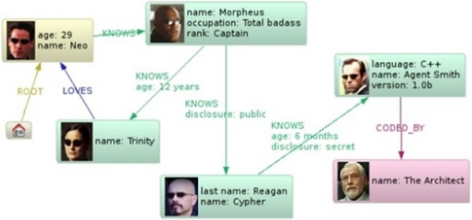
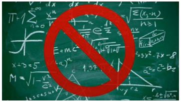
Learning Bayesian Networks with Graph Databases

Philippe Leray philippe.leray@univ-nantes.fr

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

Nantes Machine Learning Meetup, April 4, 2016

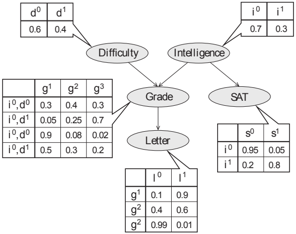




Outline ...

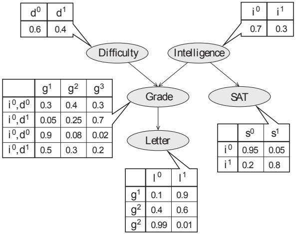
Motivations

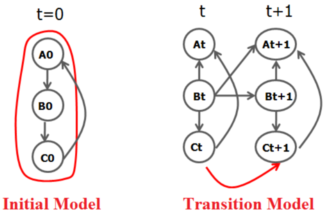
- Bayesian networks (BNs) are a powerful tool for graphical representation of the underlying knowledge in the data and reasoning with incomplete or imprecise observations.
- BNs have been extended (or generalized) in several ways, as for instance, causal BNs, dynamic BNs, relational BNs, ...



Motivations

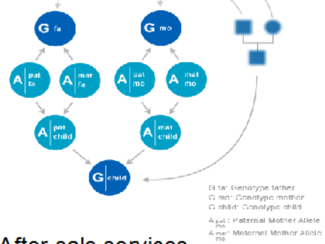
- Bayesian networks (BNs) are a powerful tool for graphical representation of the underlying knowledge in the data and reasoning with incomplete or imprecise observations.
- BNs have been extended (or generalized) in several ways, as for instance, causal BNs, dynamic BNs, relational BNs, ...



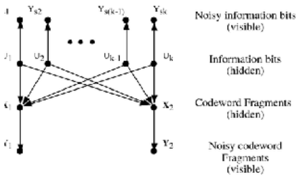


Motivations

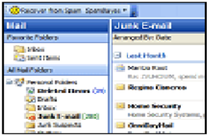
Victim identification system



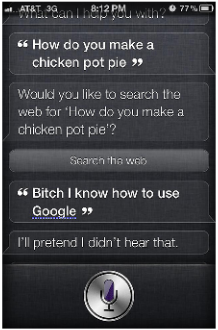
Turbo-codes (GSM, ...)



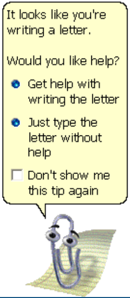
Anti Spam



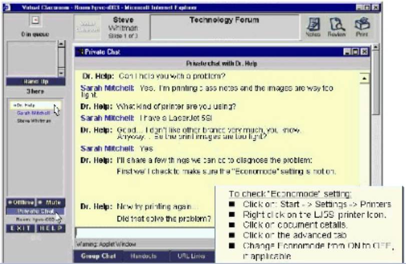
Assistant iPhone SIRI



MS Office assistant



After-sale services



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]

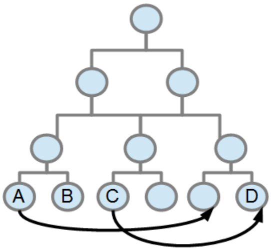
A	B	C	D	...	...	...	X_{100000}
0	1	2	3	...	...	...	7
4	6	1	0	...	...	...	5
2	3	5	6	...	...	...	4
			...				...
			...				...
			...				...
			...				...
1	3	2	6	...	...	...	7
3	8	9	0	...	...	...	1
1	2	4	5	...	...	...	3
1	4	3	7	...	...	...	2
8	5	4	3	...	...	...	4

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]

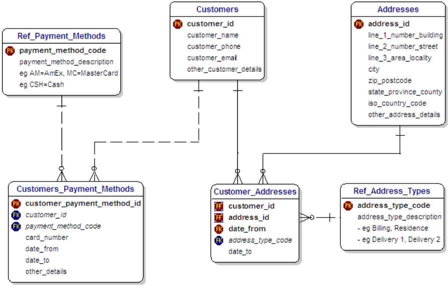
A	B	C	D
0	1	2	3
4	6	1	0
2	3	5	6
	...		
	...		
	...		
	...		
1	3	2	6
3	8	9	0
1	2	4	5
1	4	3	7
8	5	4	3



Motivations

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

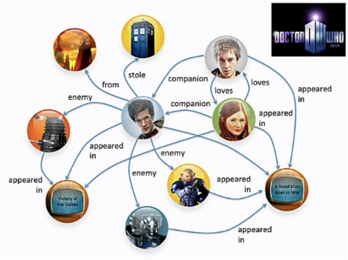
- complete /incomplete [François et al., 06]
- high n, $n >> p$ [Ammar & Leray, 11]
- stream [Yasin and Leray, 13]
- + prior knowledge / ontology [Ben Messaoud et al., 13]
- structured data [Ben Ishak, 15, Coutant, 15, Chulyadyo et al.]
-



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, 15, Coutant, 15, Chulyadyo et al.]
- not so structured data [Elabri et al.]



Motivations

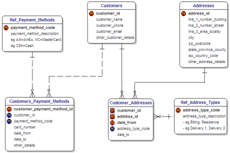
Flat data

- No relational model
- Learning probabilistic dependencies between variables

Motivations

Flat data

- No relational model
- Learning probabilistic dependencies between variables



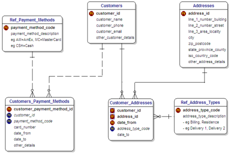
Relational DB

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

Motivations

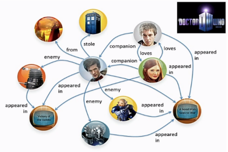
Flat data

- No relational model
- Learning probabilistic dependencies between variables



Relational DB

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

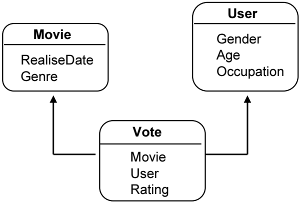


Graph DB

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

Outline ...

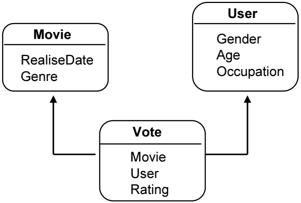
Relational schema



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

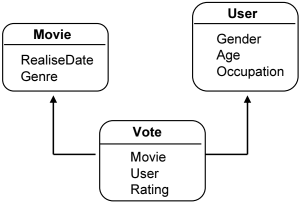
Relational schema



relational schema $\mathcal{R}$

- classes + attributes
- reference slots (e.g. $Vote.Movie$, $Vote.User$)
- 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

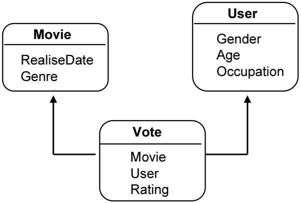
Relational schema



relational schema $\mathcal{R}$

- classes + attributes
- reference slots (e.g. $Vote.Movie$, $Vote.User$)
- 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

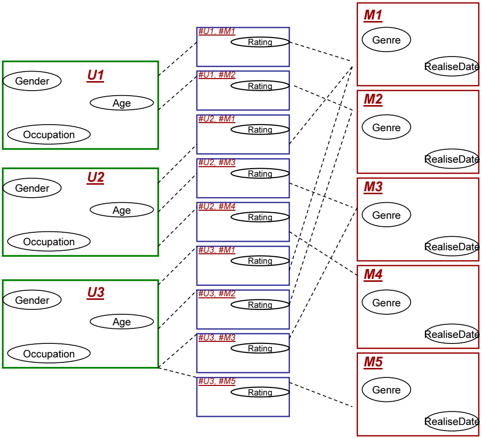
Relational schema



relational schema $\mathcal{R}$

- classes + attributes
- reference slots (e.g. $Vote.Movie$, $Vote.User$)
- 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 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 $\sigma_{\mathcal{R}}$

- Instance without attribute values

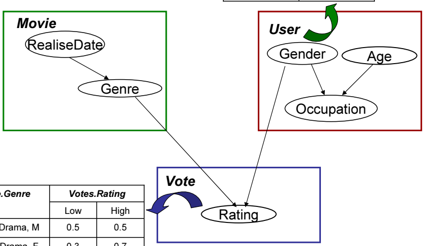
Probabilistic Relational Models

[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 $\theta_{\mathcal{S}}$

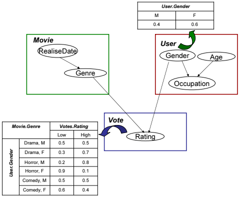


User.Gender	
M	F
0.4	0.6

Movie.Genre, User.Gender	Votes.Rating Low	Votes.Rating High
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



User.Gender	
M	F
0.4	0.6

	Movie.Genre	Votes.Rating Low	Votes.Rating 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

Aggregators

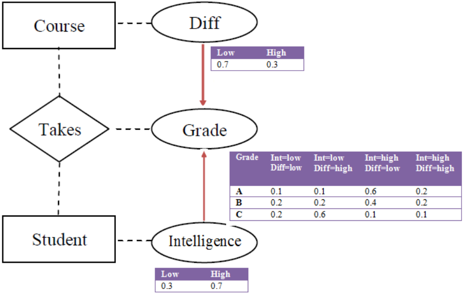
- $Vote.User.User^{-1}.Movie.genre \rightarrow Vote.rating$
- movie rating from one user can be dependent with the genre of all the movies voted by this user
  - how to describe the dependency with an unknown number of parents ?
  - solution : using an aggregated value, e.g. $\gamma = MODE$

DAPER

Another probabilistic relational model *[Heckerman & Meek, 04]*

Definition

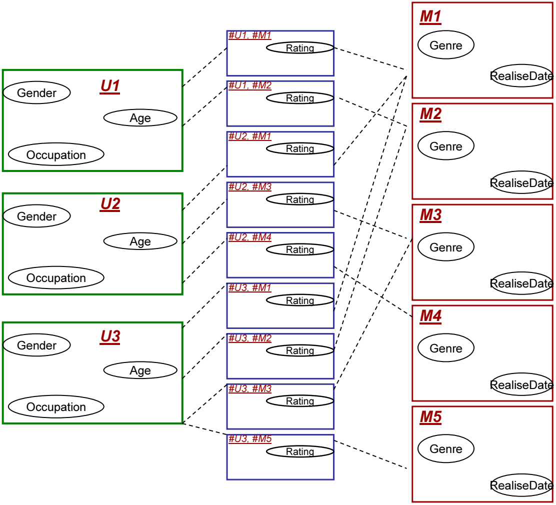
- Probabilistic model associated to an Entity-Relationship model
- Classes = { Entity classes + Relationship classes }



Learning from a relational datatase

PRM/DAPER learning

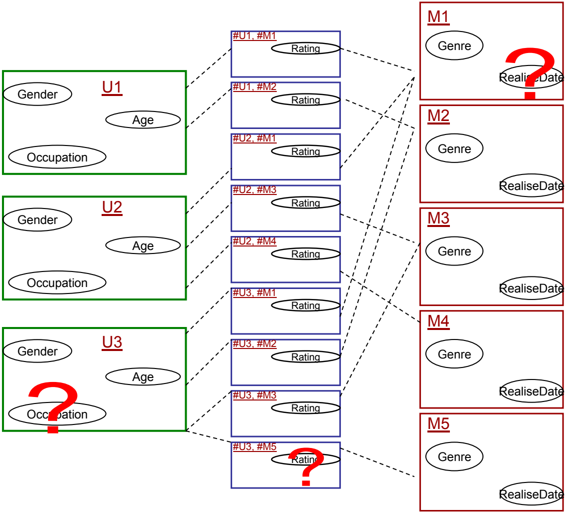
- finding the probabilistic dependencies and the probability tables from an instantiated database
- relational schema is known, but ...
- several situations / PRM extensions



Learning from a relational datatase

Attribute uncertainty

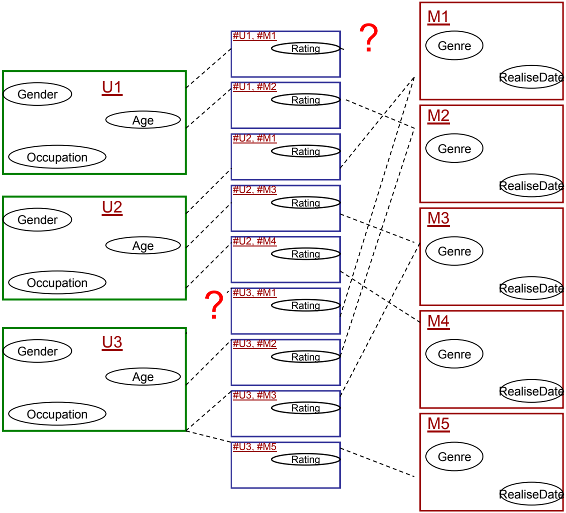
- Input : relational skeleton (all the objects and relations), some attributes
- Objective : predict only missing attributes



Learning from a relational datatase

Reference uncertainty

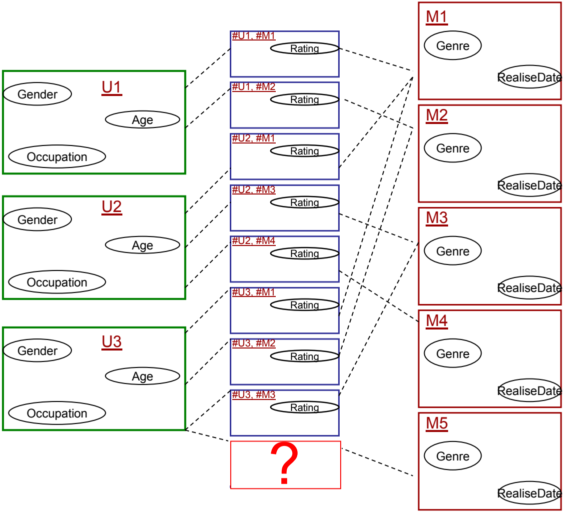
- Input : partial relational skeleton (all the objects, but some relations are missing)
- Objective : predict missing attributes and "foreign keys"



Learning from a relational datatase

Existence uncertainty

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



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]

PRM/DAPER learning with RU

Need for partitionning

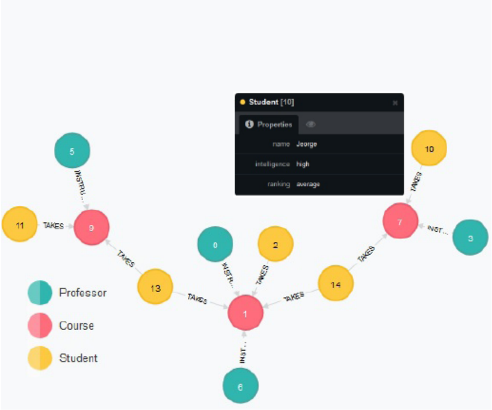
- The missing foreign key is considered as a random variable
- We need to partition the similar "target" objects in order to obtain a generic model

How to partition

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

Outline ...

Graph database

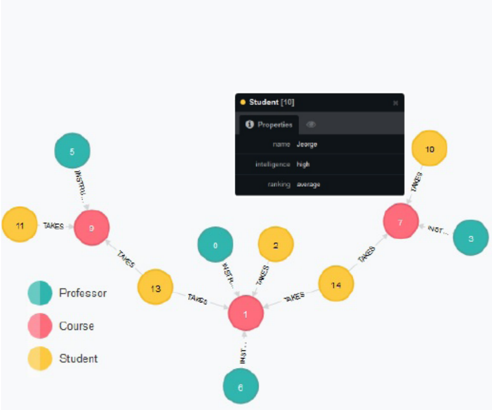


Definition

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

Properties

Graph database

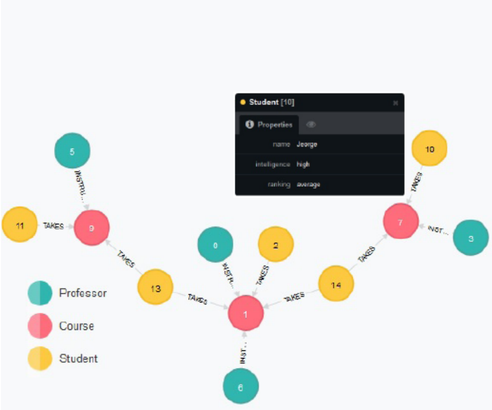


Definition

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

Properties

Graph database

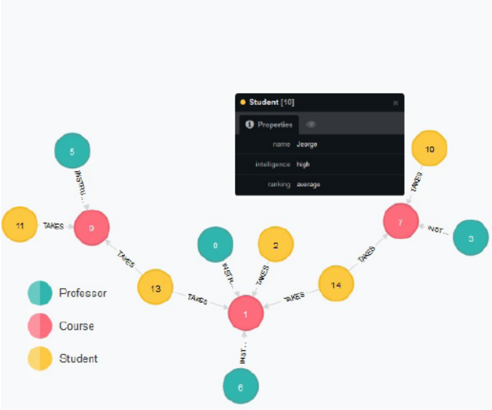


Definition

Properties

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

Graph database

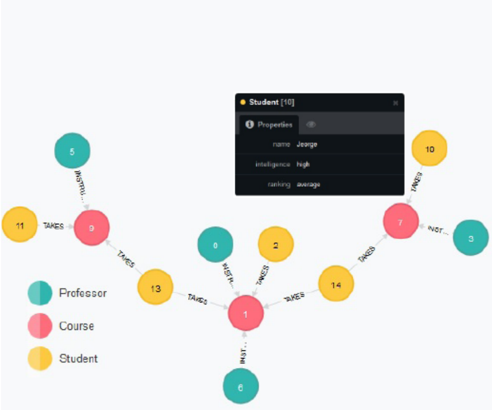


Definition

Properties

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

Learning from a Graph database

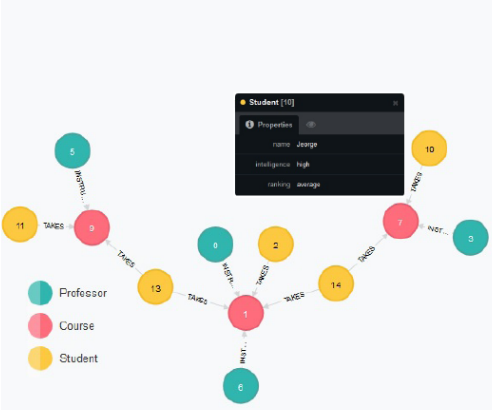


Our assumptions

- 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

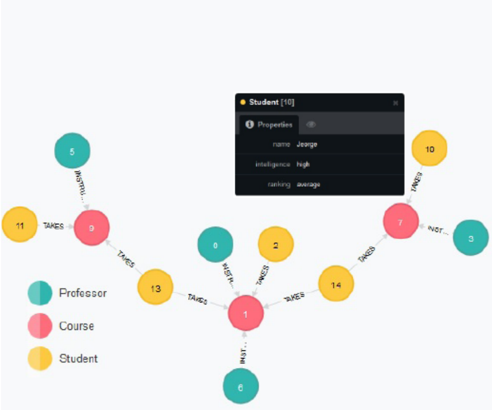


Our assumptions

- 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]

Learning from a Graph database



Our assumptions

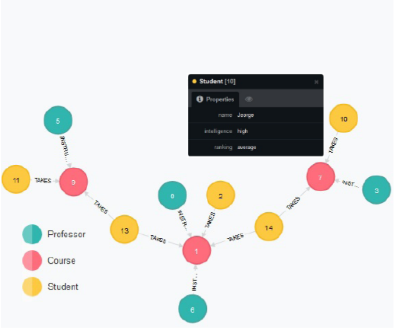
- Data is "organized" / stored by approx. following some meta/ER model.
- Use of labels in order to "type" nodes and relationships
- Otherwise, we can't do anything !

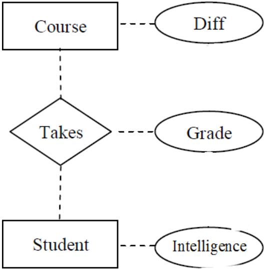
[Elabri, in progress]

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

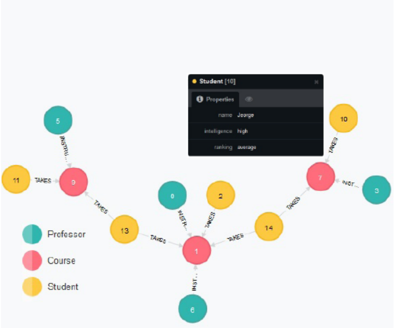


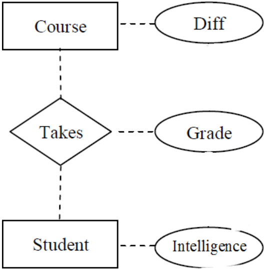


DAPER learning

ER identification from data

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





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 ?

Another track ?

Markov Logic Networks [Richardson & Domingos, 06]

- Yet another probabilistic relational model
- Relations and probabilistic dependencies are described with First Order Logic (clauses)
- Uncertainty is represented by weights over logic formulas (that can contradict themselves)

English	First-Order Logic	Clausal Form	Weight
Friends of friends are friends.	$\forall x \forall y \forall z\ \mathtt{Fr}(x,y) \wedge \mathtt{Fr}(y,z) \Rightarrow \mathtt{Fr}(x,z)$	$\neg\mathtt{Fr}(x,y) \vee \neg\mathtt{Fr}(y,z) \vee \mathtt{Fr}(x,z)$	0.7
Friendless people smoke.	$\forall x\ (\neg(\exists y\ \mathtt{Fr}(x,y)) \Rightarrow \mathtt{Sm}(x))$	$\mathtt{Fr}(x, g(x)) \vee \mathtt{Sm}(x)$	2.3
Smoking causes cancer.	$\forall x\ \mathtt{Sm}(x) \Rightarrow \mathtt{Ca}(x)$	$\neg\mathtt{Sm}(x) \vee \mathtt{Ca}(x)$	1.5
If two people are friends, either	$\forall x \forall y\ \mathtt{Fr}(x,y) \Rightarrow (\mathtt{Sm}(x) \Leftrightarrow \mathtt{Sm}(y))$	$\neg\mathtt{Fr}(x,y) \vee \mathtt{Sm}(x) \vee \neg\mathtt{Sm}(y)$,	1.1
both smoke or neither does.		$\neg\mathtt{Fr}(x,y) \vee \neg\mathtt{Sm}(x) \vee \mathtt{Sm}(y)$	1.1

Another track ?

Markov Logic Networks [Richardson & Domingos, 06]

- Yet another probabilistic relational model
- Relations and probabilistic dependencies are described with First Order Logic (clauses)
- Uncertainty is represented by weights over logic formulas (that can contradict themselves)

English	First-Order Logic	Clausal Form	Weight
Friends of friends are friends.	$\forall x \forall y \forall z\, \mathtt{Fr}(x,y) \wedge \mathtt{Fr}(y,z) \Rightarrow \mathtt{Fr}(x,z)$	$\neg\mathtt{Fr}(x,y) \vee \neg\mathtt{Fr}(y,z) \vee \mathtt{Fr}(x,z)$	0.7
Friendless people smoke.	$\forall x\, (\neg(\exists y\, \mathtt{Fr}(x,y)) \Rightarrow \mathtt{Sm}(x))$	$\mathtt{Fr}(x, g(x)) \vee \mathtt{Sm}(x)$	2.3
Smoking causes cancer.	$\forall x\, \mathtt{Sm}(x) \Rightarrow \mathtt{Ca}(x)$	$\neg\mathtt{Sm}(x) \vee \mathtt{Ca}(x)$	1.5
If two people are friends, either	$\forall x \forall y\, \mathtt{Fr}(x,y) \Rightarrow (\mathtt{Sm}(x) \Leftrightarrow \mathtt{Sm}(y))$	$\neg\mathtt{Fr}(x,y) \vee \mathtt{Sm}(x) \vee \neg\mathtt{Sm}(y)$,	1.1
both smoke or neither does.		$\neg\mathtt{Fr}(x,y) \vee \neg\mathtt{Sm}(x) \vee \mathtt{Sm}(y)$	1.1

Another track ?

Markov Logic Networks [Richardson & Domingos, 06]

- Yet another probabilistic relational model
- Relations and probabilistic dependencies are described with First Order Logic (clauses)
- Uncertainty is represented by weights over logic formulas (that can contradict themselves)

English	First-Order Logic	Clausal Form	Weight
Friends of friends are friends.	$\forall x \forall y \forall z\ \mathtt{Fr}(x,y) \wedge \mathtt{Fr}(y,z) \Rightarrow \mathtt{Fr}(x,z)$	$\neg\mathtt{Fr}(x,y) \vee \neg\mathtt{Fr}(y,z) \vee \mathtt{Fr}(x,z)$	0.7
Friendless people smoke.	$\forall x\ (\neg(\exists y\ \mathtt{Fr}(x,y)) \Rightarrow \mathtt{Sm}(x))$	$\mathtt{Fr}(x, g(x)) \vee \mathtt{Sm}(x)$	2.3
Smoking causes cancer.	$\forall x\ \mathtt{Sm}(x) \Rightarrow \mathtt{Ca}(x)$	$\neg\mathtt{Sm}(x) \vee \mathtt{Ca}(x)$	1.5
If two people are friends, either	$\forall x \forall y\ \mathtt{Fr}(x,y) \Rightarrow (\mathtt{Sm}(x) \Leftrightarrow \mathtt{Sm}(y))$	$\neg\mathtt{Fr}(x,y) \vee \mathtt{Sm}(x) \vee \neg\mathtt{Sm}(y)$,	1.1
both smoke or neither does.		$\neg\mathtt{Fr}(x,y) \vee \neg\mathtt{Sm}(x) \vee \mathtt{Sm}(y)$	1.1

Learning MLN from graph DB ?

Pros

- MLNs can handle multiple relationship signatures or exceptions
- MLN structure learning can deal in the same time with "relational schema" and probabilistic dependencies identification

Cons

Learning MLN from graph DB ?

Pros

- MLNs can handle multiple relationship signatures or exceptions
- MLN structure learning can deal in the same time with "relational schema" and probabilistic dependencies identification

Cons

- MLN structure learning complexity is huge

Outline ...

Conclusion

Visible face of this talk

- Probabilistic Relational Models = powerful tool for knowledge representation and reasoning with relational data
- Our proposition about PRM/DAPER learning from graph databases

Todo list, in progress

Conclusion

Visible face of this talk

- Probabilistic Relational Models = powerful tool for knowledge representation and reasoning with relational data
- Our proposition about PRM/DAPER learning from graph databases

Todo list, in progress

Conclusion

Visible face of this talk

Todo list, in progress

- Experimental comparison about PRM/DAPER learning from relational and graph databases
- and comparison with MLN learning ... expressivity power vs. complexity !

⇒ Implementation in our software platform PILGRIM

Conclusion

Visible face of this talk

Todo list, in progress

- Experimental comparison about PRM/DAPER learning from relational and graph databases
- and comparison with MLN learning ... expressivity power vs. complexity !

⇒ Implementation in our software platform PILGRIM

Conclusion

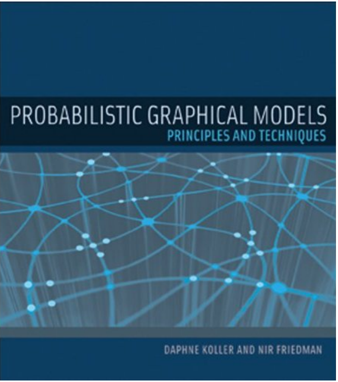
Visible face of this talk

Todo list, in progress

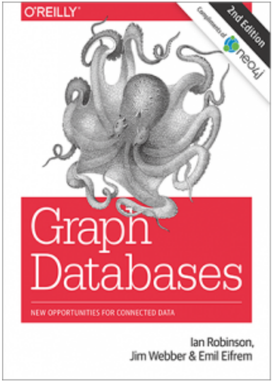
- Experimental comparison about PRM/DAPER learning from relational and graph databases
- and comparison with MLN learning ... expressivity power vs. complexity !

$\Rightarrow$ Implementation in our software platform PILGRIM

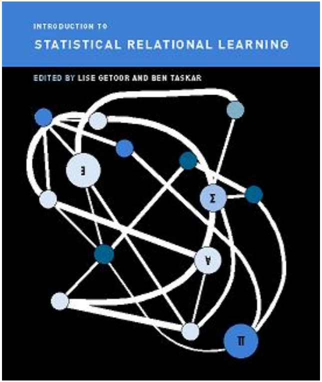
References



[Koller & Friedman, 09] MIT Press



[Robinson et al. 13] O'Reilly



[Getoor & Taskar, 07] MIT Press

Our publications

- http://tinyurl.com/PhLeray