Apprentissage de la structure des réseaux bayésiens : état de l'art et intégration de connaissances sémantiques

Montassar Ben Messaoud[12], Mouna Ben Ishak[12],
Philippe Leray[2], Nahla Ben Amor[1]

[1]LARODEC, ISG, University of Tunis, Tunisia
[2]Knowledge and Decision Team, LINA UMR 6241, Nantes, France

Journées MSTGA
7 nov. 2011, Toulouse, France

Outline

Bayesian network definition

Definition **(Pearl 1985)**

- A Bayesian network (BN) is defined by
  - one qualitative description of (conditional) dependences / independences between variables
    directed acyclic graph (DAG)
  - one quantitative description of these dependences
    conditional probability distributions (CPDs)

Example

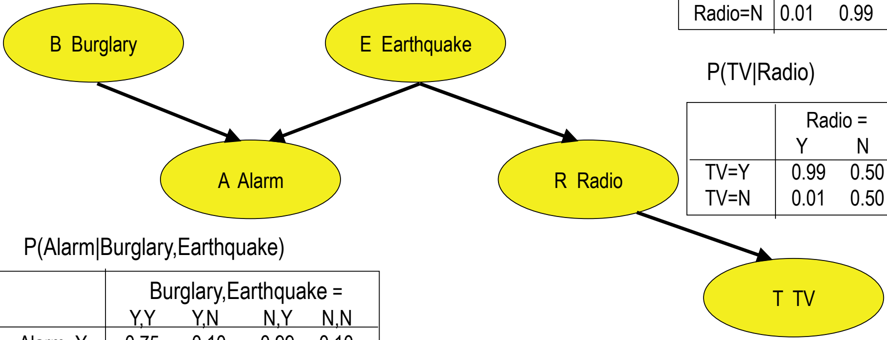
one topological order : B, E, A, R, T (not unique)

P(Radio|Earthquake)

	Earthquake = Y	Earthquake = N
Radio=Y	0.99	0.01
Radio=N	0.01	0.99

P(Burglary)=[0.001 0.999]

P(Earthquake)=[0.0001 0.9999]



P(TV|Radio)

	Radio = Y	Radio = N
TV=Y	0.99	0.50
TV=N	0.01	0.50

P(Alarm|Burglary,Earthquake)

	Burglary,Earthquake = Y,Y	Y,N	N,Y	N,N
Alarm=Y	0.75	0.10	0.99	0.10
Alarm=N	0.25	0.90	0.01	0.90

Consequence

Chain rule

$$P(S) = P(S_1) \times P(S_2|S_1) \times P(S_3|S_1, S_2) \times \cdots \times P(S_n|S_1 \ldots S_{n-1})$$

Consequence with a BN

- $P(S_i|S_1 \ldots S_{i-1}) = P(S_i|parents(S_i))$ so

$$P(S) = \Pi_{i=1}^{n} P(S_i|parents(S_i))$$

- The (global) joint probability distribution is decomposed in a product of (local) conditional distributions
- BN = compact representation of the joint distribution $P(S)$ given some information about dependence relationships between variables

Consequence

Chain rule

$P(S) = P(S_1) \times P(S_2|S_1) \times P(S_3|S_1, S_2) \times \cdots \times P(S_n|S_1 \ldots S_{n-1})$

Consequence with a BN

- $P(S_i|S_1 \ldots S_{i-1}) = P(S_i|parents(S_i))$ so

$$P(S) = \Pi_{i=1}^{n} P(S_i|parents(S_i))$$

- The (global) joint probability distribution is decomposed in a product of (local) conditional distributions
- BN = compact representation of the joint distribution $P(S)$ given some information about dependence relationships between variables

Markov equivalence

Definition

B_1 and B_2 are Markov equivalent iff both describe exactly the same conditional (in)dependence statements.

Graphical properties

- B_1 and B_2 have the same skeleton, V-structures and inferred edges.
- All the equivalent graphs (= equivalence class) can be summarized by one partially directed DAG named CPDAG or Essential Graph

Markov equivalence

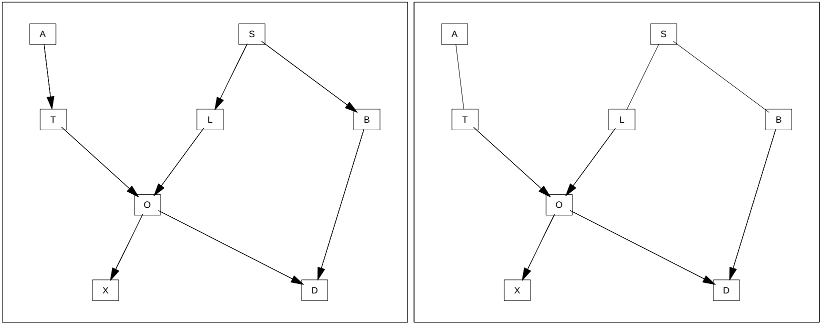
Definition

B_1 and B_2 are Markov equivalent iff both describe exactly the same conditional (in)dependence statements.

Graphical properties

- B_1 and B_2 have the same skeleton, V-structures and inferred edges.
- All the equivalent graphs (= equivalence class) can be summarized by one partially directed DAG named CPDAG or Essential Graph

Markov equivalence



Outline

BN structure learning

How to build / learn a Bayesian Network ?

1. DAG is known, how to determine the CPDs ?
   - from experts : knowledge elicitation
   - from complete data / incomplete data
2. DAG is unknown, how to determine it (or the CPDAG) ?
   - **from complete data** / incomplete data
   - latent variable discovery ?

Structure learning is a complex task

Size of the "solution" space

- the number of possible DAGs with n variables is super-exponential w.r.t n (Robinson 77)

$$NS(n) = \begin{cases} 1 \quad , & n = 0 \text{ or } 1 \\ \sum_{i=1}^{n} (-1)^{i+1} \binom{n}{i} 2^{i(n-1)} NS(n-i), & n > 1 \end{cases}$$

$$NS(5) = 29281 \qquad NS(10) = 4.2 \times 10^{18}$$

- an exhaustive search is impossible !

Structure learning algorithms

How to search a good BN ?

- Constraint-based methods
  BN = independence model
  ⇒ find CI in data in order to build the DAG
- Score-based methods
  BN = probabilistic model that must fit data as well as possible
  ⇒ search the DAG space in order to maximize a scoring function
- Hybrid methods

Constraint-based methods

How to search a good BN ?

- Constraint-based methods
  BN = independence model
  ⇒ find CI in data in order to build the DAG
- Score-based methods
  BN = probabilistic model that must fit data as well as possible
  ⇒ search the DAG space in order to maximize a scoring function
- Hybrid methods

Constraint-based methods

Two reference algorithms

- Pearl et Verma : IC, IC*
- Spirtes, Glymour et Scheines : SGS, PC, CI, FCI

Common principle

- build an undirected graph describing direct dependences between variables (χ^2 tests)
  - by adding edges (Pearl et Verma)
  - by deleting edges (SGS)
- detect V-structures (from previous statistical tests)
- propagate some edge orientation (inferred edges) in order to obtain a CPDAG

Constraint-based methods

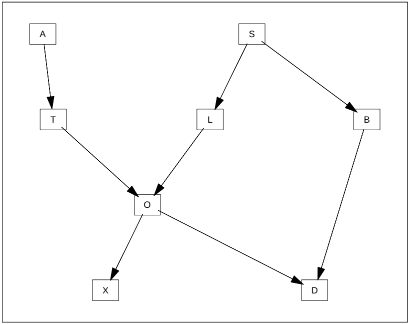
Some inconvenients

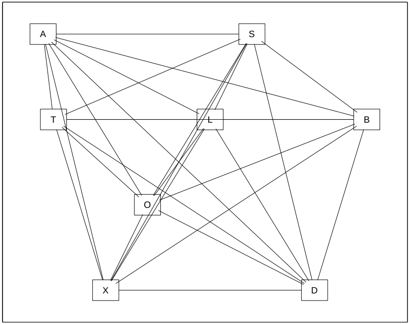
- reliability of CI test conditionally to several variables with a limited amount of data)
  - SGS heuristic : if $df < \frac{N}{10}$, then declare dependence
- Combinatorial explosion of the number of tests
  - PC heuristic : begin with order 0 ($X_A \perp X_B$) then order 1 ($X_A \perp X_B \mid X_C$), etc ...

PC algorithm

Step 0 : undirected complete graph

Left : target BN used to generate 5000 samples.

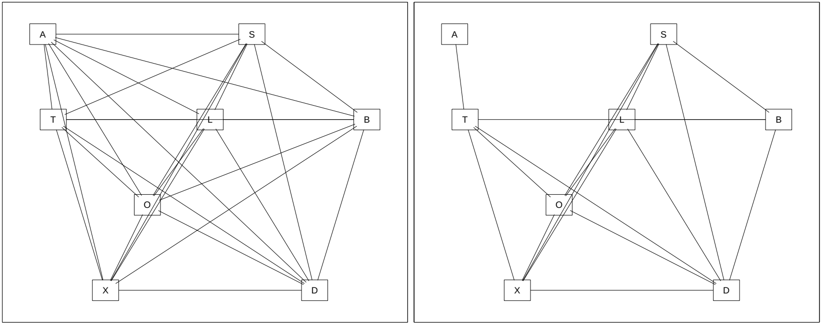




PC algorithm

Step 1a : delete all order 0 independences discovered

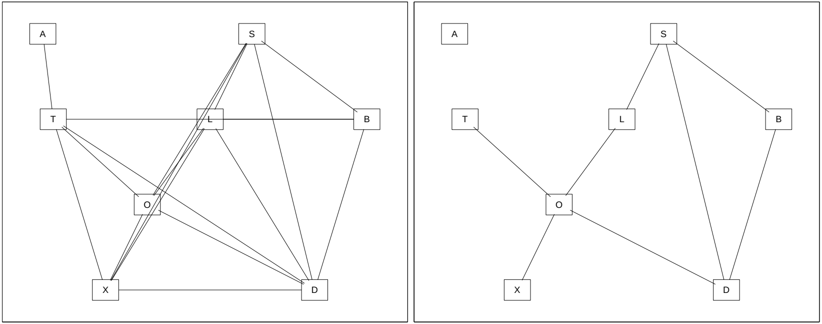
χ^2: $S \perp A$ $L \perp A$ $B \perp A$ $O \perp A$ $X \perp A$ $D \perp A$ $T \perp S$ $L \perp T$ $O \perp B$ $X \perp B$



PC algorithm

Step 1a : delete all order 1 independences discovered

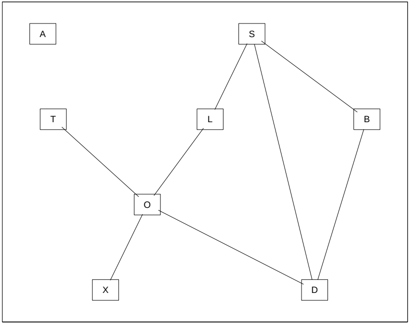
χ^2: $T \perp A | O$ $O \perp S | L$ $X \perp S | L$ $B \perp T | S$ $X \perp T | O$ $D \perp T | O$ ...

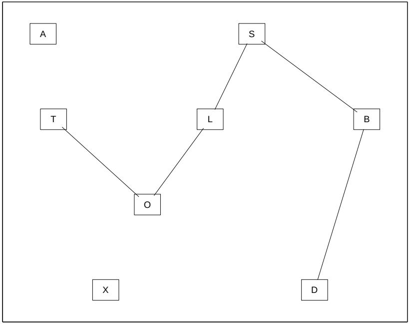


PC algorithm

Step 1a : delete all order 2 independences discovered

χ^2: $D \perp S | \{L, B\}$ $X \perp O | \{T, L\}$ $D \perp O | \{T, L\}$

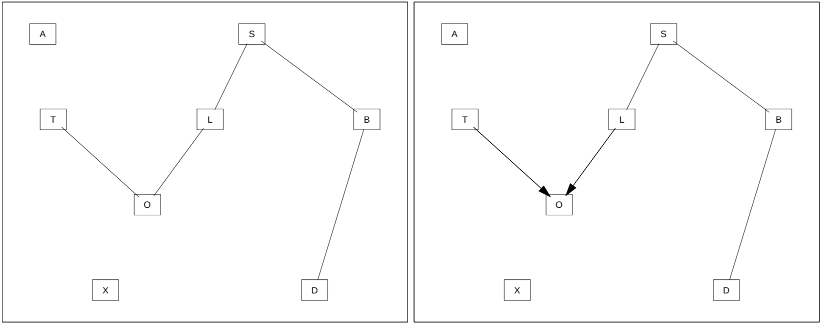




PC algorithm

Step 2 : research V-structures

χ^2 : one V-structure $T \rightarrow O \leftarrow L$ is discovered



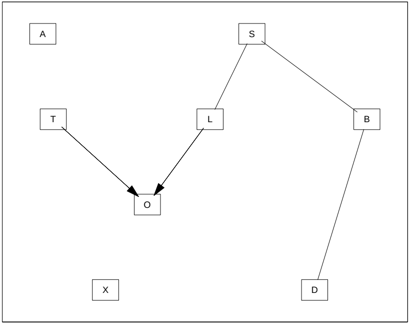
Step 3 : inferred edges

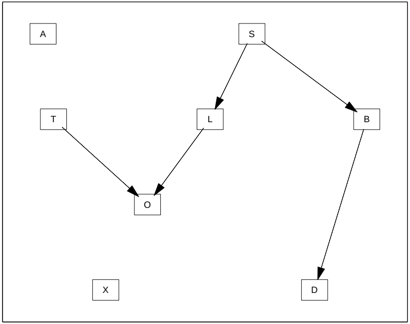
no one in this example

PC algorithm

From CPDAG to DAG

Orientation of the remaining undirected edges
(only constraint : do not create any new V-structure)

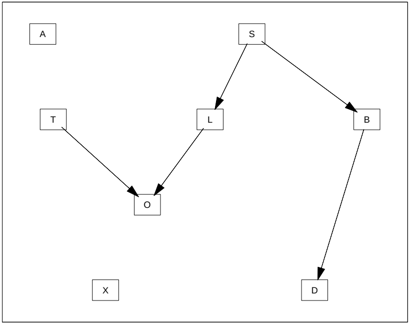


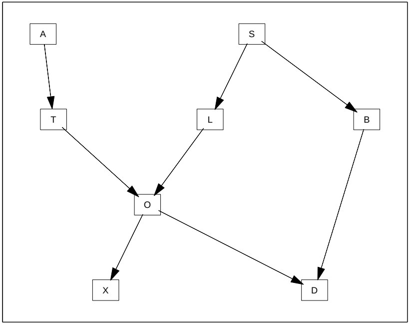


PC algorithm

Obtained DAG versus target one

χ^2 test with 5000 samples fails to discover
$A \rightarrow T$, $O \rightarrow X$ and $O \rightarrow D$





Score-based methods

How to search a good BN ?

- Constraint-based methods
  BN = independence model
  ⇒ find CI in data in order to build the DAG
- Score-based methods
  BN = probabilistic model that must fit data as well as possible
  ⇒ search the DAG space in order to maximize a scoring function
- Hybrid methods

Notion of score

General principle : Occam razor

- *Pluralitas non est ponenda sine neccesitate*
  (La pluralité (des notions) ne devrait pas être posée sans nécessité) plurality should not be posited without necessity
- *Frustra fit per plura quod potest fieri per pauciora*
  (C'est en vain que l'on fait avec plusieurs ce que l'on peut faire avec un petit nombre) It is pointless to do with more what can be done with fewer

= Parcimony principle : find a model

- fitting the data $\mathcal{D}$: likelihood : $\boxed{L(\mathcal{D}|\theta, B)}$
- the simplest possible : dimension of B : $\boxed{Dim(B)}$

Score examples

AIC and BIC

- Compromise between likelihood and complexity
- Application of AIC (Akaïke 70) and BIC (Schwartz 78) criteria

$$S_{AIC}(B,\mathcal{D}) = \log L(\mathcal{D}|\theta^{MV}, B) - Dim(B)$$

$$S_{BIC}(B,\mathcal{D}) = \log L(\mathcal{D}|\theta^{MV}, B) - \frac{1}{2} Dim(B) \log N$$

Bayesian scores : BD, BDe, BDeu

- $S_{BD}(B,\mathcal{D}) = P(B,\mathcal{D})$ (Cooper et Herskovits 92)
- $BDe = BD + score\ equivalence$ (Heckerman 94)

$$S_{BD}(B,\mathcal{D}) = P(B) \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{\Gamma(\alpha_{ij})}{\Gamma(N_{ij}+\alpha_{ij})} \prod_{k=1}^{r_i} \frac{\Gamma(N_{ijk}+\alpha_{ijk})}{\Gamma(\alpha_{ijk})}$$

Score properties

Two important properties

Decomposability

$$(Global)Score(B, \mathcal{D}) = \sum_{i=1}^{n}(local)score(X_i, pa_i)$$

Score equivalence

If two BN B_1 and B_2 are Markov equivalent then $S(B_1, \mathcal{D}) = S(B_2, \mathcal{D})$

Score properties

Two important properties

Decomposability

$$(Global)Score(B, \mathcal{D}) = \sum_{i=1}^{n} (local)score(X_i, pa_i)$$

Score equivalence

If two BN B_1 and B_2 are Markov equivalent then $S(B_1, \mathcal{D}) = S(B_2, \mathcal{D})$

Heuristic exploration of search space

Search space and heuristics

- espace $\mathcal{B}$
  - restriction to tree space : Chow&Liu, MWST
  - DAG with node ordering : K2 algorithm
  - greedy search
  - genetic algorithms, ...
- espace $\mathcal{E}$
  - greedy equivalence search

Restriction to tree space

Principle

- what is the best tree connecting all the nodes, i.e. maximizing a weight defined for each possible edge ?

Answer : maximal weighted spanning tree (MWST)

- (Chow et Liu 68) : weight = mutual information :

$$W(X_A, X_B) = \sum_{a,b} \frac{N_{ab}}{N} \log \frac{N_{ab}N}{N_{a.}N_{.b}}$$

- (Heckerman 94) : any local score :

$$W(X_A, X_B) = score(X_A, Pa(X_A) = X_B) - score(X_A, \emptyset)$$

Restriction to tree space

Principle

- what is the best tree connecting all the nodes, i.e. maximizing a weight defined for each possible edge ?

Answer : maximal weighted spanning tree (MWST)

- (Chow et Liu 68) : weight = mutual information :

$$W(X_A, X_B) = \sum_{a,b} \frac{N_{ab}}{N} \log \frac{N_{ab}N}{N_{a.}N_{.b}}$$

- (Heckerman 94) : any local score :

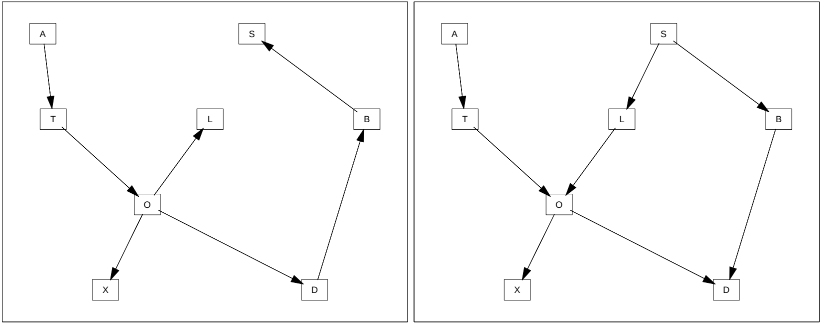
$$W(X_A, X_B) = score(X_A, Pa(X_A) = X_B) - score(X_A, \emptyset)$$

Restriction to tree space

Remarks

- MWST returns an undirected tree
- this undirected tree = CPDAG of all the directed tree with this skeleton
- Obtain a directed tree by (randomly) choosing one root and orienting the edges with a depth first search over this tree

Example : obtained DAG vs. target one



MWST can not discover cycles neither V-structures (tree space !)

Heuristic exploration of search space

Search space and heuristics

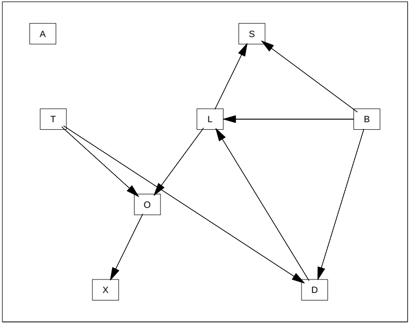
- espace $\mathcal{B}$
  - restriction to tree space : Chow&Liu, MWST
  - DAG with node ordering : K2 algorithm
  - greedy search
  - genetic algorithms, ...
- espace $\mathcal{E}$
  - greedy equivalence search

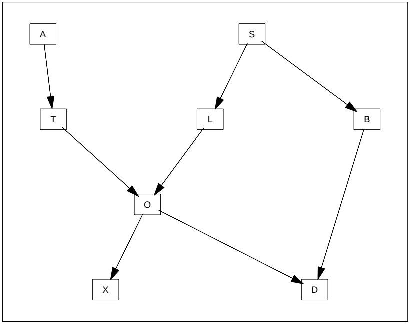
Greedy search

Principle

- Exploration of the search space with traversal operators
  - add edge
  - invert edge
  - delete edge
- and respect the DAG definition (no cycle)
- exploration can begin from any given DAG

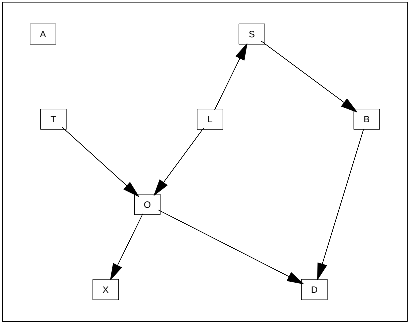
Example : obtained DAG vs. target one

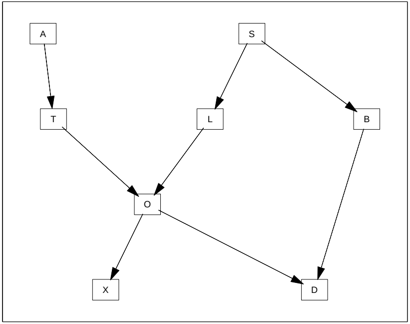




start = empty graph. GS result = local optimum :-(

Example : obtained DAG vs. target one





start = MWST result. GS result is better

Heuristic exploration of search space

Search space and heuristics

- espace $\mathcal{B}$
  - restriction to tree space : Chow&Liu, MWST
  - DAG with node ordering : K2 algorithm
  - greedy search
  - genetic algorithms, ...
- espace $\mathcal{E}$
  - greedy equivalence search

What about changing our search space

Preliminaries

- IC/PC result = CPDAG
- MWST result = CPDAG
- Score-based methods do not distinguish equivalent DAGs

Search in $\mathcal{E}$

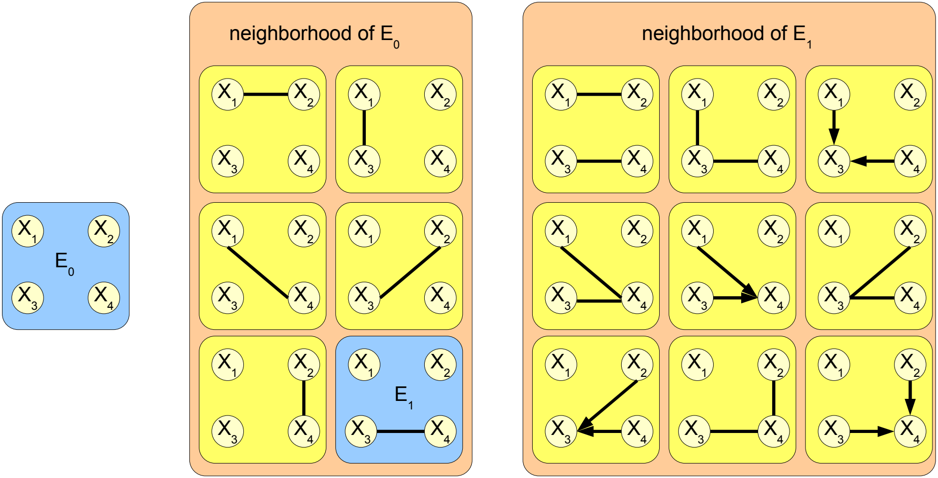
- $\mathcal{E}$ = CPDAG space
- Better properties : YES
  - 2 equivalents structures = 1 unique structure in $\mathcal{E}$
- Better size : NO
  - $\mathcal{E}$ size is quasi similar to DAG space
    (asymptotic ratio is 3,7 : Gillispie et Perlman 2001)

Greedy Equivalent Search

Principe (Chickering 2002)

- Greedy search in $\mathcal{E}$
- Phase 1 : add edges until convergence
- Phase 2 : delete edges until convergence

Add edge examples in $\mathcal{E}$



Score-based methods

How to search a good BN ?

- Constraint-based methods
  BN = independence model
  ⇒ find CI in data in order to build the DAG
- Score-based methods
  BN = probabilistic model that must fit data as well as possible
  ⇒ search the DAG space in order to maximize a scoring/fitness function
- Hybrid methods

Hybrid methods = local search methods

Local search and global learning

- search one local *neighborhood* for a given node T
- reiterate for each T
- learn the global structure with these local informations

which neighborhood ?

- $PC(T)$: Parents and Children T (without distinction)
- $MB(T)$: Markov Blanket of T - Parents, children and spouses

Hybrid methods = local search methods

Local search and global learning

- search one local *neighborhood* for a given node T
- reiterate for each T
- learn the global structure with these local informations

which neighborhood ?

- $PC(T)$: Parents and Children T (without distinction)
- $MB(T)$: Markov Blanket of T - Parents, children and spouses

Local search identification

Identification of MB(T) or PC(T)

- IAMB (Aliferis 2002)
- MMPC (Tsamardinos et al. 2003), ...

Hybrid structure learning algorithms

- MMHC (Tsamardinos et al. 2006) = MMPC + Greedy search

Outline

A BN is not a causal model

- Usual BN
  - $A \rightarrow B$ does not imply direct causal relationship between A and B,
  - only edges from the CPDAG represent causal relationships *

Confusion

- when the DAG is given by an expert, this graph is very often causal
- when the DAG is learnt from data, no reason to be causal !

- But, is it important ?
  - equivalent DAGs ⇒ same joint distribution so same result for (probabilistic) inference

  ⇒ causality is not required for (probabilistic) inference

A BN is not a causal model

- Usual BN
  - $A \rightarrow B$ does not imply direct causal relationship between A and B,
  - only edges from the CPDAG represent causal relationships *

Confusion

- when the DAG is given by an expert, this graph is very often causal
- when the DAG is learnt from data, no reason to be causal !

- But, is it important ?
  - equivalent DAGs ⇒ same joint distribution so same result for (probabilistic) inference
  - ⇒ causality is not required for (probabilistic) inference

A BN is not a causal model

- Usual BN
  - $A \rightarrow B$ does not imply direct causal relationship between A and B,
  - only edges from the CPDAG represent causal relationships *

Confusion

- when the DAG is given by an expert, this graph is very often causal
- when the DAG is learnt from data, no reason to be causal !

- But, is it important ?
  - equivalent DAGs $\Rightarrow$ same joint distribution so same result for (probabilistic) inference

  $\Rightarrow$ causality is not required for (probabilistic) inference

Causal BN

Definition

- each $A \rightarrow B$ represents ont direct causal relationship, i.e. A is the direct cause which generates B

- if causality is not required for (probabilistic) inference, what is the interest of CBNs ?

Causal BN

Definition

- each $A \rightarrow B$ represents ont direct causal relationship, i.e. A is the direct cause which generates B

- if causality is not required for (probabilistic) inference, what is the interest of CBNs ?

Intervention vs. Observation

- Probabilistic inference :
  - we observe $B = b$,
  - we compute $P(A|B = b)$
- Causal inference [Pearl 00]:
  - we manipulate/intervene upon B : $do(B = b)$

example with $A \rightarrow B$

- $P(A|do(B = b)) = P(A)$,
- $P(B|do(A = a)) = P(B|A = a)$

example with $A \leftarrow B$

- $P(A|do(B = b)) = P(A|B = b)$,
- $P(B|do(A = a)) = P(B)$

Manipulation Theorem

- How is modified the joint distribution after one manipulation $do(M = m)$?

intuitive version

- we forget the "official" causes of M (its parents in the DAG)
- we keep the fact that $M = m$ for its raising effects (to M children)

official version [Spirtes et al. 00]

$$P(v|do(m)) = \left(\prod_{V_i \in V \setminus M} P(v_i|Pa(V_i)) \right)_{M=m}$$

Causal structure learning

- usual situation : *observational* data
  - whatever the methods, the right results is the CPDAG
  - partial determination of the causal structure

How to find a full causal graph ?

- use only *experimental* data, and decide at every step the more interesting experiment to realize (*active learning* [Murphy 01], ...)
- use only *observational* data, for a very specific distribution (*LiNGAM* models [Hoyer et al. 08])

Another idea MyCaDo algorithm [Meganck, Leray & Manderick 06]

- use (already existing) observational data to find the CPDAG
- complete the orientation with experimental data

Causal structure learning

- usual situation : *observational* data
  - whatever the methods, the right results is the CPDAG
  - partial determination of the causal structure

How to find a full causal graph ?

- use only *experimental* data, and decide at every step the more interesting experiment to realize (*active learning* [Murphy 01], ...)
- use only *observational* data, for a very specific distribution (*LiNGAM* models [Hoyer et al. 08])

Another idea MyCaDo algorithm [Meganck, Leray & Manderick 06]

- use (already existing) observational data to find the CPDAG
- complete the orientation with experimental data

Causal structure learning

- usual situation : *observational* data
  - whatever the methods, the right results is the CPDAG
  - partial determination of the causal structure

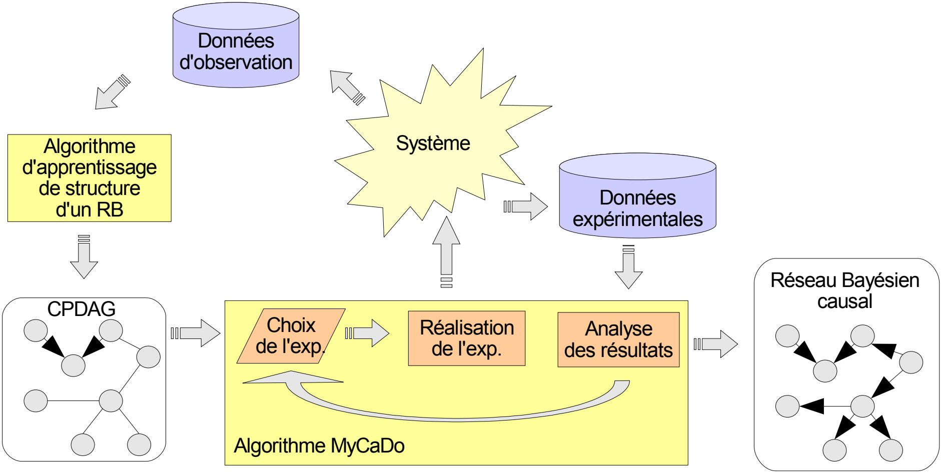
How to find a full causal graph ?

- use only *experimental* data, and decide at every step the more interesting experiment to realize (*active learning* [Murphy 01], ...)
- use only *observational* data, for a very specific distribution (*LiNGAM* models [Hoyer et al. 08])

Another idea MyCaDo algorithm [Meganck, Leray & Manderick 06]

- use (already existing) observational data to find the CPDAG
- complete the orientation with experimental data

MyCaDo algorithm



MyCaDo algorithm

1. Choice of the experiment = what variable M manipulate ?
   - the one potentially orienting more edges
   - by taking into account experiment/observation cost
2. Experimentation
   - $do(M = m)$ for all possible values m
   - Observe all candidate variables C (C–M)
3. Result analysis
   - $P(C|M)$ (observation) $\simeq P(C|do(M))$ (experiment) ?
   - equal : $C \leftarrow M$ else $M \leftarrow C$
   - orient some other edges by applying specific rules (cf. PC algo and Meek rules)

MyCaDo algorithm

1. Choice of the experiment = what variable M manipulate ?
   - the one potentially orienting more edges
   - by taking into account experiment/observation cost

2. Experimentation
   - $do(M = m)$ for all possible values m
   - Observe all candidate variables C (C–M)

3. Result analysis
   - $P(C|M)$ (observation) $\simeq P(C|do(M))$ (experiment) ?
   - equal : $C \leftarrow M$ else $M \leftarrow C$
   - orient some other edges by applying specific rules (cf. PC algo and Meek rules)

MyCaDo algorithm

1. Choice of the experiment = what variable M manipulate ?
   - the one potentially orienting more edges
   - by taking into account experiment/observation cost
2. Experimentation
   - $do(M = m)$ for all possible values m
   - Observe all candidate variables C $(C–M)$
3. Result analysis
   - $P(C|M)$ (observation) $\simeq P(C|do(M))$ (experiment) ?
   - equal : $C \leftarrow M$ else $M \leftarrow C$
   - orient some other edges by applying specific rules (cf. PC algo and Meek rules)

Outline

1. BN definition
2. BN Structure learning
3. BNs and Causality
4. **BNs and Ontologies**
5. SemCaDo algorithm
6. 2OC approach

Motivations

- Knowledge-based systems aim to make expertise available for decision making and information sharing.
- To resolve some complex problems, the combination of different knowledge-based systems can be very powerful.

Our idea

Combine Two KBSs : Bayesian Networks and Ontologies, to help both.

Motivations

- Knowledge-based systems aim to make expertise available for decision making and information sharing.
- To resolve some complex problems, the combination of different knowledge-based systems can be very powerful.

Our idea

Combine Two KBSs : Bayesian Networks and Ontologies, to help both.

Motivations

- Knowledge-based systems aim to make expertise available for decision making and information sharing.
- To resolve some complex problems, the combination of different knowledge-based systems can be very powerful.

Our idea

Combine Two KBSs : Bayesian Networks and Ontologies, to help both.

Motivations

Existing works – Bayesian Network $\Longrightarrow$ Ontology

- **BayesOWL** [Ding & Peng, 2004]
- **OntoBayes** [Yang & Calmet, 2005]
- **PR-OWL** [Costa & Laskey, 2006]

Existing works – Ontology $\Longrightarrow$ Bayesian Network

- BN "basic" construction using ontologies [Devitt & al. 2006]
- Semantical causal discovery **SemCaDo 1.0** [Ben Messaoud & al. 2009]

Our contribution : SemCaDo 2.0

- Ontology $\underset{\text{Causal Discovery}}{\Longrightarrow}$ Causal Bayesian Network $\underset{\text{Ontology Evolution}}{\Longrightarrow}$ Ontology

Motivations

Existing works – Bayesian Network $\Longrightarrow$ Ontology

- **BayesOWL** [Ding & Peng, 2004]
- **OntoBayes** [Yang & Calmet, 2005]
- **PR-OWL** [Costa & Laskey, 2006]

Existing works – Ontology $\Longrightarrow$ Bayesian Network

- BN "basic" construction using ontologies [Devitt & al. 2006]
- Semantical causal discovery **SemCaDo 1.0** [Ben Messaoud & al. 2009]

Our contribution : SemCaDo 2.0

- Ontology $\Longrightarrow$ Causal Bayesian Network $\Longrightarrow$ Ontology

  Causal Discovery   Ontology Evolution

Motivations

Existing works – Bayesian Network $\Longrightarrow$ Ontology

- **BayesOWL** [Ding & Peng, 2004]
- **OntoBayes** [Yang & Calmet, 2005]
- **PR-OWL** [Costa & Laskey, 2006]

Existing works – Ontology $\Longrightarrow$ Bayesian Network

- BN "basic" construction using ontologies [Devitt & al. 2006]
- Semantical causal discovery **SemCaDo 1.0** [Ben Messaoud & al. 2009]

Our contribution : SemCaDo 2.0

- Ontology $\Longrightarrow$ Causal Bayesian Network $\Longrightarrow$ Ontology
  Causal Discovery Ontology Evolution

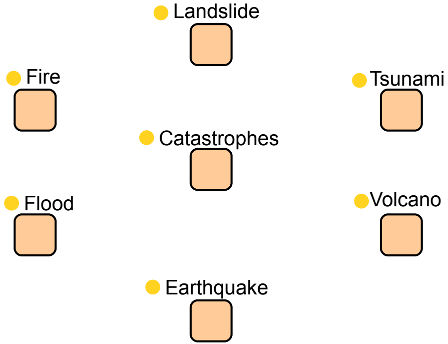
Ontology

Basics

- Definition:
  - Shared understanding within a community of people.
  - Declarative specification of entities and their relationships with each other.
  - Separate the domain knowledge from the operational knowledge.
- Construction: expertise, ontology evolution
- Reasoning: Description logic reasoners

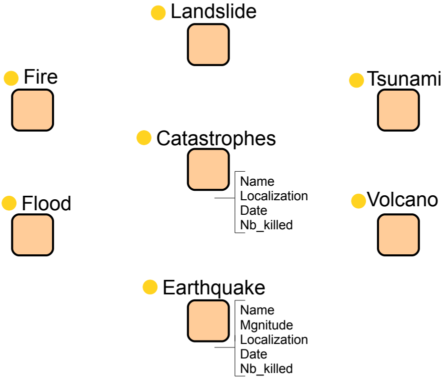
Elements of an ontology

- $\mathcal{C}$: Classes (concepts)
- $\mathcal{P}$: Attributes (properties)
- $\mathcal{H}$: Hierarchical structure (is-a, part-of relations)
- $\mathcal{R}$: Other semantic relationships
- $\mathcal{I}$: Instances (individuals)
- $\mathcal{A}$: Axioms (logic statements)



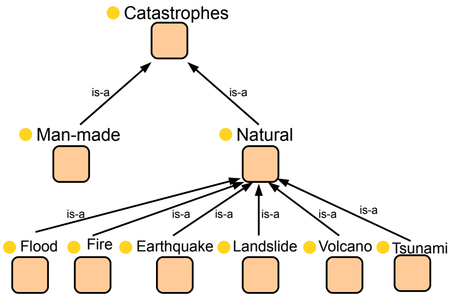
Elements of an ontology

- $\mathcal{C}$: Classes (concepts)
- $\mathcal{P}$: Attributes (properties)
- $\mathcal{H}$: Hierarchical structure (is-a, part-of relations)
- $\mathcal{R}$: Other semantic relationships
- $\mathcal{I}$: Instances (individuals)
- $\mathcal{A}$: Axioms (logic statements)



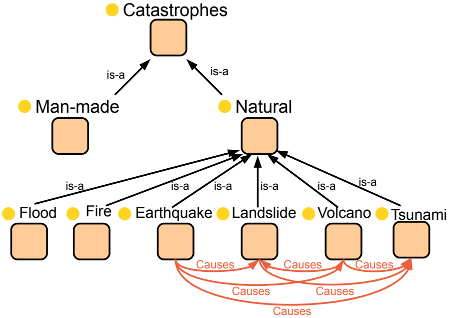
Elements of an ontology

- $\mathcal{C}$: Classes (concepts)
- $\mathcal{P}$: Attributes (properties)
- $\mathcal{H}$: Hierarchical structure (is-a, part-of relations)
- $\mathcal{R}$: Other semantic relationships
- $\mathcal{I}$: Instances (individuals)
- $\mathcal{A}$: Axioms (logic statements)



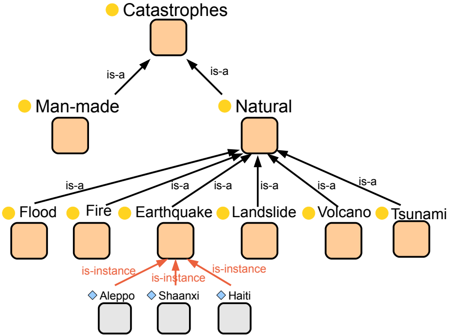
Elements of an ontology

- $\mathcal{C}$: Classes (concepts)
- $\mathcal{P}$: Attributes (properties)
- $\mathcal{H}$: Hierarchical structure (is-a, part-of relations)
- $\mathcal{R}$: Other semantic relationships
- $\mathcal{I}$: Instances (individuals)
- $\mathcal{A}$: Axioms (logic statements)



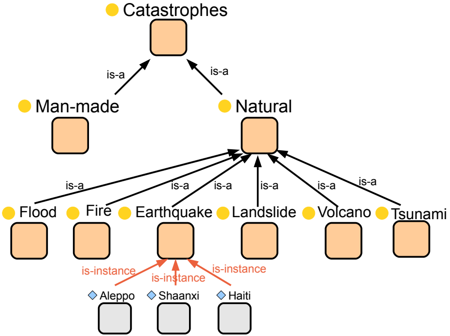
Elements of an ontology

- $\mathcal{C}$: Classes (concepts)
- $\mathcal{P}$: Attributes (properties)
- $\mathcal{H}$: Hierarchical structure (is-a, part-of relations)
- $\mathcal{R}$: Other semantic relationships
- $\mathcal{I}$: Instances (individuals)
- $\mathcal{A}$: Axioms (logic statements)



Elements of an ontology

- $\mathcal{C}$: Classes (concepts)
- $\mathcal{P}$: Attributes (properties)
- $\mathcal{H}$: Hierarchical structure (is-a, part-of relations)
- $\mathcal{R}$: Other semantic relationships
- $\mathcal{I}$: Instances (individuals)
- $\mathcal{A}$: Axioms (logic statements)



Ontology evolution

Ontology population

Get new instances of concept(s) already present in the ontology.

Ontology enrichment

Update (add or modify) concepts, properties and relations in a given ontology.

Evolution vs. revolution

- Evolution (ontology continuity): Add new knowledge.
- Revolution (ontology discontinuity): Modify existing knowledge.

Ontology evolution

Ontology population

Get new instances of concept(s) already present in the ontology.

Ontology enrichment

Update (add or modify) concepts, properties and relations in a given ontology.

Evolution vs. revolution

- Evolution (ontology continuity): Add new knowledge.
- Revolution (ontology discontinuity): Modify existing knowledge.

Ontology evolution

Ontology population

Get new instances of concept(s) already present in the ontology.

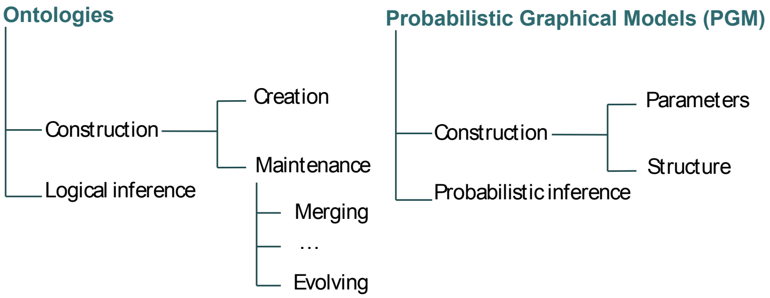
Ontology enrichment

Update (add or modify) concepts, properties and relations in a given ontology.

Evolution vs. revolution

- Evolution (ontology continuity): Add new knowledge.
- Revolution (ontology discontinuity): Modify existing knowledge.

Ontologies vs. BNs



Outline

SemCaDo assumptions

BN ⇔ Ontology general assumptions

- Nodes ⟺ Concepts
- Random variables ⟺ Concept attributes
- Causal dependencies ⟺ Semantic causal relations
- Data ⟺ Concept-attribute instances

SemCaDo specific assumptions

- Causal relations concern concepts sharing the same semantic type.
- Ontology continuity (evolution).

SemCaDo assumptions

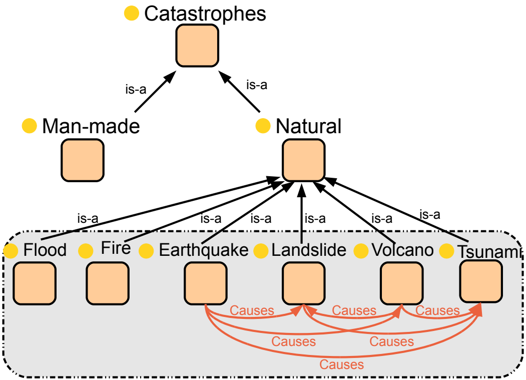
BN $\Leftrightarrow$ Ontology general assumptions

- Nodes $\Longleftrightarrow$ Concepts
- Random variables $\Longleftrightarrow$ Concept attributes
- Causal dependencies $\Longleftrightarrow$ Semantic causal relations
- Data $\Longleftrightarrow$ Concept-attribute instances

SemCaDo specific assumptions

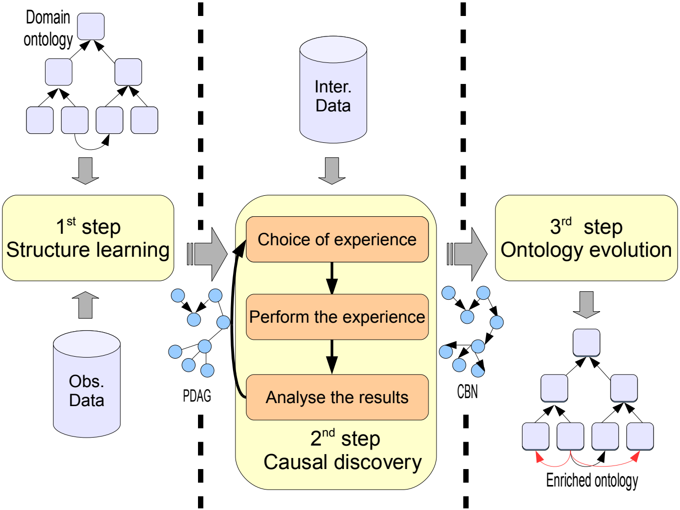
- Causal relations concern concepts sharing the same semantic type.
- Ontology continuity (evolution).

Example



Our causal BN will represent the grey part of the ontology

SemCaDo three main steps



1^{st} step: initial BN structure learning

- Extraction of the causal relationships ($\mathcal{R}_\rfloor$) from the ontology.
- Integration of these edge as constraints in the structure learning algorithm [De Campos & al., 2007].
- Continuity : these edges will not be "questioned" during learning

Interest

Ontology helps in reducing the search task complexity.

1st step: initial BN structure learning

- Extraction of the causal relationships ($\mathcal{R}_{\rfloor}$) from the ontology.
- Integration of these edge as constraints in the structure learning algorithm [De Campos & al., 2007].
- Continuity : these edges will not be "questioned" during learning

Interest

Ontology helps in reducing the search task complexity.

2nd step: serendipitous causal discovery

- Experimentations are needed in order to find the causal orientation of some edges
- Our solution : MyCaDo [Meganck & al., 2006], iterative causal discovery process
- Adaptation to take into account ontological knowledge : Rada distance on $\mathcal{H}$ between one set of concepts and their most specific common subsumer.

Interest

Ontology helps in potentially orienting the more unexpected (serendipitous) links

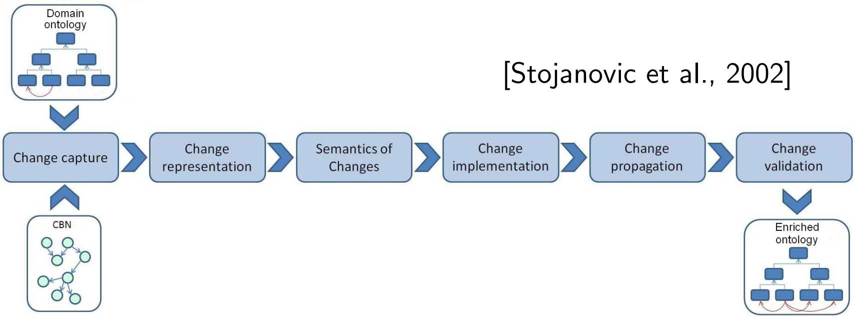
2^{nd} step: serendipitous causal discovery

- Experimentations are needed in order to find the causal orientation of some edges
- Our solution : MyCaDo [Meganck & al., 2006], iterative causal discovery process
- Adaptation to take into account ontological knowledge : Rada distance on $\mathcal{H}$ between one set of concepts and their most specific common subsumer.

Interest

Ontology helps in potentially orienting the more unexpected (serendipitous) links

3rd Step: Ontology evolution process

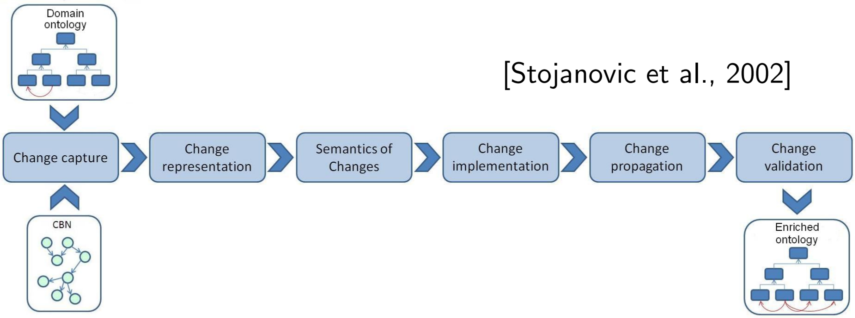


[Stojanovic et al., 2002]

Interest

BN structure learning from data helps in discovering new relations in the ontology

3rd Step: Ontology evolution process



[Stojanovic et al., 2002]

Interest

BN structure learning from data helps in discovering new relations in the ontology

Experimental study (1)

Benchmark : no existing real benchmark & system :-(

- BN graph : random generation (50 to 200 nodes).
- Ontology :
  - Causal relationships : BN edges
  - Hierarchy of concept : generation by clustering BN nodes
- Data is generated by using BN as a generative model.

Experimental protocol

- Hierarchy of concepts and 10% to 40% of existing causal relationships are given as inputs
- Semantic gain : cumulative Rada distance of the discovered relationships

Experimental study (1)

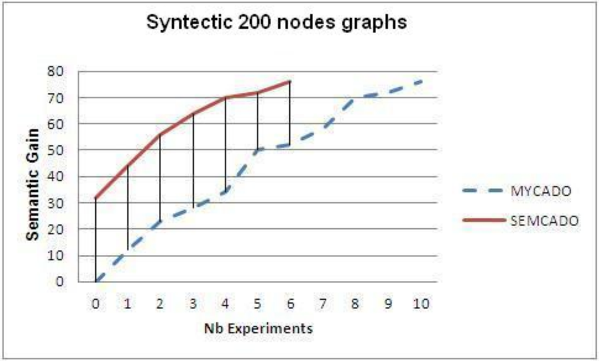
Benchmark : no existing real benchmark & system :-(

- BN graph : random generation (50 to 200 nodes).
- Ontology :
  - Causal relationships : BN edges
  - Hierarchy of concept : generation by clustering BN nodes
- Data is generated by using BN as a generative model.

Experimental protocol

- Hierarchy of concepts and 10% to 40% of existing causal relationships are given as inputs
- Semantic gain : cumulative Rada distance of the discovered relationships

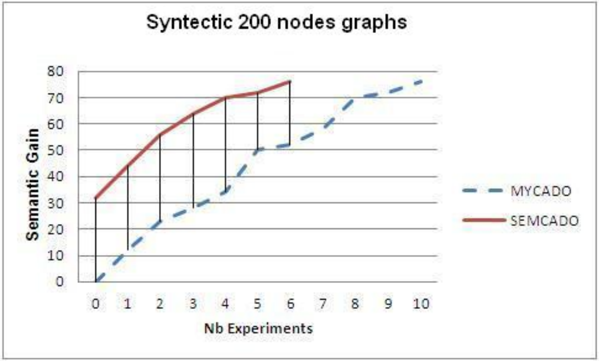
Experimental study (2)



Ontology helps structure learning

- SemCaDo performs better in less steps than MyCaDo

Experimental study (2)



Structure learning helps ontology evolution

- Original causal discoveries are discovered first and can be added to the ontology

Future works

SemCaDo 2.0

- Test SemCaDo on a real system.

SemCaDo 3.0

- Interact better with the ontology (not only $\mathcal{H}$) during the causal discovery process.
- Soften the continuity hypothesis : ontology enrichment
- Generalize to any type of semantic relations in $\mathcal{R}$.
- Extend to probabilistic relational models and probabilistic ontologies.

Future works

SemCaDo 2.0

- Test SemCaDo on a real system.

SemCaDo 3.0

- Interact better with the ontology (not only $\mathcal{H}$) during the causal discovery process.
- Soften the continuity hypothesis : ontology enrichment
- Generalize to any type of semantic relations in $\mathcal{R}$.
- Extend to probabilistic relational models and probabilistic ontologies.

Outline

Motivations

Previous works between BNs and Ontologies

- Do not explore all the expressive capabilities of ontologies.
- Focus for the most part on reasoning

Our idea

- Look for one BN extension : Object Oriented BNs.

Motivations

Previous works between BNs and Ontologies

- Do not explore all the expressive capabilities of ontologies.
- Focus for the most part on reasoning

Our idea

- Look for one BN extension : Object Oriented BNs.

Object Oriented Bayesian Networks

An extension of BNs using the object paradigm

- [Bangsø and Wuillemin, 2000a; Bangsø and Wuillemin, 2000b; Koller and Pfeffer, 1997].
- Support several aspects of the object oriented modeling. (e.g., inheritance, instantiation).
- Designed to model large and complex domains.

OOBN structure learning : OO-SEM

This algorithm [Langseth and Nielsen, 2003] is based on 2 steps

- Generation of a prior OOBN based on a prior expert knowledge
  - Grouping nodes into instantiations and instantiations into classes.
  - Giving a prior information about the candidate interfaces.
- Adaptation of Structural EM algorithm to learn the final structure.

Object Oriented Bayesian Networks

An extension of BNs using the object paradigm

- [Bangsø and Wuillemin, 2000a; Bangsø and Wuillemin, 2000b; Koller and Pfeffer, 1997].
- Support several aspects of the object oriented modeling. (e.g., inheritance, instantiation).
- Designed to model large and complex domains.

OOBN structure learning : OO-SEM

This algorithm [Langseth and Nielsen, 2003] is based on 2 steps

- Generation of a prior OOBN based on a prior expert knowledge
  - Grouping nodes into instantiations and instantiations into classes.
  - Giving a prior information about the candidate interfaces.
- Adaptation of Structural EM algorithm to learn the final structure.

OOBN structure learning

Drawbacks

- This prior knowledge is not always obvious to obtain.
- The expert should be familiar with the object oriented modeling.

Our idea

Harness ontologies representation capabilities in order to generate the prior OOBN structure.

Ontologies	OOBNs
Concepts $\mathcal{C}p$	Classes
Properties $\mathcal{P}_{cp_i}$	Real nodes
Inheritance relations $\mathcal{H}_R$	Class hierarchies
Semantic relations $\mathcal{S}_R$	Links/ Interfaces

OOBN structure learning

Drawbacks

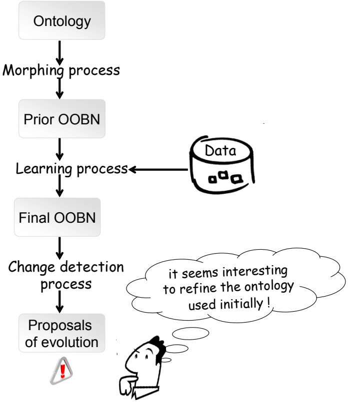
- This prior knowledge is not always obvious to obtain.
- The expert should be familiar with the object oriented modeling.

Our idea

Harness ontologies representation capabilities in order to generate the prior OOBN structure.

Ontologies	OOBNs
Concepts $\mathcal{C}p$	Classes
Properties $\mathcal{P}_{cp_i}$	Real nodes
Inheritance relations $\mathcal{H}_R$	Class hierarchies
Semantic relations $\mathcal{S}_R$	Links/ Interfaces

The 2OC approach



(1) ontology to prior OOBN

[Ben Ishak et al. 2011a]

(2) final OOBN to ontology

[Ben Ishak et al. 2011b]

Onto2PriorOOBN algorithm

Ontology based generation of a prior OOBN

- Ontology graph traversal and morphing into a prior OOBN structure.
- 3 steps
  - **Initialization step:** to generate the OOBN class and a class to each concept.
  - **Discovery step:** to define input, internal and output sets for each class of the OOBN.
  - **Closing step:** to define instances to add to the global OOBN class.

Onto2PriorOOBN algorithm

Ontology based generation of a prior OOBN

- Ontology graph traversal and morphing into a prior OOBN structure.
- 3 steps
  - **Initialization step:** to generate the OOBN class and a class to each concept.
  - **Discovery step:** to define input, internal and output sets for each class of the OOBN.
  - **Closing step:** to define instances to add to the global OOBN class.

Onto2PriorOOBN algorithm

Ontology based generation of a prior OOBN

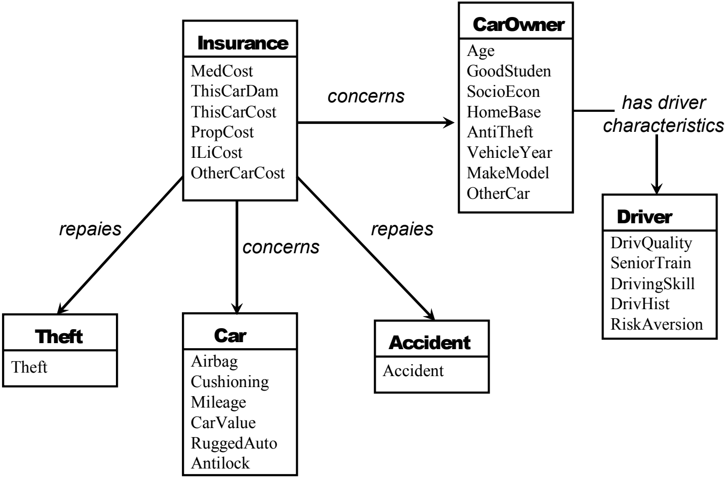
- Ontology graph traversal and morphing into a prior OOBN structure.
- 3 steps
  - **Initialization step:** to generate the OOBN class and a class to each concept.
  - **Discovery step:** to define input, internal and output sets for each class of the OOBN.
  - **Closing step:** to define instances to add to the global OOBN class.

Onto2PriorOOBN algorithm

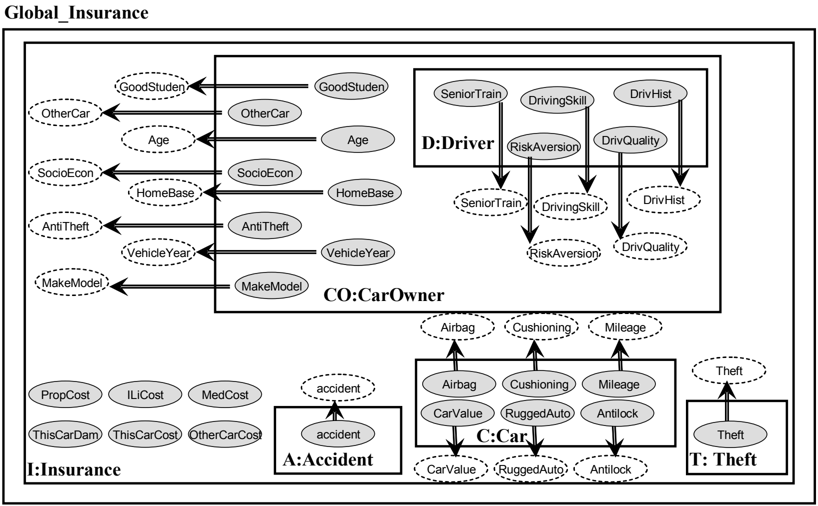
Ontology based generation of a prior OOBN

- Ontology graph traversal and morphing into a prior OOBN structure.
- 3 steps
  - **Initialization step:** to generate the OOBN class and a class to each concept.
  - **Discovery step:** to define input, internal and output sets for each class of the OOBN.
  - **Closing step:** to define instances to add to the global OOBN class.

Illustrative example : initial ontology



Illustrative example : prior OOBN



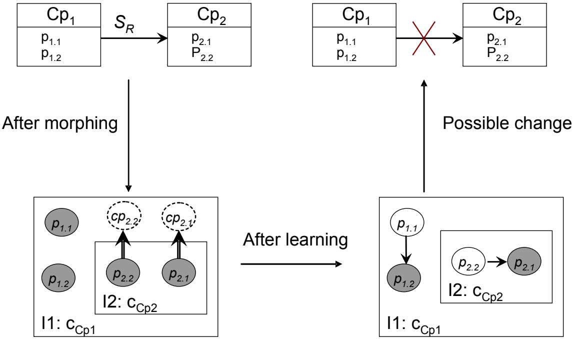
FinalOOBN2Onto

Ontology enrichment based on OOBN learning

- Part of the ontology evolution process.
- Consists in adding, removing or modifying concepts, properties and/or relations.

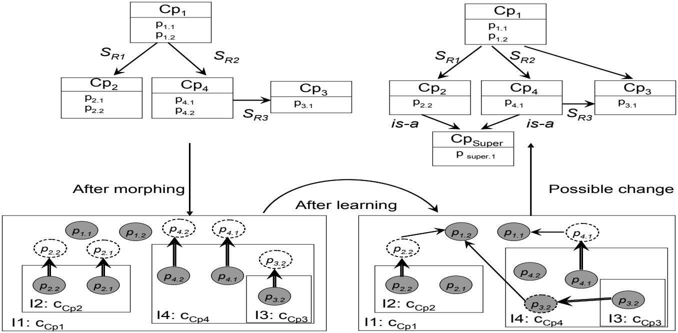
Example : remove relations

- No common interface identified between two classes
  $\Rightarrow$ their corresponding concepts should be independent.



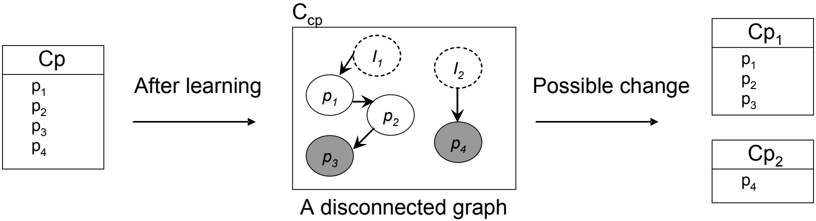
Example : add concepts / relations

- If c_{cp} communicates with only one class → Add relation.
- Otherwise, check classes similarities → Add concepts / relations.



Example : concepts redefinition

- If the class contains more than one component, then The corresponding concept may be deconstructed into more refined ones.



Concepts and relations identification

Semi-automated process

- The possible changes are communicated to an expert
- The expert semantically identify the discovered relations and / or concepts

Conclusion

We presented here two approaches for integrating semantical knowledge in order to help BN structure learning :

- SEMCADO : Causal BN, choice of "original" experiments
- O2C : Object-oriented BN, observational data

Originality

- Use an ontology instead of expert knowledge : separation between expert acquisition and structure learning
- The bidirectional benefit: a real cooperation, in both ways, between ontologies and OOBNs.

Difficulties

- No similar work or benchmark for the comparative study

Conclusion

We presented here two approaches for integrating semantical knowledge in order to help BN structure learning :

- SEMCADO : Causal BN, choice of "original" experiments
- O2C : Object-oriented BN, observational data

Originality

- Use an ontology instead of expert knowledge : separation between expert acquisition and structure learning
- The bidirectional benefit: a real cooperation, in both ways, between ontologies and OOBNs.

Difficulties

- No similar work or benchmark for the comparative study

Conclusion

We presented here two approaches for integrating semantical knowledge in order to help BN structure learning :

- SEMCADO : Causal BN, choice of "original" experiments
- O2C : Object-oriented BN, observational data

Originality

- Use an ontology instead of expert knowledge : separation between expert acquisition and structure learning
- The bidirectional benefit: a real cooperation, in both ways, between ontologies and OOBNs.

Difficulties

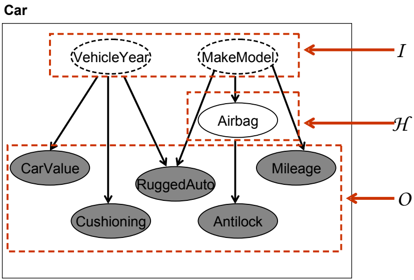
- No similar work or benchmark for the comparative study

Thank you for your attention

- Meganck, S., Leray, P., and Manderick, B. (2006). Learning causal bayesian networks from observations and experiments: A decision theoritic approach. In Proceedings of the Third International Conference, MDAI 2006, LNAI 3885, pages 58-69, Tarragona, Spain. Springer.
- Ben Messaoud, M., Leray, P., and Ben Amor, N. (2009). Integrating ontological knowledge for iterative causal discovery and vizualisation. In Proceedings of the 10th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU 2009), pages 168-179, Verona, Italy.
- Ben Messaoud, M., Leray, P., and Ben Amor, N. (2011a). Semcado: a serendipitous strategy for learning causal bayesian networks using ontologies. In Proceedings of the 11th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU 2011), pages 182-193, Belfast, Northern Ireland.
- Ben Messaoud, M., Leray, P., and Ben Amor, N. (2011b). Semcado: a serendipitous causal discovery algorithm for ontology evolution. In The IJCAI-11 Workshop on Automated Reasoning about Context and Ontology Evolution (ARCOE-11), pages 43-47, Barcelona, Spain.
- Ben Ishak, M., Leray, P., and Ben Amor, N. (2011a). Ontology-based generation of object oriented bayesian networks. In Proceedings of the 8th Bayesian Modeling Applications Workshop (BMAW-11), workshop of the 27th Conference on Uncertainty in Artificial Intelligence (UAI 2011), pages 9-17, Barcelona, Spain.
- Ben Ishak, M., Leray, P., and Ben Amor, N. (2011b). A two-way approach for probabilistic graphical models structure learning and ontology enrichment. In Proceedings of the International Conference on Knowledge Engineering and Ontology Development (KEOD 2011) part of the International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management IC3K, pages ?-?, Paris, France.

OOBN definition

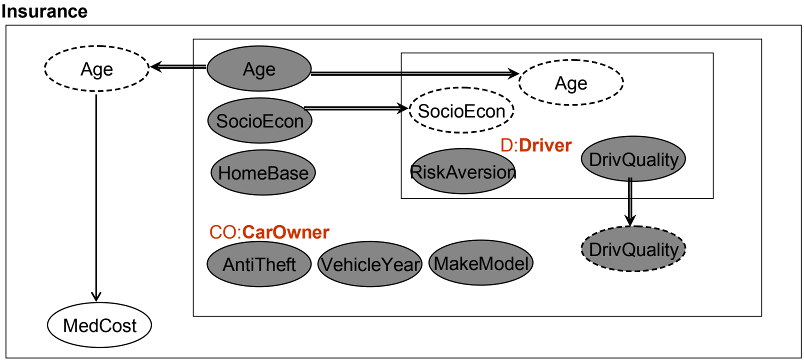
- Models the domain using fragments of a Bayesian network known as classes.
- Each class is a DAG over three sets of nodes $(\mathcal{I}, \mathcal{H}, \mathcal{O})$:



- $\mathcal{I} + \mathcal{O} =$ *the class interface.*

OOBN nodes

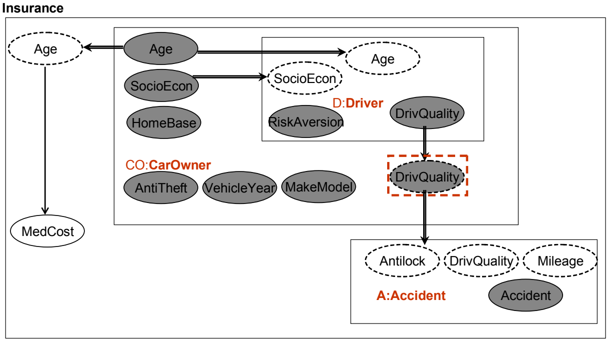
Instantiations: representing the instantiation of a class inside another class.



OOBN nodes

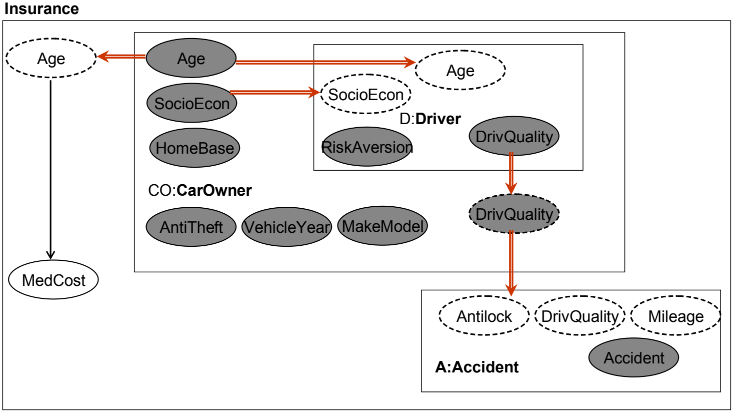
Simple nodes:

- **Reference nodes:** for specification of input and output nodes only.
- **Real nodes:** represent variables.



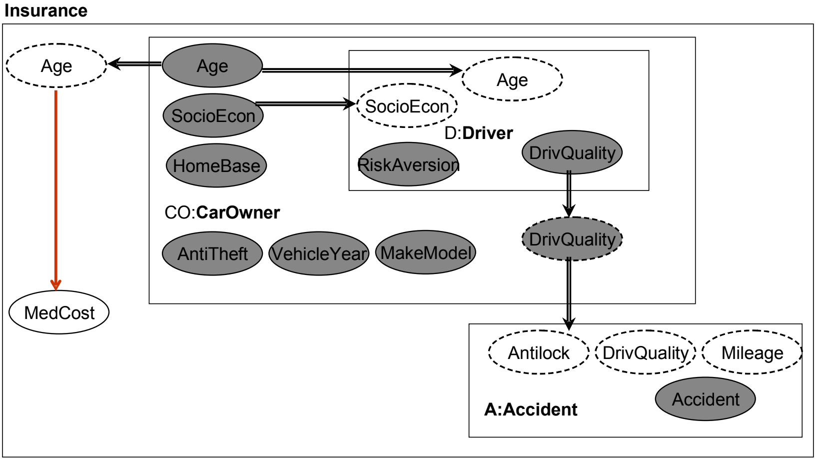
OOBN links

Reference links: to link reference or real nodes to reference nodes.



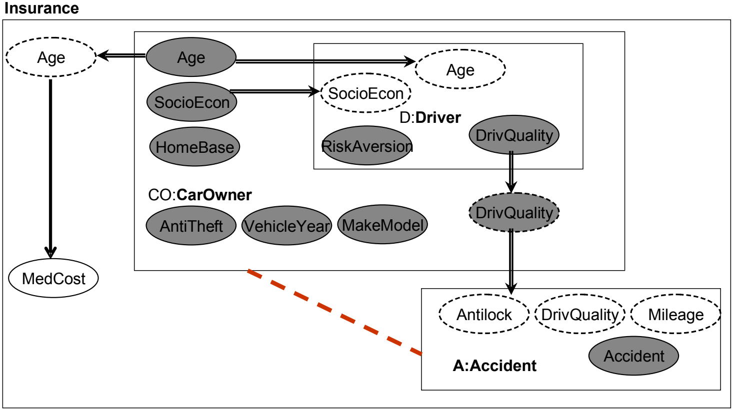
OOBN links

Directed links: to link reference or real nodes to real nodes.



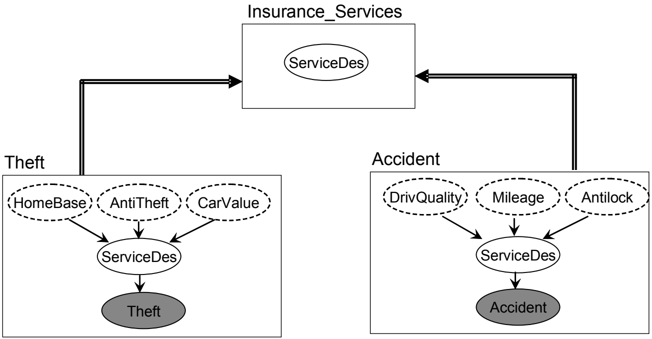
OOBN links

Construction links: to express that two nodes (or instantiations) are linked in some manner.

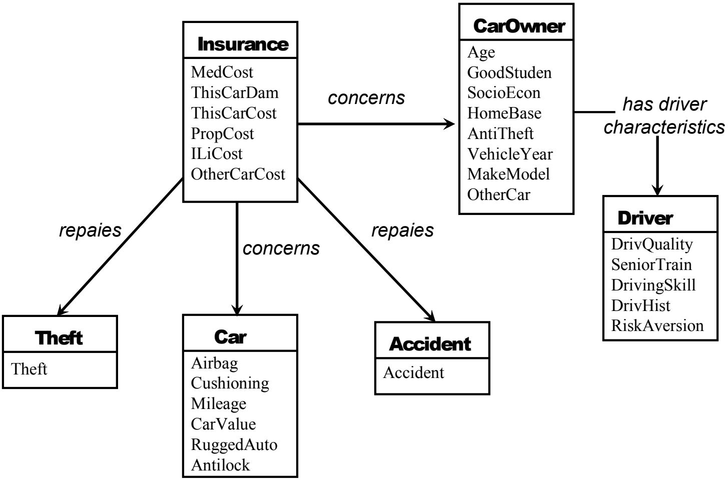


OOBN classes hierarchy

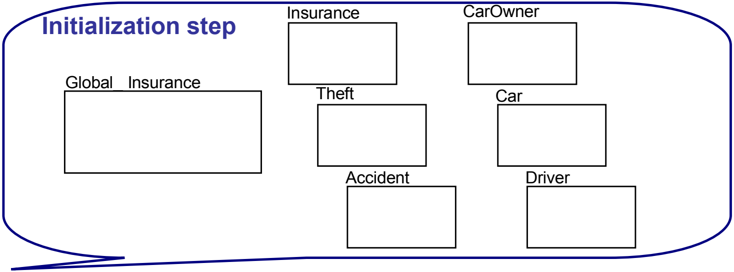
- Classes may have subclasses.
  - Subclass inherits all the superclass nodes.
  - Subclass may have additional nodes not yet represented in the superclass.



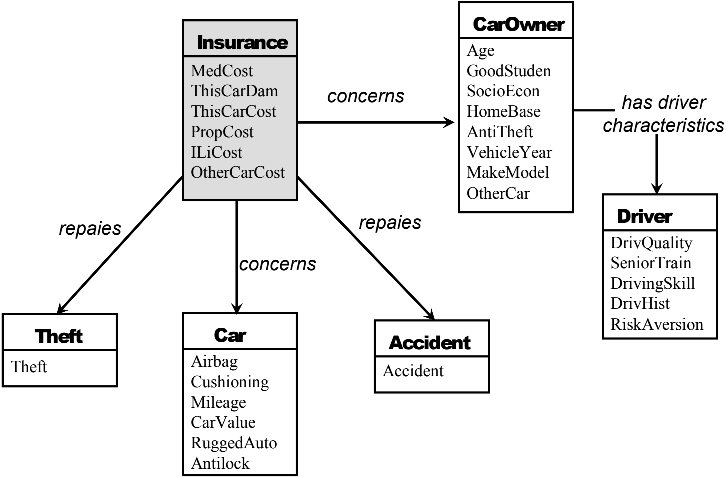
Illustrative example : more details



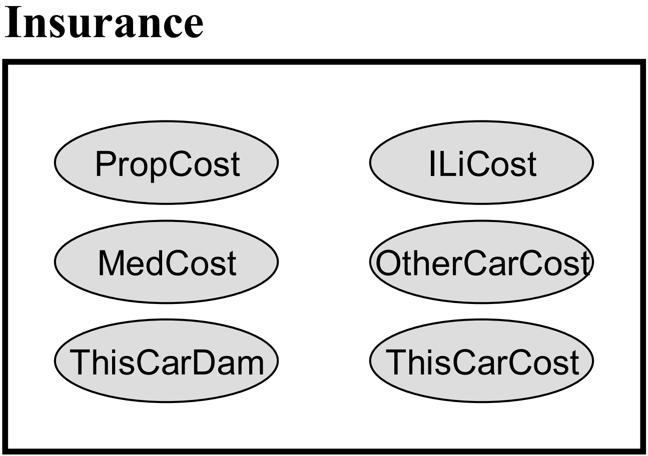
Illustrative example : more details



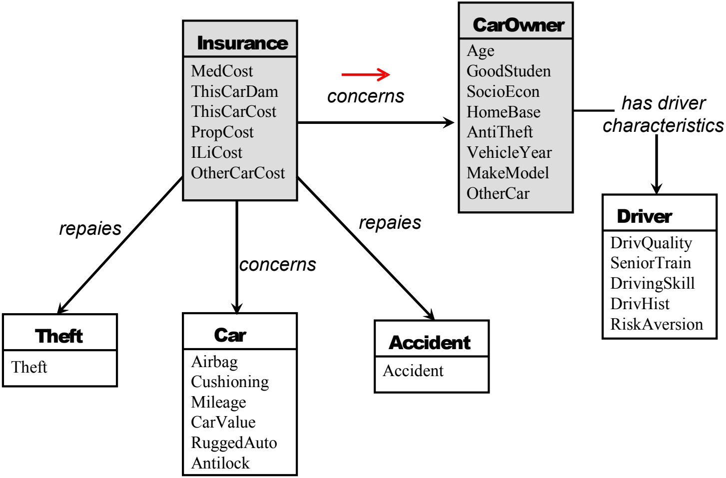
Illustrative example : more details



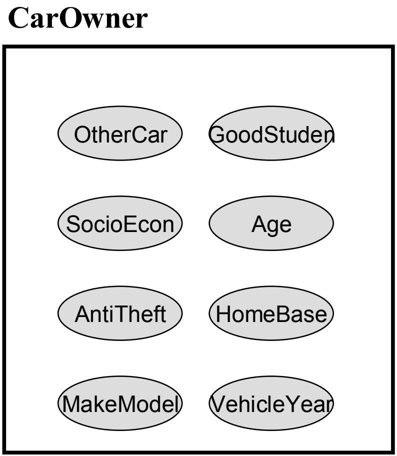
Illustrative example : more details



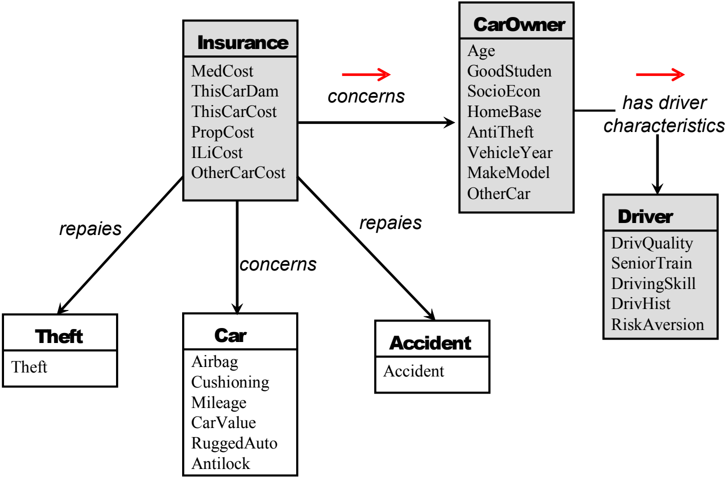
Illustrative example : more details



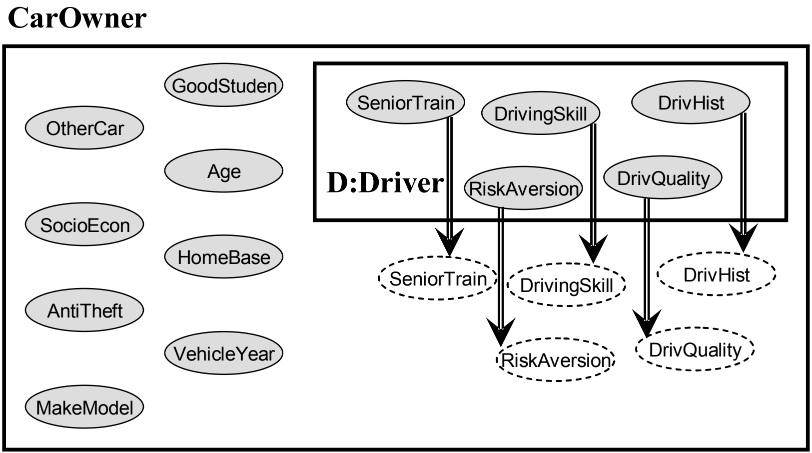
Illustrative example : more details



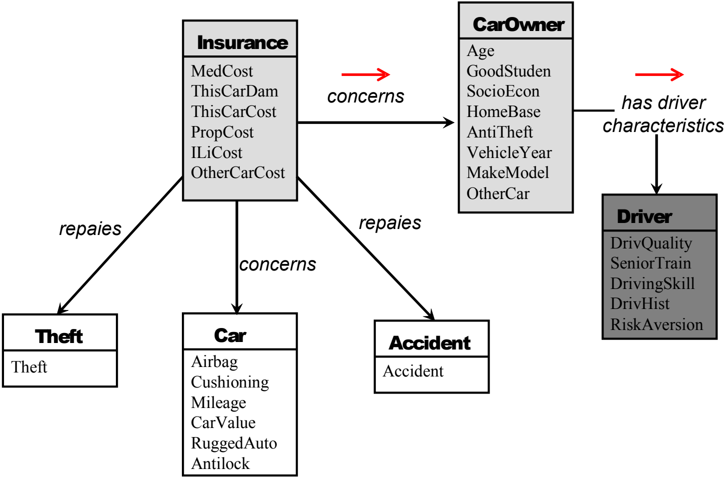
Illustrative example : more details



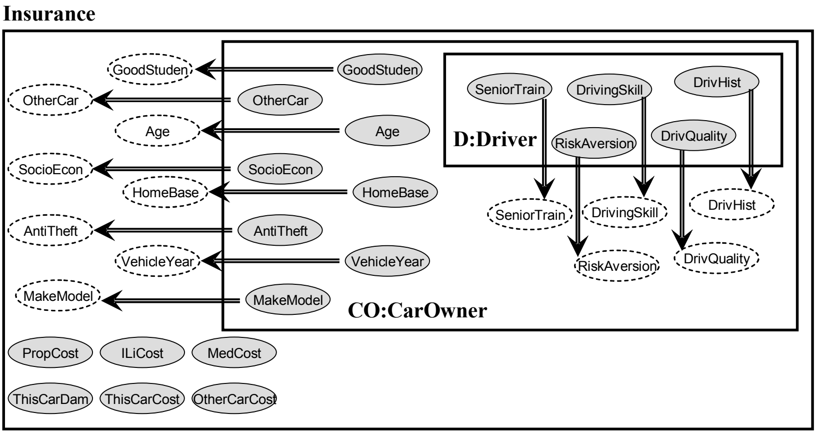
Illustrative example : more details



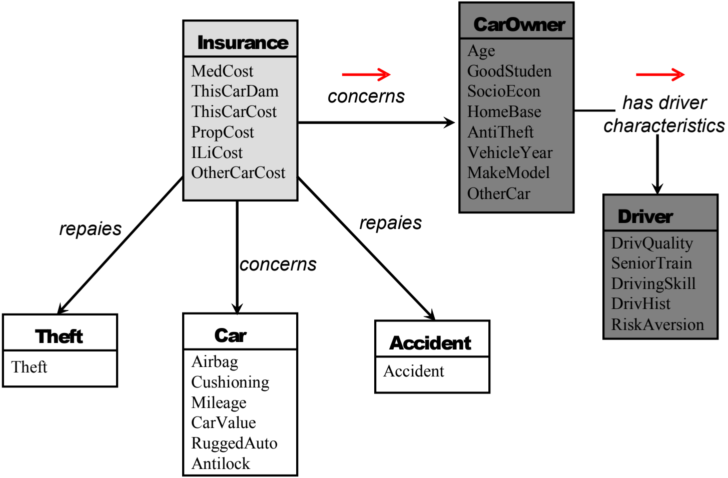
Illustrative example : more details



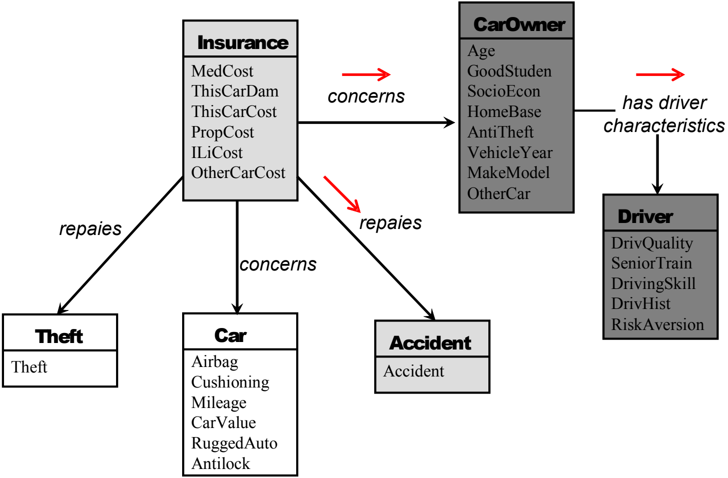
Illustrative example : more details



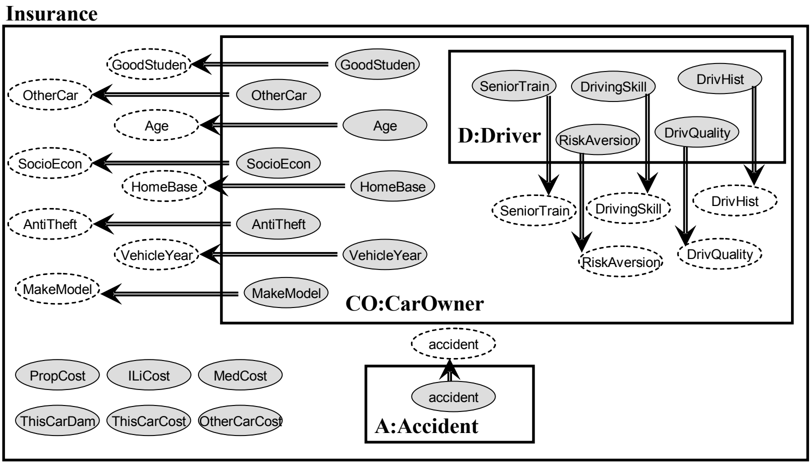
Illustrative example : more details



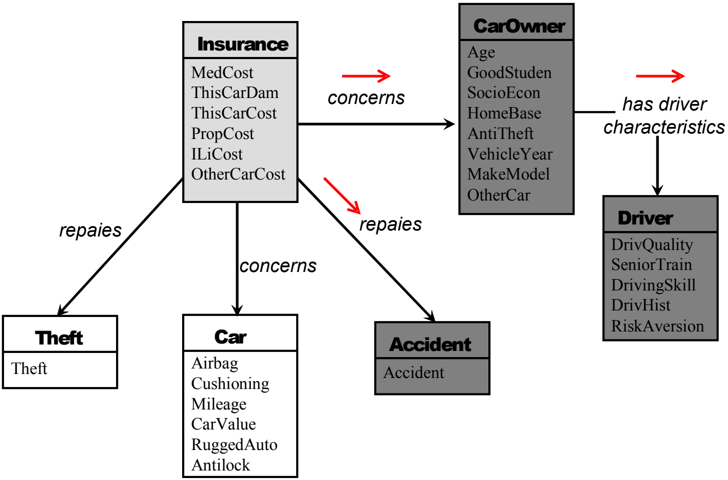
Illustrative example : more details



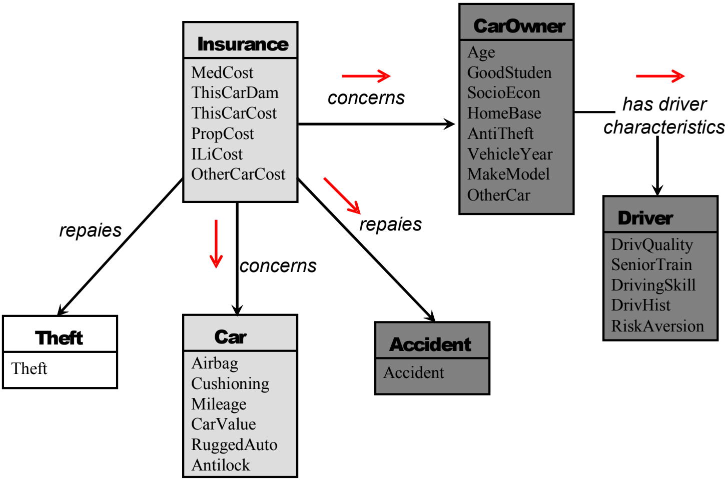
Illustrative example : more details



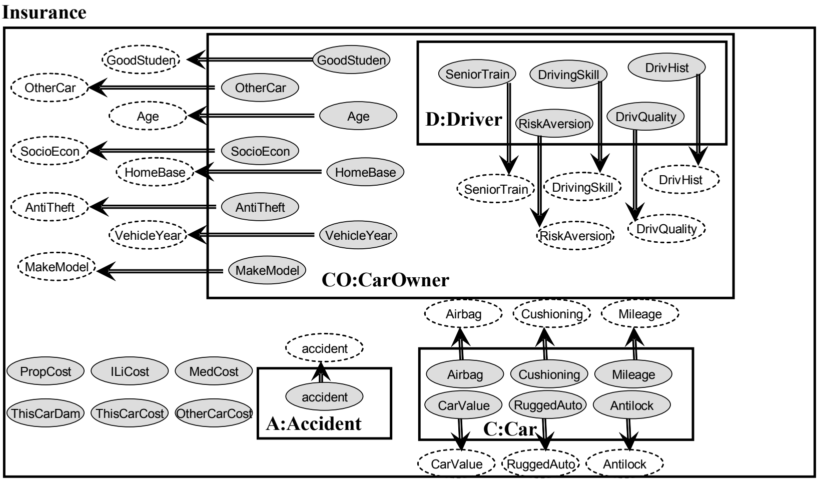
Illustrative example : more details



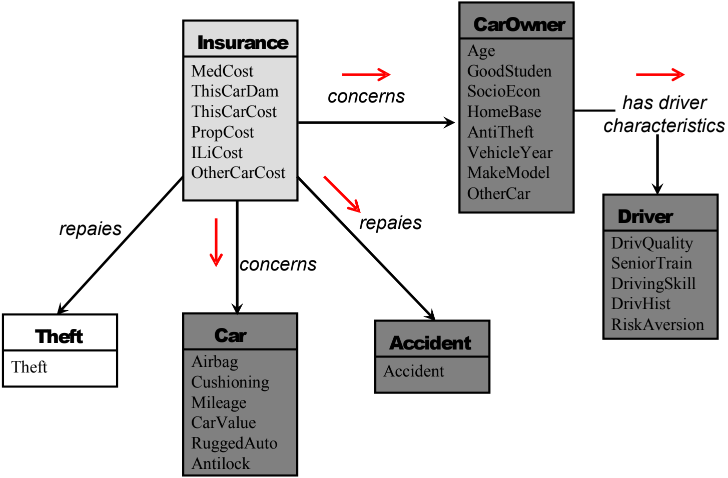
Illustrative example : more details



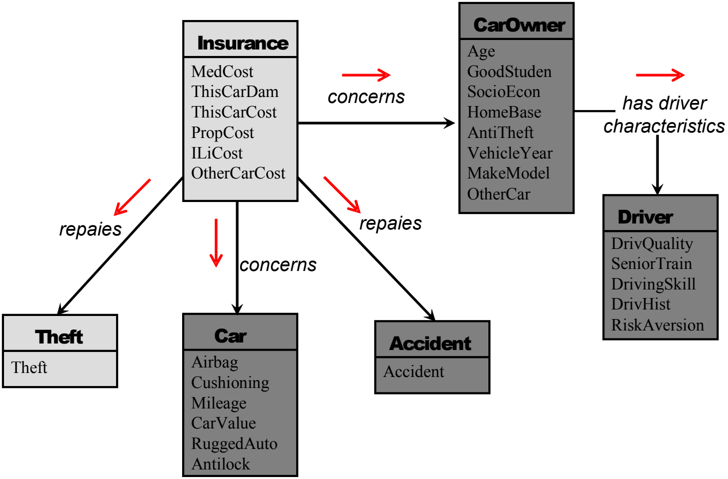
Illustrative example : more details



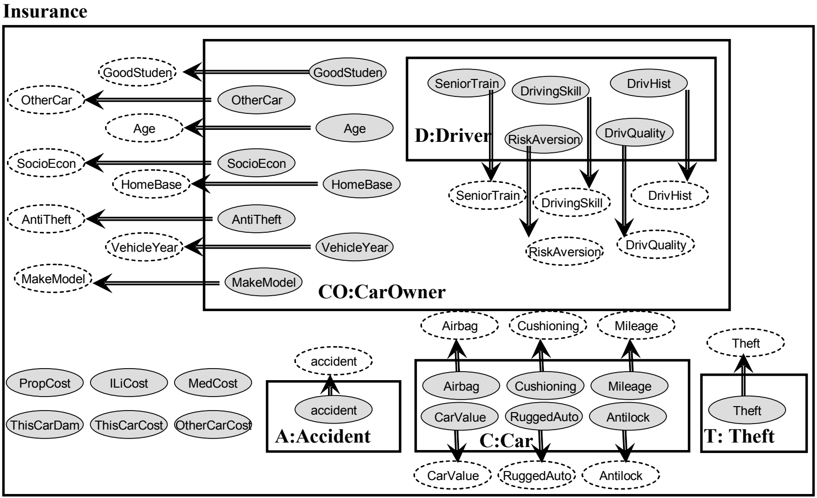
Illustrative example : more details



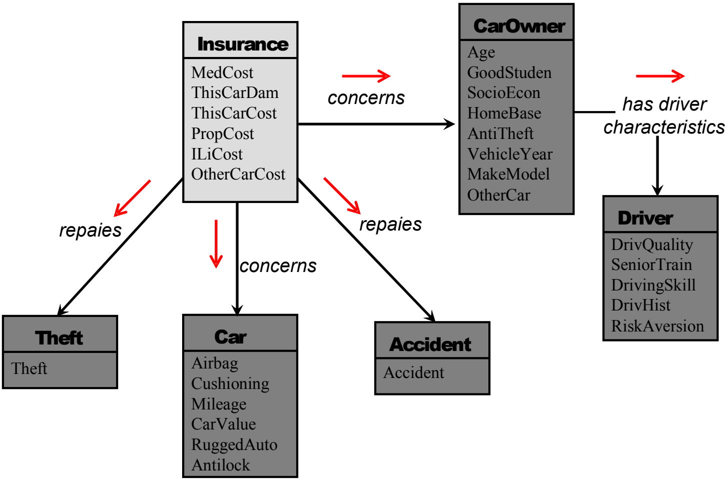
Illustrative example : more details



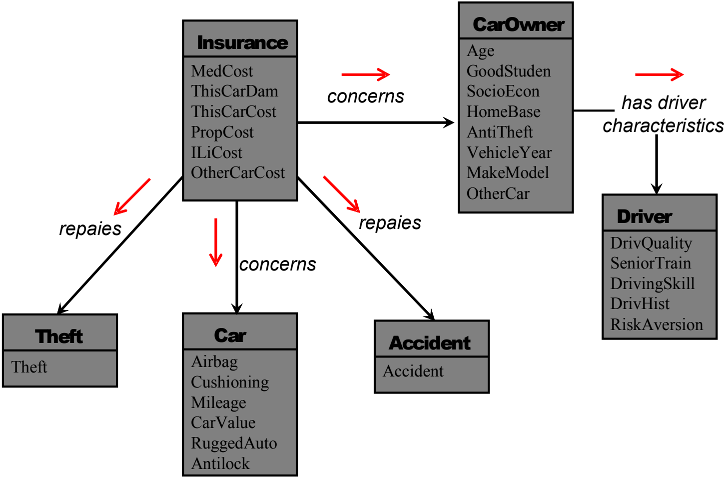
Illustrative example : more details



Illustrative example : more details



Illustrative example : more details



Illustrative example : more details