Probabilistic Databases

A Probabilistic Database

News	id	title	pr
	N1	Cannibalism reappears in Chile	0.6
	N2	Habitable planet discovered	0.7
	N3	Cyprus sinks completely	0.4
	N4	Severe shortage of beer	0.9

Sources	id	media	shares	pr
	N1	Internet Research Agency	300K	0.6
	N2	Gotham Globe	27K	0.9
	N2	The Chippwea Bugle	950K	0.8
	N3	La Cuarta	125K	0.7
	N4	Twin Peaks Gazette	350K	0.6

A Probabilistic Database

A **(tuple-independent) probabilistic database** is a pair **T** = (D,p),

where D is a database, and p : D → [0,1]

i.e., each atom in D is an independent Bernoulli random variable

(takes value 1 with probability p, and value 0 with probability 1-p)

Possible Worlds of a Probabilistic Database

News	id	title	pr
	N1	Cannibalism reappears in Chile	0.6
	N2	Habitable planet discovered	0.7
	N3	Cyprus sinks completely	0.4
	N4	Severe shortage of beer	0.9

Sources	id	media	shares	pr
	N1	Internet Research Agency	300K	0.6
	N2	Gotham Globe	27K	0.9
	N2	The Chippwea Bugle	950K	0.8
	N3	La Cuarta	125K	0.7
	N4	Twin Peaks Gazette	350K	0.6

Possible Worlds of a Probabilistic Database

News	id	title	pr
	N1	Cannibalism reappears in Chile	0.6
			0.7
	N3	Cyprus sinks completely	0.4
			0.9

Sources	id	media	shares	pr
	N1	Internet Research Agency	300K	0.6
				0.9
	N2	The Chippwea Bugle	950K	0.8
				0.7
	N4	Twin Peaks Gazette	350K	0.6

$$0.6 \times (1 - 0.7) \times 0.4 \times (1 - 0.9) \times 0.6 \times (1 - 0.9) \times 0.8 \times (1 - 0.7) \times 0.6$$

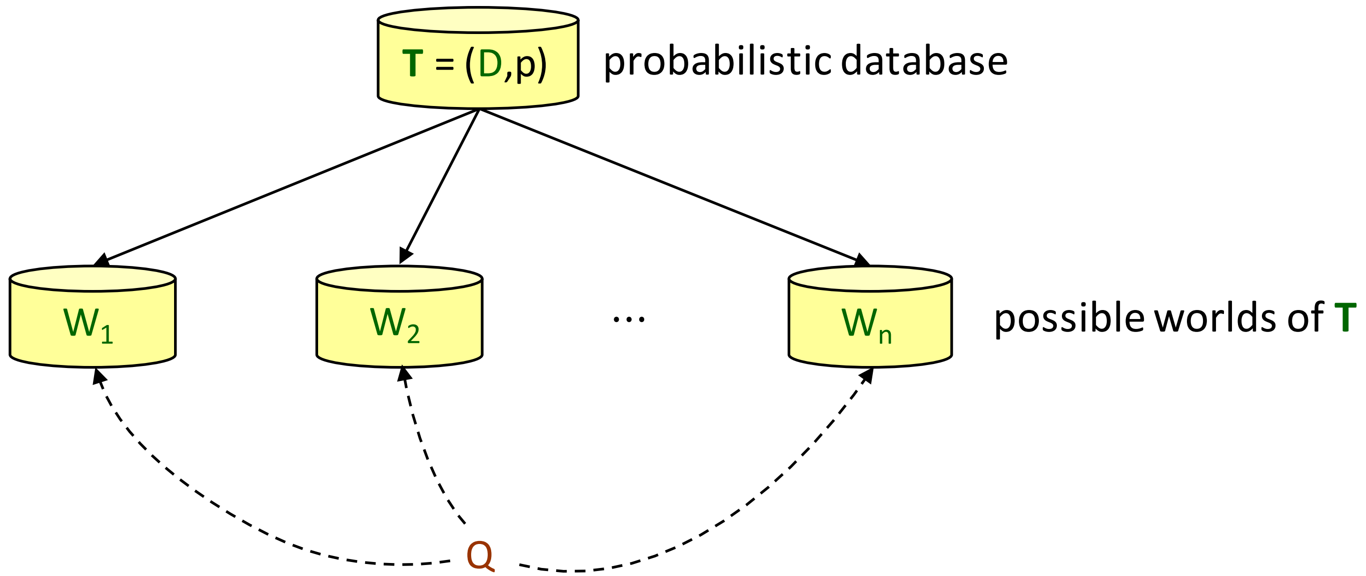
Possible Worlds of a Probabilistic Database

The **possible worlds** of a probabilistic database **T** = (D,p) are all the subsets of D

Probability of a possible world W - Pr(W) = $\prod_{\alpha \in W} p(\alpha) \cdot \prod_{\alpha \notin W} 1 - p(\alpha)$

Pr is a probability distribution over the set of possible worlds of **T** - $\sum_{W \subseteq D} \Pr(W) = 1$

Querying Probabilistic Databases



Prob(Q,**t**,**T**) = sum up the probabilities of the possible worlds of **T** that give **t** as an answer to Q

$$\text{Prob}(Q,\mathbf{t},\mathbf{T}) = \sum_{W \subseteq D \text{ and } \mathbf{t} \in Q(W)} \Pr(W)$$

Querying Probabilistic Databases

PQA(**L**)

Input: a probabilistic database **T** = (D,p), a query Q/k ∈ **L**, a tuple of constants **t** ∈ **adom**(D)^k

Output: the value Prob(Q,**t**,**T**)

we can naturally talk about the data complexity of the problem

PQA[Q](**L**) - input **T** and **t**, fixed Q

Data Complexity of PQA

Theorem: For **L** ∈ {**RA**, **DRC**, **TRC, CQ**}, PQA[Q](**L**) is #P-hard for some fixed query Q ∈ **L**.

This holds even for Boolean CQs.

This essentially means that querying probabilistic databases is a hard problem

Tackle High Data Complexity

Two main research directions:

1. Isolate classes of queries (in fact, classes of CQs) for which the problem can be solved efficiently in data complexity

2. Provide data-efficient approximations

Tackle High Data Complexity

Two main research directions:

1. Isolate classes of queries (in fact, classes of CQs) for which the problem can be solved efficiently in data complexity

2. Provide data-efficient approximations - **there exists an FPRAS for CQs**

Hierarchical CQs

A CQ Q is **hierarchical** if, for every two non-output variable x,y in Q, one of the following holds:

1. $\text{atoms}(Q,x) \subseteq \text{atoms}(Q,y)$
2. $\text{atoms}(Q,y) \subseteq \text{atoms}(Q,x)$
3. $\text{atoms}(Q,x) \cap \text{atoms}(Q,y) = \emptyset$

Theorem (Dichotomy Result): Consider a query $Q \in$ **CQ**.

- If Q is hierarchical, then PQA[Q](**CQ**) is feasible in polynomial time.
- Otherwise, PQA[Q](**CQ**) is #P-hard.

Recap

- Probabilistic databases - atoms are coming with a probability
- There are several possible worlds depending on which atoms are actually present
- Main problem of concern: compute the probability of a tuple
- Querying probabilistic databases is a hard problem (#P-hard in data complexity)
- A maximal fragment of CQs (hierarchical CQs) for which the problem is tractable
- There exists a data-efficient approximation scheme for CQs