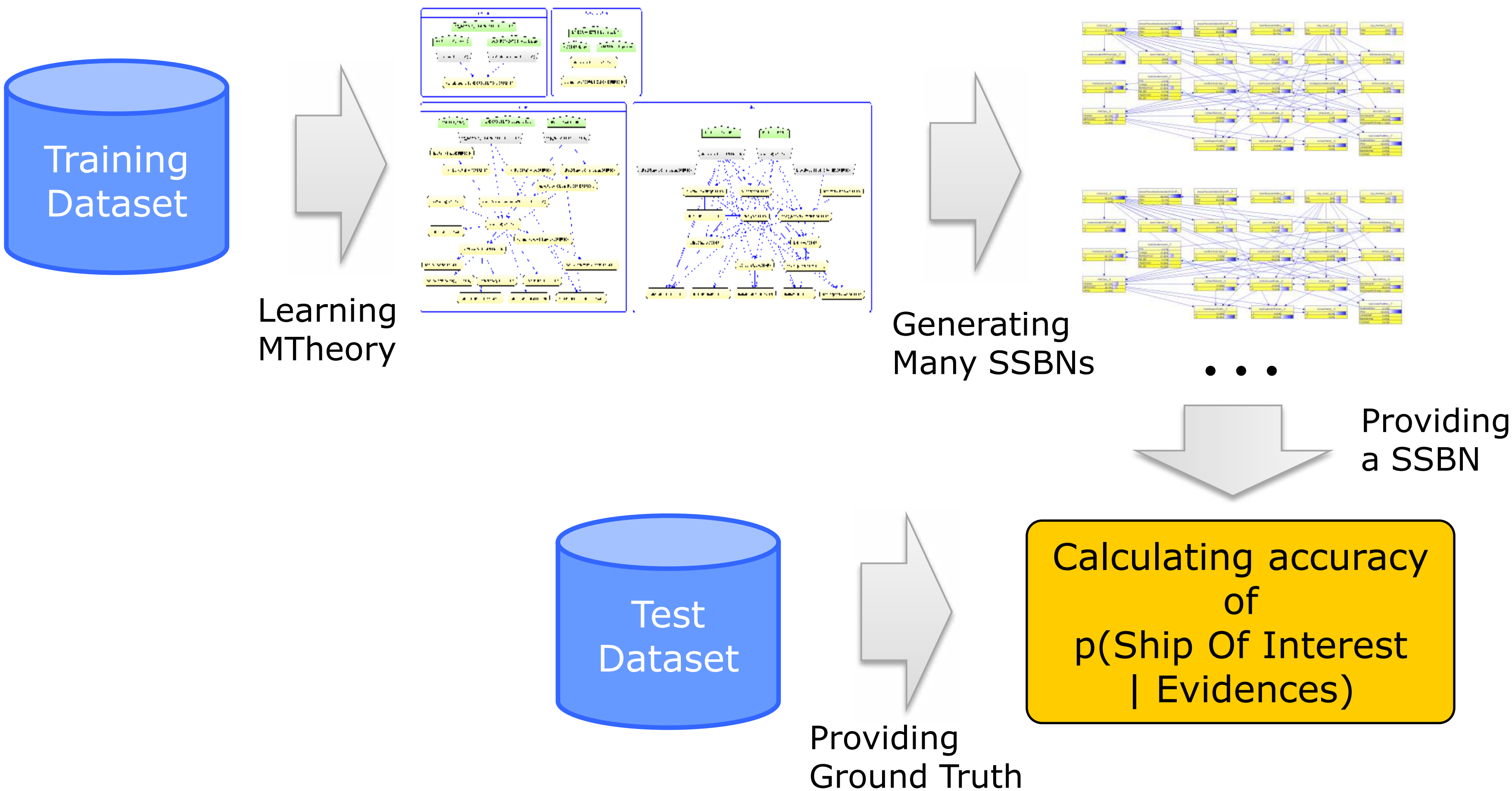


PROGNOS Simulation Module



4. Case Study

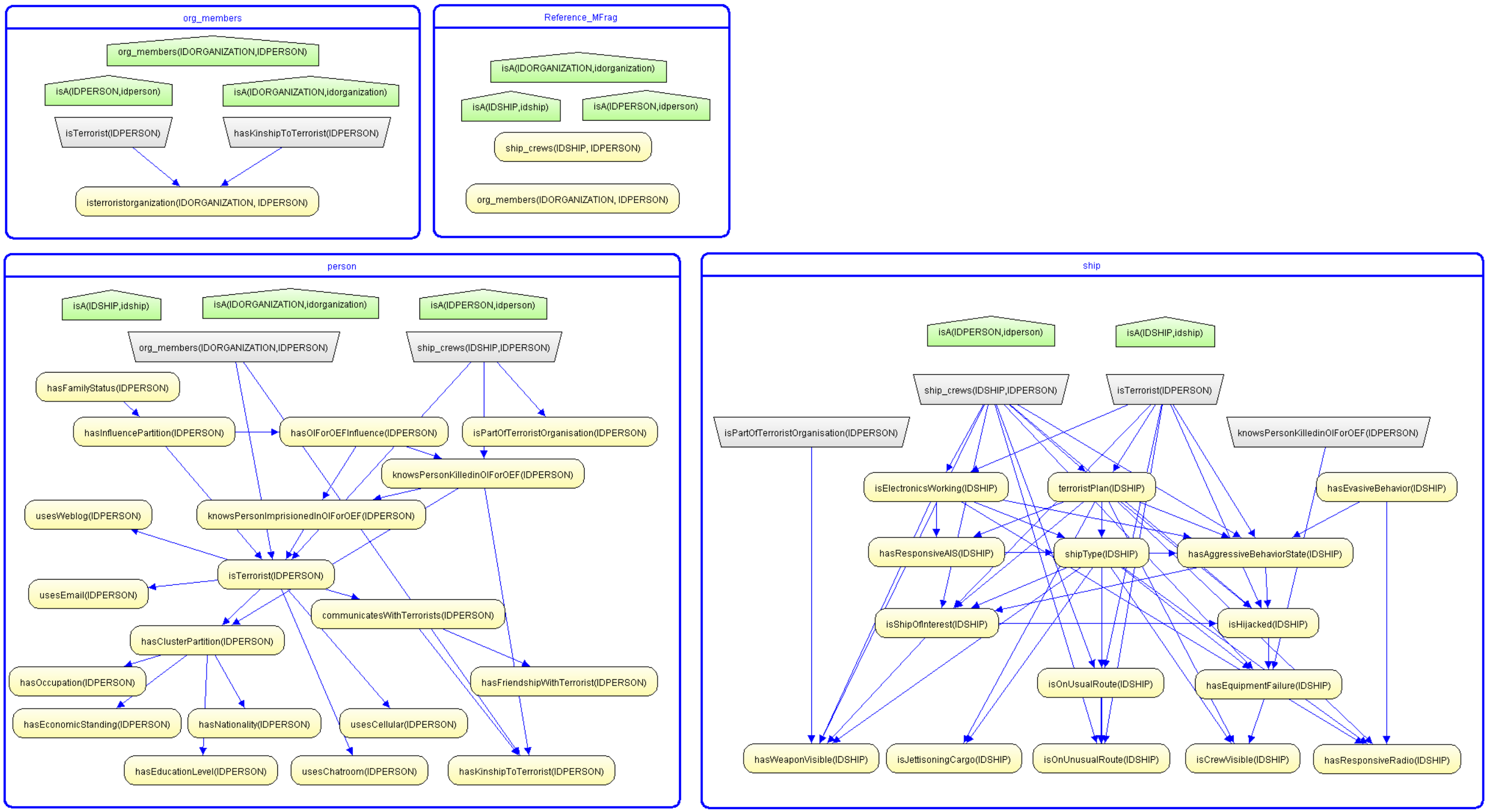
Evaluating MTheory



By calculating accuracy of p(SOI | Evidences), we evaluate the parameter of the learned MTheory, but we didn’t evaluate the structure of the MTheory

4. Case Study

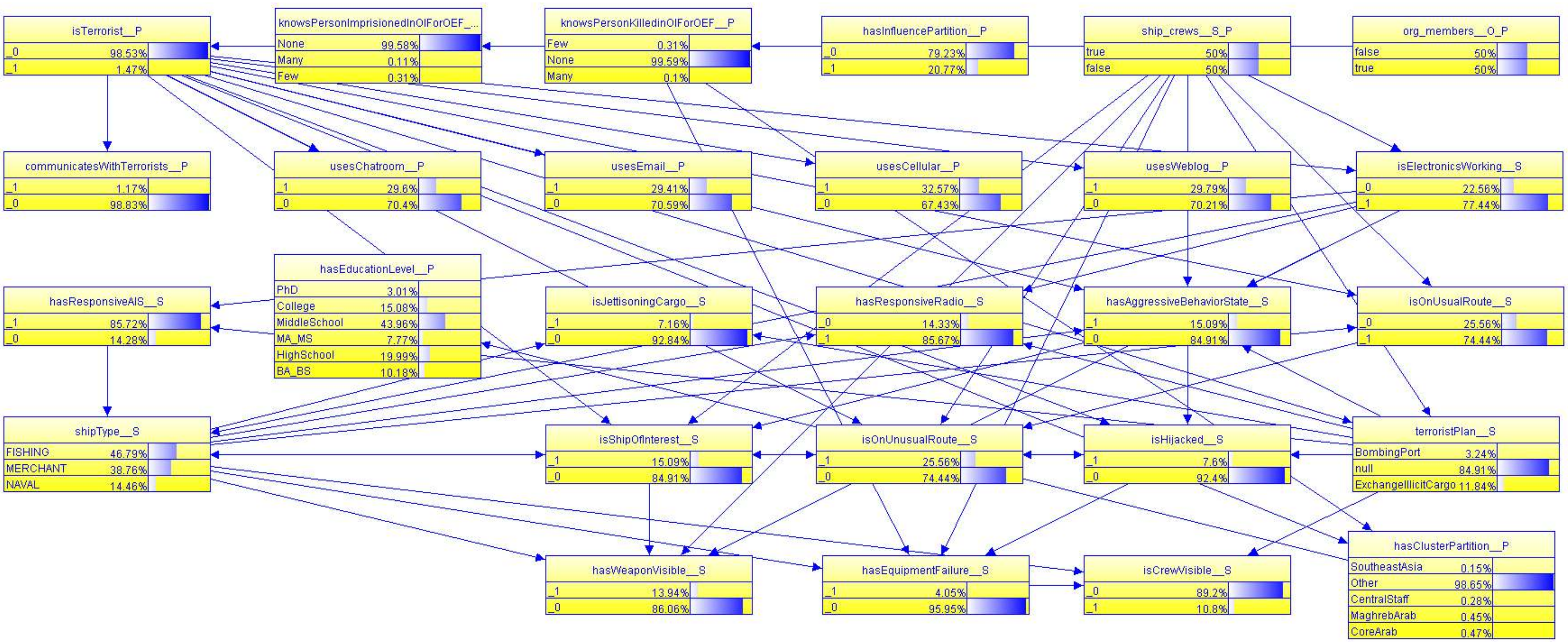
Learned PROGNOS MTheory



The default reference , org_members, person, and ship MFrag are learned from the training data set

4. Case Study

Generated SSBN from Learned PROGNOS MTheory



_1 and _0 in the state of the node means true and false respectively

The letter S, O, and P in the title of the node means Ship, Organization, and Person respectively

Accuracy of P(SOI | Evidences)

Model	AUC
Learned MTheory	0.897206546

Table 3. AUC of Learned MTheory

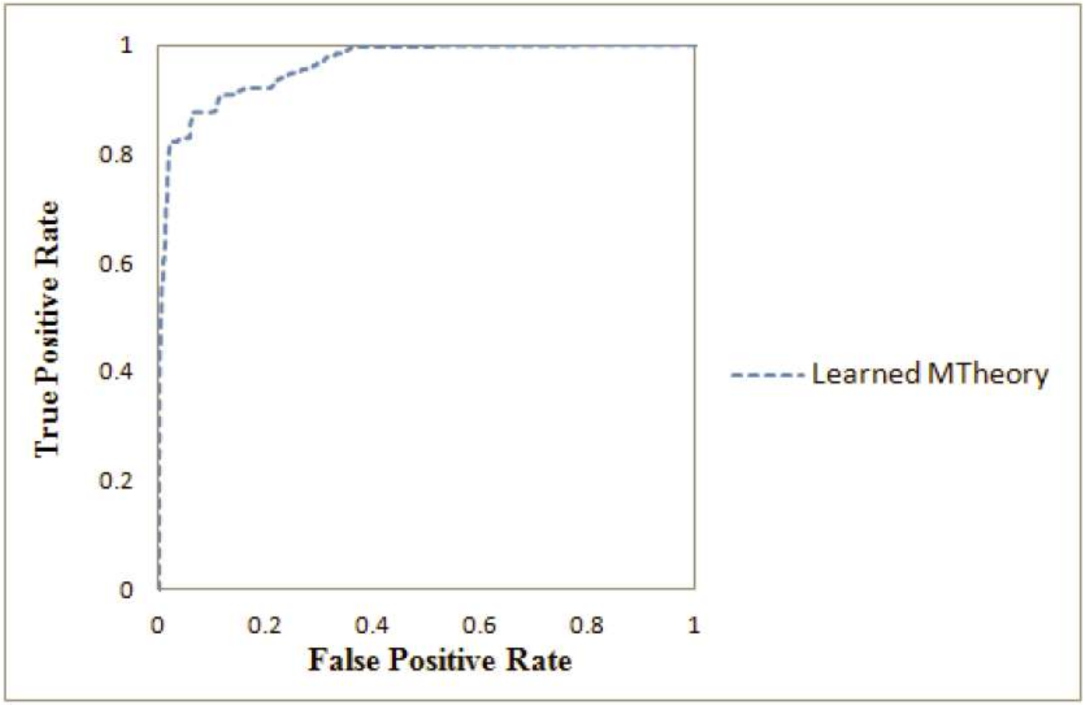


Figure 10. ROC of Learned MTheory

The learned MTheory estimated p(Ship Of Interest | Evidences) with the area under the curve (AUC), 0.897206546

5. Conclusion

- Basic MEBN Learning
  - MEBN-RM Model
  - MEBN Parameter Learning
  - MEBN Structure Learning

- Current Work
  - Hybrid random variable learning in PSAW

Thank you for viewing our presentation!

Back up 1

There remain many open research issues in this domain

1) Aggregating influence problem; how to learn an aggregating function in an aggregating situation where an instance child random variable depends on multiple instance parents which is generated from an identical class random variable?
2) Optimization of learned MTheory; how to learn an optimized structure of an MTheory without losing accuracy of query?
3) Unstructured data learning; how to learn unstructured data which isn't derived from a data model?
4) Continuous random variable learning; how to learn an MTheory which includes continuous random variables?
5) Multiple distributed data learning; how to learn an MTheory from data in multiple distributed databases?
6) Incomplete data learning; how to approximate parameters of an MTheory from missing data?
7) Learning in insufficient evidence; how to learn an MTheory from not enough observations?
8) Incremental MEBN learning; how to learn parameters of an MTheory from updated observations?

We have studied about the aggregating influence problem and continuous random variable learning in PSAW

Back up 2

- The data for learning are stored in a relational database
  - There is a single centralized database rather than multiple distributed databases
  - We do not consider learning from unstructured data
- The database contains enough observations for accurate learning
- There is no missing data
- All RVs are discrete
  - Continuous RVs are not considered
- Learning is in batch mode
  - We do not consider online incremental learning
- We do not consider the problem of aggregating influences from multiple instances of the parents of an RV

4. Background

Relational Model Example

Attributes : Key, TerrainType, UpperRegion

Primary Key: VehicleKey, TimeKey

Foreign Key: Region

Tuple: First row

VehicleObject	
Key	VehicleType
v1	Wheeled
v2	Tracked
v3	Tracked
v4	Tracked
v5	Wheeled
v6	Tracked

Region		
Key	TerrainType	UpperRegion
r1	OffRoad	null
r1_1	Road	r1
r1_2	OffRoad	r1
r2	OffRoad	null
r2_1	OffRoad	r2
r2_1_1	Road	r2_1

Time	
Key	PreviousTime
t1	null
t2	t1
t3	t2
t4	t3
t5	t4
t6	t5

Location		
VehicleKey	TimeKey	Region
v1	1	r1
v1	2	r1
v1	3	r1
v2	1	r2_1
v2	2	r2_1
v2	3	r2_1

Cardinality: 6

Domain: Wheeled, Tracked

Degree: 3

Relation: VehicleObject, Region, Time, Location

In 1969, Edgar F. Codd proposed the Relational Model (RM) as a database model based on first-order predicate logic [Codd, 1969; Codd, 1970]

Example of MEBN Structure Learning

Vehicle	
obj	VehicleType
v1	Wheeled
v2	Tracked
v3	Tracked
v4	Tracked
v5	Wheeled
v6	Tracked

Region		
rgn	TerrainType	UpperRegion
r1	OffRoad	null
r1_1	Road	r1
r1_2	OffRoad	r1
r2	OffRoad	null
r2_1	OffRoad	r2
r2_1_1	Road	r2_1

Report		
rpt	ImageTypeReort	ReportedObject
rpt1	Wheeled	v1
rpt2	Wheeled	v1
rpt3	Tracked	v1
rpt4	Tracked	v2
rpt5	Wheeled	v2
rpt6	Tracked	v2

Location		
obj	t	rgn
v1	t1	r1
v1	t2	r1
v1	t3	r1
v2	t1	r2_1
v2	t2	r2_1
v2	t3	r2_1

Entity Table

Relationship Table

4. Basic MEBN Learning

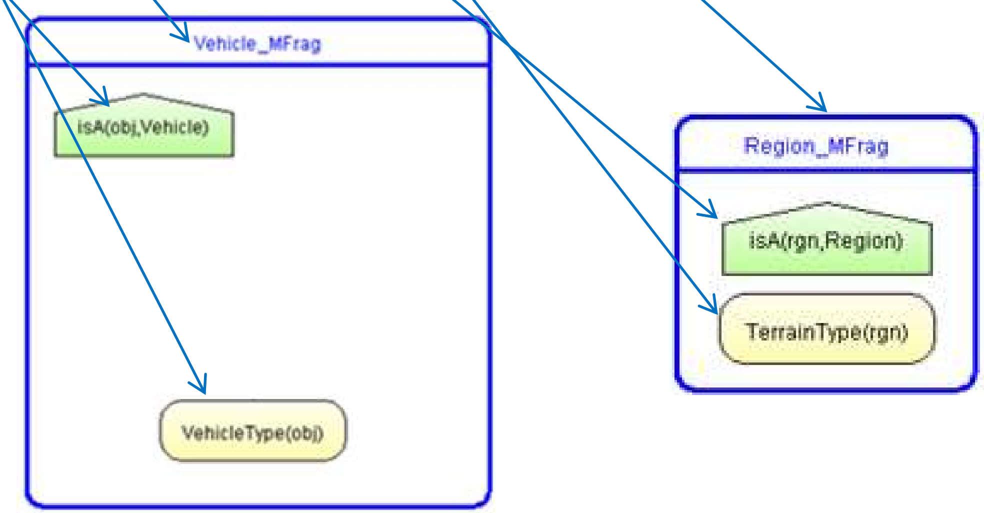
Example of MEBN Structure Learning

Vehicle	
obj	VehicleType
v1	Wheeled
v2	Tracked
v3	Tracked
v4	Tracked
v5	Wheeled
v6	Tracked

Region		
rgn	TerrainType	UpperRegion
r1	OffRoad	null
r1_1	Road	r1
r1_2	OffRoad	r1
r2	OffRoad	null
r2_1	OffRoad	r2
r2_1_1	Road	r2_1

Report		
rpt	ImageTypeReort	ReportedObject
rpt1	Wheeled	v1
rpt2	Wheeled	v1
rpt3	Tracked	v1
rpt4	Tracked	v2
rpt5	Wheeled	v2
rpt6	Tracked	v2

Location		
obj	t	rgn
v1	t1	r1
v1	t2	r1
v1	t3	r1
v2	t1	r2_1
v2	t2	r2_1
v2	t3	r2_1



1. For every entity Table, generate MFrags
2. Graph is derived by the BN structure learning Algorithm

Example of MEBN Structure Learning

Vehicle	
obj	VehicleType
v1	Wheeled
v2	Tracked
v3	Tracked
v4	Tracked
v5	Wheeled
v6	Tracked

Region		
rgn	TerrainType	UpperRegion
r1	OffRoad	null
r1_1	Road	r1
r1_2	OffRoad	r1
r2	OffRoad	null
r2_1	OffRoad	r2
r2_1_1	Road	r2_1

Report		
rpt	ImageTypeReort	ReportedObject
rpt1	Wheeled	v1
rpt2	Wheeled	v1
rpt3	Tracked	v1
rpt4	Tracked	v2
rpt5	Wheeled	v2
rpt6	Tracked	v2

Location		
obj	t	rgn
v1	t1	r1
v1	t2	r1
v1	t3	r1
v2	t1	r2_1
v2	t2	r2_1
v2	t3	r2_1

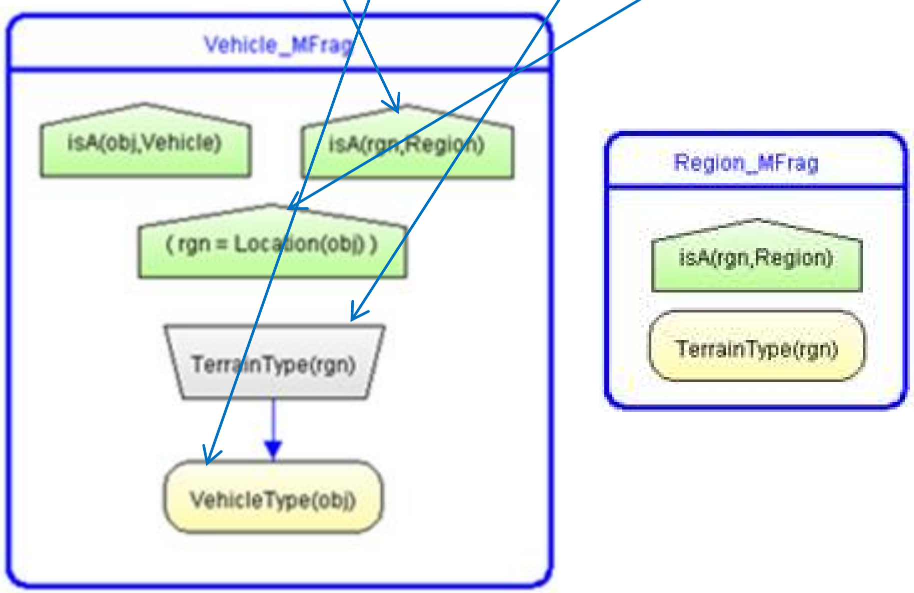
Vehicle - Location - Region			
obj	rgn	VehicleType	TerrainType
v1	r1	Wheeled	OffRoad
v2	r2_1	Tracked	OffRoad
…	…	…	…

3. For every relationship table, get Joined Table

Example of MEBN Structure Learning

Vehicle - Location - Region			
obj	rgn	VehicleType	TerrainType
v1	r1	Wheeled	OffRoad
v2	r2_1	Tracked	OffRoad
…	…	…	…

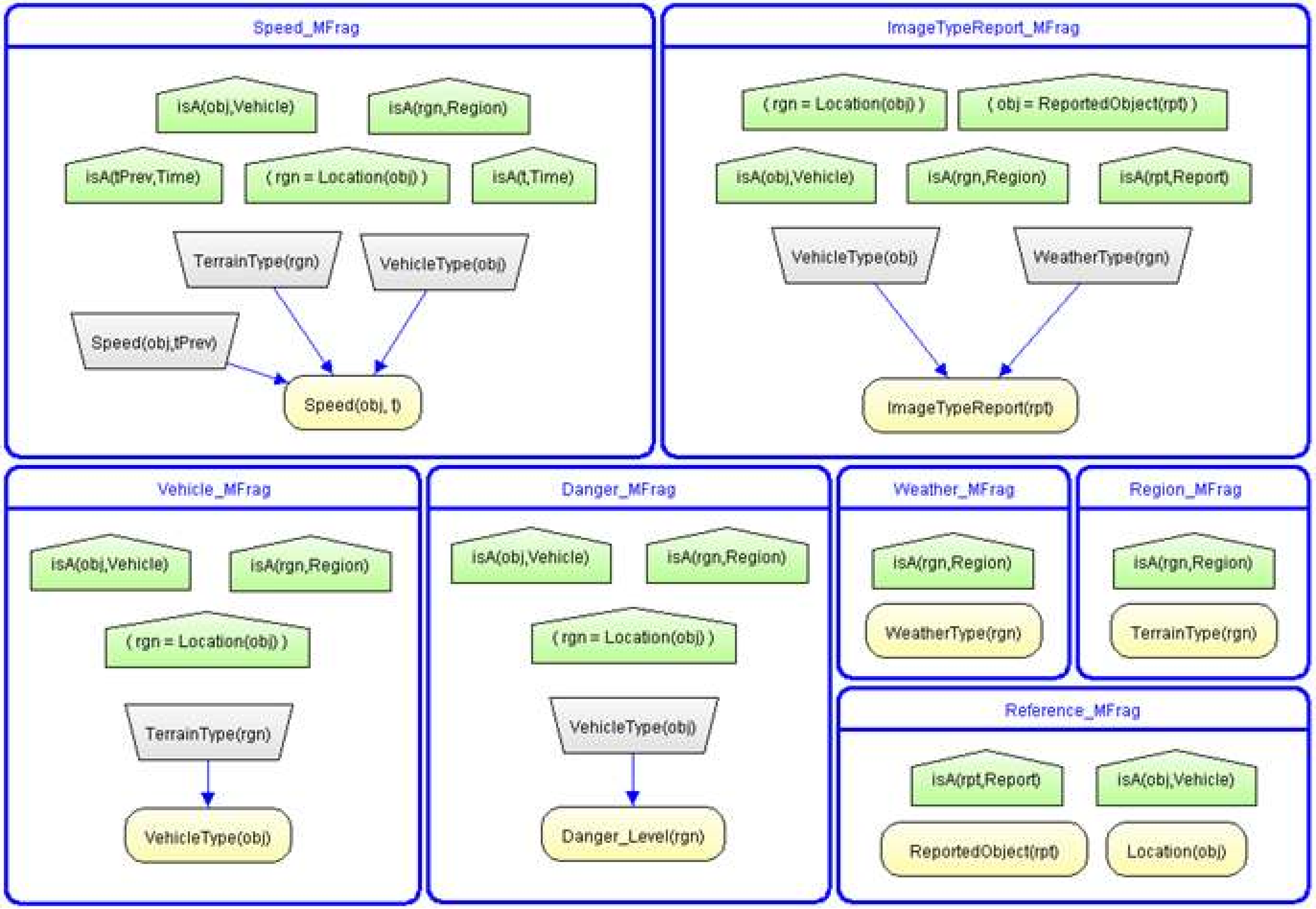
Location		
obj	t	rgn
v1	t1	r1
v1	t2	r1
v1	t3	r1
v2	t1	r2_1
v2	t2	r2_1
v2	t3	r2_1



4. Link between Joined entities
5. Add context nodes

4. Basic MEBN Learning

Example of MEBN Structure Learning



By iteration of the above process, the above MTheory can is learned.

4. Basic MEBN Learning

Basic MEBN Structure Learning

Algorithm 1: Basic Structure Learning For MEBN

```
Procedure BSL_MEBN ( DB,          // Relational database
                     BNSL_alg     // BN Structure Search algorithm
                     Sc           // Maximum size of chain
            )
  M_theory ← create a default MTheory
  M_theory ← add entities from the all keys in the tables of DB
  MF_ref ← create a default reference MFrag
  for i = 1, … until size of all tables in DB
    T_i ← get table from DB
    G_i ← search the graphs in T_i using BNSL_alg
    G_i ← revise the graph to ensure no cycle and undirected edge
    if G_i ≠ Ø then
      MF_i = createMFrag(G_i, T_i, M_theory)
  for c = 1, … until sc
    JT ← joinTables(DB, c)
    for i = 1, … until size of JT
      G_i ← search the aggregating graphs using FFS-LPD
      G_i ← search the graphs in JT_i using BNSL_alg
      G_i ← revise the graph to ensure no cycle and undirected edge
      if G_i ≠ Ø then
        for j = 1, … until size of G_i
          if any nodes in G_ij is not used for any MFrag then
            MF_ref ← create the resident node with the name of JT_i on MF_ref
            createMFrag(G_i, JT_i, M_theory)
          else
            addEdges(G_i, JT_i, Ø)
  for i = 1, … until size of all resident nodes in the MTheory
    T_ii ← get dataset related the resident node i
    calculateLPD(R_i, T_ii)
  return M_theory
```

…

Structure learning is to organize RVs into MFrags and identify parent-child relationships between nodes, given a dataset expressed in RM