Principles of Managing Uncertain Data

Lecture 9: Probabilistic Databases
Table of Contents

1 Introduction
2 Basic Probability Theory
3 Probabilistic Databases
4 Representing Probabilistic Databases
Table of Contents

1 Introduction

2 Basic Probability Theory

3 Probabilistic Databases

4 Representing Probabilistic Databases
Motivation

- Databases provide invaluable service for the management of certain data (modeling, integrity, querying, access)
- When data is probabilistic, the semantics of query answering changes dramatically, as answers have confidence levels (probabilities) that need to be computed
  - In standard terminology, querying amounts to probabilistic inference
- Hence, inference requires substantial algorithms/software outside (or on top) of the database
Probability is modeled in (understood by) the database

Query Evaluation:
- The user poses standard queries (as in deterministic data)
- The database engine answers queries by probabilistic inference
  - Rather than ordinary logical inference

Probabilistic correlations via standard database views
- Views are defined by queries
- Again, probabilistic inference provided for using the views
Historical Note: Pre-2000s

- 1980s: first probabilistic (or statistical) databases proposed
  - Examples: [LST83] [CP87]
  - Theory relating statistics (uncertain attributes) and queries
  - Limited research interest back then

- 1990s:
  - Probabilistic databases for merging DB & IR [Fuh90]
  - Additional models: attributes with prob. intervals [LLRS97],
  - *tuple-independent databases* [Zim97]
  - Still limited interest
Historical Note: 2000s On

- From 2000s: flurry of research on probabilistic databases
- Seminal article by Dalvi and Suciu [DS07], first presented at the VLDB conference [DS04].
- Probabilistic database systems: MayBMS (Cornell), MYSTIQ (UW), Orion (Purdue U.), BayesStore (UC Berkeley), PrDB (U. Maryland), Trio (Stanford), MCDB (IBM & Rice U.), SPROUT (Oxford), …
- Probabilistic DBs for text analysis [WMGH08, WMM10]
- Probabilistic DBs for data cleaning [BSI+10, HM09]
- Probabilistic data exchange/integration [DHY09, FKK11]
- Probabilistic semistructured (XML) data [NJ02, AKSS09]
- …
Efficient query evaluation on probabilistic databases

Authors: Nilesh Dalvi, Dan Suciu
Publication date: 2007/10/1
Journal: The VLDB Journal—The International Journal on Very Large Data Bases
Volume: 16
Issue: 4
Pages: 523-544
Publisher: Springer-Verlag New York, Inc.
Description: We describe a framework for supporting arbitrarily complex SQL queries with uncertain predicates. The query semantics is based on a probabilistic model and the results are ranked, much like in Information Retrieval. Our main focus is query evaluation. We describe an optimization algorithm that can compute efficiently most queries. We show, however, that the data complexity of some queries is #P-complete, which implies that these queries do not admit any efficient evaluation methods. For these queries we describe both an approximation algorithm and a Monte-Carlo simulation algorithm.

Total citations: Cited by 1111
Scholar articles:
Efficient query evaluation on probabilistic databases
N Dalvi, D Suciu - The VLDB Journal—The International Journal on Very Large Data Bases, 2007
Cited by 1111  Related articles  All 41 versions
Related CS Concepts

- Probabilistic graphical models
  - Bayesian networks (e.g., HMM)
  - Markov random fields (e.g., CRF)
- Statistical Relational Learning (SRL)
  - Bayesian Logic, Markov Logic Networks (MLN), Probabilistic Soft Logic (PSL), . . .
- Probabilistic Programming
  - Probabilistic-C, Chimple (Java), PRISM (Prolog), PyMC (Python), FACTORIE (Scala)
Features of Probabilistic Database Research

- Separation between *data* and *query*
  - In particular, data complexity
- Study of *query languages* and impact on complexity
- Focus on large volumes & complex joins
- Typically, strong independence assumptions on the data, correlation via (small) queries
Discrete Probability Spaces

- A *discrete probability space* is a pair \((\Omega, p)\) where:
  - \(\Omega\) is a nonempty, (finitely or infinitely) countable set
  - \(p : \Omega \rightarrow [0, 1]\) is a function satisfying
    \[
    \sum_{o \in \Omega} p(o) = 1
    \]

- Terminology:
  - An element \(o \in \Omega\) is called a *sample* (sometimes, *possible world*)
  - \(\Omega\) is called the *sample space*
  - \(p\) is called a *probability mass function*

- Unless otherwise specified, we will consider only discrete probability spaces, so may say just “probability space”
A probability space \((\Omega, p)\) is typically described by a “probabilistic story” that describes how \(\Omega\) and \(p\) are defined.

Very important! We need to distinguish between the **semantics** of the probability space (i.e., \(\Omega\) and \(p\)) and the **representation** of the probability space (i.e., the story itself).
Example

Description of \((\Omega, p)\):

- throw a (balanced) coin
  - if heads then drink coke
  - else
    - throw a (balanced) dice
      - if 1 or 2 then drink coffee
      - else drink milk

\[ \Omega = \{ \text{coke, coffee, milk} \} \]

\[ p(\text{coke}) = \frac{1}{2} \quad p(\text{coffee}) = \frac{1}{6} \quad p(\text{milk}) = \frac{1}{3} \]
Let \((\Omega, p)\) be a probability space

A \textit{random element} is a function from \(\Omega\) to some (arbitrary) codomain

- Examples:
  - The first letter of the chosen drink
  - The set of adequate cups for the chosen drink
  - The chosen drink

A \textit{random variable} is a random element with the codomain \(\mathbb{R}\)

- Examples:
  - The amount of calories in the chosen drink
  - The recommended temperature for the chosen drink
Let \((\Omega, p)\) be a probability space.

A *random event* (or just *event*) is simply a subset \(E\) of \(\Omega\).

For an event \(E\) we denote:

\[
\Pr_p(E) \overset{\text{def}}{=} \sum_{o \in E} p(o)
\]

- Usually \(p\) is known from the context, so we write only \(\Pr(E)\).

- Example: for \(E = \{\text{coke}, \text{milk}\}\) we have \(\Pr(E) = \frac{5}{6}\).

- We usually denote events via a constraint *language* (e.g., English or FO) over random elements.
Examples

\[ \Omega = \{ \text{coke, coffee, milk} \} \]
\[ p(\text{coke}) = \frac{1}{2} \quad p(\text{coffee}) = \frac{1}{6} \quad p(\text{milk}) = \frac{1}{3} \]

\[ \Pr(\text{drink begins with "c"}) = \Pr(\{\text{coke, coffee}\}) = \frac{5}{6} \]

\[ \Pr(\text{calories > 80}) = \Pr(\{\text{coke, milk}\}) = \frac{4}{6} \]
Conditional probability captures the probability space induced by observing partial knowledge in an original probability space.

- Often, the original space is referred to as the prior and the induced space as the posterior.

- Extremely useful concept!
  - Gives rise to inference techniques (e.g., calculation of event probabilities)
  - Modeling of natural errors (sensing devices, measure devices, natural language, etc.)
  - Modeling learning tasks in machine learning
Conditional Probability (Formal)

- Let \((\Omega, p)\) be a probability space.
- Let \(C\) be a random event, such that \(\Pr(C) > 0\).
- The subspace conditioned on \(C\) is the probability space \((\Omega, p_C)\) where \(p_C = p/\Pr(C)\) over \(C\), and 0 outside \(C\).
  - That is, for all \(o \in \Omega\) we have
    \[
    p_C(o) = \begin{cases} 
      \frac{p(o)}{\Pr(C)} & \text{if } o \in C \\
      0 & \text{otherwise}
    \end{cases}
    \]
- In simple worlds, we restrict \((\Omega, p)\) to \(C\), and normalize.
- The probability in this subspace is denoted by \(\Pr(\cdot | C)\).
- We have: \(\Pr(A|C) = \frac{\Pr(A \cap C)}{\Pr(C)}\).
Examples

\[ \Omega = \{ \text{coke, coffee, milk} \} \]

\[ p(\text{coke}) = \frac{1}{2}, \quad p(\text{coffee}) = \frac{1}{6}, \quad p(\text{milk}) = \frac{1}{3} \]

\[ \Pr(\text{coke} | > 80 \text{ calories}) = \Pr(\text{coke} | \{\text{coke, milk}\}) = \frac{1/2}{5/6} \]

\[ \Pr(> 80 \text{ calories} | \text{begins with 'c'}) \]

\[ = \Pr(\{\text{coke, milk}\} | \{\text{coke, coffee}\}) = \frac{1/2}{2/3} \]
Independence

- Let \((\Omega, p)\) be a probability space, \(A\) and \(B\) two random events.
- We say that \(A\) and \(B\) are *probabilistically independent* (or *independent events*) if

\[
\Pr(A \cap B) = \Pr(A) \times \Pr(B)
\]

- If \(\Pr(B) > 0\), then independence of \(A\) and \(B\) is equivalent to

\[
\Pr(A \mid B) = \Pr(A)
\]
Example 1

\[ \Omega = \{ \text{coke, coffee, milk, redbull, water} \} \]

\[
\begin{align*}
p(\text{coke}) &= \frac{1}{3} & p(\text{coffee}) &= \frac{1}{6} & p(\text{milk}) &= \frac{1}{6} & p(\text{redbull}) &= \frac{1}{6} & p(\text{water}) &= \frac{1}{6}
\end{align*}
\]

\[
\Pr(> 80 \text{ calories} \land \text{begins with c}) = \Pr(\text{coke}) = \frac{1}{3}
\]

\[
\begin{align*}
\Pr(> 80 \text{ calories}) &= \frac{2}{3} & \Pr(\text{begins with c}) &= \frac{1}{2}
\end{align*}
\]

\[ \Rightarrow \text{“> 80 calories” and “begins with c” are independent events} \]
Example 2

\[ \Omega = \{ \text{coke, coffee, milk, redbull, water} \} \]

\[
p(\text{coke}) = \frac{1}{3} \quad p(\text{coffee}) = \frac{1}{6} \quad p(\text{milk}) = \frac{1}{6} \quad p(\text{redbull}) = \frac{1}{6} \quad p(\text{water}) = \frac{1}{6}
\]

\[
\Pr(>80 \text{ calories } \land \text{caffeine}) = \Pr(\text{coke, redbull}) = \frac{1}{2}
\]

\[
\Pr(>80 \text{ calories}) = \frac{2}{3} \quad \Pr(\text{caffeine}) = \Pr(\text{coke, milk, redbull}) = \frac{2}{3}
\]

\[ \Rightarrow \text{ “}> 80 \text{ calories” and “caffeine” are not independent events} \]
Simple idea: events may become independent in a conditional subspace, even if correlated in the original space

Let \((\Omega, p)\) be a probability space, \(A\), \(B\) and \(C\) be three random events such that \(\Pr(C) > 0\)

We say that \(A\) and \(B\) are *conditionally independent* given \(C\) if \(A\) and \(B\) are independent events in the subspace conditioned on \(C\)

That is: \(\Pr(A \cap B \mid C') = \Pr(A \mid C') \times \Pr(B \mid C')\)
Example Revisited

\[ \Omega = \{ \text{coke, coffee, milk, redbull, water} \} \]

\[
p(\text{coke}) = \frac{1}{3} \quad p(\text{coffee}) = \frac{1}{6} \quad p(\text{milk}) = \frac{1}{6} \quad p(\text{redbull}) = \frac{1}{6} \quad p(\text{water}) = \frac{1}{6}
\]

\[
\Pr(> 80 \text{ calories} \land \text{caffeine} | \text{not coke}) = \frac{1}{4}
\]

\[
\Pr(> 80 \text{ calories} | \text{not coke}) = \frac{1}{2} \quad \Pr(\text{caffeine} | \text{not coke}) = \frac{1}{2}
\]

\[ \Rightarrow \text{"> 80 calories" and "caffeine" are conditionally independent given "not coke"} \]
On Continuous Spaces

- Continuous probability space: sample space is *uncountable*
- Examples of continuous spaces:
  - The angle of a spinner
  - The position hit by a thrown arrow
  - Choose a random number in \([0, 1]\), “uniformly”
- In this case, the probability of an event **cannot** be defined by means of probabilities of individual events
  - \(\Pr(x \in [0.1, 0.2]) = \Pr(x = 0.1) + \Pr(x = 0.101) + \ldots\)
- The modern approach is via *measure theory*
Definition of a Continuous Probability Space

- A continuous probability space is a triple \((\Omega, \mathcal{F}, p)\) where
  - \(\Omega\) is (again) the sample space
    - But this time it needs not be countable
  - \(\mathcal{F}\) is a collection of subsets of \(\Omega\)—the \textit{measurable events}
    - \(\mathcal{F}\) should be nonempty, and should satisfy closure under complement and countable unions; this is called a \(\sigma\)-\textit{algebra} over \(\Omega\)
  - \(p : \mathcal{F} \to [0, 1]\) assigns probabilities to measurable events
    - \(p\) should be a “proper” probability function, in the sense that \(p(\Omega) = 1\), and \(p\) is additive on countable unions of disjoint sets
- In the case where \(\Omega\) is countable, the discrete definition is a special case
  - \textit{What are the measurable events?}
Table of Contents

1 Introduction

2 Basic Probability Theory

3 Probabilistic Databases

4 Representing Probabilistic Databases
**Definition (Probabilistic Database)**

Let $\mathcal{S}$ be a schema. A *probabilistic database* over $\mathcal{S}$ is a probability space over instances of $\mathcal{S}$.

We will focus on *finite* spaces, where a probabilistic database is a probability space $(\Omega, p)$ with a finite $\Omega \subseteq \text{Inst}(\mathcal{S})$. 
In our running example, we have a sensing/vision system that observes faculty staff entering the building. *Observations are imprecise, and modeled as probabilistic.*
What is the probability that we observed a full course staff?
Typically, the probability space is too large to represent explicitly
  - Think about 1000 people tossing coins independently
Hence, a central aspect of a probabilistic database is its compact representation
However, compactness comes at the cost of assumptions
  - Most common assumption is that of probabilistic independence
    - Examples: Bayesian networks, probabilistic databases
  - FYI: Another popular assumption is that of symmetry
    - That is, there may be many entities represented, but they all behave exactly the same
    - Example: Markov Logic Networks
    - Analysis that uses symmetry is known as lifted inference
Studied Representations

- Probabilistic conditional instances (pc-instances)
- Tuple-independent databases
- Block-independent-disjoint (BID) databases
Recall the definition of a c-instance (representing an incomplete database)

We use here a restricted version
**Definition (Restricted c-Instance)**

Let $\mathcal{R}$ be a signature. A *restricted c-instance* over $\mathcal{R}$ is a triple $(I, X, \varphi)$ where:

- $I$ is an instance over $\mathcal{R}$
- $X$ is a set of variables $x$, each having a finite domain $\text{dom}(x)$
- $\varphi$ maps every fact $f$ of $I$ to a propositional formula $\varphi(f)$ over *atomic formulas* of the form “$x = v$”
  - $x \in X$ and $v \in \text{dom}(x)$
Example Notation

- When every $x \in X$ has the domain $\{\text{true, false}\}$, we write “$x$” instead of “$x = \text{true}$”
- Also, we omit $\varphi(f)$ if it is true
Example

<table>
<thead>
<tr>
<th>Lecturer</th>
<th>ObsLects</th>
<th>Observation</th>
<th>Courses</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ahuva</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Avia</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TA</th>
<th>ObsTAs</th>
<th>Observation</th>
<th>Courses</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asma</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Alon</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Course</th>
<th></th>
<th>Lecturer</th>
<th>TA</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB</td>
<td></td>
<td>Ahuva</td>
<td>Alon</td>
</tr>
<tr>
<td>PL</td>
<td></td>
<td>Avia</td>
<td>Asma</td>
</tr>
</tbody>
</table>

\[ x = y = z = \text{true} \]

\[ x = y = z = \text{false} \]
Semantics of a Restricted c-Instance

- Let $\mathcal{I} = (I, X, \varphi)$ be a restricted c-instance
- A valuation over $\mathcal{I}$ is a function $\mu$ that maps every variable $x \in X$ to a value $\mu(x) \in \text{dom}(x)$
- The valuation $\mu$ defines an ordinary instance $\mu(\mathcal{I})$:
  \[
  \mu(\mathcal{I}) \overset{\text{def}}{=} \{ f \in I \mid \mu \models \varphi(f) \}
  \]
- Denote by $M_X$ the set of all valuations over $X$
- The incomplete instance $[\mathcal{I}]$ defined by $\mathcal{I}$ is:
  \[
  [\mathcal{I}] \overset{\text{def}}{=} \{ \mu(\mathcal{I}) \mid \mu \in M_X \} \]
Definition (pc-Instance)

Let $\mathcal{S}$ be a schema. A pc-instance over $\mathcal{S}$ is a quadruple $(I, X, \varphi, \pi)$ where:

- $(I, X, \varphi)$ is a restricted c-instance
- $\pi$ assigns to each variable a distribution over values
  - More precisely, $\pi$ is a function that assigns a number $\pi(x, v) \in [0, 1]$ to each $x \in X$ and $v \in \text{dom}(x)$, such that $\sum_{v \in \text{dom}(x)} \pi(x, v) = 1$ for all $x \in X$. 
**Example**

<table>
<thead>
<tr>
<th>ObsLects</th>
<th>ObsTAs</th>
<th>Courses</th>
</tr>
</thead>
<tbody>
<tr>
<td>lecturer</td>
<td>ta</td>
<td>course</td>
</tr>
<tr>
<td>Ahuva</td>
<td>Asma</td>
<td>DB</td>
</tr>
<tr>
<td>Avia</td>
<td>Alon</td>
<td>PL</td>
</tr>
</tbody>
</table>

\[
\begin{align*}
\pi(x, \text{true}) & = 0.5 & \pi(y, \text{true}) & = 0.3 \\
\pi(x, \text{false}) & = 0.5 & \pi(y, \text{false}) & = 0.7 \\
\pi(z, \text{true}) & = 0.4 & \pi(z, \text{false}) & = 0.6
\end{align*}
\]
Probabilistic Valuation

- Let $\mathcal{P} = (I, X, \varphi, \pi)$ be a pc-instance.
- We assume that a valuation $\mu \in M_X$ is randomly generated by independently assigning a random value for each variable $x \in X$.
- Hence, the probability of a valuation $\mu$, which we denote by $\pi(\mu)$, is given by:

$$
\pi(\mu) \overset{\text{def}}{=} \prod_{x \in X} \pi(x, \mu(x))
$$
Semantics of a pc-Instance

- Let $\mathcal{P} = (I, X, \varphi, \pi)$ be a pc-instance, and denote $\mathcal{I} = (I, X, \varphi)$.
- $\mathcal{P}$ represents the probabilistic database in which a sample is created in two steps:
  1. Generate a random valuation $\mu$.
  2. Generate $\mu(\mathcal{I})$.
- Formally, the probabilistic database defined by $\mathcal{P}$, denoted $[\mathcal{P}]$, is $(\Omega, p)$ where:
  - $\Omega = [\mathcal{I}]$
  - $p(J) = \sum_{\mu \in M_X | \mu(\mathcal{I}) = J} \pi(\mu)$.
### Example

<table>
<thead>
<tr>
<th>( \text{ObsLects} )</th>
<th>( \text{ObsTAs} )</th>
<th>( \text{Courses} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>lecturer</td>
<td>ta</td>
<td>course</td>
</tr>
<tr>
<td>Ahuva</td>
<td>x</td>
<td>DB</td>
</tr>
<tr>
<td>Avia</td>
<td>( \neg x \lor y )</td>
<td>PL</td>
</tr>
<tr>
<td></td>
<td>Asma</td>
<td>Ahuva</td>
</tr>
<tr>
<td></td>
<td>Alon</td>
<td>Avia</td>
</tr>
</tbody>
</table>

\[ \pi(x, \text{true}) = 0.5 \quad \pi(y, \text{true}) = 0.3 \quad \pi(z, \text{true}) = 0.4 \]
\[ \pi(x, \text{false}) = 0.5 \quad \pi(y, \text{false}) = 0.7 \quad \pi(z, \text{false}) = 0.6 \]

\[ \mu_1(x) = \text{false} \quad \mu_1(y) = \mu_1(z) = \text{true} \]
\[ \pi(\mu_1) = 0.5 \times 0.3 \times 0.4 \]

\[ J = \mu_1(\mathcal{I}) : \]

<table>
<thead>
<tr>
<th>( \text{ObsLects} )</th>
<th>( \text{ObsTAs} )</th>
<th>( \text{Courses} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>lecturer</td>
<td>ta</td>
<td>course</td>
</tr>
<tr>
<td>Avia</td>
<td></td>
<td>DB</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PL</td>
</tr>
<tr>
<td></td>
<td>Alon</td>
<td>Ahuva</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Avia</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Alon</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Asma</td>
</tr>
</tbody>
</table>
Example

<table>
<thead>
<tr>
<th>ObsLects</th>
<th></th>
<th>ObsTAs</th>
<th></th>
<th>Courses</th>
</tr>
</thead>
<tbody>
<tr>
<td>lecturer</td>
<td></td>
<td>ta</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ahuva</td>
<td>x</td>
<td>Asma</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>Avia</td>
<td>¬x ∨ y</td>
<td>Alon</td>
<td>¬x ∨ z</td>
<td></td>
</tr>
</tbody>
</table>

\[
\begin{align*}
\pi(x, \text{true}) &= 0.5 \\
\pi(x, \text{false}) &= 0.5 \\
\pi(y, \text{true}) &= 0.3 \\
\pi(y, \text{false}) &= 0.7 \\
\pi(z, \text{true}) &= 0.4 \\
\pi(z, \text{false}) &= 0.6
\end{align*}
\]

\[\mu_2(x) = \mu_2(y) = \text{false} \quad \mu_2(z) = \text{true} \quad \pi(\mu_2) = 0.5 \times 0.7 \times 0.4\]

\[J = \mu_2(\mathcal{I}):\]

<table>
<thead>
<tr>
<th>ObsLects</th>
<th></th>
<th>ObsTAs</th>
<th></th>
<th>Courses</th>
</tr>
</thead>
<tbody>
<tr>
<td>lecturer</td>
<td></td>
<td>ta</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Avia</td>
<td></td>
<td>Alon</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[
\begin{align*}
\mu_2(\mathcal{I}) &= \mu_2(\mathcal{I})
\end{align*}
\]
### Example

<table>
<thead>
<tr>
<th>ObsLects</th>
<th>ObsTAs</th>
<th>Courses</th>
</tr>
</thead>
<tbody>
<tr>
<td>lecturer</td>
<td>ta</td>
<td>course</td>
</tr>
<tr>
<td>Ahuva</td>
<td>x</td>
<td>DB</td>
</tr>
<tr>
<td>Avia</td>
<td>¬x ∨ y</td>
<td>PL</td>
</tr>
</tbody>
</table>

\[ \pi(x, \text{true}) = 0.5 \quad \pi(y, \text{true}) = 0.3 \]
\[ \pi(x, \text{false}) = 0.5 \quad \pi(y, \text{false}) = 0.7 \]

\[ \mu_3(x) = \text{false} \quad \mu_3(y) = \text{true} \quad \mu_3(z) = \text{false} \]
\[ \pi(\mu_3) = 0.5 \times 0.3 \times 0.6 \]

\[ J = \mu_3(\mathcal{I}) : \]

<table>
<thead>
<tr>
<th>ObsLects</th>
<th>ObsTAs</th>
<th>Courses</th>
</tr>
</thead>
<tbody>
<tr>
<td>lecturer</td>
<td></td>
<td>course</td>
</tr>
<tr>
<td></td>
<td>ta</td>
<td>lecturer</td>
</tr>
<tr>
<td>Avia</td>
<td></td>
<td>Ahuva</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Alon</td>
</tr>
<tr>
<td></td>
<td>ta</td>
<td>ta</td>
</tr>
<tr>
<td></td>
<td></td>
<td>lecturer</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ahuva</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Asma</td>
</tr>
</tbody>
</table>

\[ \pi(x, \text{true}) = 0.5 \quad \pi(y, \text{true}) = 0.3 \]
\[ \pi(x, \text{false}) = 0.5 \quad \pi(y, \text{false}) = 0.7 \]

\[ \mu_3(x) = \text{false} \quad \mu_3(y) = \text{true} \quad \mu_3(z) = \text{false} \]
\[ \pi(\mu_3) = 0.5 \times 0.3 \times 0.6 \]
### Example

**ObsLects**  |  **ObsTAs**  |  **Courses**
---|---|---
| lecturer | ta | course | lecturer | ta |
| Ahuva |  \(x\) |  DB | | |
| Avia | \(\neg x \lor y\) |  PL | Ahuva | |
| |  Asma | | | Alon |
| |  Alon | | | |

\[
\pi(x, \text{true}) = 0.5 \quad \pi(y, \text{true}) = 0.3 \quad \pi(z, \text{true}) = 0.4 \\
\pi(x, \text{false}) = 0.5 \quad \pi(y, \text{false}) = 0.7 \quad \pi(z, \text{false}) = 0.6
\]

\[
\mu_4(x) = \mu_4(y) = \mu_4(z) = \text{false} \quad \pi(\mu_4) = 0.5 \times 0.7 \times 0.6
\]

**J** = \(\mu_4(\mathcal{I})\):

| **ObsLects**  |  **ObsTAs**  |  **Courses**
---|---|---
| lecturer | ta | course | lecturer | ta |
| |  |  DB | Ahuva | |
| |  |  PL | | Alon |
| Avia | | | | |
| |  Alon | | | |
| |  |  | | |

\[
\mu_4(x) = \mu_4(y) = \mu_4(z) = \text{false} \quad \pi(\mu_4) = 0.5 \times 0.7 \times 0.6
\]
Example

$\mathcal{I}$:

<table>
<thead>
<tr>
<th>$\text{ObsLects}$</th>
<th>$\text{ObsTAs}$</th>
<th>$\text{Courses}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>lecturer</td>
<td>ta</td>
<td>course</td>
</tr>
<tr>
<td>Ahuva</td>
<td>Asma</td>
<td>DB</td>
</tr>
<tr>
<td>Avia</td>
<td>Alon</td>
<td>PL</td>
</tr>
<tr>
<td>$x$</td>
<td>$x$</td>
<td>Ahuva</td>
</tr>
<tr>
<td>$\neg x \vee y$</td>
<td>$\neg x \vee z$</td>
<td>Avia</td>
</tr>
</tbody>
</table>

$\pi(x, \text{true}) = 0.5$  $\pi(y, \text{true}) = 0.3$
$\pi(x, \text{false}) = 0.5$ $\pi(y, \text{false}) = 0.7$

$\pi(z, \text{true}) = 0.4$
$\pi(z, \text{false}) = 0.6$

$p(J) = \pi(\mu_1) + \pi(\mu_2) + \pi(\mu_3) + \pi(\mu_4)$
In a *Block-Independent-Disjoint Database*, or *BID database* for short, each relation is divided into *blocks* of tuples.

To generate a random database:

- Consider each block *independently*
- Choose at most one tuple from each block; hence, tuples in the same block are *disjoint*
Formal Definition

- A BID database can be represented as a special case of a pc-instance
- Specifically, it is a pc-instance \( \mathcal{P} = (I, X, \varphi, \pi) \) where:
  - Each \( \varphi(f) \) is an atomic condition (of the form \( x = a \))
  - No two facts use the same atomic condition
    - That is, if \( f \neq f' \), then \( \varphi(f) \) and \( \varphi(f') \) use either different \( x \)s or different \( a \)s
  - Different relations use disjoint sets of variables
### Example

<table>
<thead>
<tr>
<th>Lecturer</th>
<th>$x$</th>
<th>$y$</th>
<th>$z$</th>
<th>$w$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ahuva</td>
<td>0</td>
<td></td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Avia</td>
<td>1</td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Aviv</td>
<td></td>
<td>0</td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>Adi</td>
<td></td>
<td>1</td>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Course</th>
<th>Lecturer</th>
<th>TA</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB</td>
<td>Ahuva</td>
<td>Alon</td>
</tr>
<tr>
<td>PL</td>
<td>Avia</td>
<td>Asma</td>
</tr>
</tbody>
</table>

\[
\begin{align*}
\pi(x, 0) &= 0.5 & \pi(x, 1) &= 0.5 \\
\pi(y, 0) &= 0.3 & \pi(y, 1) &= 0.5 & \pi(y, 2) &= 0.2 \\
\pi(z, 0) &= 0.4 & \pi(z, 1) &= 0.3 & \pi(z, 2) &= 0.3 \\
\pi(w, 0) &= 0.9 & \pi(w, 1) &= 0.1 
\end{align*}
\]
Due to the special form of a BID database, we can use a more intuitive representation, by assigning a primary key to each relation.

More formally:

- We have schema is $S = (\mathcal{R}, \Sigma)$, where $\Sigma$ consists primary keys, one per relation name.
- A BID database over $S$ is a pair $(I, \pi)$, where:
  - $I$ is an instance over $\mathcal{R}$
  - $\pi : I \rightarrow [0, 1]$ is a function that satisfies $\sum_{f \in B} \pi(f) \leq 1$ for every block $B$ of facts of the same relation and key.
Example

<table>
<thead>
<tr>
<th>ObsLects</th>
<th>ObsTAs</th>
<th>Courses</th>
</tr>
</thead>
<tbody>
<tr>
<td>lecturer</td>
<td>ta</td>
<td>course</td>
</tr>
<tr>
<td>Ahuva</td>
<td>Asma</td>
<td>DB</td>
</tr>
<tr>
<td></td>
<td>Alon</td>
<td>Ahuva</td>
</tr>
<tr>
<td>Avia</td>
<td>Avner</td>
<td>Alon</td>
</tr>
<tr>
<td></td>
<td>Alma</td>
<td>Avia</td>
</tr>
<tr>
<td>Aviv</td>
<td></td>
<td>Asma</td>
</tr>
<tr>
<td>Adi</td>
<td></td>
<td>Alon</td>
</tr>
</tbody>
</table>

\[
\begin{align*}
\pi(x, 0) &= 0.5 \\
\pi(x, 1) &= 0.5 \\
\pi(y, 0) &= 0.3 \\
\pi(y, 1) &= 0.5 \\
\pi(y, 2) &= 0.2 \\
\pi(z, 0) &= 0.4 \\
\pi(z, 1) &= 0.3 \\
\pi(z, 2) &= 0.3 \\
\pi(w, 0) &= 0.9 \\
\pi(w, 1) &= 0.1
\end{align*}
\]
In a tuple-independent database, a random instance is generated considering each tuple independently and choosing whether or not this tuple should be included.
A tuple-independent database can be represented as a special case of a pc-instance. Specifically, it is a pc-instance $\mathcal{P} = (I, X, \varphi, \pi)$ where:

- Each $\varphi(f)$ is an atomic condition (of the form $x = a$)
- Each fact uses a distinct variable
### Example

<table>
<thead>
<tr>
<th>Lecturer</th>
<th>x = 0</th>
<th>y = 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ahuva</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Avia</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TA</th>
<th>z = 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asma</td>
<td></td>
</tr>
<tr>
<td>Alon</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Course</th>
<th>lecturer</th>
<th>ta</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB</td>
<td>Ahuva</td>
<td></td>
</tr>
<tr>
<td>PL</td>
<td>Avia</td>
<td>Asma</td>
</tr>
</tbody>
</table>

\[
\begin{align*}
\pi(x, 0) &= 0.5 & \pi(x, 1) &= 0.5 \\
\pi(y, 0) &= 0.3 & \pi(y, 1) &= 0.7 \\
\pi(z, 0) &= 0.4 & \pi(z, 1) &= 0.3 & \pi(z, 2) &= 0.3 \\
\pi(w, 0) &= 0.9 & \pi(w, 1) &= 0.1
\end{align*}
\]
Simpler Representation

- Due to the special form of a tuple-independent database, we can again use a more intuitive representation.
- A tuple-independent database $\mathcal{P}$ over a signature $\mathcal{R}$ is a pair $(I, \pi)$, where:
  - $I$ is an instance over $\mathcal{R}$
  - $\pi : I \rightarrow [0, 1]$
- The probability space $\llbracket \mathcal{P} \rrbracket$ is $(\Omega, p)$, where $\Omega$ consists of all the subinstances $I$, and $p$ is as follows:

$$p(J) = \prod_{f \in J} \pi(f) \times \prod_{f \in I \setminus J} (1 - \pi(f))$$
### Example

<table>
<thead>
<tr>
<th>ObsLects</th>
<th>ObsTAs</th>
<th>Courses</th>
</tr>
</thead>
<tbody>
<tr>
<td>lecturer</td>
<td>ta</td>
<td>course</td>
</tr>
<tr>
<td>Ahuva</td>
<td>Asma</td>
<td>DB</td>
</tr>
<tr>
<td></td>
<td>Alon</td>
<td>Avia</td>
</tr>
<tr>
<td>Avia</td>
<td></td>
<td>PL</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>z = 0</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>lecturer</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ahuva</td>
<td>0.5</td>
<td></td>
</tr>
<tr>
<td>Avia</td>
<td>0.3</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ta</th>
<th>w = 0</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Asma</td>
<td>0.4</td>
<td></td>
</tr>
<tr>
<td>Alon</td>
<td>0.9</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>course</th>
<th>lecturer</th>
<th>ta</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB</td>
<td>Ahuva</td>
<td>Alon</td>
</tr>
<tr>
<td>PL</td>
<td>Avia</td>
<td>Asma</td>
</tr>
</tbody>
</table>


References III

- Oktie Hassanzadeh and Renée J. Miller, *Creating probabilistic databases from duplicated data*, VLDB J. **18** (2009), no. 5, 1141–1166.


End of lecture 9

Probabilistic Databases