Data acquisition, extraction, and storage

Probabilistic Databases

Pierre Senellart

ÉCOLE NORMALE SUPÉRIEURE | PSL

13 December 2024

The material in these slides has been designed in large part by Antoine Amarilli and Silviu Maniu.

Outline

Possible worlds

Remember from class on data provenance:

- Fix a finite set of possible tuples of same arity
- A possible world: a subset of the possible tuples

(Finite) uncertain/incomplete relation: set of possible worlds (see also class on incomplete databases)

Possible worlds

Remember from class on data provenance:

- Fix a finite set of possible tuples of same arity
- A possible world: a subset of the possible tuples

(Finite) uncertain/incomplete relation: set of possible worlds (see also class on incomplete databases)

U_1

date	teacher	room
04	Michaël	S08
04	Pierre	S08
04	Pierre	S06
11	Michaël	S08
11	Michaël	S06
11	Pierre	S08

U_2

date	teacher	room
04	Michaël	S08
04	Pierre	S08
04	Pierre	S06
11	Michaël	S08
11	Michaël	S06
11	Pierre	S08

Probabilistic instances

- Support $\mathcal{U}$: uncertain relation

Probabilistic instances

- Support $\mathcal{U}$: uncertain relation
- Probability distribution π on $\mathcal{U}$:

Probabilistic instances

- Support $\mathcal{U}$: uncertain relation
- Probability distribution π on $\mathcal{U}$:
  - Function from $\mathcal{U}$ to reals in $[0, 1]$
  - It must sum up to 1: $\sum_{I \in \mathcal{U}} \pi(I) = 1$

Probabilistic instances

- Support $\mathcal{U}$: uncertain relation
- Probability distribution π on $\mathcal{U}$:
  - Function from $\mathcal{U}$ to reals in $[0, 1]$
  - It must sum up to 1: $\sum_{I \in \mathcal{U}} \pi(I) = 1$

U_1

date	teacher	room
04	Michaël	S08
04	Pierre	S08
04	Pierre	S06
11	Michaël	S08
11	Michaël	S06
11	Pierre	S08

U_2

date	teacher	room
04	Michaël	S08
04	Pierre	S08
04	Pierre	S06
11	Michaël	S08
11	Michaël	S06
11	Pierre	S08

Probabilistic instances

- Support $\mathcal{U}$: uncertain relation
- Probability distribution π on $\mathcal{U}$:
  - Function from $\mathcal{U}$ to reals in $[0,1]$
  - It must sum up to 1: $\sum_{I \in \mathcal{U}} \pi(I) = 1$

U_1

date	teacher	room
04	Michaël	S08
04	Pierre	S08
04	Pierre	S06
11	Michaël	S08
11	Michaël	S06
11	Pierre	S08

$\pi(U_1) = 0.8$

U_2

date	teacher	room
04	Michaël	S08
04	Pierre	S08
04	Pierre	S06
11	Michaël	S08
11	Michaël	S06
11	Pierre	S08

$\pi(U_2) = 0.2$

Relational algebra on uncertain instances

- Extend relational algebra operators to uncertain instances
- The possible worlds of the result should be...
  - take all possible worlds in the supports of the inputs
  - apply the operation and get the possible outputs

Relational algebra on uncertain instances

- Extend relational algebra operators to uncertain instances
- The possible worlds of the result should be...
  - take all possible worlds in the supports of the inputs
  - apply the operation and get the possible outputs

U_1

04	P.	S08
11	P.	S06

U_2

11	M.	S08

Relational algebra on uncertain instances

- Extend relational algebra operators to uncertain instances
- The possible worlds of the result should be...
  - take all possible worlds in the supports of the inputs
  - apply the operation and get the possible outputs

U_1

04	P.	S08
11	P.	S06

$\cup$

U_2

11	M.	S08

Relational algebra on uncertain instances

- Extend relational algebra operators to uncertain instances
- The possible worlds of the result should be...
  - take all possible worlds in the supports of the inputs
  - apply the operation and get the possible outputs

U_1

04	P.	S08
11	P.	S06

U_2

11	M.	S08

$\bigcup$

V_1

V_2

11	M.	S08

Relational algebra on uncertain instances

- Extend relational algebra operators to uncertain instances
- The possible worlds of the result should be...
  - take all possible worlds in the supports of the inputs
  - apply the operation and get the possible outputs

U_1

04	P.	S08
11	P.	S06

U_2

11	M.	S08

$\bigcup$

V_1

V_2

11	M.	S08

$=$

Relational algebra on uncertain instances

- Extend relational algebra operators to uncertain instances
- The possible worlds of the result should be...
  - take all possible worlds in the supports of the inputs
  - apply the operation and get the possible outputs

U_1

04	P.	S08
11	P.	S06

U_2

11	M.	S08

$\cup$

V_1

V_2

11	M.	S08

$=$

04	P.	S08
11	P.	S06

04	P.	S08
11	P.	S06
11	M.	S08

11	M.	S08

Relational algebra on probabilistic instances

- Let's adapt relational algebra to probabilistic instances
- The possible worlds of the result should be...

Relational algebra on probabilistic instances

- Let's adapt relational algebra to probabilistic instances
- The possible worlds of the result should be...
  - take all possible worlds of the inputs
  - apply the operation and get a possible output

Relational algebra on probabilistic instances

- Let's adapt relational algebra to probabilistic instances
- The possible worlds of the result should be...
  - take all possible worlds of the inputs
  - apply the operation and get a possible output
- The probability of each possible world should be...
  - consider all input possible worlds that give it
  - sum up their probabilities

Example of relational algebra on probabilistic instances

Example of relational algebra on probabilistic instances

U_1

04	P.	S08
11	P.	S06

U_2

11	M.	S08

Example of relational algebra on probabilistic instances

U_1

04	P.	S08
11	P.	S06

$\bigcup$

U_2

11	M.	S08

Example of relational algebra on probabilistic instances

U_1

04	P.	S08
11	P.	S06

V_1

$\bigcup$

U_2

11	M.	S08

V_2

11	M.	S08

Example of relational algebra on probabilistic instances

U_1

04	P.	S08
11	P.	S06

$\pi(U_1) = 0.8$

U_2

11	M.	S08

$\pi(U_2) = 0.2$

$\bigcup$

V_1

$\pi(V_1) = 0.9$

V_2

11	M.	S08

$\pi(V_2) = 0.1$

Example of relational algebra on probabilistic instances

U_1

04	P.	S08
11	P.	S06

$\pi(U_1) = 0.8$

U_2

11	M.	S08

$\pi(U_2) = 0.2$

$\bigcup$

V_1

$\pi(V_1) = 0.9$

V_2

11	M.	S08

$\pi(V_2) = 0.1$

$=$

Example of relational algebra on probabilistic instances

U_1

04	P.	S08
11	P.	S06

$\pi(U_1) = 0.8$

U_2

11	M.	S08

$\pi(U_2) = 0.2$

$\bigcup$

V_1

$\pi(V_1) = 0.9$

V_2

11	M.	S08

$\pi(V_2) = 0.1$

$=$

W_1

04	P.	S08
11	P.	S06

Example of relational algebra on probabilistic instances

U_1

04	P.	S08
11	P.	S06

$\pi(U_1) = 0.8$

U_2

11	M.	S08

$\pi(U_2) = 0.2$

$\bigcup$

V_1

(empty)

$\pi(V_1) = 0.9$

V_2

11	M.	S08

$\pi(V_2) = 0.1$

$=$

W_1

04	P.	S08
11	P.	S06

W_2

04	P.	S08
11	P.	S06
11	M.	S08

Example of relational algebra on probabilistic instances

U_1

04	P.	S08
11	P.	S06

$\pi(U_1) = 0.8$

U_2

11	M.	S08

$\pi(U_2) = 0.2$

$\cup$

V_1

$\pi(V_1) = 0.9$

V_2

11	M.	S08

$\pi(V_2) = 0.1$

$=$

W_1

04	P.	S08
11	P.	S06

W_2

04	P.	S08
11	P.	S06
11	M.	S08

W_3

11	M.	S08

Example of relational algebra on probabilistic instances

U_1

04	P.	S08
11	P.	S06

$\pi(U_1) = 0.8$

U_2

11	M.	S08

$\pi(U_2) = 0.2$

$\bigcup$

V_1

$\pi(V_1) = 0.9$

V_2

11	M.	S08

$\pi(V_2) = 0.1$

$=$

W_1

04	P.	S08
11	P.	S06

$\pi(W_1) = 0.8 \times 0.9$

W_2

04	P.	S08
11	P.	S06
11	M.	S08

W_3

11	M.	S08

Example of relational algebra on probabilistic instances

U_1

04	P.	S08
11	P.	S06

$\pi(U_1) = 0.8$

U_2

11	M.	S08

$\pi(U_2) = 0.2$

$\cup$

V_1

$\pi(V_1) = 0.9$

V_2

11	M.	S08

$\pi(V_2) = 0.1$

$=$

W_1

04	P.	S08
11	P.	S06

$\pi(W_1) = 0.8 \times 0.9$

W_2

04	P.	S08
11	P.	S06
11	M.	S08

$\pi(W_2) = 0.8 \times 0.1$

W_3

11	M.	S08

Example of relational algebra on probabilistic instances

U_1

04	P.	S08
11	P.	S06

$\pi(U_1) = 0.8$

U_2

11	M.	S08

$\pi(U_2) = 0.2$

$\cup$

V_1

$\pi(V_1) = 0.9$

V_2

11	M.	S08

$\pi(V_2) = 0.1$

$=$

W_1

04	P.	S08
11	P.	S06

$\pi(W_1) = 0.8 \times 0.9$

W_2

04	P.	S08
11	P.	S06
11	M.	S08

$\pi(W_2) = 0.8 \times 0.1$

W_3

11	M.	S08

$\pi(W_3) = 0.2 \times 0.9$

Example of relational algebra on probabilistic instances

U_1

04	P.	S08
11	P.	S06

$\pi(U_1) = 0.8$

U_2

11	M.	S08

$\pi(U_2) = 0.2$

$\cup$

V_1

$\pi(V_1) = 0.9$

V_2

11	M.	S08

$\pi(V_2) = 0.1$

$=$

W_1

04	P.	S08
11	P.	S06

$\pi(W_1) = 0.8 \times 0.9$

W_2

04	P.	S08
11	P.	S06
11	M.	S08

$\pi(W_2) = 0.8 \times 0.1$

W_3

11	M.	S08

$$\pi(W_3) = 0.2 \times 0.9 + 0.2 \times 0.1$$

Representation system

- Remember that if we have N possible tuples

Representation system

- Remember that if we have N possible tuples
  $\rightarrow$ there are 2^N possible instances

Representation system

- Remember that if we have N possible tuples
  - $\rightarrow$ there are 2^N possible instances
  - $\rightarrow$ there are 2^{2^N} possible uncertain instances

Representation system

- Remember that if we have N possible tuples
  - → there are 2^N possible instances
  - → there are 2^{2^N} possible uncertain instances
  - → writing out an uncertain instance is exponential

Representation system

- Remember that if we have N possible tuples
  - → there are 2^N possible instances
  - → there are 2^{2^N} possible uncertain instances
  - → writing out an uncertain instance is exponential

→ Boolean provenance is a concise way to represent uncertain instances

Representation system

- Remember that if we have N possible tuples
  - → there are 2^N possible instances
  - → there are 2^{2^N} possible uncertain instances
  - → writing out an uncertain instance is exponential

→ Boolean provenance is a concise way to represent uncertain instances

- For probabilistic instances:
  - → there are infinitely many possible instances
  - → writing out a probabilistic instance is still exponential

Representation system

- Remember that if we have N possible tuples
  - → there are 2^N possible instances
  - → there are 2^{2^N} possible uncertain instances
  - → writing out an uncertain instance is exponential

→ Boolean provenance is a concise way to represent uncertain instances

- For probabilistic instances:
  - → there are infinitely many possible instances
  - → writing out a probabilistic instance is still exponential

→ How to represent probabilistic instances?

Outline

Probabilistic Relational Models

Probabilistic instances

TID, BID

pc-tables

Probabilistic query evaluation

Complexity

Conclusion

Tuple-independent databases

- The simplest model: tuple-independent databases
- Annotate each instance fact with a probability

Tuple-independent databases

- The simplest model: tuple-independent databases
- Annotate each instance fact with a probability

U

date	teacher	room
04	Michaël	S08
04	Pierre	S08
11	Michaël	S08

Tuple-independent databases

- The simplest model: tuple-independent databases
- Annotate each instance fact with a probability

U

date	teacher	room	
04	Michaël	S08	0.8
04	Pierre	S08	0.2
11	Michaël	S08	1

Tuple-independent databases

- The simplest model: tuple-independent databases
- Annotate each instance fact with a probability

U

date	teacher	room	
04	Michaël	S08	0.8
04	Pierre	S08	0.2
11	Michaël	S08	1

→ Assume independence between tuples
(Michaël and Pierre may teach at the same time)

Semantics of TID

- Each tuple is kept or discarded with the probability
- Probabilistic choices are independent across tuples

Semantics of TID

- Each tuple is kept or discarded with the probability
- Probabilistic choices are independent across tuples

U

date	teacher	room	
04	Michaël	S08	0.8
04	Pierre	S08	0.2
11	Michaël	S08	1

Semantics of TID

- Each tuple is kept or discarded with the probability
- Probabilistic choices are independent across tuples

U

date	teacher	room	
04	Michaël	S08	0.8
04	Pierre	S08	0.2
11	Michaël	S08	1

U

date	teacher	room

Semantics of TID

- Each tuple is kept or discarded with the probability
- Probabilistic choices are independent across tuples

U

date	teacher	room	
04	Michaël	S08	0.8
04	Pierre	S08	0.2
11	Michaël	S08	1

U

date	teacher	room
04	Michaël	S08

Semantics of TID

- Each tuple is kept or discarded with the probability
- Probabilistic choices are independent across tuples

U

date	teacher	room	
04	Michaël	S08	0.8
04	Pierre	S08	0.2
11	Michaël	S08	1

U

date	teacher	room
04	Michaël	S08
04	Pierre	S08

Semantics of TID

- Each tuple is kept or discarded with the probability
- Probabilistic choices are independent across tuples

U

date	teacher	room	
04	Michaël	S08	0.8
04	Pierre	S08	0.2
11	Michaël	S08	1

U

date	teacher	room
04	Michaël	S08
04	Pierre	S08
11	Michaël	S08

Semantics of TID

- Each tuple is kept or discarded with the probability
- Probabilistic choices are independent across tuples

U

date	teacher	room	
04	Michaël	S08	0.8
04	Pierre	S08	0.2
11	Michaël	S08	1

U

date	teacher	room
04	Michaël	S08
04	Pierre	S08
11	Michaël	S08

What's the probability of this outcome?

Semantics of TID

- Each tuple is kept or discarded with the probability
- Probabilistic choices are independent across tuples

U

date	teacher	room	
04	Michaël	S08	0.8
04	Pierre	S08	0.2
11	Michaël	S08	1

U

date	teacher	room
04	Michaël	S08
04	Pierre	S08
11	Michaël	S08

What's the probability of this outcome?

0.8

Semantics of TID

- Each tuple is kept or discarded with the probability
- Probabilistic choices are independent across tuples

U

date	teacher	room	
04	Michaël	S08	0.8
04	Pierre	S08	0.2
11	Michaël	S08	1

U

date	teacher	room
04	Michaël	S08
04	Pierre	S08
11	Michaël	S08

What's the probability of this outcome?

$$0.8 \times$$

Semantics of TID

- Each tuple is kept or discarded with the probability
- Probabilistic choices are independent across tuples

U

date	teacher	room	
04	Michaël	S08	0.8
04	Pierre	S08	0.2
11	Michaël	S08	1

U

date	teacher	room
04	Michaël	S08
04	Pierre	S08
11	Michaël	S08

What's the probability of this outcome?

$$0.8 \times (1 - 0.2)$$

Semantics of TID

- Each tuple is kept or discarded with the probability
- Probabilistic choices are independent across tuples

U

date	teacher	room	
04	Michaël	S08	0.8
04	Pierre	S08	0.2
11	Michaël	S08	1

U

date	teacher	room
04	Michaël	S08
04	Pierre	S08
11	Michaël	S08

What's the probability of this outcome?

$$0.8 \times (1 - 0.2) \times 1$$

Getting a probability distribution

The semantics of a TID instance is a probabilistic instance. . .

$\rightarrow$ the possible worlds are the subsets

Getting a probability distribution

The semantics of a TID instance is a probabilistic instance...

$\rightarrow$ the possible worlds are the subsets

$\rightarrow$ always keeping tuples with probability 1

Getting a probability distribution

The semantics of a TID instance is a probabilistic instance...

→ the possible worlds are the subsets

→ always keeping tuples with probability 1

Formally, for a TID instance I, the probability of J:

Getting a probability distribution

The semantics of a TID instance is a probabilistic instance...

$\rightarrow$ the possible worlds are the subsets

$\rightarrow$ always keeping tuples with probability 1

Formally, for a TID instance I, the probability of J:

- we must have $J \subseteq I$
- product of $p_{\mathbf{t}}$ for each tuple $\mathbf{t}$ kept in J
- product of $1 - p_{\mathbf{t}}$ for each tuple $\mathbf{t}$ not kept in J

Is it a probability distribution?

Do the probabilities always sum to 1?

- Let N be the number of tuples
- There are 2^N possible worlds
- They are all products of p_i or $1 - p_i$ for each $1 \leq i \leq N$

→ This is the result of expanding the expression:
$(p_1 + (1 - p_1)) \times \cdots \times (p_n + (1 - p_n))$

- All factors are equal to 1, so the probabilities sum to 1

Strong representation system

Uncertain instance: set of possible worlds

Strong representation system

Uncertain instance: set of possible worlds

Uncertainty framework: concise way to represent uncertain instances

Strong representation system

Uncertain instance:	set of possible worlds
Uncertainty framework:	concise way to represent uncertain instances
Query language:	here, relational algebra

Strong representation system

Uncertain instance: set of possible worlds

Uncertainty framework: concise way to represent uncertain instances

Query language: here, relational algebra

Definition (Strong representation system)

For any query in the language,

Strong representation system

Uncertain instance: set of possible worlds

Uncertainty framework: concise way to represent uncertain instances

Query language: here, relational algebra

Definition (Strong representation system)

For any query in the language,
on uncertain instances represented in the framework,

Strong representation system

Uncertain instance: set of possible worlds

Uncertainty framework: concise way to represent uncertain instances

Query language: here, relational algebra

Definition (Strong representation system)

For any query in the language,
on uncertain instances represented in the framework,
the uncertain instance obtained by evaluating the query

Strong representation system

Uncertain instance: set of possible worlds

Uncertainty framework: concise way to represent uncertain instances

Query language: here, relational algebra

Definition (Strong representation system)

For any query in the language,
on uncertain instances represented in the framework,
the uncertain instance obtained by evaluating the query
can also be represented in the framework

Strong representation system

Uncertain instance:	set of possible worlds
Uncertainty framework:	concise way to represent uncertain instances
Query language:	here, relational algebra

Definition (Strong representation system)

For any query in the language,
on uncertain instances represented in the framework,
the uncertain instance obtained by evaluating the query
can also be represented in the framework

→ Are TID instances a strong representation system?

Implementing select

U

date	teacher	room	
04	Michaël	S06	0.8
04	Pierre	S06	0.2
11	Michaël	S06	1

Implementing select

U

date	teacher	room	
04	Michaël	S06	0.8
04	Pierre	S06	0.2
11	Michaël	S06	1

$\sigma_{\text{teacher}=\text{“Michaël”}}(U)$

date	teacher	room

Implementing select

U

date	teacher	room	
04	Michaël	S06	0.8
04	Pierre	S06	0.2
11	Michaël	S06	1

$\sigma_{\text{teacher}=\text{“Michaël”}}(U)$

date	teacher	room	
04	Michaël	S06	0.8
11	Michaël	S06	1

Implementing select

U

date	teacher	room	
04	Michaël	S06	0.8
04	Pierre	S06	0.2
11	Michaël	S06	1

$\sigma_{\text{teacher}=\text{“Michaël”}}(U)$

date	teacher	room	
04	Michaël	S06	0.8
11	Michaël	S06	1

→ Is this correct? ...

Implementing select

U

date	teacher	room	
04	Michaël	S06	0.8
04	Pierre	S06	0.2
11	Michaël	S06	1

$\sigma_{\text{teacher}=\text{“Michaël”}}(U)$

date	teacher	room	
04	Michaël	S06	0.8
11	Michaël	S06	1

→ Is this correct? ... So far, so good.

Implementing project

U

date	teacher	room	
04	Michaël	S06	0.8
04	Pierre	S06	0.2
11	Michaël	S06	1
11	Pierre	S06	0.1
18	Michaël	S06	0.9

Implementing project

U

date	teacher	room	
04	Michaël	S06	0.8
04	Pierre	S06	0.2
11	Michaël	S06	1
11	Pierre	S06	0.1
18	Michaël	S06	0.9

$\pi_{\mathsf{date}}(U)$

date

Implementing project

U

date	teacher	room	
04	Michaël	S06	0.8
04	Pierre	S06	0.2
11	Michaël	S06	1
11	Pierre	S06	0.1
18	Michaël	S06	0.9

$\pi_{\text{date}}(U)$

date
04
11
18

Implementing project

U

date	**teacher**	**room**	
04	Michaël	S06	0.8
04	Pierre	S06	0.2
11	Michaël	S06	1
11	Pierre	S06	0.1
18	Michaël	S06	0.9

$\pi_{\text{date}}(U)$

date	
04	
11	
18	0.9

Implementing project

U

date	teacher	room	
04	Michaël	S06	0.8
04	Pierre	S06	0.2
11	Michaël	S06	1
11	Pierre	S06	0.1
18	Michaël	S06	0.9

$\pi_{\text{date}}(U)$

date	
04	
11	1
18	0.9

Implementing project

U

date	teacher	room	
04	Michaël	S06	0.8
04	Pierre	S06	0.2
11	Michaël	S06	1
11	Pierre	S06	0.1
18	Michaël	S06	0.9

$\pi_{\text{date}}(U)$

date	
04	$1-(1-0.2)\times(1-0.8)$
11	1
18	0.9

Implementing project

U

date	teacher	room	
04	Michaël	S06	0.8
04	Pierre	S06	0.2
11	Michaël	S06	1
11	Pierre	S06	0.1
18	Michaël	S06	0.9

$\pi_{\text{date}}(U)$

date	
04	$1 - (1 - 0.2) \times (1 - 0.8)$
11	1
18	0.9

→ Is this correct? ...

Implementing project

U

date	teacher	room	
04	Michaël	S06	0.8
04	Pierre	S06	0.2
11	Michaël	S06	1
11	Pierre	S06	0.1
18	Michaël	S06	0.9

$\pi_{\text{date}}(U)$

date	
04	$1-(1-0.2)\times(1-0.8)$
11	1
18	0.9

→ Is this correct? ... So far, so good.

Implementing join

U

date	teacher	room	
04	Michaël	S06	0.8
04	Pierre	S06	0.2

Implementing join

U

date	teacher	room	
04	Michaël	S06	0.8
04	Pierre	S06	0.2

Repair

room	cause	
S06	leopard	0.1

Implementing join

U

date	teacher	room	
04	Michaël	S06	0.8
04	Pierre	S06	0.2

Repair

room	cause	
S06	leopard	0.1

$U \bowtie$ Repair

date	teacher	room	cause

Implementing join

U

date	teacher	room	
04	Michaël	S06	0.8
04	Pierre	S06	0.2

Repair

room	cause	
S06	leopard	0.1

$U \bowtie$ Repair

date	teacher	room	cause
04	Michaël	S06	leopard
04	Pierre	S06	leopard

Implementing join

U

date	teacher	room	
04	Michaël	S06	0.8
04	Pierre	S06	0.2

Repair

room	cause	
S06	leopard	0.1

$U \bowtie$ Repair

date	teacher	room	cause	
04	Michaël	S06	leopard	0.8×0.1
04	Pierre	S06	leopard	0.2×0.1

Implementing join

U

date	teacher	room	
04	Michaël	S06	0.8
04	Pierre	S06	0.2

Repair

room	cause	
S06	leopard	0.1

$U \bowtie$ Repair

date	teacher	room	cause	
04	Michaël	S06	leopard	0.8×0.1
04	Pierre	S06	leopard	0.2×0.1

$\rightarrow$ Is this correct?

Implementing join . . . OR NOT!

U

date	teacher	room	
04	Michaël	S06	0.8
04	Pierre	S06	0.2

Repair

room	cause	
S06	leopard	0.1

$U \bowtie$ Repair

date	teacher	room	cause	
04	Michaël	S06	leopard	0.8×0.1
04	Pierre	S06	leopard	0.2×0.1

→ Is this correct?

→ It's WRONG!

Why is it wrong?

U

date	teacher	room	
04	Michaël	S06	1
04	Pierre	S06	1

Why is it wrong?

U

date	teacher	room	
04	Michaël	S06	1
04	Pierre	S06	1

Repair

room	cause	
S06	leopard	1/2

Why is it wrong?

U

date	teacher	room	
04	Michaël	S06	1
04	Pierre	S06	1

Repair

room	cause	
S06	leopard	1/2

$U \bowtie$ Repair

date	teacher	room	cause
04	Michaël	S06	leopard
04	Pierre	S06	leopard

Why is it wrong?

U

date	teacher	room	
04	Michaël	S06	1
04	Pierre	S06	1

Repair

room	cause	
S06	leopard	1/2

$U \bowtie$ Repair

date	teacher	room	cause	
04	Michaël	S06	leopard	1/2
04	Pierre	S06	leopard	1/2

Why is it wrong?

U

date	teacher	room	
04	Michaël	S06	1
04	Pierre	S06	1

Repair

room	cause	
S06	leopard	1/2

$U \bowtie$ Repair

date	teacher	room	cause	
04	Michaël	S06	leopard	1/2
04	Pierre	S06	leopard	1/2

→ The two tuples are not independent!

→ The first is there iff the second is there.

Why does it matter?

$U \bowtie \text{Repair}$

date	teacher	room	cause	
04	Michaël	S06	leopard	1/2
04	Pierre	S06	leopard	1/2

Why does it matter?

$U \bowtie \text{Repair}$

date	**teacher**	**room**	**cause**	
04	Michaël	S06	leopard	1/2
04	Pierre	S06	leopard	1/2

$\pi_{\textbf{room}}(U \bowtie \text{Repair})$

room
S06

Why does it matter?

$U \bowtie \text{Repair}$

date	**teacher**	**room**	**cause**	
04	Michaël	S06	leopard	$1/2$
04	Pierre	S06	leopard	$1/2$

$\pi_{\textbf{room}}(U \bowtie \text{Repair})$

room	
S06	$1-(1-1/2)\times(1-1/2)$

Why does it matter?

$U \bowtie \text{Repair}$

date	**teacher**	**room**	**cause**	
04	Michaël	S06	leopard	$1/2$
04	Pierre	S06	leopard	$1/2$

$\pi_{\textbf{room}}(U \bowtie \text{Repair})$

room	
S06	$1-(1-1/2)\times(1-1/2)$

$\rightarrow$ Probability of $3/4$...

Why does it matter?

$U \bowtie \text{Repair}$

date	teacher	room	cause	
04	Michaël	S06	leopard	$1/2$
04	Pierre	S06	leopard	$1/2$

$\pi_{\textbf{room}}(U \bowtie \text{Repair})$

room	
S06	$1-(1-1/2)\times(1-1/2)$

→ Probability of $3/4$...

→ But the leopard had probability $1/2$!

TID are not a strong representation system

- The result of a query on TID may not be a TID

→ We will see that the correlations can be complex

TID are not a strong representation system

- The result of a query on TID may not be a TID

→ We will see that the correlations can be complex

- How to evaluate queries on a TID then?

→ List all possible worlds and count the probabilities

Query evaluation done right

U

date	teacher	room	
04	Michaël	S06	0.8
04	Pierre	S06	0.2

Query evaluation done right

U

date	teacher	room	
04	Michaël	S06	0.8
04	Pierre	S06	0.2

Repair

room	cause	
S06	leopard	0.1

Query evaluation done right

U

date	teacher	room	
04	Michaël	S06	0.8
04	Pierre	S06	0.2

Repair

room	cause	
S06	leopard	0.1

$\pi_{\textbf{room}}(U \bowtie \text{Repair})$

room
S06

Query evaluation done right

U

date	teacher	room	
04	Michaël	S06	0.8
04	Pierre	S06	0.2

Repair

room	cause	
S06	leopard	0.1

$\pi_{\textbf{room}}(U \bowtie \text{Repair})$

room	
S06	???

Query evaluation done right

U

date	teacher	room	
04	Michaël	S06	0.8
04	Pierre	S06	0.2

Repair

room	cause	
S06	leopard	0.1

$\pi_{\textbf{room}}(U \bowtie \text{Repair})$

room	
S06	???

- Either there is no leopard and then no result...

Query evaluation done right

U

date	teacher	room	
04	Michaël	S06	0.8
04	Pierre	S06	0.2

Repair

room	cause	
S06	leopard	0.1

$\pi_{\mathbf{room}}(U \bowtie \text{Repair})$

room	
S06	???

- Either there is no leopard and then no result…
- Or there is a leopard and then…

Query evaluation done right

U

date	**teacher**	**room**	
04	Michaël	S06	0.8
04	Pierre	S06	0.2

Repair

room	**cause**	
S06	leopard	0.1

$\pi_{\textbf{room}}(U \bowtie \text{Repair})$

room	
S06	???

- Either there is no leopard and then no result...
- Or there is a leopard and then...
  - Non-empty result:

Query evaluation done right

U

date	teacher	room	
04	Michaël	S06	0.8
04	Pierre	S06	0.2

Repair

room	cause	
S06	leopard	0.1

$\pi_{\text{room}}(U \bowtie \text{Repair})$

room	
S06	???

- Either there is no leopard and then no result...
- Or there is a leopard and then...
  - Non-empty result: $1 - (1 - 0.8) \times (1 - 0.2)$

Query evaluation done right

U

date	teacher	room	
04	Michaël	S06	0.8
04	Pierre	S06	0.2

Repair

room	cause	
S06	leopard	0.1

$\pi_{\mathbf{room}}(U \bowtie \text{Repair})$

room	
S06	???

- Either there is no leopard and then no result...
- Or there is a leopard and then...
  - Non-empty result: $1 - (1 - 0.8) \times (1 - 0.2) = 0.84$

Query evaluation done right

U

date	teacher	room	
04	Michaël	S06	0.8
04	Pierre	S06	0.2

Repair

room	cause	
S06	leopard	0.1

$\pi_{\textbf{room}}(U \bowtie \text{Repair})$

room
S06

- Either there is no leopard and then no result...
- Or there is a leopard and then...
  - Non-empty result: $1 - (1 - 0.8) \times (1 - 0.2) = 0.84$
- The query probability is:

Query evaluation done right

U

date	teacher	room	
04	Michaël	S06	0.8
04	Pierre	S06	0.2

Repair

room	cause	
S06	leopard	0.1

$\pi_{\mathbf{room}}(U \bowtie \text{Repair})$

room	
S06	0.084

- Either there is no leopard and then no result...
- Or there is a leopard and then...
  - Non-empty result: $1 - (1 - 0.8) \times (1 - 0.2) = 0.84$
- The query probability is: 0.1×0.84

Expressiveness of TID

Can we represent all probabilistic instances with TID?

Expressiveness of TID

Can we represent all probabilistic instances with TID?

"The class is taught by Pierre or Michaël or no one but not both"

Expressiveness of TID

Can we represent all probabilistic instances with TID?

"The class is taught by Pierre or Michaël or no one but not both"

U_1

teacher
Michaël

$\pi(U_1) = 0.8$

Expressiveness of TID

Can we represent all probabilistic instances with TID?

"The class is taught by Pierre or Michaël or no one but not both"

U_1

teacher
Michaël

$\pi(U_1) = 0.8$

U_2

teacher
Pierre

$\pi(U_2) = 0.1$

Expressiveness of TID

Can we represent all probabilistic instances with TID?

"The class is taught by Pierre or Michaël or no one but not both"

U_1

teacher
Michaël

$\pi(U_1) = 0.8$

U_2

teacher
Pierre

$\pi(U_2) = 0.1$

U_3

teacher

$\pi(U_3) = 0.1$

Expressiveness of TID

Can we represent all probabilistic instances with TID?

"The class is taught by Pierre or Michaël or no one but not both"

U_1

teacher
Michaël

$\pi(U_1) = 0.8$

U_2

teacher
Pierre

$\pi(U_2) = 0.1$

U_3

teacher

$\pi(U_3) = 0.1$

U

teacher
Pierre
Michaël

Expressiveness of TID

Can we represent all probabilistic instances with TID?

"The class is taught by Pierre or Michaël or no one but not both"

U_1

teacher
Michaël

$\pi(U_1) = 0.8$

U_2

teacher
Pierre

$\pi(U_2) = 0.1$

U_3

teacher

$\pi(U_3) = 0.1$

U

teacher	
Pierre	0.1
Michaël	

Expressiveness of TID

Can we represent all probabilistic instances with TID?

"The class is taught by Pierre or Michaël or no one but not both"

U_1

teacher
Michaël

$\pi(U_1) = 0.8$

U_2

teacher
Pierre

$\pi(U_2) = 0.1$

U_3

teacher

$\pi(U_3) = 0.1$

U

teacher	
Pierre	0.1
Michaël	0.8

Expressiveness of TID

Can we represent all probabilistic instances with TID?

"The class is taught by Pierre or Michaël or no one but not both"

U_1

teacher
Michaël

$\pi(U_1) = 0.8$

U_2

teacher
Pierre

$\pi(U_2) = 0.1$

U_3

teacher

$\pi(U_3) = 0.1$

U

teacher	
Pierre	0.1
Michaël	0.8

→ We cannot forbid that both teach the class!

Block-independent disjoint instances

- A more expressive framework than TID
- Call some attributes the key (underlined)

Block-independent disjoint instances

- A more expressive framework than TID
- Call some attributes the key (underlined)

U

mon	day	teacher	room
Jan	04	Michaël	S08
Jan	04	Pierre	S08
Jan	11	Michaël	S06
Jan	11	Pierre	S08

Block-independent disjoint instances

- A more expressive framework than TID
- Call some attributes the key (underlined)

U

mon	day	teacher	room
Jan	04	Michaël	S08
Jan	04	Pierre	S08
Jan	11	Michaël	S06
Jan	11	Pierre	S08

- The blocks are the sets of tuples with the same key

Block-independent disjoint instances

- A more expressive framework than TID
- Call some attributes the key (underlined)

U

<u>**mon**</u>	<u>**day**</u>	**teacher**	**room**
Jan	04	Michaël	S08
Jan	04	Pierre	S08
Jan	11	Michaël	S06
Jan	11	Pierre	S08

- The blocks are the sets of tuples with the same key
- Each tuple has a probability

Block-independent disjoint instances

- A more expressive framework than TID
- Call some attributes the key (<u>underlined</u>)

U

<u>mon</u>	<u>day</u>	teacher	room	
Jan	04	Michaël	S08	0.9
Jan	04	Pierre	S08	0.1
Jan	11	Michaël	S06	0.8
Jan	11	Pierre	S08	0.1

- The blocks are the sets of tuples with the same key
- Each tuple has a probability

Block-independent disjoint instances

- A more expressive framework than TID
- Call some attributes the key (underlined)

U

<u>mon</u>	<u>day</u>	teacher	room	
Jan	04	Michaël	S08	0.9
Jan	04	Pierre	S08	0.1
Jan	11	Michaël	S06	0.8
Jan	11	Pierre	S08	0.1

- The blocks are the sets of tuples with the same key
- Each tuple has a probability
- Probabilities must sum to ≤ 1 in each block

BID semantics

U

mon	day	teacher	room	
Jan	04	Michaël	S08	0.9
Jan	04	Pierre	S08	0.1
Jan	11	Michaël	S06	0.8
Jan	11	Pierre	S08	0.1

BID semantics

U

<u>**mon**</u>	<u>**day**</u>	**teacher**	**room**	
Jan	04	Michaël	S08	0.9
Jan	04	Pierre	S08	0.1
Jan	11	Michaël	S06	0.8
Jan	11	Pierre	S08	0.1

- For each block:

BID semantics

U

<u>mon</u>	<u>day</u>	teacher	room	
Jan	04	Michaël	S08	0.9
Jan	04	Pierre	S08	0.1
Jan	11	Michaël	S06	0.8
Jan	11	Pierre	S08	0.1

- For each block:
  - Pick one tuple according to probabilities

BID semantics

U

mon	**day**	**teacher**	**room**	
Jan	04	Michaël	S08	0.9
Jan	04	Pierre	S08	0.1
Jan	11	Michaël	S06	0.8
Jan	11	Pierre	S08	0.1

- For each block:
  - Pick one tuple according to probabilities
  - Possibly no tuple if probabilities are < 1

BID semantics

U

<u>mon</u>	<u>day</u>	teacher	room	
Jan	04	Michaël	S08	0.9
Jan	04	Pierre	S08	0.1
Jan	11	Michaël	S06	0.8
Jan	11	Pierre	S08	0.1

- For each block:
  - Pick one tuple according to probabilities
  - Possibly no tuple if probabilities are < 1

$\rightarrow$ Do choices independently in each block

BID semantics

U

mon	day	teacher	room	
Jan	04	Michaël	S08	0.9
Jan	04	Pierre	S08	0.1
Jan	11	Michaël	S06	0.8
Jan	11	Pierre	S08	0.1

U

mon	day	teacher	room

- For each block:
  - Pick one tuple according to probabilities
  - Possibly no tuple if probabilities are < 1

→ Do choices independently in each block

BID semantics

U

<u>mon</u>	<u>day</u>	teacher	room	
Jan	04	Michaël	S08	0.9
Jan	04	Pierre	S08	0.1
Jan	11	Michaël	S06	0.8
Jan	11	Pierre	S08	0.1

U

<u>mon</u>	<u>day</u>	teacher	room
Jan	04	Michaël	S08
Jan	04	Pierre	S08

- For each block:
  - Pick one tuple according to probabilities
  - Possibly no tuple if probabilities are < 1

→ Do choices independently in each block

BID semantics

U

<u>**mon**</u>	<u>**day**</u>	**teacher**	**room**	
Jan	04	Michaël	S08	0.9
Jan	04	Pierre	S08	0.1
Jan	11	Michaël	S06	0.8
Jan	11	Pierre	S08	0.1

U

<u>**mon**</u>	<u>**day**</u>	**teacher**	**room**
Jan	04	Michaël	S08
Jan	04	Pierre	S08
Jan	11	Michaël	S06
Jan	11	Pierre	S08

- For each block:
  - Pick one tuple according to probabilities
  - Possibly no tuple if probabilities are < 1

$\rightarrow$ Do choices independently in each block

BID captures TID

- Each TID can be expressed as a BID...

BID captures TID

- Each TID can be expressed as a BID…
  - → Take all attributes as key
  - → Each block contains a single tuple

BID captures TID

- Each TID can be expressed as a BID...
  - → Take <u>all</u> <u>attributes</u> as key
  - → Each block contains a single tuple

U

<u>date</u>	**<u>teacher</u>**	**<u>room</u>**	
04	Michaël	S08	0.8
04	Pierre	S08	0.2
11	Michaël	S08	1

Expressiveness of BID

Can we represent all probabilistic instances with BID?

Expressiveness of BID

Can we represent all probabilistic instances with BID?

"The class is taught by exactly two among Pierre, Michaël, Antoine."

Expressiveness of BID

Can we represent all probabilistic instances with BID?

"The class is taught by exactly two among Pierre, Michaël, Antoine."

U_1

teacher
Michaël
Antoine

$\pi(U_1) = 0.8$

Expressiveness of BID

Can we represent all probabilistic instances with BID?

"The class is taught by exactly two among Pierre, Michaël, Antoine."

U_1

teacher
Michaël
Antoine
$\pi(U_1) = 0.8$

U_2

teacher
Pierre
Antoine
$\pi(U_2) = 0.1$

Expressiveness of BID

Can we represent all probabilistic instances with BID?

"The class is taught by exactly two among Pierre, Michaël, Antoine."

U_1

teacher
Michaël
Antoine

$\pi(U_1) = 0.8$

U_2

teacher
Pierre
Antoine

$\pi(U_2) = 0.1$

U_3

teacher
Pierre
Michaël

$\pi(U_3) = 0.1$

Expressiveness of BID

Can we represent all probabilistic instances with BID?

"The class is taught by exactly two among Pierre, Michaël, Antoine."

U_1

teacher
Michaël
Antoine

$\pi(U_1) = 0.8$

U_2

teacher
Pierre
Antoine

$\pi(U_2) = 0.1$

U_3

teacher
Pierre
Michaël

$\pi(U_3) = 0.1$

→ If **teacher** is not a key, then only one tuple

Expressiveness of BID

Can we represent all probabilistic instances with BID?

"The class is taught by exactly two among Pierre, Michaël, Antoine."

U_1

teacher
Michaël
Antoine

$\pi(U_1) = 0.8$

U_2

teacher
Pierre
Antoine

$\pi(U_2) = 0.1$

U_3

teacher
Pierre
Michaël

$\pi(U_3) = 0.1$

→ If **teacher** is not a key, then only one tuple

→ If **teacher** is a key **teacher**, then TID

Expressiveness of BID

Can we represent all probabilistic instances with BID?

"The class is taught by exactly two among Pierre, Michaël, Antoine."

U_1

teacher
Michaël
Antoine

$\pi(U_1) = 0.8$

U_2

teacher
Pierre
Antoine

$\pi(U_2) = 0.1$

U_3

teacher
Pierre
Michaël

$\pi(U_3) = 0.1$

→ If **teacher** is not a key, then only one tuple

→ If **teacher** is a key <u>**teacher**</u>, then TID

→ We cannot represent this probabilistic instance as a BID

Expressiveness of BID

Can we represent all probabilistic instances with BID?

"The class is taught by exactly two among Pierre, Michaël, Antoine."

U_1

teacher
Michaël
Antoine

$\pi(U_1) = 0.8$

U_2

teacher
Pierre
Antoine

$\pi(U_2) = 0.1$

U_3

teacher
Pierre
Michaël

$\pi(U_3) = 0.1$

→ If **teacher** is not a key, then only one tuple

→ If **teacher** is a key **teacher**, then TID

→ We cannot represent this probabilistic instance as a BID

→ It is not a strong representation system either

  → Same counterexample as for TID

Outline

Boolean c-tables

Boolean c-tables (Boolean provenance):

- Set of Boolean variables $x_1, x_2, \ldots$
- Each tuple has a condition: Variables, Boolean operators

Boolean c-tables

Boolean c-tables (Boolean provenance):

- Set of Boolean variables $x_1, x_2, \ldots$
- Each tuple has a condition: Variables, Boolean operators

date	teacher	room	
04	Michaël	C42	$\neg x_1$
04	Pierre	C42	x_1
11	Michaël	S08	$x_2 \wedge \neg x_1$
11	Pierre	S08	$x_2 \wedge x_1$
11	Michaël	S06	$\neg x_2 \wedge \neg x_1$
11	Pierre	S06	$\neg x_2 \wedge x_1$

x_1 Michaël is sick

x_2 Projector in S08 is working

pc-tables

A (Boolean) pc-table is a Boolean c-table
plus a probability p_i *for each* x_i
indicating the independent probability that x_i *is true.*

pc-tables

A (Boolean) pc-table is a Boolean c-table plus a probability p_i for each x_i indicating the independent probability that x_i is true.

Formally:

- A Boolean valuation ν of the x_i maps each to 0 or 1
  - Possible world of the Boolean c-instance under ν
  - The possible worlds are the worlds over all valuations

pc-tables

A (Boolean) pc-table is a Boolean c-table plus a probability p_i for each x_i indicating the independent probability that x_i is true.

Formally:

- A Boolean valuation ν of the x_i maps each to 0 or 1
  - Possible world of the Boolean c-instance under ν
  - The possible worlds are the worlds over all valuations
- The probability of valuation ν is:

pc-tables

A (Boolean) pc-table is a Boolean c-table plus a probability p_i for each x_i indicating the independent probability that x_i is true.

Formally:

- A Boolean valuation ν of the x_i maps each to 0 or 1
  - Possible world of the Boolean c-instance under ν
  - The possible worlds are the worlds over all valuations
- The probability of valuation ν is:
  - Product of the p_i for the x_i with $\nu(x_i) = 1$

pc-tables

A (Boolean) pc-table is a Boolean c-table plus a probability p_i for each x_i indicating the independent probability that x_i is true.

Formally:

- A Boolean valuation ν of the x_i maps each to 0 or 1
  - Possible world of the Boolean c-instance under ν
  - The possible worlds are the worlds over all valuations
- The probability of valuation ν is:
  - Product of the p_i for the x_i with $\nu(x_i) = 1$
  - Product of the $1 - p_i$ for the x_i with $\nu(x_i) = 0$

pc-tables

A (Boolean) pc-table is a Boolean c-table plus a probability p_i for each x_i indicating the independent probability that x_i is true.

Formally:

- A Boolean valuation ν of the x_i maps each to 0 or 1
  - Possible world of the Boolean c-instance under ν
  - The possible worlds are the worlds over all valuations
- The probability of valuation ν is:
  - Product of the p_i for the x_i with $\nu(x_i) = 1$
  - Product of the $1 - p_i$ for the x_i with $\nu(x_i) = 0$
  - Sounds familiar?

pc-tables

A (Boolean) pc-table is a Boolean c-table plus a probability p_i for each x_i indicating the independent probability that x_i is true.

Formally:

- A Boolean valuation ν of the x_i maps each to 0 or 1
  - Possible world of the Boolean c-instance under ν
  - The possible worlds are the worlds over all valuations
- The probability of valuation ν is:
  - Product of the p_i for the x_i with $\nu(x_i) = 1$
  - Product of the $1 - p_i$ for the x_i with $\nu(x_i) = 0$
  - Sounds familiar?

    → Yeah, it's like TID instances!

pc-table example

date	teacher	room	
04	Michaël	C42	$\neg x_1$
04	Pierre	C42	x_1
11	Michaël	S08	$x_2 \wedge \neg x_1$
11	Pierre	S08	$x_2 \wedge x_1$
11	Michaël	S06	$\neg x_2 \wedge \neg x_1$
11	Pierre	S06	$\neg x_2 \wedge x_1$

pc-table example

date	teacher	room	
04	Michaël	C42	$\neg x_1$
04	Pierre	C42	x_1
11	Michaël	S08	$x_2 \land \neg x_1$
11	Pierre	S08	$x_2 \land x_1$
11	Michaël	S06	$\neg x_2 \land \neg x_1$
11	Pierre	S06	$\neg x_2 \land x_1$

x_1 Michaël is sick

x_2 Projector in S08 is working

pc-table example

date	teacher	room	
04	Michaël	C42	$\neg x_1$
04	Pierre	C42	x_1
11	Michaël	S08	$x_2 \wedge \neg x_1$
11	Pierre	S08	$x_2 \wedge x_1$
11	Michaël	S06	$\neg x_2 \wedge \neg x_1$
11	Pierre	S06	$\neg x_2 \wedge x_1$

x_1 Michaël is sick

$\rightarrow$ Probability 0.1

x_2 Projector in S08 is working

$\rightarrow$ Probability 0.2

pc-table semantics example

date	teacher	room	$x_1 : 0.1, x_2 : 0.2$
04	Michaël	C42	$\neg x_1$
04	Pierre	C42	x_1
11	Michaël	S08	$x_2 \land \neg x_1$
11	Pierre	S08	$x_2 \land x_1$
11	Michaël	S06	$\neg x_2 \land \neg x_1$
11	Pierre	S06	$\neg x_2 \land x_1$

pc-table semantics example

date	teacher	room	$x_1 : 0.1, x_2 : 0.2$
04	Michaël	C42	$\neg x_1$
04	Pierre	C42	x_1
11	Michaël	S08	$x_2 \wedge \neg x_1$
11	Pierre	S08	$x_2 \wedge x_1$
11	Michaël	S06	$\neg x_2 \wedge \neg x_1$
11	Pierre	S06	$\neg x_2 \wedge x_1$

- Take ν mapping x_1 to 0 and x_2 to 1

pc-table semantics example

date	teacher	room	$x_1 : 0.1, x_2 : 0.2$
04	Michaël	C42	$\neg x_1$
04	Pierre	C42	x_1
11	Michaël	S08	$x_2 \land \neg x_1$
11	Pierre	S08	$x_2 \land x_1$
11	Michaël	S06	$\neg x_2 \land \neg x_1$
11	Pierre	S06	$\neg x_2 \land x_1$

- Take ν mapping x_1 to 0 and x_2 to 1
- Probability of ν:

pc-table semantics example

date	teacher	room	$x_1 : 0.1,\ x_2 : 0.2$
04	Michaël	C42	$\neg x_1$
04	Pierre	C42	x_1
11	Michaël	S08	$x_2 \land \neg x_1$
11	Pierre	S08	$x_2 \land x_1$
11	Michaël	S06	$\neg x_2 \land \neg x_1$
11	Pierre	S06	$\neg x_2 \land x_1$

- Take ν mapping x_1 to 0 and x_2 to 1
- Probability of ν: $(1 - 0.1) \times 0.2 = 0.18$

pc-table semantics example

date	teacher	room	$x_1 : 0.1, x_2 : 0.2$
04	Michaël	C42	$\neg x_1$
04	Pierre	C42	x_1
11	Michaël	S08	$x_2 \land \neg x_1$
11	Pierre	S08	$x_2 \land x_1$
11	Michaël	S06	$\neg x_2 \land \neg x_1$
11	Pierre	S06	$\neg x_2 \land x_1$

- Take ν mapping x_1 to 0 and x_2 to 1
- Probability of ν: $(1 - 0.1) \times 0.2 = 0.18$
- Evaluate the conditions

pc-table semantics example

date	teacher	room	$x_1 : 0.1, x_2 : 0.2$
04	Michaël	C42	$\neg x_1$
04	Pierre	C42	x_1
11	Michaël	S08	$x_2 \wedge \neg x_1$
11	Pierre	S08	$x_2 \wedge x_1$
11	Michaël	S06	$\neg x_2 \wedge \neg x_1$
11	Pierre	S06	$\neg x_2 \wedge x_1$

date	teacher	room
04	Michaël	C42
04	Pierre	C42
11	Michaël	S08
11	Pierre	S08
11	Michaël	S06
11	Pierre	S06

- Take ν mapping x_1 to 0 and x_2 to 1
- Probability of ν: $(1 - 0.1) \times 0.2 = 0.18$
- Evaluate the conditions

pc-table semantics example

date	teacher	room	$x_1 : 0.1,\ x_2 : 0.2$
04	Michaël	C42	$\neg x_1$
04	Pierre	C42	x_1
11	Michaël	S08	$x_2 \wedge \neg x_1$
11	Pierre	S08	$x_2 \wedge x_1$
11	Michaël	S06	$\neg x_2 \wedge \neg x_1$
11	Pierre	S06	$\neg x_2 \wedge x_1$

date	teacher	room
04	Michaël	C42
04	Pierre	C42
11	Michaël	S08
11	Pierre	S08
11	Michaël	S06
11	Pierre	S06

- Take ν mapping x_1 to 0 and x_2 to 1
- Probability of ν: $(1 - 0.1) \times 0.2 = 0.18$
- Evaluate the conditions

→ Probability of possible world: sum over the valuations

pc-table semantics example

date	teacher	room	$x_1 : 0.1,\ x_2 : 0.2$
04	Michaël	C42	$\neg x_1$
04	Pierre	C42	x_1
11	Michaël	S08	$x_2 \land \neg x_1$
11	Pierre	S08	$x_2 \land x_1$
11	Michaël	S06	$\neg x_2 \land \neg x_1$
11	Pierre	S06	$\neg x_2 \land x_1$

date	teacher	room
04	Michaël	C42
04	Pierre	C42
11	Michaël	S08
11	Pierre	S08
11	Michaël	S06
11	Pierre	S06

- Take ν mapping x_1 to 0 and x_2 to 1
- Probability of ν: $(1 - 0.1) \times 0.2 = 0.18$
- Evaluate the conditions

→ Probability of possible world: sum over the valuations

→ Here: only this valuation, 0.18

pc-tables capture TID

Give each tuple its own variable:

U

date	**teacher**	**room**
04	Michaël	S08
04	Pierre	S08
11	Michaël	S08

pc-tables capture TID

Give each tuple its own variable:

U

date	teacher	room	
04	Michaël	S08	x_1
04	Pierre	S08	x_2
11	Michaël	S08	x_3

pc-tables capture TID

Give each tuple its own variable:

U

date	teacher	room	
04	Michaël	S08	x_1
04	Pierre	S08	x_2
11	Michaël	S08	x_3

→ Give each variable the probability of the tuple

pc-tables capture mutually exclusive

U

mon	day	teacher	room	
Jan	04	Michaël	S08	$x = 1$
Jan	04	Pierre	S08	$x = 2$
Jan	04	Antoine	S08	$x = 3$

pc-tables capture mutually exclusive

U

mon	day	teacher	room	
Jan	04	Michaël	S08	$x = 1$
Jan	04	Pierre	S08	$x = 2$
Jan	04	Antoine	S08	$x = 3$

- Give a probability to each value of x, summing up to 1

pc-tables capture mutually exclusive

U

mon	**day**	**teacher**	**room**	
Jan	04	Michaël	S08	$x = 1$
Jan	04	Pierre	S08	$x = 2$
Jan	04	Antoine	S08	$x = 3$

- Give a probability to each value of x, summing up to 1

  → Example: x has probability:

    - 0.8 to be 1
    - 0.1 to be 2
    - 0.1 to be 3

pc-tables capture mutually exclusive

U

mon	day	teacher	room	
Jan	04	Michaël	S08	$x = 1$
Jan	04	Pierre	S08	$x = 2$
Jan	04	Antoine	S08	$x = 3$

- Give a probability to each value of x, summing up to 1
  → Example: x has probability:
    - 0.8 to be 1
    - 0.1 to be 2
    - 0.1 to be 3
- Rewriting from non-Boolean to Boolean...

Rewriting non-Boolean to Boolean

U

mon	**day**	**teacher**	**room**	
Jan	04	Michaël	S08	$x = 00$
Jan	04	Pierre	S08	$x = 01$
Jan	04	Antoine	S08	$x = 10$

Rewriting non-Boolean to Boolean

U

mon	day	teacher	room	
Jan	04	Michaël	S08	$x = 00$
Jan	04	Pierre	S08	$x = 01$
Jan	04	Antoine	S08	$x = 10$

U

mon	day	teacher	room	
Jan	04	Michaël	S08	$\neg x_1 \land \neg x_2$
Jan	04	Pierre	S08	$\neg x_1 \land x_2$
Jan	04	Antoine	S08	$x_1 \land \neg x_2$

Rewriting non-Boolean to Boolean

U

mon	day	teacher	room	
Jan	04	Michaël	S08	$x = 00$
Jan	04	Pierre	S08	$x = 01$
Jan	04	Antoine	S08	$x = 10$

U

mon	day	teacher	room	
Jan	04	Michaël	S08	$\neg x_1 \wedge \neg x_2$
Jan	04	Pierre	S08	$\neg x_1 \wedge x_2$
Jan	04	Antoine	S08	$x_1 \wedge \neg x_2$

→ How to choose the probabilities?

Choosing the probabilities

- We start with the probabilities:
  - $x = 00$ has probability 0.8
  - $x = 01$ has probability 0.1
  - $x = 10$ has probability 0.1
  - $x = 11$ has probability 0

Choosing the probabilities

- We start with the probabilities:
  - $x = 00$ has probability 0.8
  - $x = 01$ has probability 0.1
  - $x = 10$ has probability 0.1
  - $x = 11$ has probability 0
- See the rewriting as a decision tree:
  Either the first bit is 0 or it is 1:
  - if the first bit is 0, then either the second is 0 or it is 1
  - if the first bit is 1, then either the second is 0 or it is 1

Choosing the probabilities

- We start with the probabilities:
  - $x = 00$ has probability 0.8
  - $x = 01$ has probability 0.1
  - $x = 10$ has probability 0.1
  - $x = 11$ has probability 0
- See the rewriting as a decision tree:
  Either the first bit is 0 or it is 1:
  - if the first bit is 0, then either the second is 0 or it is 1
  - if the first bit is 1, then either the second is 0 or it is 1
- Use variable x_1 for the first choice, proba 0.1
  - If $x_1 = 0$ use variable x_2 for the second choice, proba...

Choosing the probabilities

- We start with the probabilities:
  - $x = 00$ has probability 0.8
  - $x = 01$ has probability 0.1
  - $x = 10$ has probability 0.1
  - $x = 11$ has probability 0
- See the rewriting as a decision tree:
  Either the first bit is 0 or it is 1:
  - if the first bit is 0, then either the second is 0 or it is 1
  - if the first bit is 1, then either the second is 0 or it is 1
- Use variable x_1 for the first choice, proba 0.1
  - If $x_1 = 0$ use variable x_2 for the second choice, proba... 1/9

Choosing the probabilities

- We start with the probabilities:
  - $x = 00$ has probability 0.8
  - $x = 01$ has probability 0.1
  - $x = 10$ has probability 0.1
  - $x = 11$ has probability 0
- See the rewriting as a decision tree:
  Either the first bit is 0 or it is 1:
  - if the first bit is 0, then either the second is 0 or it is 1
  - if the first bit is 1, then either the second is 0 or it is 1
- Use variable x_1 for the first choice, proba 0.1
  - If $x_1 = 0$ use variable x_2 for the second choice, proba... 1/9
  - If $x_1 = 1$ use variable x'_2 for the second choice, proba...

Choosing the probabilities

- We start with the probabilities:
  - $x = 00$ has probability 0.8
  - $x = 01$ has probability 0.1
  - $x = 10$ has probability 0.1
  - $x = 11$ has probability 0
- See the rewriting as a decision tree:
  Either the first bit is 0 or it is 1:
  - if the first bit is 0, then either the second is 0 or it is 1
  - if the first bit is 1, then either the second is 0 or it is 1
- Use variable x_1 for the first choice, proba 0.1
  - If $x_1 = 0$ use variable x_2 for the second choice, proba... 1/9
  - If $x_1 = 1$ use variable x'_2 for the second choice, proba... 0

Converting mutually exclusive to pc-tables

mon	day	teacher	room	
Jan	04	Michaël	S08	$x = 00$
Jan	04	Pierre	S08	$x = 01$
Jan	04	Antoine	S08	$x = 10$

- Probabilities: x has proba 0.8 to be 1, 0.1 to be 2, 0.1 to be 3

→ Rewriting:

Converting mutually exclusive to pc-tables

mon	day	teacher	room	
Jan	04	Michaël	S08	$x = 00$
Jan	04	Pierre	S08	$x = 01$
Jan	04	Antoine	S08	$x = 10$

- Probabilities: x has proba 0.8 to be 1, 0.1 to be 2, 0.1 to be 3

→ Rewriting:

mon	day	teacher	room	
Jan	04	Michaël	S08	$\neg x_1 \wedge \neg x_2$
Jan	04	Pierre	S08	$\neg x_1 \wedge x_2$
Jan	04	Antoine	S08	$x_1 \wedge \neg x_2'$

→ x_1 has proba 0.1, x_2 has proba $1/9$, x_2' has proba 0

Capturing BID with pc-tables

- This process generalizes: create decision trees

Capturing BID with pc-tables

- This process generalizes: create decision trees
- We can capture BID by doing this in each block

Capturing BID with pc-tables

- This process generalizes: create decision trees
- We can capture BID by doing this in each block

day	teacher	room	
04	Michaël	S08	0.9
04	Pierre	S08	0.1
11	Michaël	S06	0.8
11	Pierre	S08	0.1

Capturing BID with pc-tables

- This process generalizes: create decision trees
- We can capture BID by doing this in each block

day	**teacher**	**room**	
04	Michaël	S08	0.9
04	Pierre	S08	0.1
11	Michaël	S06	0.8
11	Pierre	S08	0.1

day	**teacher**	**room**	
04	Michaël	S08	$\neg x_1$
04	Pierre	S08	x_1
11	Michaël	S06	$\neg y_1 \land \neg y_2$
11	Pierre	S08	$\neg y_1 \land y_2$

Capturing BID with pc-tables

- This process generalizes: create decision trees
- We can capture BID by doing this in each block

day	teacher	room	
04	Michaël	S08	0.9
04	Pierre	S08	0.1
11	Michaël	S06	0.8
11	Pierre	S08	0.1

day	teacher	room	
04	Michaël	S08	$\neg x_1$
04	Pierre	S08	x_1
11	Michaël	S06	$\neg y_1 \wedge \neg y_2$
11	Pierre	S08	$\neg y_1 \wedge y_2$

x_1 has probability 0.1

y_1 has probability 0.1

y_2 has probability 1/9

Strong representation system

- Boolean c-tables are a strong representation system for the relational algebra (just use m-semiring provenance computation rules)
- Further, each valuation of the output
  is the output for the same valuation of the inputs
  → assuming that variables in the input relations are different
  → this preserves probabilities

Strong representation system

- Boolean c-tables are a strong representation system for the relational algebra (just use m-semiring provenance computation rules)
- Further, each valuation of the output is the output for the same valuation of the inputs
  - → assuming that variables in the input relations are different
  - → this preserves probabilities

→ pc-tables are a strong representation system

Capturing all probabilistic instances

- Support $\mathcal{U}$: uncertain relation
- Here, set of subsets of a finite set of tuples
- Probability distribution π on $\mathcal{U}$

Capturing all probabilistic instances

- Support $\mathcal{U}$: uncertain relation
- Here, set of subsets of a finite set of tuples
- Probability distribution π on $\mathcal{U}$

→ Can any probabilistic instance be represented by a pc-table?

Capturing uncertain instances with Boolean c-tables (1)

- Number the possible worlds in binary
- For each tuple, write the possible worlds where it appears

Capturing uncertain instances with Boolean c-tables (1)

- Number the possible worlds in binary
- For each tuple, write the possible worlds where it appears

00

v	**w**
a	d
b	e
c	f

01

v	**w**
a	d
b	e
c	f

10

v	**w**
a	d
b	e
c	f

11

v	**w**
a	d
b	e
c	f

Capturing uncertain instances with Boolean c-tables (1)

- Number the possible worlds in binary
- For each tuple, write the possible worlds where it appears

00

v	w
a	d
b	e
c	f

01

v	w
a	d
b	e
c	f

10

v	w
a	d
b	e
c	f

11

v	w
a	d
b	e
c	f

v	w	
a	d	$x = 00 \vee x = 01 \vee x = 10 \vee x = 11$
b	e	$x = 01$
c	f	$x = 01 \vee x = 10 \vee x = 11$

Capturing uncertain instances with Boolean c-tables (1)

- Number the possible worlds in binary
- For each tuple, write the possible worlds where it appears

00		01		10		11	
v	**w**	**v**	**w**	**v**	**w**	**v**	**w**
a	d	a	d	a	d	a	d
b	e	b	e	b	e	b	e
c	f	c	f	c	f	c	f

v	**w**	
a	d	$x = 00 \lor x = 01 \lor x = 10 \lor x = 11$
b	e	$x = 01$
c	f	$x = 01 \lor x = 10 \lor x = 11$

→ We can also do this with pc-tables

Capturing uncertain instances with Boolean c-tables (2)

Second step: reduce to binary:

v	w	
a	d	$x = 00 \lor x = 01 \lor x = 10 \lor x = 11$
b	e	$x = 01$
c	f	$x = 01 \lor x = 10 \lor x = 11$

Capturing uncertain instances with Boolean c-tables (2)

Second step: reduce to binary:

v	w	
a	d	$x = 00 \lor x = 01 \lor x = 10 \lor x = 11$
b	e	$x = 01$
c	f	$x = 01 \lor x = 10 \lor x = 11$

v	w	
a	d	$\neg x_1 \land \neg x_2 \lor \neg x_1 \land x_2 \lor x_1 \land \neg x_2 \lor x_1 \land x_2$
b	e	$\neg x_1 \land x_2$
c	f	$\neg x_1 \land x_2 \lor x_1 \land \neg x_2 \lor x_1 \land x_2$

Capturing uncertain instances with Boolean c-tables (2)

Second step: reduce to binary:

v	w	
a	d	$x = 00 \lor x = 01 \lor x = 10 \lor x = 11$
b	e	$x = 01$
c	f	$x = 01 \lor x = 10 \lor x = 11$

v	w	
a	d	$\neg x_1 \land \neg x_2 \lor \neg x_1 \land x_2 \lor x_1 \land \neg x_2 \lor x_1 \land x_2$
b	e	$\neg x_1 \land x_2$
c	f	$\neg x_1 \land x_2 \lor x_1 \land \neg x_2 \lor x_1 \land x_2$

- For pc-instances, how to choose the probabilities?

Capturing uncertain instances with Boolean c-tables (2)

Second step: reduce to binary:

v	w	
a	d	$x = 00 \lor x = 01 \lor x = 10 \lor x = 11$
b	e	$x = 01$
c	f	$x = 01 \lor x = 10 \lor x = 11$

v	w	
a	d	$\neg x_1 \land \neg x_2 \lor \neg x_1 \land x_2 \lor x_1 \land \neg x_2 \lor x_1 \land x_2$
b	e	$\neg x_1 \land x_2$
c	f	$\neg x_1 \land x_2 \lor x_1 \land \neg x_2 \lor x_1 \land x_2$

- For pc-instances, how to choose the probabilities?

→ We have seen this: this is encoding a mutually exclusive choice

Outline

Probabilistic query evaluation

- Inputs:
  - a database D of TID instances
  - a relational algebra query Q
  - a result tuple $\vec{t}$

Probabilistic query evaluation

- Inputs:
  - a database D of TID instances
  - a relational algebra query Q
  - a result tuple $\vec{t}$
- Output : what is the probability that $\vec{t}$ is in $Q(D)$?

Probabilistic query evaluation

- Inputs:
  - a database D of TID instances
  - a relational algebra query Q
  - a result tuple $\vec{t}$
- Output : what is the probability that $\vec{t}$ is in $Q(D)$?

$\rightarrow$ What is the marginal probability of obtaining $\vec{t}$ as a result?

Probabilistic query evaluation example

TID instance U

date	prof	
04	S	0.8
04	A	0.2

Query Q

$\pi_{\mathbf{prof}}(U)$

Tuple $\vec{t}$

S

Probabilistic query evaluation example

TID instance U

date	prof	
04	S	0.8
04	A	0.2

Query Q

$\pi_{\textbf{prof}}(U)$

Tuple $\vec{t}$

S

→ The marginal probability is 0.8

Marginal probabilities vs TID representations

Here's another example:

TID instance U'

date	prof	
04	S	1
04	A	1

TID instance V

date	
04	0.5

Query Q

$\pi_{\mathbf{prof}}(U' \bowtie V)$

Tuple $\vec{t}$

S
and
A

Marginal probabilities vs TID representations

Here's another example:

TID instance U'

date	prof	
04	S	1
04	A	1

TID instance V

date	
04	0.5

Query Q

$\pi_{\mathbf{prof}}(U' \bowtie V)$

Tuple $\vec{t}$

S
and
A

→ The marginal probability of S is 0.5

→ The marginal probability of A is also 0.5

Marginal probabilities vs TID representations

Here's another example:

TID instance U'

date	prof	
04	S	1
04	A	1

TID instance V

date	
04	0.5

Query Q

$\pi_{\textbf{prof}}(U' \bowtie V)$

Tuple $\vec{t}$

S
and
A

- → The marginal probability of S is 0.5
- → The marginal probability of A is also 0.5
- → Caution: It does not mean that the result is the TID instance at the right!

prof	
S	0.5
A	0.5

Motivation for probabilistic query evaluation

- Answers the intuitive question "what is the probability of this"?
- Often more interesting than the correlations between worlds

Motivation for probabilistic query evaluation

- Answers the intuitive question “what is the probability of this”?
- Often more interesting than the correlations between worlds

→ How to compute these probabilities?

Naive probabilistic query evaluation

- Compute the probabilistic instance represented by the input
  - → Finite number of possible worlds

Naive probabilistic query evaluation

- Compute the probabilistic instance represented by the input
  - → Finite number of possible worlds
- Run the query over each possible world
  - → Check if the result tuple is in the output

Naive probabilistic query evaluation

- Compute the probabilistic instance represented by the input
  → Finite number of possible worlds
- Run the query over each possible world
  → Check if the result tuple is in the output
- Sum the probabilities of all worlds that contain the output tuple

Naive probabilistic query evaluation example

TID instance U

date	prof	
04	S	0.8
04	A	0.2

Query Q

$\pi_{\mathbf{prof}}(U)$

Tuple $\vec{t}$

S

Naive probabilistic query evaluation example

TID instance U

date	prof	
04	S	0.8
04	A	0.2

Query Q

$\pi_{\mathbf{prof}}(U)$

Tuple $\vec{t}$

S

Probabilistic relation $Q(U)$:

prof
S
A
0.8×0.2

prof
A
$(1 - 0.8) \times 0.2$

prof
S
$0.8 \times (1 - 0.2)$

prof
$(1 - 0.8) \times (1 - 0.2)$

Naive probabilistic query evaluation example

TID instance U

date	prof	
04	S	0.8
04	A	0.2

Query Q

$\pi_{\mathbf{prof}}(U)$

Tuple $\vec{t}$

S

Probabilistic relation $Q(U)$:

prof
S
A

0.8×0.2

prof
S
A

$(1 - 0.8) \times 0.2$

prof
S
A

$0.8 \times (1 - 0.2)$

prof
S
A

$(1 - 0.8) \times (1 - 0.2)$

Total probability that $\vec{t}$ is in $Q(U)$: $0.8 \times 0.2 + 0.8 \times (1 - 0.2) = 0.8$

Naive evaluation advantages and drawbacks

- Naive evaluation is always possible

Naive evaluation advantages and drawbacks

- Naive evaluation is always possible
- However, it takes exponential time in general
  - → Even if the query output has few possible worlds!
  - → Feasible if the input has few possible worlds (few tuples)

Naive evaluation advantages and drawbacks

- Naive evaluation is always possible
- However, it takes exponential time in general
  - → Even if the query output has few possible worlds!
  - → Feasible if the input has few possible worlds (few tuples)
- Probabilistic query evaluation is computationally intractable so it is unlikely that we can beat naive evaluation in general

Naive evaluation advantages and drawbacks

- Naive evaluation is always possible
- However, it takes exponential time in general
  - → Even if the query output has few possible worlds!
  - → Feasible if the input has few possible worlds (few tuples)
- Probabilistic query evaluation is computationally intractable so it is unlikely that we can beat naive evaluation in general
  - → More efficient methods for special cases

Outline

Extensional evaluation idea

- Sometimes we can compute the probabilities at each step:

Extensional evaluation idea

- Sometimes we can compute the probabilities at each step:

U

date	prof	
04	S	0.8
04	A	0.2

Extensional evaluation idea

- Sometimes we can compute the probabilities at each step:

U

date	prof	
04	S	0.8
04	A	0.2

V

student	
1	0.4
2	0.6

Extensional evaluation idea

- Sometimes we can compute the probabilities at each step:

U

date	prof	
04	S	0.8
04	A	0.2

V

student	
1	0.4
2	0.6

$U \times V$

date	prof	student
04	S	1
04	S	2
04	A	1
04	A	2

Extensional evaluation idea

- Sometimes we can compute the probabilities at each step:

U

date	prof	
04	S	0.8
04	A	0.2

V

student	
1	0.4
2	0.6

$U \times V$

date	prof	student	
04	S	1	0.8×0.4
04	S	2	
04	A	1	
04	A	2	

Extensional evaluation idea

- Sometimes we can compute the probabilities at each step:

U

date	prof	
04	S	0.8
04	A	0.2

V

student	
1	0.4
2	0.6

$U \times V$

date	prof	student	
04	S	1	0.8×0.4
04	S	2	0.8×0.6
04	A	1	
04	A	2	

Extensional evaluation idea

- Sometimes we can compute the probabilities at each step:

U

date	prof	
04	S	0.8
04	A	0.2

V

student	
1	0.4
2	0.6

$U \times V$

date	prof	student	
04	S	1	0.8×0.4
04	S	2	0.8×0.6
04	A	1	0.2×0.4
04	A	2	

Extensional evaluation idea

- Sometimes we can compute the probabilities at each step:

U

date	prof	
04	S	0.8
04	A	0.2

V

student	
1	0.4
2	0.6

$U \times V$

date	prof	student	
04	S	1	0.8×0.4
04	S	2	0.8×0.6
04	A	1	0.2×0.4
04	A	2	0.2×0.6

Query independence

- We say that queries Q and Q' are syntactically independent if no relation is used in both Q and Q'
  - → Example: $Q = R \bowtie S$ and $Q' = \pi_{\mathbf{a}}(T \times U)$
  - → Intuition: the tuples in Q and Q' are independent

Query independence

- We say that queries Q and Q' are syntactically independent if no relation is used in both Q and Q'
  - → Example: $Q = R \bowtie S$ and $Q' = \pi_{\mathbf{a}}(T \times U)$
  - → Intuition: the tuples in Q and Q' are independent
- Independent join: if Q and Q' are syntactically independent then we can compute $Q \bowtie Q'$ and $Q \times Q'$: multiply the probabilities

Query independence

- We say that queries Q and Q' are syntactically independent if no relation is used in both Q and Q'
  - → Example: $Q = R \bowtie S$ and $Q' = \pi_{\mathbf{a}}(T \times U)$
  - → Intuition: the tuples in Q and Q' are independent

- Independent join: if Q and Q' are syntactically independent then we can compute $Q \bowtie Q'$ and $Q \times Q'$: multiply the probabilities

U

date	pr	
04	S	0.8
04	A	0.2

Query independence

- We say that queries Q and Q' are syntactically independent if no relation is used in both Q and Q'
  - → Example: $Q = R \bowtie S$ and $Q' = \pi_{\mathbf{a}}(T \times U)$
  - → Intuition: the tuples in Q and Q' are independent

- Independent join: if Q and Q' are syntactically independent then we can compute $Q \bowtie Q'$ and $Q \times Q'$: multiply the probabilities

U

date	pr	
04	S	0.8
04	A	0.2

V

pr	st	
A	1	0.4
S	2	0.6

Query independence

- We say that queries Q and Q' are syntactically independent if no relation is used in both Q and Q'
  - → Example: $Q = R \bowtie S$ and $Q' = \pi_{\mathbf{a}}(T \times U)$
  - → Intuition: the tuples in Q and Q' are independent

- Independent join: if Q and Q' are syntactically independent then we can compute $Q \bowtie Q'$ and $Q \times Q'$: multiply the probabilities

U

date	pr	
04	S	0.8
04	A	0.2

V

pr	st	
A	1	0.4
S	2	0.6

$U \bowtie V$

date	pr	st
04	S	2
04	A	1

Query independence

- We say that queries Q and Q' are syntactically independent if no relation is used in both Q and Q'
  - → Example: $Q = R \bowtie S$ and $Q' = \pi_{\mathbf{a}}(T \times U)$
  - → Intuition: the tuples in Q and Q' are independent
- Independent join: if Q and Q' are syntactically independent then we can compute $Q \bowtie Q'$ and $Q \times Q'$: multiply the probabilities

U

date	pr	
04	S	0.8
04	A	0.2

V

pr	st	
A	1	0.4
S	2	0.6

$U \bowtie V$

date	pr	st	
04	S	2	0.8×0.6
04	A	1	

Query independence

- We say that queries Q and Q' are syntactically independent if no relation is used in both Q and Q'
  → Example: $Q = R \bowtie S$ and $Q' = \pi_{\mathbf{a}}(T \times U)$
  → Intuition: the tuples in Q and Q' are independent

- Independent join: if Q and Q' are syntactically independent then we can compute $Q \bowtie Q'$ and $Q \times Q'$: multiply the probabilities

U

date	pr	
04	S	0.8
04	A	0.2

V

pr	st	
A	1	0.4
S	2	0.6

$U \bowtie V$

date	pr	st	
04	S	2	0.8×0.6
04	A	1	0.2×0.4

More query independence

- Independent union: for syntactically independent Q and Q' we can compute $Q \cup Q'$ using the rule for independent OR

More query independence

- Independent union: for syntactically independent Q and Q' we can compute $Q \cup Q'$ using the rule for independent OR

U

date	prof	
04	S	0.8
04	A	0.2

More query independence

- Independent union: for syntactically independent Q and Q' we can compute $Q \cup Q'$ using the rule for independent OR

U

date	**prof**	
04	S	0.8
04	A	0.2

V

date	**prof**	
04	A	0.4
11	A	0.2

More query independence

- Independent union: for syntactically independent Q and Q' we can compute $Q \cup Q'$ using the rule for independent OR

U

date	prof	
04	S	0.8
04	A	0.2

V

date	prof	
04	A	0.4
11	A	0.2

$U \cup V$

date	prof
04	S
04	A
11	A

More query independence

- Independent union: for syntactically independent Q and Q' we can compute $Q \cup Q'$ using the rule for independent OR

U

date	prof	
04	S	0.8
04	A	0.2

V

date	prof	
04	A	0.4
11	A	0.2

$U \cup V$

date	prof	
04	S	0.8
04	A	
11	A	

More query independence

- Independent union: for syntactically independent Q and Q' we can compute $Q \cup Q'$ using the rule for independent OR

U

date	prof	
04	S	0.8
04	A	0.2

V

date	prof	
04	A	0.4
11	A	0.2

$U \cup V$

date	prof	
04	S	0.8
04	A	$1-(1-0.2)\times(1-0.4)$
11	A	

More query independence

- Independent union: for syntactically independent Q and Q' we can compute $Q \cup Q'$ using the rule for independent OR

U

date	prof	
04	S	0.8
04	A	0.2

V

date	prof	
04	A	0.4
11	A	0.2

$U \cup V$

date	prof	
04	S	0.8
04	A	$1-(1-0.2)\times(1-0.4)$
11	A	0.2

Selection

Selection can just be applied in the straightforward way:

Selection

Selection can just be applied in the straightforward way:

U

date	prof	
04	S	0.8
04	A	0.2

Selection

Selection can just be applied in the straightforward way:

U

date	prof	
04	S	0.8
04	A	0.2

$\sigma_{\mathbf{prof}=\text{“S”}}(U)$

date	prof

Selection

Selection can just be applied in the straightforward way:

U

date	prof	
04	S	0.8
04	A	0.2

$\sigma_{\mathbf{prof}=\text{"S"}}(U)$

date	prof	
04	S	0.8

Independent projection

- Self-join-free conjunctive query: a join ($\bowtie$) of projections (π) that does not use the same relation name twice:
  - $\rightarrow$ Example: $Q = R \bowtie S \bowtie \pi_{\mathbf{a}}(T)$

Independent projection

- Self-join-free conjunctive query: a join ($\bowtie$) of projections (π) that does not use the same relation name twice:
  - $\rightarrow$ Example: $Q = R \bowtie S \bowtie \pi_{\mathbf{a}}(T)$
- A separator is an attribute that occurs in all tables of the join:
  - $\rightarrow$ Example: if $R(\mathbf{a}, \mathbf{b}), S(\mathbf{a}), T(\mathbf{a}, \mathbf{c})$ then $\mathbf{a}$ is a separator of Q

Independent projection

- Self-join-free conjunctive query: a join ($\bowtie$) of projections (π) that does not use the same relation name twice:
  - → Example: $Q = R \bowtie S \bowtie \pi_{\mathbf{a}}(T)$
- A separator is an attribute that occurs in all tables of the join:
  - → Example: if $R(\mathbf{a}, \mathbf{b}), S(\mathbf{a}), T(\mathbf{a}, \mathbf{c})$ then $\mathbf{a}$ is a separator of Q
- If Q is a self-join-free conjunctive query and $\mathbf{a}$ is a separator then $\pi_{-\mathbf{a}}(Q)$ (projecting away the attribute $\mathbf{a}$) can be computed using independent OR

Independent projection example

U

date	prof	
04	S	0.8
04	A	0.2
11	S	0.4
11	A	0.6

Independent projection example

U

date	prof	
04	S	0.8
04	A	0.2
11	S	0.4
11	A	0.6

$\pi_{\mathbf{date}}(U)$

date
04
11

Independent projection example

U

date	prof	
04	S	0.8
04	A	0.2
11	S	0.4
11	A	0.6

$\pi_{\mathbf{date}}(U)$

date	
04	$1-(1-0.8)\times(1-0.2)$
11	

Independent projection example

U		
date	**prof**	
04	S	0.8
04	A	0.2
11	S	0.4
11	A	0.6

$\pi_{\mathbf{date}}(U)$	
date	
04	$1-(1-0.8)\times(1-0.2)$
11	$1-(1-0.4)\times(1-0.6)$

Another independent projection example

Consider the two tables:

U

date	prof	
04	S	1/2

Another independent projection example

Consider the two tables:

U

date	prof	
04	S	1/2

V

prof	cause	
S	illness	1/2
S	bahamas	1/2

Another independent projection example

Consider the two tables:

U

date	prof	
04	S	1/2

V

prof	cause	
S	illness	1/2
S	bahamas	1/2

- Query: $Q(U, V) = \pi_{\mathbf{date,prof}}(U \bowtie V)$
- Can be rewritten as: $U \bowtie \pi_{-\mathbf{cause}}(V)$

Another independent projection example

Consider the two tables:

U

date	prof	
04	S	1/2

V

prof	cause	
S	illness	1/2
S	bahamas	1/2

- Query: $Q(U, V) = \pi_{\mathbf{date,prof}}(U \bowtie V)$
- Can be rewritten as: $U \bowtie \pi_{-\mathbf{cause}}(V)$

$\pi_{-\mathbf{cause}}(V)$

prof	
S	3/4

$U \bowtie \pi_{-\mathbf{cause}}(V)$

date	prof	
04	S	3/8

The choice of plan matters!

- Query: $Q(U, V) = \pi_{\mathbf{date,prof}}(U \bowtie V)$
- Can be rewritten as $\pi_{-\mathbf{cause}}(U \bowtie V)$ instead of $U \bowtie \pi_{-\mathbf{cause}}(V)$

The choice of plan matters!

- Query: $Q(U, V) = \pi_{\mathbf{date,prof}}(U \bowtie V)$
- Can be rewritten as $\pi_{-\mathbf{cause}}(U \bowtie V)$ instead of $U \bowtie \pi_{-\mathbf{cause}}(V)$

U

date	prof	
04	S	1/2

The choice of plan matters!

- Query: $Q(U, V) = \pi_{\textbf{date},\textbf{prof}}(U \bowtie V)$
- Can be rewritten as $\pi_{-\textbf{cause}}(U \bowtie V)$ instead of $U \bowtie \pi_{-\textbf{cause}}(V)$

U

date	**prof**	
04	S	1/2

V

prof	**cause**	
S	illness	1/2
S	bahamas	1/2

The choice of plan matters!

- Query: $Q(U, V) = \pi_{\mathbf{date},\mathbf{prof}}(U \bowtie V)$
- Can be rewritten as $\pi_{-\mathbf{cause}}(U \bowtie V)$ instead of $U \bowtie \pi_{-\mathbf{cause}}(V)$

U

date	prof	
04	S	1/2

V

prof	cause	
S	illness	1/2
S	bahamas	1/2

$U \bowtie V$

date	prof	cause
04	S	illness
04	S	bahamas

The choice of plan matters!

- Query: $Q(U, V) = \pi_{\textbf{date,prof}}(U \bowtie V)$
- Can be rewritten as $\pi_{-\textbf{cause}}(U \bowtie V)$ instead of $U \bowtie \pi_{-\textbf{cause}}(V)$

U

date	**prof**	
04	S	1/2

V

prof	**cause**	
S	illness	1/2
S	bahamas	1/2

$U \bowtie V$

date	**prof**	**cause**	
04	S	illness	1/4
04	S	bahamas	

The choice of plan matters!

- Query: $Q(U, V) = \pi_{\mathbf{date,prof}}(U \bowtie V)$
- Can be rewritten as $\pi_{-\mathbf{cause}}(U \bowtie V)$ instead of $U \bowtie \pi_{-\mathbf{cause}}(V)$

U

date	**prof**	
04	S	1/2

V

prof	**cause**	
S	illness	1/2
S	bahamas	1/2

$U \bowtie V$

date	**prof**	**cause**	
04	S	illness	1/4
04	S	bahamas	1/4

The choice of plan matters!

- Query: $Q(U, V) = \pi_{\textbf{date,prof}}(U \bowtie V)$
- Can be rewritten as $\pi_{-\textbf{cause}}(U \bowtie V)$ instead of $U \bowtie \pi_{-\textbf{cause}}(V)$

U

date	**prof**	
04	S	1/2

V

prof	**cause**	
S	illness	1/2
S	bahamas	1/2

$U \bowtie V$

date	**prof**	**cause**	
04	S	illness	1/4
04	S	bahamas	1/4

$\pi_{-\textbf{cause}}(U \bowtie V)$

date	**prof**
04	S

The choice of plan matters!

- Query: $Q(U, V) = \pi_{\mathbf{date,prof}}(U \bowtie V)$
- Can be rewritten as $\pi_{-\mathbf{cause}}(U \bowtie V)$ instead of $U \bowtie \pi_{-\mathbf{cause}}(V)$

U

date	**prof**	
04	S	1/2

V

prof	**cause**	
S	illness	1/2
S	bahamas	1/2

$U \bowtie V$

date	**prof**	**cause**	
04	S	illness	1/4
04	S	bahamas	1/4

$\pi_{-\mathbf{cause}}(U \bowtie V)$

date	**prof**	
04	S	7/16 ??

The choice of plan matters!

- Query: $Q(U, V) = \pi_{\mathbf{date,prof}}(U \bowtie V)$
- Can be rewritten as $\pi_{-\mathbf{cause}}(U \bowtie V)$ instead of $U \bowtie \pi_{-\mathbf{cause}}(V)$

U

date	prof	
04	S	1/2

V

prof	cause	
S	illness	1/2
S	bahamas	1/2

$U \bowtie V$

date	prof	cause	
04	S	illness	1/4
04	S	bahamas	1/4

$\pi_{-\mathbf{cause}}(U \bowtie V)$

date	prof	
04	S	7/16 ??

→ The last projection is not independent, so incorrect result!

Safe plans

- A safe plan for a query Q is a way to implement Q using the extensional operators:

Safe plans

- A safe plan for a query Q is a way to implement Q using the extensional operators:
  - It must use them correctly, e.g., respecting independence

Safe plans

- A safe plan for a query Q is a way to implement Q using the extensional operators:
  - It must use them correctly, e.g., respecting independence
  - It must be equivalent to the desired query Q

Safe plans

- A safe plan for a query Q is a way to implement Q using the extensional operators:
  - It must use them correctly, e.g., respecting independence
  - It must be equivalent to the desired query Q

→ With a safe plan, we can compute the marginal probability of all query results

Do all queries have a safe plan?

- Relations $R(\mathbf{a}), S(\mathbf{a}, \mathbf{b}), T(\mathbf{b})$

Do all queries have a safe plan?

- Relations $R(\mathbf{a}), S(\mathbf{a}, \mathbf{b}), T(\mathbf{b})$
- Query $Q = \pi_{-\mathbf{a}}(\pi_{-\mathbf{b}}(R \bowtie S \bowtie T))$

Do all queries have a safe plan?

- Relations $R(\mathbf{a}), S(\mathbf{a}, \mathbf{b}), T(\mathbf{b})$
- Query $Q = \pi_{-\mathbf{a}}(\pi_{-\mathbf{b}}(R \bowtie S \bowtie T))$
- Does Q have a safe plan?

Do all queries have a safe plan?

- Relations $R(\mathbf{a}), S(\mathbf{a}, \mathbf{b}), T(\mathbf{b})$
- Query $Q = \pi_{-\mathbf{a}}(\pi_{-\mathbf{b}}(R \bowtie S \bowtie T))$
- Does Q have a safe plan?
  - If we do the joins first then no projection is independent

Do all queries have a safe plan?

- Relations $R(\mathbf{a}), S(\mathbf{a}, \mathbf{b}), T(\mathbf{b})$
- Query $Q = \pi_{-\mathbf{a}}(\pi_{-\mathbf{b}}(R \bowtie S \bowtie T))$
- Does Q have a safe plan?
  - If we do the joins first then no projection is independent
  - If we write Q as $\pi_{-\mathbf{a}}(R \bowtie \pi_{-\mathbf{b}}(S \bowtie T)))$
    then the projection is not safe

Do all queries have a safe plan?

- Relations $R(\mathbf{a}), S(\mathbf{a}, \mathbf{b}), T(\mathbf{b})$
- Query $Q = \pi_{-\mathbf{a}}(\pi_{-\mathbf{b}}(R \bowtie S \bowtie T))$
- Does Q have a safe plan?
  - If we do the joins first then no projection is independent
  - If we write Q as $\pi_{-\mathbf{a}}(R \bowtie \pi_{-\mathbf{b}}(S \bowtie T)))$ then the projection is not safe
  - Same problem for $\pi_{-\mathbf{b}}(\pi_{-\mathbf{a}}(R \bowtie S) \bowtie T)$

Do all queries have a safe plan?

- Relations $R(\mathbf{a}), S(\mathbf{a}, \mathbf{b}), T(\mathbf{b})$
- Query $Q = \pi_{-\mathbf{a}}(\pi_{-\mathbf{b}}(R \bowtie S \bowtie T))$
- Does Q have a safe plan?
  - If we do the joins first then no projection is independent
  - If we write Q as $\pi_{-\mathbf{a}}(R \bowtie \pi_{-\mathbf{b}}(S \bowtie T)))$
    then the projection is not safe
  - Same problem for $\pi_{-\mathbf{b}}(\pi_{-\mathbf{a}}(R \bowtie S) \bowtie T)$

→ In fact Q is intractable and it has no safe plan

Extensional query evaluation summary

- Extensional query evaluation:
  - Express the query as a safe plan with the extensional operators
  - Compute the query results and their probabilities via the plan
  - The probabilities are correct because the plan is safe

Extensional query evaluation summary

- Extensional query evaluation:
  - Express the query as a safe plan with the extensional operators
  - Compute the query results and their probabilities via the plan
  - The probabilities are correct because the plan is safe

- Summary of safe operators:
  - Product and join of syntactically independent queries

    → Product of independent probabilities
  - Union of syntactically independent queries

    → Independent OR of the probabilities
  - Projecting away a separator attribute

    → Independent OR because the tuples in each group are independent
  - Applying selection in the straightforward way
  - Also other rules: negation, inclusion-exclusion, etc.

Extensional query evaluation summary

- Extensional query evaluation:
  - Express the query as a safe plan with the extensional operators
  - Compute the query results and their probabilities via the plan
  - The probabilities are correct because the plan is safe
- Summary of safe operators:
  - Product and join of syntactically independent queries

    → Product of independent probabilities
  - Union of syntactically independent queries

    → Independent OR of the probabilities
  - Projecting away a separator attribute

    → Independent OR because the tuples in each group are independent
  - Applying selection in the straightforward way
  - Also other rules: negation, inclusion-exclusion, etc.

→ Not all queries have safe plans

Outline

Idea of intensional query evaluation

- We cannot always compute directly the probabilities of results
- Idea:
  - Compute a lineage expression (Boolean provenance!) for each output tuple
    describing the possible worlds where it appears
  - Compute the probability of these lineage expressions

Idea of intensional query evaluation

- We cannot always compute directly the probabilities of results
- Idea:
  - Compute a lineage expression (Boolean provenance!) for each output tuple
    describing the possible worlds where it appears
  - Compute the probability of these lineage expressions
- Advantages:
  - Intensional evaluation is always possible (but not always efficient)
  - Intensional evaluation is more modular:
    - → Compute the lineage expression (no probabilities)
    - → Use any model counting method or software

Idea of intensional query evaluation

- We cannot always compute directly the probabilities of results
- Idea:
  - Compute a lineage expression (Boolean provenance!) for each output tuple
    describing the possible worlds where it appears
  - Compute the probability of these lineage expressions
- Advantages:
  - Intensional evaluation is always possible (but not always efficient)
  - Intensional evaluation is more modular:
    - → Compute the lineage expression (no probabilities)
    - → Use any model counting method or software
- Disadvantages:
  - Two steps: (1.) compute the lineage; (2.) compute the probability
  - The lineage expression loses information about the query

Reminder: pc-tables

Remember that a TID is a special case of a pc-table:

U

date	**prof**	$x_1 : 0.8, x_2 : 0.2$
04	S	x_1
04	A	x_2

Remember that pc-tables are a strong representation system (same rules as for pc-tables for relational algebra operators)

pc-table query example

U

date	**prof**	$x_1 : 0.8,\ x_2 : 0.2$
04	S	x_1
04	A	x_2

pc-table query example

U

date	prof	$x_1 : 0.8,\ x_2 : 0.2$
04	S	x_1
04	A	x_2

$\pi_{\mathbf{date}}(U)$

date	$x_1 : 0.8,\ x_2 : 0.2$
04	$x_1 \vee x_2$

Lineage expression

$\pi_{\mathbf{date}}(U)$	
date	$x_1 : 0.8,\ x_2 : 0.2$
04	$x_1 \vee x_2$

- The lineage expression $x_1 \vee x_2$ describes the possible worlds where the tuple 04 appears.

Lineage expression

$\pi_{\textbf{date}}(U)$	
date	$x_1 : 0.8$, $x_2 : 0.2$
04	$x_1 \vee x_2$

- The lineage expression $x_1 \vee x_2$ describes the possible worlds where the tuple 04 appears.
- The probability that $x_1 \vee x_2$ is true is exactly the probability that this tuple is in the result

Intensional query evaluation

- Translate the TID to a pc-table

Intensional query evaluation

- Translate the TID to a pc-table
- Evaluate the query on the pc-table using c-table rules

Intensional query evaluation

- Translate the TID to a pc-table
- Evaluate the query on the pc-table using c-table rules
- Compute the probability $P(\phi)$ of the lineage expression ϕ of the output tuple under consideration

Intensional query evaluation

- Translate the TID to a pc-table
- Evaluate the query on the pc-table using c-table rules
- Compute the probability $P(\phi)$ of the lineage expression ϕ of the output tuple under consideration

→ We have reduced probabilistic query evaluation to computing the probability that a Boolean formula is true

How to compute the probability of a lineage expression?

Many ways to compute the probability $P(\phi)$:

- Naive method: enumerate all possibilities (exponential)

How to compute the probability of a lineage expression?

Many ways to compute the probability $\mathrm{P}(\phi)$:

- Naive method: enumerate all possibilities (exponential)
- Use some simple intensional rules
  - → e.g., $\mathrm{P}(\phi(x,y,z) \wedge \psi(x',y',z')) = \mathrm{P}(\phi(x,y,z)) \times \mathrm{P}(\psi(x',y',z'))$ thanks to independence

How to compute the probability of a lineage expression?

Many ways to compute the probability $\mathrm{P}(\phi)$:

- Naive method: enumerate all possibilities (exponential)
- Use some simple intensional rules
  → e.g., $\mathrm{P}(\phi(x, y, z) \wedge \psi(x', y', z')) = \mathrm{P}(\phi(x, y, z)) \times \mathrm{P}(\psi(x', y', z'))$ thanks to independence
- Compile the lineage expression in a tractable formalism
  - read-once formulas
  - tractable circuit classes
  - binary decision diagrams

How to compute the probability of a lineage expression?

Many ways to compute the probability $\mathrm{P}(\phi)$:

- Naive method: enumerate all possibilities (exponential)
- Use some simple intensional rules
  → e.g., $\mathrm{P}(\phi(x,y,z) \wedge \psi(x',y',z')) = \mathrm{P}(\phi(x,y,z)) \times \mathrm{P}(\psi(x',y',z'))$ thanks to independence
- Compile the lineage expression in a tractable formalism
  - read-once formulas
  - tractable circuit classes
  - binary decision diagrams
- Approximate the probability of the lineage expression

How to compute the probability of a lineage expression?

Many ways to compute the probability $P(\phi)$:

- Naive method: enumerate all possibilities (exponential)
- Use some simple intensional rules
  $\rightarrow$ e.g., $P(\phi(x, y, z) \wedge \psi(x', y', z')) = P(\phi(x, y, z)) \times P(\psi(x', y', z'))$ thanks to independence
- Compile the lineage expression in a tractable formalism
  - read-once formulas
  - tractable circuit classes
  - binary decision diagrams
- Approximate the probability of the lineage expression
- Use an external weighted model counter

Naive evaluation

Example: formula $\phi = x_1 \vee x_2$ with $\mathrm{P}(x_1 = 1) = 0.8$ and $\mathrm{P}(x_2 = 1) = 0.2$

- If x_1 is true and x_2 is true, the formula is true
  - $\rightarrow$ Probability 0.8×0.2

Naive evaluation

Example: formula $\phi = x_1 \vee x_2$ with $\mathrm{P}(x_1 = 1) = 0.8$ and $\mathrm{P}(x_2 = 1) = 0.2$

- If x_1 is true and x_2 is true, the formula is true
  - → Probability 0.8×0.2
- If x_1 is true and x_2 is false, the formula is true
  - → Probability $0.8 \times (1 - 0.2)$

Naive evaluation

Example: formula $\phi = x_1 \lor x_2$ with $\mathrm{P}(x_1 = 1) = 0.8$ and $\mathrm{P}(x_2 = 1) = 0.2$

- If x_1 is true and x_2 is true, the formula is true
  $\rightarrow$ Probability 0.8×0.2
- If x_1 is true and x_2 is false, the formula is true
  $\rightarrow$ Probability $0.8 \times (1 - 0.2)$
- If x_1 is false and x_2 is true, the formula is true
  $\rightarrow$ Probability $(1 - 0.8) \times 0.2$

Naive evaluation

Example: formula $\phi = x_1 \vee x_2$ with $\mathrm{P}(x_1 = 1) = 0.8$ and $\mathrm{P}(x_2 = 1) = 0.2$

- If x_1 is true and x_2 is true, the formula is true
  → Probability 0.8×0.2
- If x_1 is true and x_2 is false, the formula is true
  → Probability $0.8 \times (1 - 0.2)$
- If x_1 is false and x_2 is true, the formula is true
  → Probability $(1 - 0.8) \times 0.2$
- If x_1 is false and x_2 is false, the formula is false
  → Probability $(1 - 0.8) \times (1 - 0.2)$

Naive evaluation

Example: formula $\phi = x_1 \lor x_2$ with $\mathrm{P}(x_1 = 1) = 0.8$ and $\mathrm{P}(x_2 = 1) = 0.2$

- If x_1 is true and x_2 is true, the formula is true
  $\rightarrow$ Probability 0.8×0.2
- If x_1 is true and x_2 is false, the formula is true
  $\rightarrow$ Probability $0.8 \times (1 - 0.2)$
- If x_1 is false and x_2 is true, the formula is true
  $\rightarrow$ Probability $(1 - 0.8) \times 0.2$
- If x_1 is false and x_2 is false, the formula is false
  $\rightarrow$ Probability $(1 - 0.8) \times (1 - 0.2)$

$\rightarrow$ $\mathrm{P}(\phi) = 0.8 \times 0.2 + 0.8 \times (1 - 0.2) + (1 - 0.8) \times 0.2$

Naive evaluation

Example: formula $\phi = x_1 \vee x_2$ with $\mathrm{P}(x_1 = 1) = 0.8$ and $\mathrm{P}(x_2 = 1) = 0.2$

- If x_1 is true and x_2 is true, the formula is true
  - $\rightarrow$ Probability 0.8×0.2
- If x_1 is true and x_2 is false, the formula is true
  - $\rightarrow$ Probability $0.8 \times (1 - 0.2)$
- If x_1 is false and x_2 is true, the formula is true
  - $\rightarrow$ Probability $(1 - 0.8) \times 0.2$
- If x_1 is false and x_2 is false, the formula is false
  - $\rightarrow$ Probability $(1 - 0.8) \times (1 - 0.2)$

$\rightarrow$ $\mathrm{P}(\phi) = 0.8 \times 0.2 + 0.8 \times (1 - 0.2) + (1 - 0.8) \times 0.2 = 0.84$

Intensional rule definitions

- ϕ and ψ are syntactically independent if they have no variables in common
  - → E.g., $\phi(x, y, z)$ and $\psi(x', y', z')$

Intensional rule definitions

- ϕ and ψ are syntactically independent if they have no variables in common
  → E.g., $\phi(x, y, z)$ and $\psi(x', y', z')$
- ϕ and ψ are mutually exclusive if $\phi \land \psi$ is unsatisfiable
  → E.g., $\phi = x \land y$ and $\psi = \neg x \land (y \lor z)$

Intensional rule definitions

- ϕ and ψ are syntactically independent if they have no variables in common
  → E.g., $\phi(x, y, z)$ and $\psi(x', y', z')$
- ϕ and ψ are mutually exclusive if $\phi \land \psi$ is unsatisfiable
  → E.g., $\phi = x \land y$ and $\psi = \neg x \land (y \lor z)$
- $\phi_{|x=0}$ is the result of replacing x by 0 in ϕ (and likewise for $\phi_{|x=1}$)
  → E.g., for $\phi = \neg x \land (y \lor z)$, we have $\phi_{|x=0} = y \lor z$ and $\phi_{|x=1} = \bot$

Intensional rules

- Independent AND: if ϕ and ψ are syntactically independent then: $\mathrm{P}(\phi \wedge \psi) = \mathrm{P}(\phi) \times \mathrm{P}(\psi)$

Intensional rules

- Independent AND: if ϕ and ψ are syntactically independent then: $\mathrm{P}(\phi \wedge \psi) = \mathrm{P}(\phi) \times \mathrm{P}(\psi)$
- Independent OR: if ϕ and ψ are syntactically independent then: $\mathrm{P}(\phi \vee \psi) = 1 - (1 - \mathrm{P}(\phi)) \times (1 - \mathrm{P}(\psi))$

Intensional rules

- Independent AND: if ϕ and ψ are syntactically independent then: $\mathrm{P}(\phi \wedge \psi) = \mathrm{P}(\phi) \times \mathrm{P}(\psi)$
- Independent OR: if ϕ and ψ are syntactically independent then: $\mathrm{P}(\phi \vee \psi) = 1 - (1 - \mathrm{P}(\phi)) \times (1 - \mathrm{P}(\psi))$
- Mutually exclusive OR: if ϕ and ψ are mutually exclusive then: $\mathrm{P}(\phi \vee \psi) = \mathrm{P}(\phi) + \mathrm{P}(\psi)$

Intensional rules

- Independent AND: if ϕ and ψ are syntactically independent then: $\mathrm{P}(\phi \wedge \psi) = \mathrm{P}(\phi) \times \mathrm{P}(\psi)$
- Independent OR: if ϕ and ψ are syntactically independent then: $\mathrm{P}(\phi \vee \psi) = 1 - (1 - \mathrm{P}(\phi)) \times (1 - \mathrm{P}(\psi))$
- Mutually exclusive OR: if ϕ and ψ are mutually exclusive then: $\mathrm{P}(\phi \vee \psi) = \mathrm{P}(\phi) + \mathrm{P}(\psi)$
- Negation: for any ϕ, we have $\mathrm{P}(\neg\phi) = 1 - \mathrm{P}(\phi)$

Intensional rules

- Independent AND: if ϕ and ψ are syntactically independent then: $\mathrm{P}(\phi \wedge \psi) = \mathrm{P}(\phi) \times \mathrm{P}(\psi)$
- Independent OR: if ϕ and ψ are syntactically independent then: $\mathrm{P}(\phi \vee \psi) = 1 - (1 - \mathrm{P}(\phi)) \times (1 - \mathrm{P}(\psi))$
- Mutually exclusive OR: if ϕ and ψ are mutually exclusive then: $\mathrm{P}(\phi \vee \psi) = \mathrm{P}(\phi) + \mathrm{P}(\psi)$
- Negation: for any ϕ, we have $\mathrm{P}(\neg\phi) = 1 - \mathrm{P}(\phi)$
- Shannon expansion: for any ϕ and variable x, we have: $\mathrm{P}(\phi) = \mathrm{P}(x = 0) \times \mathrm{P}(\phi_{|x=0}) + \mathrm{P}(x = 1) \times \mathrm{P}(\phi_{|x=1})$

Application of intensional rules

- We can always compute probabilities with intensional rules

→ But Shannon expansions are costly and may be exponential

- The efficiency of these rules depends:
  - on how the lineage is written
  - on the order in which they are applied
- Note that these rules are a bit similar to the extensional rules

Tractable lineage formalisms

- Read-once formula: each variable occurs at most once
  → If the lineage is written in this way, we can compute the probability with independent AND, independent OR, negation

Tractable lineage formalisms

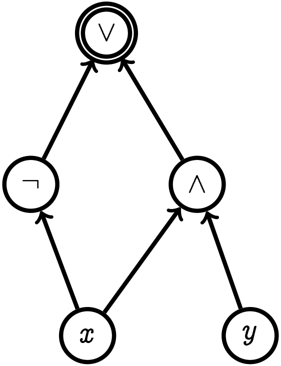
- Read-once formula: each variable occurs at most once
  → If the lineage is written in this way, we can compute the probability with independent AND, independent OR, negation
- Tractable Boolean circuit representations of lineages

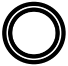
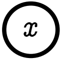
Tractable lineage formalisms

- Read-once formula: each variable occurs at most once
  - → If the lineage is written in this way, we can compute the probability with independent AND, independent OR, negation
- Tractable Boolean circuit representations of lineages
- Tractable representations as Binary decision diagrams

Boolean circuit representations

Circuits are just a way to represent Boolean formulas while factoring common subexpressions (more concise)

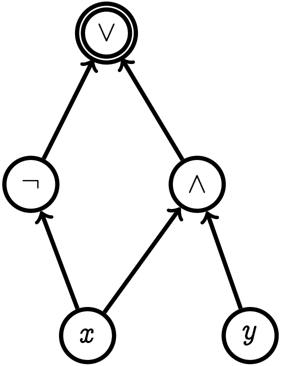


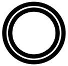
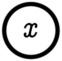
- Directed acyclic graph of gates
- Output gate: 
- Variable gates: 

- Internal gates: 

Boolean circuit representations

Circuits are just a way to represent Boolean formulas while factoring common subexpressions (more concise)



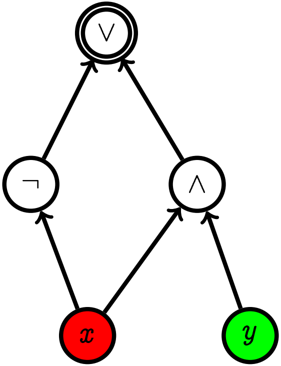
- Directed acyclic graph of gates
- Output gate: 
- Variable gates: 

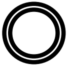
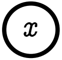
- Internal gates: 

- Valuation: function from variables to $\{0, 1\}$
  Example: $\nu = \{x \mapsto 0,\ y \mapsto 1\}\ldots$

Boolean circuit representations

Circuits are just a way to represent Boolean formulas while factoring common subexpressions (more concise)



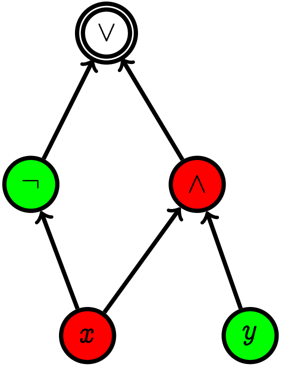
- Directed acyclic graph of gates
- Output gate: 
- Variable gates: 

- Internal gates: 

- Valuation: function from variables to $\{0, 1\}$
  Example: $\nu = \{x \mapsto 0,\ y \mapsto 1\}\ldots$

Boolean circuit representations

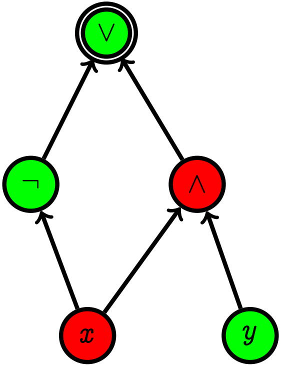
Circuits are just a way to represent Boolean formulas while factoring common subexpressions (more concise)

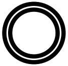
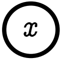


- Directed acyclic graph of gates
- Output gate:
- Variable gates: x
- Internal gates: $\vee$ $\wedge$ $\neg$
- Valuation: function from variables to $\{0, 1\}$
  Example: $\nu = \{x \mapsto 0,\ y \mapsto 1\}\dots$

Boolean circuit representations

Circuits are just a way to represent Boolean formulas while factoring common subexpressions (more concise)



- Directed acyclic graph of gates
- Output gate: 
- Variable gates: 

- Internal gates: 

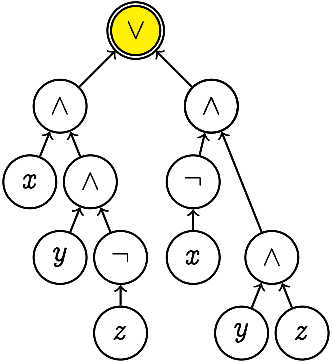
- Valuation: function from variables to $\{0, 1\}$
  Example: $\nu = \{x \mapsto 0,\ y \mapsto 1\}\ldots$ mapped to 1

Circuit restrictions

Tractable circuit class: d-DNNF:

- $\vee$ are all deterministic:

The inputs are mutually exclusive
(= no valuation ν makes two inputs simultaneously evaluate to 1)



Circuit restrictions

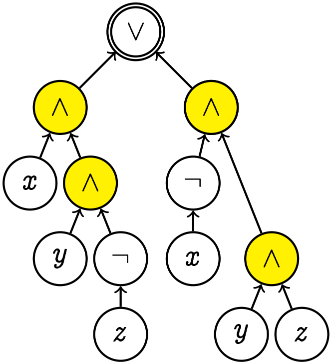
Tractable circuit class: d-DNNF:

- (∨) are all deterministic:

The inputs are mutually exclusive
(= no valuation ν makes two inputs simultaneously evaluate to 1)

- (∧) are all decomposable:

The inputs are independent
(= no variable x has a path to two different inputs)



Circuit restrictions

Tractable circuit class: d-DNNF:

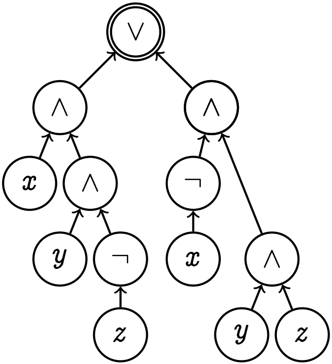
- $\vee$ are all deterministic:

The inputs are mutually exclusive
(= no valuation ν makes two inputs simultaneously evaluate to 1)

- $\wedge$ are all decomposable:

The inputs are independent
(= no variable x has a path to two different inputs)

$\rightarrow$ We can compute the probability of a d-DNNF with the intensional rules



Ordered Binary Decision Diagram (OBDD)

OBDD for a Boolean query Q on database I:
ordered decision diagram on the facts of I to decide whether Q holds

Ordered Binary Decision Diagram (OBDD)

OBDD for a Boolean query Q on database I:
ordered decision diagram on the facts of I to decide whether Q holds

$$Q : \pi_{\emptyset}(R \bowtie S \bowtie T)$$

R	
a	r_1
b	r_2
c	r_3

S		
a	a	s_1
b	v	s_2
b	w	s_3

T	
v	t_1
w	t_2
b	t_3

Ordered Binary Decision Diagram (OBDD)

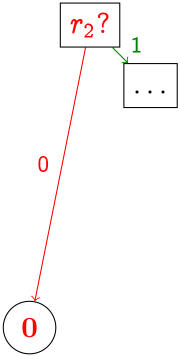
OBDD for a Boolean query Q on database I:
ordered decision diagram on the facts of I to decide whether Q holds

$$Q : \pi_{\emptyset}(R \bowtie S \bowtie T)$$

R	
a	r_1
b	r_2
c	r_3

S		
a	a	s_1
b	v	s_2
b	w	s_3

T	
v	t_1
w	t_2
b	t_3



Ordered Binary Decision Diagram (OBDD)

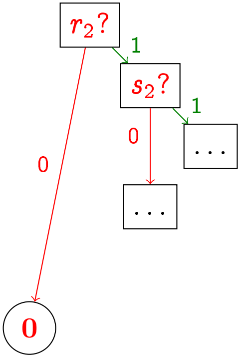
OBDD for a Boolean query Q on database I:
ordered decision diagram on the facts of I to decide whether Q holds

$$Q : \pi_\emptyset(R \bowtie S \bowtie T)$$

R	
a	r_1
b	r_2
c	r_3

S		
a	a	s_1
b	v	s_2
b	w	s_3

T	
v	t_1
w	t_2
b	t_3



Ordered Binary Decision Diagram (OBDD)

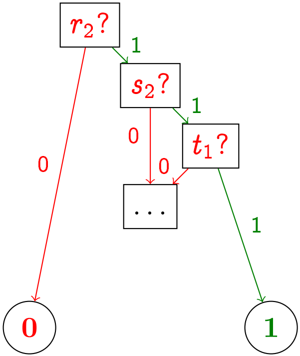
OBDD for a Boolean query Q on database I:
ordered decision diagram on the facts of I to decide whether Q holds

$$Q : \pi_\emptyset(R \bowtie S \bowtie T)$$

R	
a	r_1
b	r_2
c	r_3

S		
a	a	s_1
b	v	s_2
b	w	s_3

T	
v	t_1
w	t_2
b	t_3



Ordered Binary Decision Diagram (OBDD)

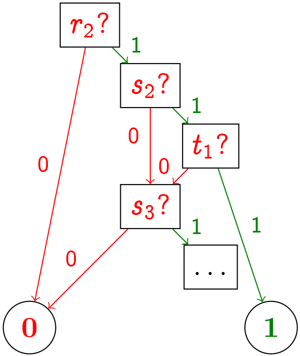
OBDD for a Boolean query Q on database I:
ordered decision diagram on the facts of I to decide whether Q holds

$$Q : \pi_{\emptyset}(R \bowtie S \bowtie T)$$

R	
a	r_1
b	r_2
c	r_3

S		
a	a	s_1
b	v	s_2
b	w	s_3

T	
v	t_1
w	t_2
b	t_3



Ordered Binary Decision Diagram (OBDD)

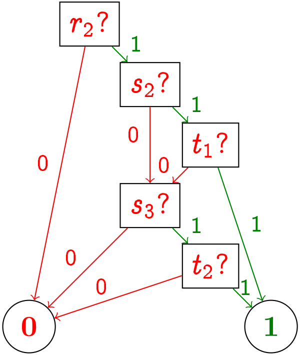
OBDD for a Boolean query Q on database I:
ordered decision diagram on the facts of I to decide whether Q holds

$$Q : \pi_\emptyset(R \bowtie S \bowtie T)$$

R	
a	r_1
b	r_2
c	r_3

S		
a	a	s_1
b	v	s_2
b	w	s_3

T	
v	t_1
w	t_2
b	t_3



Ordered Binary Decision Diagram (OBDD)

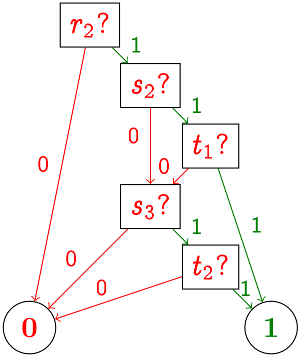
OBDD for a Boolean query Q on database I:
ordered decision diagram on the facts of I to decide whether Q holds

$$Q : \pi_\emptyset(R \bowtie S \bowtie T)$$

R	
a	r_1
b	r_2
c	r_3

S		
a	a	s_1
b	v	s_2
b	w	s_3

T	
v	t_1
w	t_2
b	t_3



→ We can compute the probability of an OBDD bottom-up

Approximation

- When it's too hard to compute the exact probability, we can approximate it

Approximation

- When it's too hard to compute the exact probability, we can approximate it
- One possibility is to compute a lower bound and upper bound:
  - $\max(\mathrm{P}(\phi), \mathrm{P}(\psi)) \leq \mathrm{P}(\phi \vee \psi) \leq \min(\mathrm{P}(\phi) + \mathrm{P}(\psi), 1)$
  - $\max(0, \mathrm{P}(\phi) + \mathrm{P}(\psi) - 1) \leq \mathrm{P}(\phi \wedge \psi) \leq \min(\mathrm{P}(\phi), \mathrm{P}(\psi))$ (by duality)
  - $\mathrm{P}(\neg\phi) = 1 - \mathrm{P}(\phi)$ (reminder)

Approximation by sampling

Another possibility is to approximate via Monte-Carlo sampling:

- Pick a random valuation according to the variable probabilities:
  - → Set $x_1 = 0$ with probability on $\mathrm{P}(x_1 = 0)$ and $x_1 = 1$ otherwise
  - → Repeat for the other variables

Approximation by sampling

Another possibility is to approximate via Monte-Carlo sampling:

- Pick a random valuation according to the variable probabilities:
  - → Set $x_1 = 0$ with probability on $\mathrm{P}(x_1 = 0)$ and $x_1 = 1$ otherwise
  - → Repeat for the other variables
- Evaluate the lineage formula ϕ under this valuation

Approximation by sampling

Another possibility is to approximate via Monte-Carlo sampling:

- Pick a random valuation according to the variable probabilities:
  - → Set $x_1 = 0$ with probability on $\mathrm{P}(x_1 = 0)$ and $x_1 = 1$ otherwise
  - → Repeat for the other variables
- Evaluate the lineage formula ϕ under this valuation
- Approximate the probability of the formula ϕ as the proportion of times when it was true

Approximation by sampling

Another possibility is to approximate via Monte-Carlo sampling:

- Pick a random valuation according to the variable probabilities:
  - → Set $x_1 = 0$ with probability on $\mathrm{P}(x_1 = 0)$ and $x_1 = 1$ otherwise
  - → Repeat for the other variables
- Evaluate the lineage formula ϕ under this valuation
- Approximate the probability of the formula ϕ as the proportion of times when it was true
- Theoretical guarantees: on how many samples suffice so that, with high probability, the estimated probability is almost correct

Using external tools

- Specialized software to compute the probability of a formula: weighted model counters
- Examples (ongoing research):
  - c2d: `http://reasoning.cs.ucla.edu/c2d/download.php`
  - d4: `https://www.cril.univ-artois.fr/KC/d4.html`
  - dsharp: `https://bitbucket.org/haz/dsharp`

Outline

Probabilistic Relational Models

Probabilistic query evaluation

Complexity

Conclusion

Complexity of probabilistic query evaluation (PQE)

What is the data complexity of probabilistic query evaluation on TID depending on the class $\mathcal{Q}$ of queries and class $\mathcal{I}$ of instances?

Complexity of probabilistic query evaluation (PQE)

What is the data complexity of probabilistic query evaluation on TID depending on the class $\mathcal{Q}$ of queries and class $\mathcal{I}$ of instances?

- Existing dichotomy result: [Dalvi and Suciu, 2012]
  - $\mathcal{Q}$ are (unions of) conjunctive queries, $\mathcal{I}$ is all TID instances
  - There is a class $\mathcal{S} \subseteq \mathcal{Q}$ of safe queries

Complexity of probabilistic query evaluation (PQE)

What is the data complexity of probabilistic query evaluation on TID depending on the class $\mathcal{Q}$ of queries and class $\mathcal{I}$ of instances?

- Existing dichotomy result: [Dalvi and Suciu, 2012]
  - $\mathcal{Q}$ are (unions of) conjunctive queries, $\mathcal{I}$ is all TID instances
  - There is a class $\mathcal{S} \subseteq \mathcal{Q}$ of safe queries
  - PQE is PTIME for any $q \in \mathcal{S}$ on all instances

Complexity of probabilistic query evaluation (PQE)

What is the data complexity of probabilistic query evaluation on TID depending on the class $\mathcal{Q}$ of queries and class $\mathcal{I}$ of instances?

- Existing dichotomy result: [Dalvi and Suciu, 2012]
  - $\mathcal{Q}$ are (unions of) conjunctive queries, $\mathcal{I}$ is all TID instances
  - There is a class $\mathcal{S} \subseteq \mathcal{Q}$ of safe queries
  - PQE is PTIME for any $q \in \mathcal{S}$ on all instances
  - PQE is #P-hard for any $q \in \mathcal{Q} \backslash \mathcal{S}$ on all instances

Complexity of probabilistic query evaluation (PQE)

What is the data complexity of probabilistic query evaluation on TID depending on the class $\mathcal{Q}$ of queries and class $\mathcal{I}$ of instances?

- Existing dichotomy result: [Dalvi and Suciu, 2012]
  - $\mathcal{Q}$ are (unions of) conjunctive queries, $\mathcal{I}$ is all TID instances
  - There is a class $\mathcal{S} \subseteq \mathcal{Q}$ of safe queries
  - PQE is PTIME for any $q \in \mathcal{S}$ on all instances
  - PQE is #P-hard for any $q \in \mathcal{Q} \backslash \mathcal{S}$ on all instances
  - $q : \exists x\ y\ R(x) \wedge S(x, y) \wedge T(y)$ is unsafe!

Complexity of probabilistic query evaluation (PQE)

What is the data complexity of probabilistic query evaluation on TID depending on the class $\mathcal{Q}$ of queries and class $\mathcal{I}$ of instances?

- Existing dichotomy result: [Dalvi and Suciu, 2012]
  - $\mathcal{Q}$ are (unions of) conjunctive queries, $\mathcal{I}$ is all TID instances
  - There is a class $\mathcal{S} \subseteq \mathcal{Q}$ of safe queries
  - PQE is PTIME for any $q \in \mathcal{S}$ on all instances
  - PQE is #P-hard for any $q \in \mathcal{Q} \backslash \mathcal{S}$ on all instances
  - $q : \exists x \, y \; R(x) \wedge S(x, y) \wedge T(y)$ is unsafe!

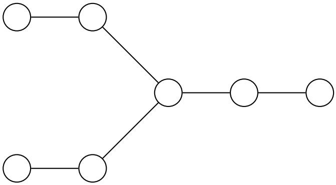
Is there a smaller class $\mathcal{I}$ such that PQE is tractable for a larger $\mathcal{Q}$?

Trees and treelike instances

- Idea: let $\mathcal{I}$ be treelike instances (constant bound on treewidth)

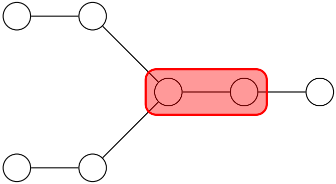
Trees and treelike instances

- Idea: let $\mathcal{I}$ be treelike instances (constant bound on treewidth)



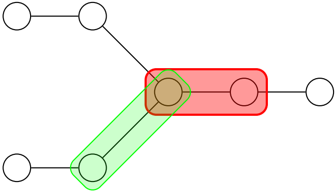
Trees and treelike instances

- Idea: let $\mathcal{I}$ be treelike instances (constant bound on treewidth)



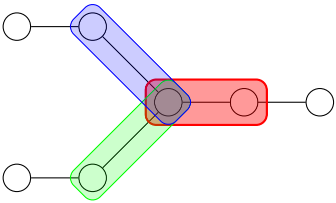
Trees and treelike instances

- Idea: let $\mathcal{I}$ be treelike instances (constant bound on treewidth)



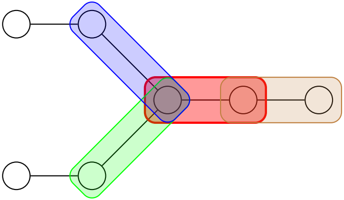
Trees and treelike instances

- Idea: let $\mathcal{I}$ be treelike instances (constant bound on treewidth)



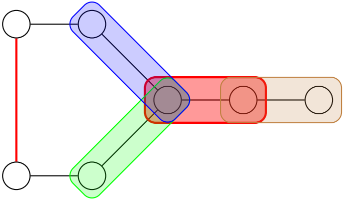
Trees and treelike instances

- Idea: let $\mathcal{I}$ be treelike instances (constant bound on treewidth)



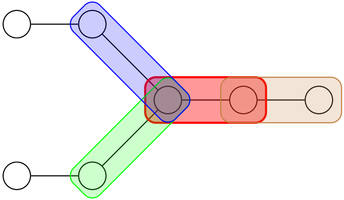
Trees and treelike instances

- Idea: let $\mathcal{I}$ be treelike instances (constant bound on treewidth)



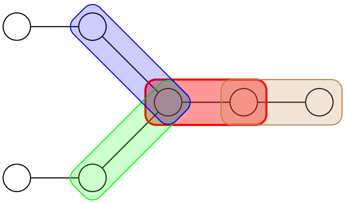
Trees and treelike instances

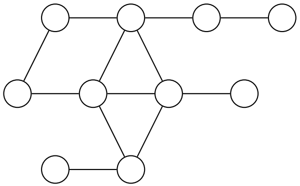
- Idea: let $\mathcal{I}$ be treelike instances (constant bound on treewidth)



Trees and treelike instances

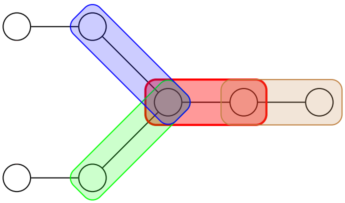
- Idea: let $\mathcal{I}$ be treelike instances (constant bound on treewidth)

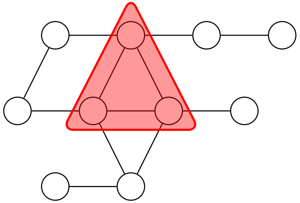




Trees and treelike instances

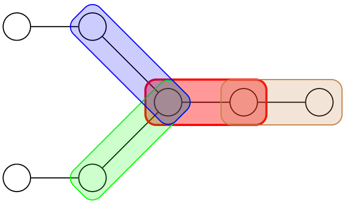
- Idea: let $\mathcal{I}$ be treelike instances (constant bound on treewidth)

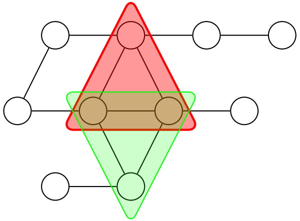




Trees and treelike instances

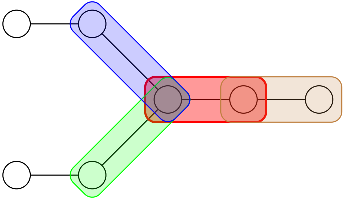
- Idea: let $\mathcal{I}$ be treelike instances (constant bound on treewidth)

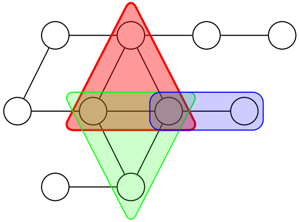




Trees and treelike instances

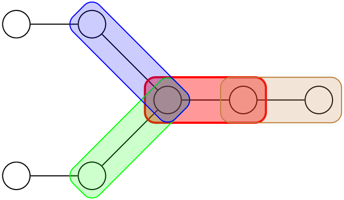
- Idea: let $\mathcal{I}$ be treelike instances (constant bound on treewidth)

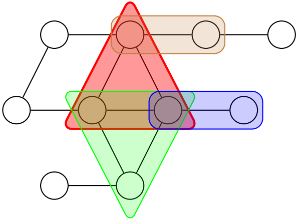




Trees and treelike instances

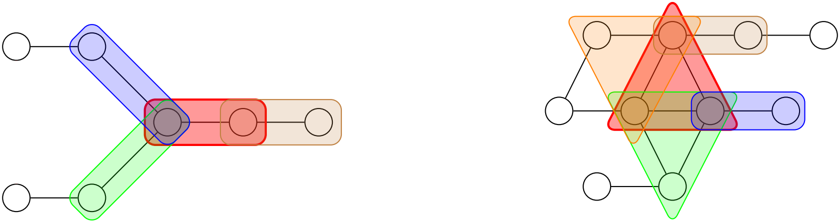
- Idea: let $\mathcal{I}$ be treelike instances (constant bound on treewidth)





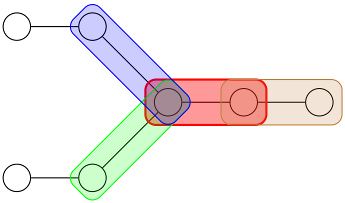
Trees and treelike instances

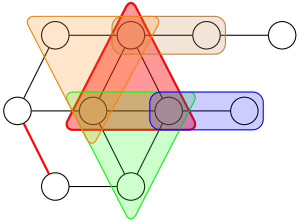
- Idea: let $\mathcal{I}$ be treelike instances (constant bound on treewidth)



Trees and treelike instances

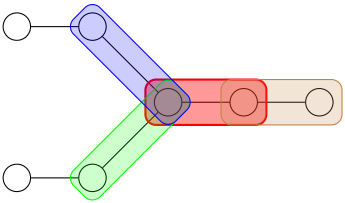
- Idea: let $\mathcal{I}$ be treelike instances (constant bound on treewidth)

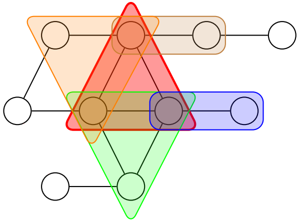




Trees and treelike instances

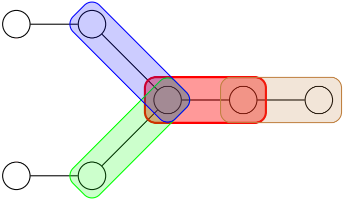
- Idea: let $\mathcal{I}$ be treelike instances (constant bound on treewidth)

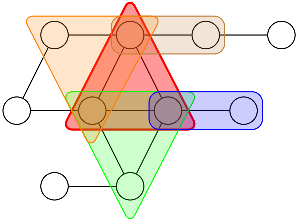




Trees and treelike instances

- Idea: let $\mathcal{I}$ be treelike instances (constant bound on treewidth)

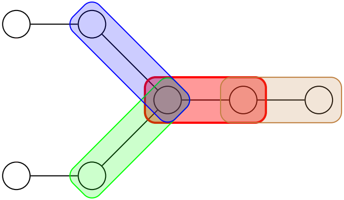


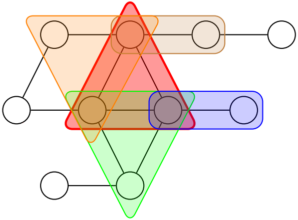


  - Trees have treewidth 1
  - Cycles have treewidth 2
  - k-cliques and $(k-1)$-grids have treewidth $k-1$

Trees and treelike instances

- Idea: let $\mathcal{I}$ be treelike instances (constant bound on treewidth)





  - Trees have treewidth 1
  - Cycles have treewidth 2
  - k-cliques and $(k-1)$-grids have treewidth $k-1$

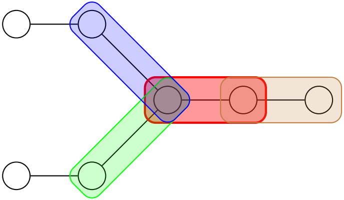
→ Known results [Courcelle, 1990]:

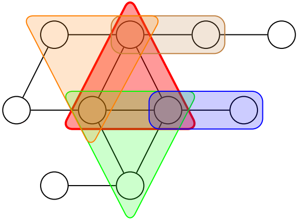
  - $\mathcal{I}$: treelike instances; $\mathcal{Q}$: monadic second-order queries

  → non-probabilistic QE is in linear time

Trees and treelike instances

- Idea: let $\mathcal{I}$ be treelike instances (constant bound on treewidth)





  - Trees have treewidth 1
  - Cycles have treewidth 2
  - k-cliques and $(k-1)$-grids have treewidth $k-1$

→ Known results [Courcelle, 1990]:

  - $\mathcal{I}$: treelike instances; $\mathcal{Q}$: monadic second-order queries

  → non-probabilistic QE is in linear time

→ Does this extend to probabilistic QE?

Dichotomy for PQE

An instance-based dichotomy result:

Upper bound. [Amarilli et al., 2015]

For $\mathcal{I}$ the treelike instances and $\mathcal{Q}$ the MSO queries

$\rightarrow$ PQE is in linear time modulo arithmetic costs

Dichotomy for PQE

An instance-based dichotomy result:

Upper bound. [Amarilli et al., 2015]

For $\mathcal{I}$ the treelike instances and $\mathcal{Q}$ the MSO queries

$\rightarrow$ PQE is in linear time modulo arithmetic costs

- Also for expressive provenance representations
- Also with bounded-treewidth correlations

Dichotomy for PQE

An instance-based dichotomy result:

Upper bound. [Amarilli et al., 2015]

For $\mathcal{I}$ the treelike instances and $\mathcal{Q}$ the MSO queries

→ PQE is in linear time modulo arithmetic costs

- Also for expressive provenance representations
- Also with bounded-treewidth correlations

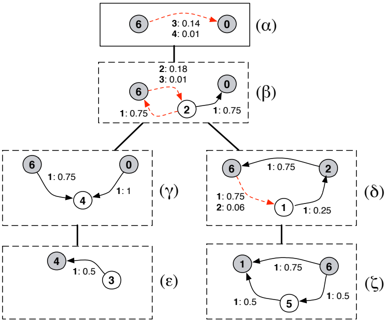
Lower bound. [Amarilli et al., 2016]

For any unbounded-tw family $\mathcal{I}$ and $\mathcal{Q}$ the FO queries

→ PQE is #P-hard under RP reductions assuming:

- High-tw instances in $\mathcal{I}$ are easily constructible
- Signature arity is 2 (graphs)

Application: Efficient querying of uncertain graphs [Maniu et al., 2017]



- Problem: Optimize query evaluation on probabilistic graphs
- Challenge: Real graph data is not treelike
- Methodology: Build partial tree decompositions and use different query evaluation techniques on treelike parts and on the rest of the data

Outline

Probabilistic Relational Models

Probabilistic query evaluation

Complexity

Conclusion

Summary: Probabilistic Models

We have seen relational formalisms for probabilistic instances:

- TID, a simple model with independent probabilities on tuples
- BID, adding blocks with mutually exclusive choices
- pc-tables, i.e., Boolean c-tables with probabilities on variables
  - → pc-tables can capture any probabilistic instance

Summary: Probabilistic Models

We have seen relational formalisms for probabilistic instances:

- TID, a simple model with independent probabilities on tuples
- BID, adding blocks with mutually exclusive choices
- pc-tables, i.e., Boolean c-tables with probabilities on variables
  - → pc-tables can capture any probabilistic instance

Summary: Probabilistic Query Evaluation

- We have seen probabilistic query evaluation on TID instances: compute the marginal probability of each query output tuple

Summary: Probabilistic Query Evaluation

- We have seen probabilistic query evaluation on TID instances: compute the marginal probability of each query output tuple
- We can enumerate naively all possible worlds, but it is inefficient

Summary: Probabilistic Query Evaluation

- We have seen probabilistic query evaluation on TID instances: compute the marginal probability of each query output tuple
- We can enumerate naively all possible worlds, but it is inefficient
- For some queries, we can do better:

Summary: Probabilistic Query Evaluation

- We have seen probabilistic query evaluation on TID instances: compute the marginal probability of each query output tuple
- We can enumerate naively all possible worlds, but it is inefficient
- For some queries, we can do better:
  - Extensional evaluation: find a safe plan so we can correctly compute all probabilities as the query is being evaluated

Summary: Probabilistic Query Evaluation

- We have seen probabilistic query evaluation on TID instances: compute the marginal probability of each query output tuple

- We can enumerate naively all possible worlds, but it is inefficient

- For some queries, we can do better:
  - Extensional evaluation: find a safe plan so we can correctly compute all probabilities as the query is being evaluated
  - Intensional evaluation:
    - compute the lineage of each result via pc-tables
    - compute the probability of each lineage expression

References I

Abiteboul, S., Hull, R., and Vianu, V. (1995).
Foundations of Databases.
Addison-Wesley.
http://webdam.inria.fr/Alice/pdfs/all.pdf.

Amarilli, A., Bourhis, P., and Senellart, P. (2015).
Provenance circuits for trees and treelike instances.
In *Proc. ICALP*, pages 56–68, Kyoto, Japan.

Amarilli, A., Bourhis, P., and Senellart, P. (2016).
Tractable lineages on treelike instances: Limits and extensions.
In *Proc. PODS*, San Francisco, USA.

References II

Barbará, D., Garcia-Molina, H., and Porter, D. (1992).
The management of probabilistic data.
IEEE Transactions on Knowledge and Data Engineering, 4(5).
http://www.iai.uni-bonn.de/III/lehre/AG/IntelligenteDatenbanken/Seminar/SS05/Literatur/%5BBGP92%5DProbData_IEEE_TKDE.pdf.

Courcelle, B. (1990).
The monadic second-order logic of graphs. I. Recognizable sets of finite graphs.
Inf. Comput., 85(1).

References III

Dalvi, N. and Suciu, D. (2012).
The dichotomy of probabilistic inference for unions of conjunctive queries.
J. ACM, 59(6).

Dalvi, N. N. and Suciu, D. (2007).
Efficient query evaluation on probabilistic databases.
VLDB Journal.
http://www.vldb.org/conf/2004/RS22P1.PDF.

Green, T. J. and Tannen, V. (2006).
Models for incomplete and probabilistic information.
IEEE Data Eng. Bull.
http://sites.computer.org/debull/A06mar/green.ps.

References IV

Huang, J., Antova, L., Koch, C., and Olteanu, D. (2009).
MayBMS: a probabilistic database management system.
In *SIGMOD*.
https://www.cs.ox.ac.uk/dan.olteanu/papers/hako-sigmod09.pdf.

Lakshmanan, L. V. S., Leone, N., Ross, R. B., and Subrahmanian, V. S. (1997).
ProbView: A flexible probabilistic database system.
ACM Transactions on Database Systems.
http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.53.293&rep=rep1&type=pdf.

References V

Maniu, S., Cheng, R., and Senellart, P. (2017).
An indexing framework for queries on probabilistic graphs.
ACM Transactions on Database Systems, 42(2):13:1–13:34.

Ré, C. and Suciu, D. (2007).
Materialized views in probabilistic databases: for information exchange and query optimization.
In *VLDB*.
http://www.cs.stanford.edu/people/chrismre/papers/prob_materialized_views_TR.pdf.

Suciu, D., Olteanu, D., Ré, C., and Koch, C. (2011).
Probabilistic Databases.
Morgan & Claypool.
Unavailable online.