Advances in Learning with Bayesian Networks

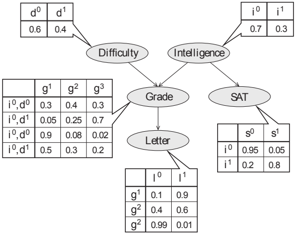
Philippe Leray
philippe.leray@univ-nantes.fr

DUKe (Data User Knowledge) research group, LINA UMR 6241, Nantes, France

Nantes Machine Learning Meetup, July 6, 2015, Nantes, France

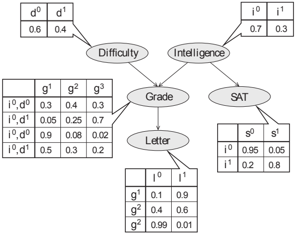
Motivations

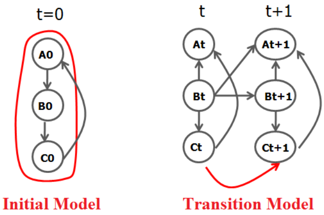
- Bayesian networks (BNs) are a powerful tool for graphical representation of the underlying knowledge in the data and reasoning with incomplete or imprecise observations.
- BNs have been extended (or generalized) in several ways, as for instance, causal BNs, dynamic BNs, relational BNs, ...



Motivations

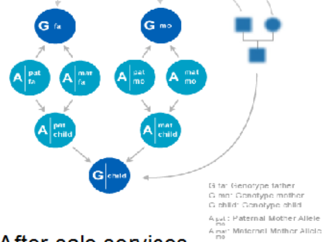
- Bayesian networks (BNs) are a powerful tool for graphical representation of the underlying knowledge in the data and reasoning with incomplete or imprecise observations.
- BNs have been extended (or generalized) in several ways, as for instance, causal BNs, dynamic BNs, relational BNs, ...



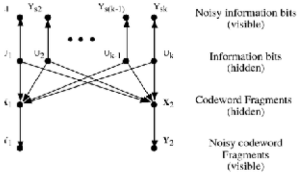


Motivations

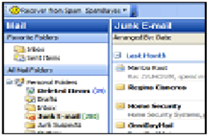
Victim identification system



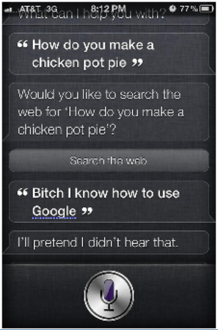
Turbo-codes (GSM, ...)



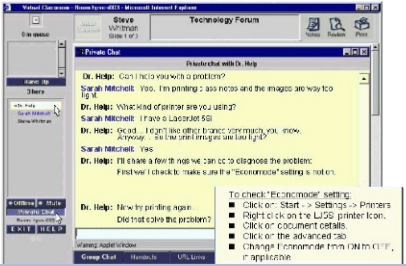
Anti Spam



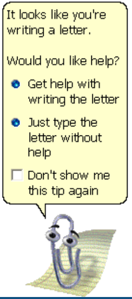
Assistant iPhone SIRI



After-sale services



MS Office assistant



Motivations

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

- complete

A	B	C	D
0	1	2	3
4	6	1	0
2	3	5	6
...			
...			
...			
1	3	2	6
3	8	9	0
1	2	4	5
1	4	3	7
8	5	4	3

Motivations

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

- complete /incomplete [François et al. 06]

A	B	C	D
0	1	2	3
4	?	1	0
2	3	5	?
	...		
	...		
	...		
1	3	?	6
3	8	9	0
1	2	4	?
1	?	3	7
8	5	4	3

Motivations

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

- complete /incomplete [François et al. 06]
- high n,
-
-
-
-

A	B	C	D	...	...	...	X_{100000}
0	1	2	3	...	...	...	7
4	6	1	0	...	...	...	5
2	3	5	6	...	...	...	4
			...				...
			...				...
			...				...
			...				...
1	3	2	6	...	...	...	7
3	8	9	0	...	...	...	1
1	2	4	5	...	...	...	3
1	4	3	7	...	...	...	2
8	5	4	3	...	...	...	4

Motivations

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

- complete /incomplete [François et al. 06]
- high n, $n >> p$ [Ammar & Leray, 11]

A	B	C	D	...	...	...	X_{100000}
0	1	2	3	...	...	...	7
4	6	1	0	...	...	...	5
2	3	5	6	...	...	...	4

Motivations

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

- complete /incomplete [François et al. 06]
- high n, $n >> p$ [Ammar & Leray, 11]
- stream [Yasin and Leray, 13]

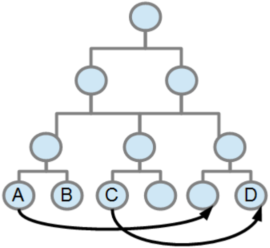
A	B	C	D	...	...	...	X_{100000}
0	1	2	3	...	...	...	7
4	6	1	0	...	...	...	5
2	3	5	6	...	...	...	4
			...				...
			...				...
			...				...
			...				...
1	3	2	6	...	...	...	7
3	8	9	0	...	...	...	1
1	2	4	5	...	...	...	3
1	4	3	7	...	...	...	2
8	5	4	3	...	...	...	4

Motivations

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

- complete /incomplete [François et al. 06]
- high n, $n >> p$ [Ammar & Leray, 11]
- stream [Yasin and Leray, 13]
- + prior knowledge / ontology [Ben Messaoud et al., 13]
-
-

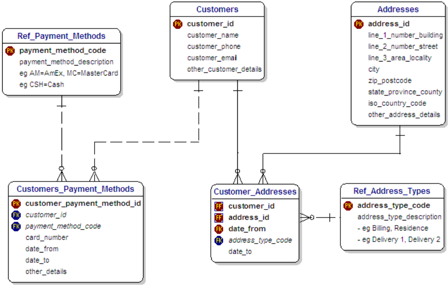
A	B	C	D
0	1	2	3
4	6	1	0
2	3	5	6
...			
...			
...			
...			
1	3	2	6
3	8	9	0
1	2	4	5
1	4	3	7
8	5	4	3



Motivations

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

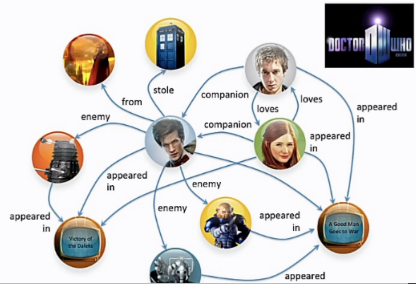
- complete /incomplete [François et al. 06]
- high n, $n >> p$ [Ammar & Leray, 11]
- stream [Yasin and Leray, 13]
- + prior knowledge / ontology [Ben Messaoud et al., 13]
- structured data [Ben Ishak, Coutant, Chulyadyo et al.]



Motivations

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

- complete /incomplete [François et al. 06]
- high n, $n >> p$ [Ammar & Leray, 11]
- stream [Yasin and Leray, 13]
- + prior knowledge / ontology [Ben Messaoud et al., 13]
- structured data [Ben Ishak, Coutant, Chulyadyo et al.]
- not so structured data [Elabri et al.]



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

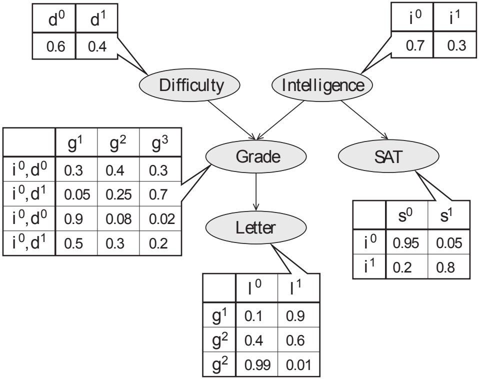
- how to learn BNs in such various contexts ?
- state of the art : founding algorithms and recent ones
- pointing out our contributions in this field

Outline ...

Bayesian network [Pearl, 1985]

Definition

- $\mathcal{G}$ qualitative description of conditional dependences / independences between variables
  directed acyclic graph (DAG)
- Θ quantitative description of these dependences
  conditional probability distributions (CPDs)



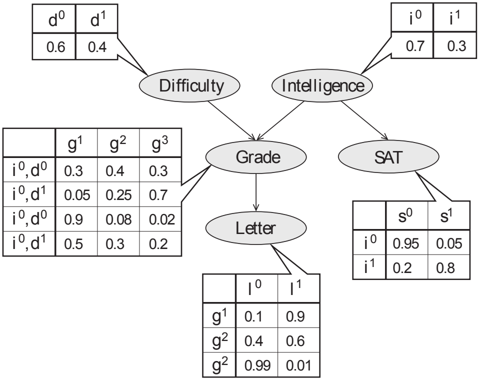
Main property

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



d^0	d^1
0.6	0.4

i^0	i^1
0.7	0.3

	g^1	g^2	g^3
i^0,d^0	0.3	0.4	0.3
i^0,d^1	0.05	0.25	0.7
i^0,d^0	0.9	0.08	0.02
i^0,d^1	0.5	0.3	0.2

	s^0	s^1
i^0	0.95	0.05
i^1	0.2	0.8

	l^0	l^1
g^1	0.1	0.9
g^2	0.4	0.6
g^2	0.99	0.01

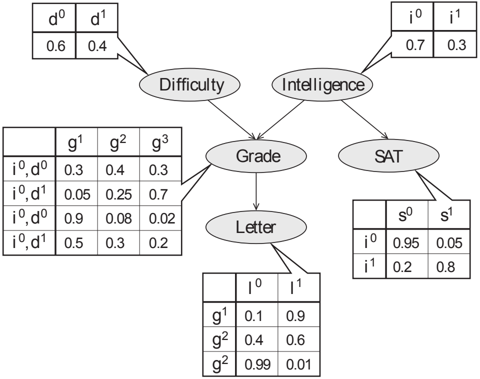
Main property

- 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 ⊖

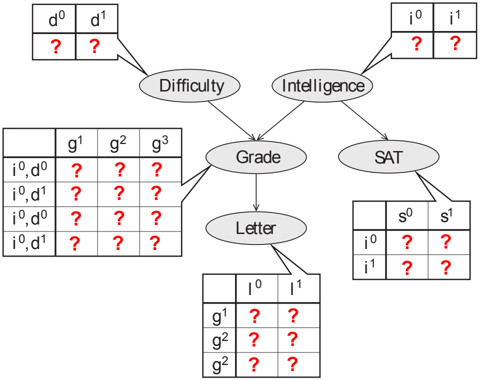
- parameter learning / G given
- structure learning



One model... but two learning tasks

BN = graph G and set of CPDs ⊖

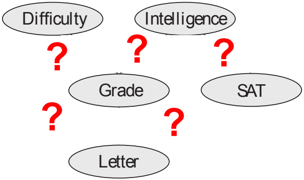
- parameter learning / G given
- structure learning



One model... but two learning tasks

BN = graph G and set of CPDs Θ

- parameter learning / G given
- structure learning



Parameter learning (generative)

Complete data $\mathcal{D}$

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

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

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

Other approaches $P(\theta) \sim$ Dirichlet(α)

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

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

Parameter learning (generative)

Complete data $\mathcal{D}$

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

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

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

Other approaches $P(\theta) \sim$ Dirichlet(α)

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

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

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

$$\theta_{i,j,k} = \frac{N^{old}\theta^{old}_{i,j,k} + N_{i,j,k}}{N^{old} + N}$$

- 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

$$\theta_{i,j,k} = \frac{N^{old}\theta^{old}_{i,j,k} + N_{i,j,k}}{N^{old} + N}$$

- 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 :-(

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]

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

- 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]

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

- 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)

Structure learning (generative / complete)

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
  $\Rightarrow$ 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 $n < 1000$)

Hybrid/ local search methods

Structure learning (generative / complete)

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 $n < 1000$)

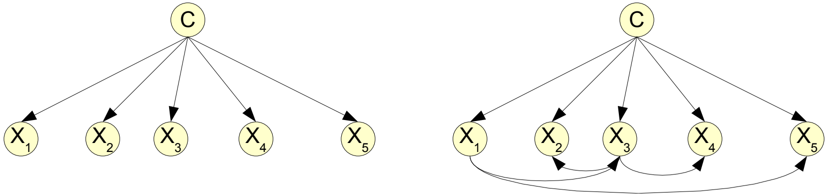
Hybrid/ local search methods

- 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
- ...



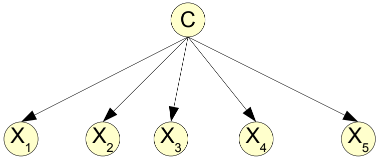
Structure learning

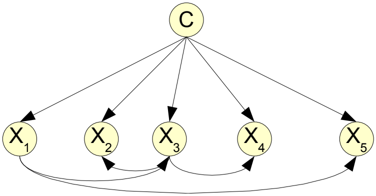
- usually, the structure is learned in a generative way
- the parameters are then tuned in a discriminative way

Structure learning (discriminative)

Specific structures

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





Structure learning

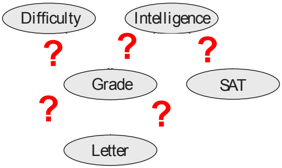
- usually, the structure is learned in a generative way
- the parameters are then tuned in a discriminative way

Structure learning

Incomplete data

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

A	B	C	D
0	1	2	3
4	?	1	0
2	3	5	?
	...		
	...		
	...		
1	3	?	6
3	8	9	0
1	2	4	?
1	?	3	7
8	5	4	3



Structure learning

$n >> p$

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

A	B	C	D	...	...	...	X_{100000}
0	1	2	3	...	...	...	7
4	6	1	0	...	...	...	5
2	3	5	6	...	...	...	4

Structure learning

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]

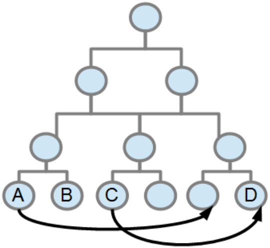
A	B	C	D	...	...	...	X_{100000}
0	1	2	3	...	...	...	7
4	6	1	0	...	...	...	5
2	3	5	6	...	...	...	4
			...				...
			...				...
			...				...
			...				...
1	3	2	6	...	...	...	7
3	8	9	0	...	...	...	1
1	2	4	5	...	...	...	3
1	4	3	7	...	...	...	2
8	5	4	3	...	...	...	4

Structure learning

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]

A	B	C	D
0	1	2	3
4	6	1	0
2	3	5	6
	...		
	...		
	...		
	...		
1	3	2	6
3	8	9	0
1	2	4	5
1	4	3	7
8	5	4	3

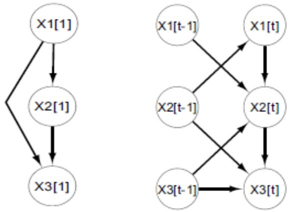


Outline ...

Dynamic Bayesian networks (DBNs)

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

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



(a) Prior network (b) Transition network

Simplified k-TBN

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

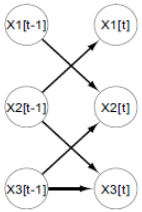
Dynamic Bayesian networks (DBNs)

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

- $k-1$ 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]



(c) Transition network with only inter time-slice arcs

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 (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 "unsimplified" 2-TBNs with high n [Trabelsi et al., 13]

DBN structure learning (generative)

Score-based methods

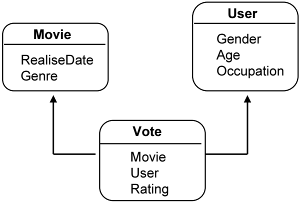
- dynamic Greedy Search [Friedman et al., 98], genetic algorithm [Gao et al., 07], dynamic Simulated Annealing [Hartemink, 05], ...
- for k-TBN (G_0 and $G_{\rightarrow}$ learning)
- but not scalable (high n)

Hybrid methods

- [Dojer, 06] [Vinh et al., 12] for simplified k-TBN, but often limited to $k = 2$ for scalability
- dynamic MMHC for "unsimplified" 2-TBNs with high n [Trabelsi et al., 13]

Outline ...

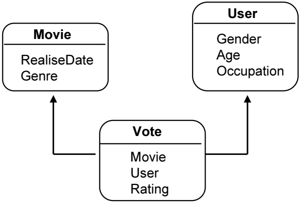
Relational schema



A relational schema $\mathcal{R}$

- classes + relational variables
- reference slots (e.g., *Vote.Movie*, *Vote.User*)
- slot chain = a sequence of reference slots
  - allow to walk in the relational schema to create new variables
  - ex: $Vote.User.User^{-1}.Movie$: all the movies voted by a particular user

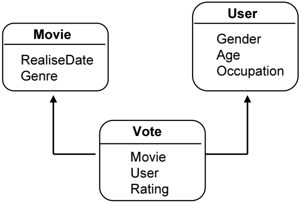
Relational schema



A relational schema $\mathcal{R}$

- classes + relational variables
- reference slots (e.g., $Vote.Movie$, $Vote.User$)
- slot chain = a sequence of reference slots
  - allow to walk in the relational schema to create new variables
  - ex: $Vote.User.User^{-1}.Movie$: all the movies voted by a particular user

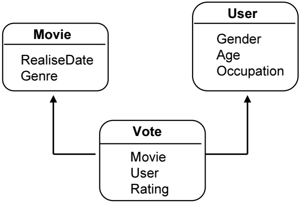
Relational schema



A relational schema $\mathcal{R}$

- classes + relational variables
- reference slots (e.g., *Vote.Movie*, *Vote.User*)
- slot chain = a sequence of reference slots
  - allow to walk in the relational schema to create new variables
  - ex: $Vote.User.User^{-1}.Movie$: all the movies voted by a particular user

Relational schema



A relational schema $\mathcal{R}$

- classes + relational variables
- reference slots (e.g., $Vote.Movie$, $Vote.User$)
- slot chain = a sequence of reference slots
  - allow to walk in the relational schema to create new variables
  - ex: $Vote.User.User^{-1}.Movie$: all the movies voted by a particular user

Probabilistic Relational Models

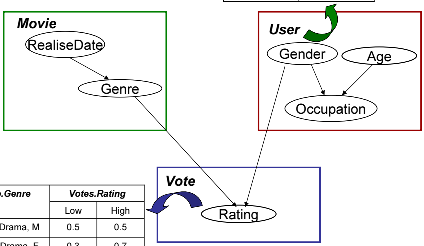
[Koller & Pfeffer, 98]

Definition

A PRM Π associated to $\mathcal{R}$:

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

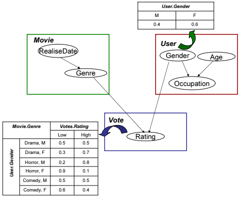
User.Gender	
M	F
0.4	0.6



Movie.Genre		Votes.Rating	
		Low	High
User.Gender	Drama, M	0.5	0.5
	Drama, F	0.3	0.7
	Horror, M	0.2	0.8
	Horror, F	0.9	0.1
	Comedy, M	0.5	0.5
	Comedy, F	0.6	0.4

Probabilistic Relational Models

Definition



Aggregators

- $Vote.User.User^{-1}.Movie.genre \rightarrow Vote.rating$
- 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. $\gamma = MODE$

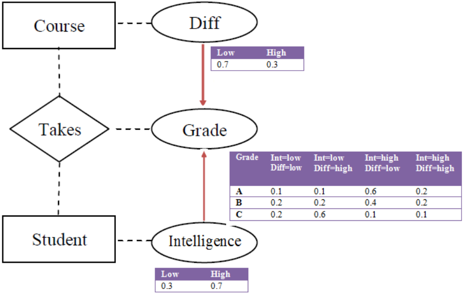
DAPER

Another probabilistic relational model

[Heckerman & Meek, 04]

Definition

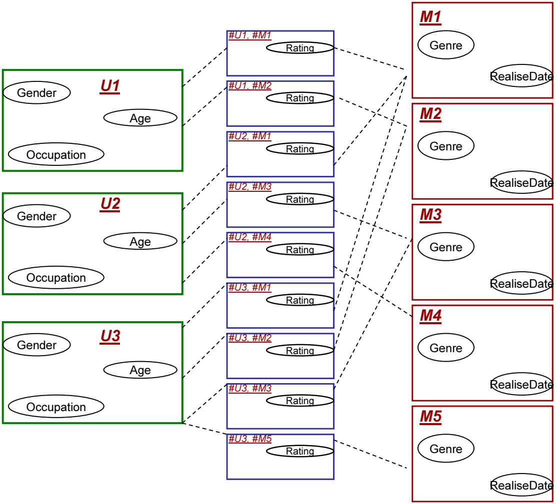
Probabilistic model associated to an Entity-Relationship model



Learning from a relational datatase

GBN

- PRM/DAPER
  learning = finding the probabilistic dependencies and the probability tables from an instantiated database
- the relational schema/ER model is given



PRM/DAPER structure learning

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 structure learning

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 structure learning

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 structure learning

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

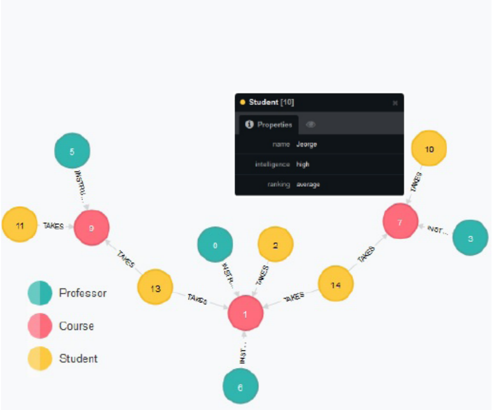
Score-based methods

- Greedy search [Getoor et al., 07]

Hybrid methods

- relational MMHC [Ben Ishak et al., 15]

Graph database



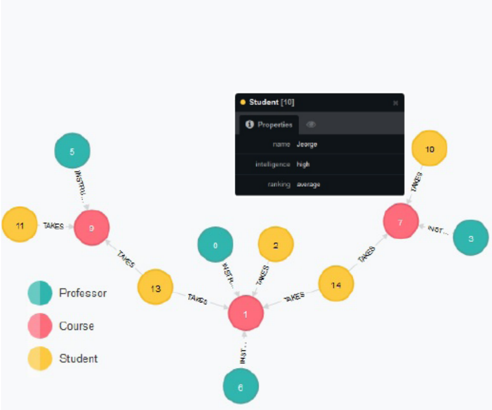
Definition

- Data is organized as a graph, with "labelled" nodes and relationships
- Attributes can be associated to both.
- Seems nice for ER model but ...

Schema-free

Elabri et al., in progress

Graph database



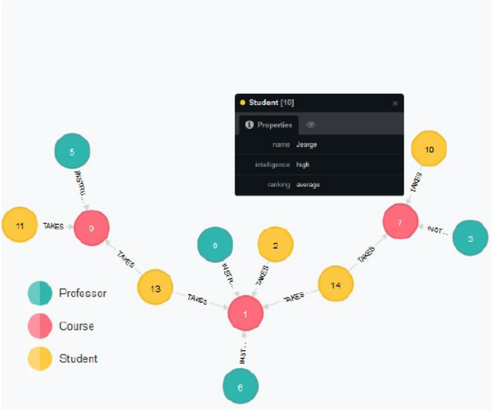
Definition

- Data is organized as a graph, with "labelled" nodes and relationships
- Attributes can be associated to both.
- Seems nice for ER model but ...

Schema-free

Elabri et al., in progress

Graph database



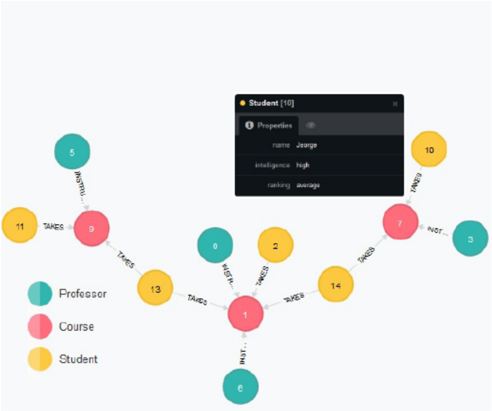
Definition

- Data is organized as a graph, with "labelled" nodes and relationships
- Attributes can be associated to both.
- Seems nice for ER model but ...

Schema-free

Elabri et al., in progress

Graph database



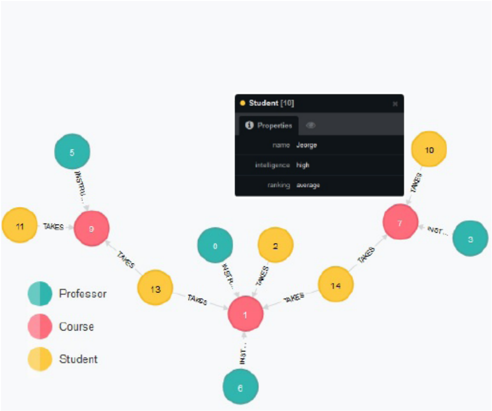
Definition

Schema-free

- Only data, no "relational schema"
- No warranty that the data has been "stored" by following some meta/ER model.

Elabri et al., in progress

Graph database



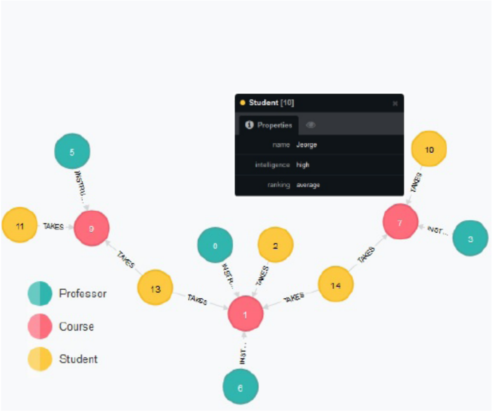
Definition

Schema-free

- Only data, no "relational schema"
- No warranty that the data has been "stored" by following some meta/ER model.

Elabri et al., in progress

Graph database



Definition

Schema-free

Elabri et al., in progress

- Learning a probabilistic relational model from a graph DB
- Extension to Markov Logic Networks

Outline ...

Conclusion

Visible face of this talk

- BNs = powerful tool for knowledge representation and reasoning

⇒ interest in using structure learning algorithms for knowledge discovery

- BN structure learning is NP-hard, even for "usual" BN/data
- but we want to learn more and more complex models with more and more complex data

⇒ many works in progress in order to develop such learning algorithms

Hidden face of this talk

Conclusion

Visible face of this talk

- BNs = powerful tool for knowledge representation and reasoning

$\Rightarrow$ interest in using structure learning algorithms for knowledge discovery

- BN structure learning is NP-hard, even for "usual" BN/data
- but we want to learn more and more complex models with more and more complex data

$\Rightarrow$ many works in progress in order to develop such learning algorithms

Hidden face of this talk

Conclusion

Visible face of this talk

Hidden face of this talk

- BN learning and causality : causal discovery
- BN learning tools : no unified programming tools, often limited to simple BN models / simple data

⇒ coming soon : PILGRIM our GPL platform in C++, dealing with BN, DBN, RBN, incremental data, ...

- BN versus other probabilistic graphical models : Qualitative probabilistic models, Markov random fields, Conditional random fields, Deep belief networks, ...

Conclusion

Visible face of this talk

Hidden face of this talk

- BN learning and causality : causal discovery
- BN learning tools : no unified programming tools, often limited to simple BN models / simple data

⇒ coming soon : PILGRIM our GPL platform in C++, dealing with BN, DBN, RBN, incremental data, ...

- BN versus other probabilistic graphical models : Qualitative probabilistic models, Markov random fields, Conditional random fields, Deep belief networks, ...

Conclusion

Visible face of this talk

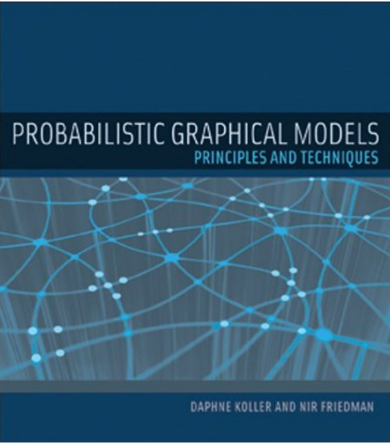
Hidden face of this talk

- BN learning and causality : causal discovery
- BN learning tools : no unified programming tools, often limited to simple BN models / simple data

$\Rightarrow$ coming soon : PILGRIM our GPL platform in C++, dealing with BN, DBN, RBN, incremental data, ...

- BN versus other probabilistic graphical models : Qualitative probabilistic models, Markov random fields, Conditional random fields, Deep belief networks, ...

References



One starting point

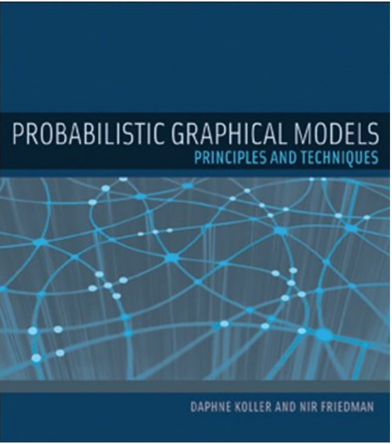
- [Koller & Friedman, 09] Probabilistic Graphical Models: Principles and Techniques. MIT Press.

Our publications

- `http://tinyurl.com/PhLeray`

Thank you for your attention

References



One starting point

- [Koller & Friedman, 09] Probabilistic Graphical Models: Principles and Techniques. MIT Press.

Our publications

- `http://tinyurl.com/PhLeray`

Thank you for your attention