Advances in Learning with Bayesian Networks

Philippe Leray
philippe.leray@univ-nantes.fr

DUKe (Data User Knowledge) research group, LS2N UMR 6004, Nantes, France

Séminaire IRT SystemX, 19 jan. 2017
1er janvier 2017, fusion de 2 UMR

- IRCCyN (Institut de Recherche en Communications et Cybernétique de Nantes)
- LINA (Laboratoire d’Informatique de Nantes Atlantique)

- 450 personnes (≈ 215 permanents)
- 5 axes thématiques
  - Conception et Conduite de Systèmes
  - Robotique, Procédés, Calcul
  - Signaux, Images, Ergonomie et Langues
  - Science du Logiciel et des Systèmes Distribués
  - Science des Données et de la Décision

- 5 établissements support
  - Ecole Centrale de Nantes
  - Université de Nantes
  - Institut Mines Telecom Atlantique
  - CNRS, INRIA
Research group in Data Sciences

- 16 permanents
- 3 associates
- 12 PhD students
- 2 postdoc/engineers
DUKe scientific approach

Proposing « agile », « user in the loop »
Data-mining / Machine Learning algorithms

Applications and valorisation
Digital Humanity, BioInformatics, Business Intelligence
Motivations

- Bayesian networks (BNs) are a powerful tool for graphical representation of the underlying knowledge in the data and reasoning with incomplete or imprecise observations.

- BNs have been extended (or generalized) in several ways, as for instance, causal BNs, dynamic BNs, relational BNs, ...
Motivations

- Bayesian networks (BNs) are a powerful tool for graphical representation of the underlying knowledge in the data and reasoning with incomplete or imprecise observations.

- BNs have been extended (or generalized) in several ways, as for instance, causal BNs, dynamic BNs, relational BNs, ...
Motivations

Victim identification system

Turbo-codes (GSM, ...)

Anti Spam

MS Office assistant

It looks like you're writing a letter.

Would you like help?
- Get help with writing the letter
- Just type the letter without help

Don't show me this tip again

Assistant iPhone SIRI

“ How do you make a chicken pot pie ”

Would you like to search the web for ‘How do you make a chicken pot pie’?

“ Bitch I know how to use Google ”

I'll pretend I didn't hear that.
Motivations

We would like to learn a BN from data... but which kind of data?

- complete
Motivations

We would like to learn a BN from data... but which kind of data?

- complete / incomplete [François 06]
Motivations

We would like to learn a BN from data... but which kind of data?

- complete / incomplete [François 06]
- high $n$,
Motivations

We would like to learn a BN from data... but which kind of data?

- complete / incomplete [François 06]
- high $n, n \gg p$ [Ammar 11]

\[
\begin{array}{cccc|c}
A & B & C & D & X \\
0 & 1 & 2 & 3 & 7 \\
4 & 6 & 1 & 0 & 5 \\
2 & 3 & 5 & 6 & 4 \\
\end{array}
\]
Motivations

We would like to learn a BN from data... but which kind of data?

- complete /incomplete [François 06]
- high $n$, $n \gg p$ [Ammar 11]
- stream [Yasin 13]
Motivations

We would like to learn a BN from data... but which kind of data?

- complete / incomplete [François 06]
- high $n$, $n \gg p$ [Ammar 11]
- stream [Yasin 13]
- + prior knowledge / ontology [Ben Messaoud 13]
Motivations

We would like to learn a BN from data... but which kind of data?

- complete / incomplete [François 06]
- high $n$, $n \gg p$ [Ammar 11]
- stream [Yasin 13]
- + prior knowledge / ontology [Ben Messaoud 13]
- structured data [Ben Ishak 15, Coutant 15, Chulyadyo 16]
Motivations

We would like to learn a BN from data... but which kind of data?

- complete /incomplete [François 06]
- high $n, n \gg p$ [Ammar 11]
- stream [Yasin 13]
- + prior knowledge / ontology [Ben Messaoud 13]
- structured data [Ben Ishak 15, Coutant 15, Chulyadyo 16]
- not so structured data [Elabri]
Motivations

Even the learning task can differ: generative

- modeling $P(X, Y)$
- no target variable
- more general model
- better behavior with incomplete data

Objectives of this talk

- how to learn BNs in such various contexts?
- state of the art: founding algorithms and recent ones
- pointing out our contributions in this field
Motivations

Even the learning task can differ: generative vs. discriminative

- modeling $P(X, Y)$
- no target variable
- more general model
- better behavior with incomplete data

- modeling $P(Y|X)$
- one target variable $Y$
- dedicated model

Objectives of this talk

- how to learn BNs in such various contexts?
- state of the art: founding algorithms and recent ones
- pointing out our contributions in this field
Motivations

Even the learning task can differ: generative vs. discriminative

- modeling $P(X, Y)$
- no target variable
- more general model
- better behavior with incomplete data

- modeling $P(Y|X)$
- one target variable $Y$
- dedicated model

Objectives of this talk

- how to learn BNs in such various contexts?
- state of the art: founding algorithms and recent ones
- pointing out our contributions in this field
Outline ... 

1. **BN learning**
   - Definition
   - Parameter learning
   - Structure learning

2. **Dynamic BN learning**
   - Definition
   - Learning

3. **Relational BN learning**
   - Definitions
   - Learning with a relational DB
   - Learning with a Graph DB

4. The end

Philippe Leray

Advances in Learning with Bayesian Networks 9/45
Bayesian network

[Pearl, 1985]

Definition

G qualitative description of conditional dependences / independences between variables
directed acyclic graph (DAG)

Θ quantitative description of these dependences conditional probability distributions (CPDs)

Main property

- the global model is decomposed into a set of local conditional models
Bayesian network

[Pearl, 1985]

**Definition**
- \( G \) qualitative description of conditional dependences / independences between variables
directed acyclic graph (DAG)
- \( \Theta \) quantitative description of these dependences conditional probability distributions (CPDs)

**Main property**
- the global model is decomposed into a set of local conditional models
One model... but two learning tasks

BN = graph $G$ and set of CPDs $\Theta$

- parameter learning / $G$ given
- structure learning

Philippe Leray

Advances in Learning with Bayesian Networks 11/45
One model... but two learning tasks

\[ BN = \text{graph } G \text{ and set of CPDs } \Theta \]

- parameter learning / \( G \) given
- structure learning

Philippe Leray

Advances in Learning with Bayesian Networks 11/45
One model... but two learning tasks

BN = graph $G$ and set of CPDs $\Theta$

- parameter learning / $G$ given
- structure learning
Parameter learning (generative)

**Complete data \( \mathcal{D} \)**

- **max. of likelihood (ML)**: \( \hat{\theta}^{MV} = \arg\max P(\mathcal{D}|\theta) \)
- **closed-form solution**:

\[
\hat{P}(X_i = x_k|Pa(X_i) = x_j) = \hat{\theta}_{i,j,k}^{MV} = \frac{N_{i,j,k}}{\sum_k N_{i,j,k}}
\]

\( N_{i,j,k} = \text{nb of occurrences of } \{X_i = x_k \text{ and } Pa(X_i) = x_j\} \)

**Other approaches**

- **max. a posteriori (MAP)**: \( \hat{\theta}^{MAP} = \arg\max P(\theta|\mathcal{D}) \)
- **expectation a posteriori (EAP)**: \( \hat{\theta}^{EAP} = \mathbb{E}(P(\theta|\mathcal{D})) \)

\[
\hat{\theta}_{i,j,k}^{MAP} = \frac{N_{i,j,k} + \alpha_{i,j,k}-1}{\sum_k (N_{i,j,k} + \alpha_{i,j,k}-1)}
\]

\[
\hat{\theta}_{i,j,k}^{EAP} = \frac{N_{i,j,k} + \alpha_{i,j,k}}{\sum_k (N_{i,j,k} + \alpha_{i,j,k})}
\]
Parameter learning (generative)

**Complete data** $\mathcal{D}$

- **max. of likelihood (ML)**: $\hat{\theta}^{MV} = \text{argmax} \ P(\mathcal{D}|\theta)$
- **closed-form solution**:
  \[
  \hat{P}(X_i = x_k | \text{Pa}(X_i) = x_j) = \hat{\theta}_{i,j,k}^{MV} = \frac{N_{i,j,k}}{\sum_{k} N_{i,j,k}}
  \]
  \[N_{i,j,k} = \text{nb of occurrences of } \{X_i = x_k \text{ and } \text{Pa}(X_i) = x_j\}\]

**Other approaches**

- **max. a posteriori (MAP)**: $\hat{\theta}^{MAP} = \text{argmax} \ P(\theta|\mathcal{D})$
  \[\hat{\theta}_{i,j,k}^{MAP} = \frac{N_{i,j,k}+\alpha_{i,j,k}-1}{\sum_{k}(N_{i,j,k}+\alpha_{i,j,k}-1)}\]

- **expectation a posteriori (EAP)**: $\hat{\theta}^{EAP} = \mathbb{E}(P(\theta|\mathcal{D}))$
  \[\hat{\theta}_{i,j,k}^{EAP} = \frac{N_{i,j,k}+\alpha_{i,j,k}}{\sum_{k}(N_{i,j,k}+\alpha_{i,j,k})}\]

$P(\theta) \sim \text{Dirichlet}(\alpha)$
Parameter learning (generative)

**Incomplete data**

- no closed-form solution
- EM (iterative) algorithm [Dempster, 77], convergence to a local optimum

**Incremental data**

- advantages of sufficient statistics

\[ \theta_{i,j,k} = \frac{N^\text{old} \theta_{i,j,k}^\text{old} + N_{i,j,k}}{N^\text{old} + N} \]

- this Bayesian updating can include a forgetting factor
**Parameter learning (generative)**

**Incomplete data**
- no closed-form solution
- EM (iterative) algorithm [Dempster, 77], convergence to a local optimum

**Incremental data**
- advantages of sufficient statistics
  \[
  \theta_{i,j,k} = \frac{N^{old} \theta_{i,j,k}^{old} + N_{i,j,k}}{N^{old} + N}
  \]
- this Bayesian updating can include a forgetting factor
Parameter learning (discriminative)

**Complete data**
- no closed-form
- iterative algorithms such as gradient descent

**Incomplete data**
- no closed-form
- iterative algorithms + EM :-(

Philippe Leray

Advances in Learning with Bayesian Networks 14/45
Parameter learning (discriminative)

Complete data
- no closed-form
- iterative algorithms such as gradient descent

Incomplete data
- no closed-form
- iterative algorithms + EM :-(
BN 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]
  
  $NS(5) = 29281$ \quad $NS(10) = 4.2 \times 10^{18}$

- An exhaustive search is impossible for realistic $n$!

  One thousand millenniums $= 3.2 \times 10^{13}$ seconds

**Identifiability**

- Data can only help finding (conditional) dependences / independences
- Markov Equivalence: several graphs describe the same dependence statements
- Causal Sufficiency: do we know all the explaining variables?
BN 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]
  \[ NS(5) = 29281 \quad NS(10) = 4.2 \times 10^{18} \]
- an exhaustive search is impossible for realistic $n$!

One thousand millenniums $= 3.2 \times 10^{13}$ seconds

**Identifiability**
- data can only help finding (conditional) dependences / independences
- Markov Equivalence: several graphs describe the same dependence statements
- causal Sufficiency: do we know all the explaining variables?
Structure learning (generative / complete)

### Constraint-based methods
- BN = independence model
  - ⇒ find CI in data in order to build the DAG
  - ex: IC [Pearl & Verma, 91], PC [Spirtes et al., 93]
- problem: reliability of CI statistical tests (ok for $n < 100$)

### Score-based methods
- BN = probabilistic model that must fit data as well as possible
- problem: size of search space (ok for $n < 1000$)

### Hybrid/ local search methods
- local search / neighbor identification (statistical tests)
- global (score) optimization
- usually for scalability reasons (ok for high $n$)
**Constraint-based methods**

- BN = independence model
- problem: reliability of CI statistical tests (ok for $n < 100$)

**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
  - ex: Maximum Weighted Spanning Tree [Chow & Liu, 68], Greedy Search [Chickering, 95], evolutionary approaches [Larranaga et al., 96] [Wang & Yang, 10]
- problem: size of search space (ok for $n < 1000$)
### Constraint-based methods
- BN = independence model
- problem: reliability of CI statistical tests (ok for $n < 100$)

### Score-based methods
- BN = probabilistic model that must fit data as well as possible
- problem: size of search space (ok for $n < 1000$)

### Hybrid/ local search methods
- local search / neighbor identification (statistical tests)
- global (score) optimization
- usually for scalability reasons (ok for high $n$)
- ex: MMHC algorithm [Tsamardinos et al., 06]
Structure learning (discriminative)

Specific structures

- naive Bayes, augmented naive Bayes
- multi-nets
- ...

Structure learning

- usually, the structure is learned in a generative way
- the parameters are then tuned in a discriminative way
Structure learning (discriminative)

Specific structures
- naive Bayes, augmented naive Bayes
- multi-nets
- ...

Structure learning
- usually, the structure is learned in a generative way
- the parameters are then tuned in a discriminative way
Incomplete data

- hybridization of previous structure learning methods and EM
- ex: Structural EM [Friedman, 97]
  \[ \simeq \text{Greedy Search } + \text{EM} \]
- problem: convergence

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>?</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>5</td>
<td>?</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>?</td>
<td>6</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td>9</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>4</td>
<td>?</td>
</tr>
<tr>
<td>1</td>
<td>?</td>
<td>3</td>
<td>7</td>
</tr>
<tr>
<td>8</td>
<td>5</td>
<td>4</td>
<td>3</td>
</tr>
</tbody>
</table>
Structure learning

$n >> p$

- robustness and complexity issues
- application of Perturb & Combine principle
- ex: mixture of randomly perturbed trees
  [Ammar & Leray, 11]

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>...</th>
<th>...</th>
<th>...</th>
<th>X_{10000}</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>7</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>1</td>
<td>0</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>5</td>
<td>6</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>4</td>
</tr>
</tbody>
</table>
Incremental learning and data streams

- Bayesian updating is easy for parameters
- Bayesian updating is complex for structure learning
- and other constraints related to data streams (limited storage, ...)
- ex: incremental MMHC [Yasin and Leray, 13]

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>...</th>
<th>...</th>
<th>...</th>
<th>X_{10000}</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>7</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>1</td>
<td>0</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>5</td>
<td>6</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>2</td>
<td>6</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>7</td>
</tr>
<tr>
<td>3</td>
<td>8</td>
<td>9</td>
<td>0</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>4</td>
<td>5</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>3</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>3</td>
<td>7</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>2</td>
</tr>
<tr>
<td>8</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>4</td>
</tr>
</tbody>
</table>
Integration of prior knowledge
- in order to reduce search space: white list, black list, node ordering [Campos & Castellano, 07]
- interaction with ontologies [Ben Messaoud et al., 13]
Outline ...

1. BN learning
   - Definition
   - Parameter learning
   - Structure learning

2. Dynamic BN learning
   - Definition
   - Learning

3. Relational BN learning
   - Definitions
   - Learning with a relational DB
   - Learning with a Graph DB

4. The end
Dynamic Bayesian networks (DBNs)

k slices temporal BN (k-TBN) [Murphy, 02]

- $k - 1$ Markov order
- prior graph $G_0$ + transition graph $G_→$
- for example: 2-TBNs model [Dean & Kanazawa, 89]

Simplified k-TBN

- k-TBN with only temporal edges [Dojer, 06][Vinh et al, 12]
Dynamic Bayesian networks (DBNs)

**k slices temporal BN (k-TBN)**

- $k - 1$ Markov order
- Prior graph $G_0$ + transition graph $G_{\rightarrow}$
- For example: 2-TBNs model [Dean & Kanazawa, 89]

**Simplified k-TBN**

- k-TBN with only temporal edges [Dojer, 06][Vinh et al, 12]
Score-based methods

- dynamic Greedy Search [Friedman et al., 98], genetic algorithm [Gao et al., 07], dynamic Simulated Annealing [Hartemink, 05], ...
- for k-TBN (\(G_0\) and \(G_\rightarrow\) learning)
- but not scalable (high \(n\))

Hybrid methods

- [Dojer, 06] [Vinh et al., 12] for simplified k-TBN, but often limited to \(k = 2\) for scalability
- dynamic MMHC for ”unspecified” 2-TBNs with high \(n\) [Trabelsi et al., 13]
DBN structure learning (generative)

Score-based methods
- dynamic Greedy Search [Friedman et al., 98], genetic algorithm [Gao et al., 07], dynamic Simulated Annealing [Hartemink, 05], ...
- for k-TBN (G₀ and G→ learning)
- but not scalable (high n)

Hybrid methods
- [Dojer, 06] [Vinh et al., 12] for simplified k-TBN, but often limited to k = 2 for scalability
- dynamic MMHC for ”un-simplified” 2-TBNs with high n [Trabelsi et al., 13]
Outline ...

1. BN learning
   - Definition
   - Parameter learning
   - Structure learning

2. Dynamic BN learning
   - Definition
   - Learning

3. Relational BN learning
   - Definitions
   - Learning with a relational DB
   - Learning with a Graph DB

4. The end
Motivations

Flat data

- No relational model
- Learning probabilistic dependencies between variables
Motivations

Flat data

- No relational model
- Learning probabilistic dependencies between variables

Relational DB

- Relational schema is given
- Learning prob. dep. between variables, but more complex!
Motivations

Flat data
- No relational model
- Learning probabilistic dependencies between variables

Relational DB
- Relational schema is given
- Learning prob. dep. between variables, but more complex!

Graph DB
- Relational schema?
- Learning prob. dep. between variables?
Relational schema

relational schema $\mathcal{R}$

- classes + attributes
- reference slots (e.g. Vote.Movie, Vote.User)
- inverse reference slots (e.g. User.User$^{-1}$)
- slot chain = a sequence of (inverse) reference slots
  
  ex: Vote.User.User$^{-1}$.Movie all the movies voted by a particular user
Relational schema

**relational schema** $\mathcal{R}$

- classes + attributes
- reference slots (e.g. $\text{Vote.Movie, Vote.User}$)
- inverse reference slots (e.g. $\text{User.User}^{-1}$)
- slot chain = a sequence of (inverse) reference slots
  - ex: $\text{Vote.User.User}^{-1}.\text{Movie}$: all the movies voted by a particular user
Relational schema

relational schema $\mathcal{R}$

- classes + attributes
- reference slots (e.g. $\text{Vote.Movie}$, $\text{Vote.User}$)
- inverse reference slots (e.g. $\text{User.User}^{-1}$)
- slot chain = a sequence of (inverse) reference slots
  - ex: $\text{Vote.User.User}^{-1}.\text{Movie}$: all the movies voted by a particular user

<table>
<thead>
<tr>
<th>Movie</th>
<th>User</th>
<th>Vote</th>
</tr>
</thead>
<tbody>
<tr>
<td>RealiseDate</td>
<td>Gender</td>
<td>Movie</td>
</tr>
<tr>
<td>Genre</td>
<td>Age</td>
<td>User</td>
</tr>
<tr>
<td>Occupation</td>
<td>Rating</td>
<td>User</td>
</tr>
</tbody>
</table>

Movie
- RealiseDate
- Genre

User
- Gender
- Age
- Occupation

Vote
- Movie
- User
- Rating
Relational schema

**relational schema** \( R \)

- classes + attributes
- reference slots (e.g. \( \text{Vote.Movie}, \text{Vote.User} \))
- inverse reference slots (e.g. \( \text{User.User}^{-1} \))
- slot chain = a sequence of (inverse) reference slots
  - ex: \( \text{Vote.User.User}^{-1}.\text{Movie} \): all the movies voted by a particular user
Relational skeleton

Instance $\mathcal{I}$
- Set of objects for each class
- With a value for each reference slot and each attribute
- == a "populated" database

Relational skeleton $\sigma_R$
- Instance without attribute values
Probabilistic Relational Models

[Koller & Pfeffer, 98]

Definition

A PRM $\Pi$ associated to $\mathcal{R}$:

- a qualitative dependency structure $\mathcal{S}$ (with possible long slot chains and aggregation functions)
- a set of parameters $\theta_S$

<table>
<thead>
<tr>
<th>User.Gender</th>
<th>M</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.4</td>
<td>0.6</td>
<td></td>
</tr>
</tbody>
</table>
Definition

Aggregators

- \( \text{Vote.} \text{User.} \text{User}^{-1}. \text{Movie.genre} \rightarrow \text{Vote.rating} \)

  movie rating from one user can be dependent with the genre of all the movies voted by this user
  
  - how to describe the dependency with an unknown number of parents?
  - solution: using an aggregated value, e.g. \( \gamma = \text{MODE} \)
Another probabilistic relational model [Heckerman & Meek, 04]

**Definition**

- Probabilistic model associated to an Entity-Relationship model
- Classes = \{ Entity classes + Relationship classes \}
Learning from a relational database

**PRM/DAPER learning**

- finding the probabilistic dependencies and the probability tables from an instantiated database
- relational schema is known, but ...
- several situations / PRM extensions
Learning from a relational database

Attribute uncertainty

- Input: relational skeleton (all the objects and relations), some attributes
- Objective: predict only missing attributes
Learning from a relational database

Reference uncertainty
- Input: partial relational skeleton (all the objects, but some relations are missing)
- Objective: predict missing attributes and "foreign keys"
Learning from a relational database

Existence uncertainty

- Input: partial relational skeleton (all the entity objects, but some relationship objects are missing)
- Objective: predict existence of relationships between entity objects

BN learning
Dynamic BN learning
Relational BN learning
The end

Philippe Leray
Advances in Learning with Bayesian Networks 35/45
PRM/DAPER learning with AU

Relational variables

- finding new variables by exploring the relational schema
- ex: student.reg.grade, registration.course.reg.grade, registration.student reg.course.reg.grade, ...

⇒ adding another dimension in the search space
⇒ limitation to a given maximal slot chain length

Constraint-based methods

- relational PC [Maier et al., 10] relational CD [Maier et al., 13]
- don’t deal with aggregation functions

Score-based methods

- Greedy search [Getoor et al., 07]
PRM/DAPER learning with AU

Relational variables

⇒ adding another dimension in the search space
⇒ limitation to a given maximal slot chain length

Constraint-based methods

- relational PC [Maier et al., 10] relational CD [Maier et al., 13]
- don’t deal with aggregation functions

Score-based methods

- Greedy search [Getoor et al., 07]

Hybrid methods

- relational MMHC [Ben Ishak et al., 15]
PRM/DAPER learning with AU

Relational variables
- adding another dimension in the search space
- limitation to a given maximal slot chain length

Constraint-based methods
- relational PC [Maier et al., 10] relational CD [Maier et al., 13]
- don’t deal with aggregation functions

Score-based methods
- Greedy search [Getoor et al., 07]

Hybrid methods
- relational MMHC [Ben Ishak et al., 15]
PRM/DAPER learning with AU

Relational variables

- adding another dimension in the search space
- limitation to a given maximal slot chain length

Constraint-based methods

- relational PC [Maier et al., 10] relational CD [Maier et al., 13]
- don’t deal with aggregation functions

Score-based methods

- Greedy search [Getoor et al., 07]

Hybrid methods

- relational MMHC [Ben Ishak et al., 15]
Need for partitionning
- The missing foreign key is considered as a random variable.
- We need to partition the similar "target" objects in order to obtain a generic model.

How to partition
- With object attributes [Getoor et al.] = clustering.
- With relational information = graph partitionning.
- With both: [Coutant et al., 15]
Graph database

Definition
- Data is described in a graph, with nodes and relationships
- Attributes can be associated to both.

Properties
Graph database

**Definition**

- Data is described in a graph, with nodes and relationships
- Attributes can be associated to both.
Graph database

Definition

Properties

- Scalability / large data (no join operation, only graph traversal)
- Schema-free, no relational schema
Graph database

**Definition**

**Properties**
- Scalability / large data (no join operation, only graph traversal)
- Schema-free, no relational schema
Our assumptions

- Data is "organized" / stored by approx. following some meta/ER model.
- Use of labels in order to "type" nodes and relationships
- Otherwise, we can't do anything!

[Elabri, in progress]
Learning from a Graph database

Our assumptions

- Data is "organized" / stored by approx. following some meta/ER model.
- Use of labels in order to "type" nodes and relationships
- Otherwise, we can’t do anything!

[Elabri, in progress]
Learning from a Graph database

Our assumptions

- Data is "organized" / stored by approx. following some meta/ER model.
- Use of labels in order to "type" nodes and relationships
- Otherwise, we can’t do anything!

[Elabri, in progress]
ER identification from data

- E = node labels, R = relationship labels
- choosing only the most frequent signature $(E_i \times E_j)$ for each $R$
DAPER learning

**ER identification from data**

- E = node labels, R = relationship labels
- Choosing only the most frequent signature \((E_i \times E_j)\) for each R

![Diagram of a Bayesian Network](image)

[Diagram showing a network with nodes labeled as Professor, Course, Student, and relationships Takes and Grade]

Philippe Leray

Advances in Learning with Bayesian Networks 40/45
DAPER learning

DAPER structure learning

Once ER model is identified, we can learn the probabilistic dependencies:

- **Attribute uncertainty**: predicting attribute value only
- **Reference uncertainty**: predicting the target node for an existing relation?
- **Existence uncertainty**: predicting a relationship between two existing nodes?
DAPER learning

DAPER structure learning

Once ER model is identified, we can learn the probabilistic dependencies:

- **Attribute uncertainty**: predicting attribute value only
- **Reference uncertainty**: predicting the target node for an existing relation?
- **Existence uncertainty**: predicting a relationship between two existing nodes?
DAPER learning

**DAPER structure learning**

Once ER model is identified, we can learn the probabilistic dependencies:

- **Attribute uncertainty**: predicting attribute value only
- **Reference uncertainty**: predicting the target node for an existing relation?
- **Existence uncertainty**: predicting a relationship between two existing nodes?
Outline ...

1. BN learning
   - Definition
   - Parameter learning
   - Structure learning

2. Dynamic BN learning
   - Definition
   - Learning

3. Relational BN learning
   - Definitions
   - Learning with a relational DB
   - Learning with a Graph DB

4. The end
Conclusion

Visible face of this talk

- Bayesian networks = powerful tool for knowledge representation and reasoning with data
- Our contributions about BN learning in several contexts

Todo list, in progress
Conclusion

Visible face of this talk

- Bayesian networks = powerful tool for knowledge representation and reasoning with data
- Our contributions about BN learning in several contexts

Todo list, in progress
Conclusion

Visible face of this talk

Todo list, in progress

- Dealing with all the problems in the same time :-) 
- Interacting with some probabilistic & logic frameworks 
- Implementation in our software platform PILGRIM
Conclusion

Visible face of this talk

Todo list, in progress
- Dealing with all the problems in the same time :-)
- Interacting with some probabilistic & logic frameworks
- Implementation in our software platform PILGRIM
Conclusion

Visible face of this talk

Todo list, in progress

- Dealing with all the problems in the same time :-)  
- Interacting with some probabilistic & logic frameworks  
- Implementation in our software platform PILGRIM
BN Learning wrt. 7 Vs of Big Data

- **Volume**: scalable algorithms and map-reduce implementations
- **Variety**: flat data, SQL, graph databases, ...
- **Velocity/Variability**: incremental anytime learning, non stationary data
- **Visualization**: for user interaction
- **Veracity**: does the user give accurate data?
- **Value**: of data...
References

**One starting point**


**Our publications**

- http://tinyurl.com/PhLeray

Thank you for your attention
References

One starting point

[Koller & Friedman, 09]

Our publications

http://tinyurl.com/PhLeray

Thank you for your attention