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
Databases provide invaluable service for the management of *certain* data (modeling, integrity, querying, access)
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
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
Probabilistic Databases to Bridge the Gap

- Probability is modeled in (understood by) the database
Probabilistic Databases to Bridge the Gap

- 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 Databases to Bridge the Gap

- 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* Nilesch 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 ..., 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
Table of Contents

1 Introduction

2 Basic Probability Theory

3 Probabilistic Databases

4 Representing Probabilistic Databases
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
  \[
  \sum_{o \in \Omega} p(o) = 1
  \]
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*
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.
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
```
Example

Description of $(\Omega, p)$:

- **Scenario 1**: If heads then drink coke; else throw a (balanced) dice.
  - If 1 or 2 then drink coffee; else drink milk.
- **Scenario 2**: If heads then drink coke; else throw a (balanced) dice.
  - If 1 or 2 then drink coffee; else drink milk.

\[
\begin{align*}
\Omega &= \{\text{coke, coffee, milk}\} \\
p(\text{coke}) &= \frac{1}{2} \\
p(\text{coffee}) &= \frac{1}{6} \\
p(\text{milk}) &= \frac{1}{3}
\end{align*}
\]
Random Elements and Random Variables

- Let \((\Omega, p)\) be a probability space.
- A *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
Random Elements and Random Variables

- 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 $\Omega$
Event

- 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:

$$\Pr_p(E) \overset{\text{def}}{=} \sum_{o \in E} p(o)$$
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 \(\Pr(E)\).
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 \(E = \{\text{coke, milk}\}\) we have \(\Pr(E) = \frac{5}{6}\).
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 $E = \{\text{coke, milk} \}$ we have $\Pr(E) = \frac{5}{6}$.

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

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

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

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

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

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

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

Pr(drink begins with “c”) = Pr(\{coke, coffee\}) = \frac{5}{6}

Pr(calories > 80) = Pr(\{coke, milk\}) = \frac{4}{6}
Conditional Probability

- 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.
Conditional Probability

- 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 $C$
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
    \[
    p_C(o) = \begin{cases} 
    \frac{p(o)}{\Pr(C)} & \text{if } o \in C \\
    0 & \text{otherwise}
    \end{cases}
    \]
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
    \[
p_C(o) = \begin{cases} 
p(o) / \Pr(C) & \text{if } o \in C \\
    0 & \text{otherwise}
    \end{cases}
    \]
- In simple worlds, we restrict $(\Omega, p)$ to $C$, and normalize
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 $\Pr(\cdot \mid C)$. 
Conditional Probability (Formal)

- Let \((\Omega, p)\) be a probability space
- Let \(C\) be a random event, such that \(\Pr(C) > 0\)
- The \textit{subspace conditioned on} \(C\) is the probability space \((\Omega, p_C)\) where \(p_C = \frac{p}{\Pr(C)}\) 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 \textit{restrict} \((\Omega, p)\) to \(C\), and \textit{normalize}
- The probability in this subspace is denoted by \(\Pr(\cdot \mid C)\)
- We have: \(\Pr(A \mid C) = \frac{\Pr(A \cap C)}{\Pr(C)}\)
Examples

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

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

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

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

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

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

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

\[ = \Pr(\{\text{coke, milk}\} | \{\text{coke, coffee}\}) = \frac{1/2}{2/3} \]
Let \((\Omega, p)\) be a probability space, \(A\) and \(B\) two random events.

We say that \(A\) and \(B\) are \textit{probabilistically independent} (or \textit{independent events}) if

\[
\Pr(A \cap B) = \Pr(A) \times \Pr(B)
\]
Let \((\Omega, p)\) be a probability space, \(A\) and \(B\) two random events. We say that \(A\) and \(B\) are \textit{probabilistically independent} (or \textit{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

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

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

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

\[ \Pr(> 80 \text{ calories}) = \frac{2}{3} \quad \Pr(\text{begins with c}) = \frac{1}{2} \]
Example 1

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

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

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

\[ \Pr(> 80 \text{ calories}) = \frac{2}{3} \quad \Pr(\text{begins with c}) = \frac{1}{2} \]

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

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

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

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

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

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

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

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

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

\[ \Rightarrow \text{“}>80 \text{ calories” and \ “caffeine” are not independent events} \]
Conditional Independence

- Simple idea: events may become independent in a conditional subspace, even if correlated in the original space
Conditional Independence

- 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

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

\[ \Pr(>80 \text{ calories} \land \text{caffeine} \mid \text{not coke}) = \frac{1}{4} \]
\[ \Pr(>80 \text{ calories} \mid \text{not coke}) = \frac{1}{2} \quad \Pr(\text{caffeine} \mid \text{not coke}) = \frac{1}{2} \]
Example Revisited

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

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

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

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

\[ \Rightarrow "> 80 \text{ calories}" \text{ and } "\text{caffeine}" \text{ are conditionally independent given } "\text{not coke}" \]
Continuous probability space: sample space is *uncountable*
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”
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
  - $\Pr(x \in [0.1, 0.2]) = \Pr(x = 0.1) + \Pr(x = 0.101) + \ldots$
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 *measurable events*
    - \(\mathcal{F}\) should be nonempty, and should satisfy closure under complement and countable unions; this is called a \(\sigma\)-algebra over \(\Omega\)
  - \(p : \mathcal{F} \rightarrow [0, 1]\) 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
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 *measurable events*
    - \(\mathcal{F}\) should be nonempty, and should satisfy closure under complement and countable unions; this is called a \(\sigma\)-algebra over \(\Omega\)
  - \(p : \mathcal{F} \rightarrow [0, 1]\) 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
  - *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 $S$ be a schema. A *probabilistic database* over $S$ is a probability space over instances of $S$. 
**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 $\Omega \subseteq \text{Inst}(\mathcal{S})$. 
Running Example

In our running example, we have a sensing/vision system that observes faculty staff entering the building.
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?
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.
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
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
Table of Contents

1 Introduction

2 Basic Probability Theory

3 Probabilistic Databases

4 Representing Probabilistic Databases
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)
Restricted c-Instances

- 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 \{true, false\}, we write “$x$” instead of “$x = \text{true}$”
Example Notation

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

**ObsLects**

<table>
<thead>
<tr>
<th>lecturer</th>
<th>x</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ahuva</td>
<td></td>
</tr>
<tr>
<td>Avia</td>
<td></td>
</tr>
</tbody>
</table>

\[ \neg x \lor y \]

**ObsTAs**

<table>
<thead>
<tr>
<th>ta</th>
<th>x</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asma</td>
<td></td>
</tr>
<tr>
<td>Alon</td>
<td></td>
</tr>
</tbody>
</table>

\[ \neg x \lor z \]

**Courses**

<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>
Example

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

| Courses |
|---|---|---|
| course | lecturer | ta |
| DB      | Ahuva   | Alon |
| PL      | Avia    | Asma |

$x = y = z = \text{true}$
### Example

#### ObsLects

<table>
<thead>
<tr>
<th>lecturer</th>
<th>x [\neg x \lor y]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ahuva</td>
<td></td>
</tr>
<tr>
<td>Avia</td>
<td></td>
</tr>
</tbody>
</table>

#### ObsTAs

<table>
<thead>
<tr>
<th>ta</th>
<th>x [\neg x \lor z]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asma</td>
<td></td>
</tr>
<tr>
<td>Alon</td>
<td></td>
</tr>
</tbody>
</table>

#### Courses

<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>

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

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

<table>
<thead>
<tr>
<th>Courses</th>
</tr>
</thead>
<tbody>
<tr>
<td>course</td>
</tr>
<tr>
<td>DB</td>
</tr>
<tr>
<td>PL</td>
</tr>
</tbody>
</table>

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

Let $\mathcal{I} = (I, X, \varphi)$ be a restricted $c$-instance.
Semantics of a Restricted \( c \)-Instance

- Let \( I = (I, X, \varphi) \) be a restricted \( c \)-instance
- A *valuation* over \( I \) is a function \( \mu \) that maps every variable \( x \in X \) to a value \( \mu(x) \in \text{dom}(x) \)
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 $\mu(\mathcal{I})$:

$$\mu(\mathcal{I}) \overset{\text{def}}{=} \{ f \in I | \mu \models \varphi(f) \}$$
Semantics of a Restricted c-Instance

- Let $\mathcal{I} = (I, X, \varphi)$ be a restricted c-instance
- A \textit{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 $X$
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 $\llbracket \mathcal{I} \rrbracket$ defined by $\mathcal{I}$ is:
  \[
  \llbracket \mathcal{I} \rrbracket \overset{\text{def}}{=} \{ \mu(\mathcal{I}) \mid \mu \in M_X \} 
  \]
**Definition (pc-Instance)**

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

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

- $(I, X, \varphi)$ is a restricted $c$-instance
**Definition (pc-Instance)**

Let $S$ be a schema. A *pc-instance* over $S$ 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>lecturer</th>
<th>ObsLects</th>
<th>ta</th>
<th>ObsTAs</th>
<th>Course</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ahuva</td>
<td>$x$</td>
<td>Asma</td>
<td>$x$</td>
<td>DB</td>
</tr>
<tr>
<td>Avia</td>
<td>$\neg x \lor y$</td>
<td>Alon</td>
<td>$\neg x \lor z$</td>
<td>PL</td>
</tr>
</tbody>
</table>

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

- Let $\mathcal{P} = (I, X, \varphi, \pi)$ be a pc-instance
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 \textit{independently} assigning a random value for each variable $x \in X$
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 $\mathcal{I} = (I, X, \varphi)$.
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 $\mu(\mathcal{I})$. 

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 $\llbracket \mathcal{P} \rrbracket$, is $(\Omega, p)$ where:
  - $\Omega = \llbracket \mathcal{I} \rrbracket$
  - $p(J) = \sum_{\mu \in M_X | \mu(\mathcal{I}) = J} \pi(\mu)$
Example

<table>
<thead>
<tr>
<th>(\text{I} :)</th>
<th>(\text{ObsLects})</th>
<th>(\text{ObsTAs})</th>
<th>(\text{Courses})</th>
</tr>
</thead>
<tbody>
<tr>
<td>(x)</td>
<td>Ahuva</td>
<td>Asma</td>
<td>DB</td>
</tr>
<tr>
<td>(\neg x \lor y)</td>
<td>Avia</td>
<td>Alon</td>
<td>PL</td>
</tr>
<tr>
<td>(\pi(x, \text{true}) = 0.5)</td>
<td>(\pi(y, \text{false}) = 0.5)</td>
<td>(\pi(z, \text{true}) = 0.4)</td>
<td>(\pi(z, \text{false}) = 0.6)</td>
</tr>
</tbody>
</table>
Example

\[ \mathcal{I}: \]

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

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

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

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

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

<table>
<thead>
<tr>
<th>Lecturer</th>
<th>ObsLects</th>
<th>ObsTAs</th>
<th>Courses</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$x$</td>
<td>$x$</td>
<td>course</td>
</tr>
<tr>
<td>Ahuva</td>
<td>$\neg x \lor y$</td>
<td>Asma</td>
<td>lecturer</td>
</tr>
<tr>
<td>Avia</td>
<td></td>
<td>Alon</td>
<td>ta</td>
</tr>
</tbody>
</table>

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

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

\[
J = \mu_2(I) :
\]

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

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

### pc-Instances

<table>
<thead>
<tr>
<th>Lecturer</th>
<th>$x$</th>
<th>$\neg x \lor y$</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>$x$</th>
<th>$\neg x \lor z$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asma</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Alon</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Courses

<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>Alon</td>
</tr>
</tbody>
</table>

### Example

$I$:

$\pi(x, \text{true}) = 0.5 \quad \pi(y, \text{true}) = 0.3 \quad \pi(z, \text{true}) = 0.4$

$\pi(x, \text{false}) = 0.5 \quad \pi(y, \text{false}) = 0.7 \quad \pi(z, \text{false}) = 0.6$

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

$J = \mu_3(I)$:

<table>
<thead>
<tr>
<th>Lecturer</th>
<th>$x$</th>
<th>$\neg x \lor z$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avia</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Alon</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TA</th>
<th>$x$</th>
<th>$\neg x \lor z$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asma</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Alon</td>
<td></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>Alon</td>
</tr>
</tbody>
</table>

### Example

#### ObsLects
<table>
<thead>
<tr>
<th>lecturer</th>
<th>x</th>
<th>¬x ∨ y</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ahuva</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Avia</td>
<td>x</td>
<td></td>
</tr>
</tbody>
</table>

#### ObsTAs
<table>
<thead>
<tr>
<th>ta</th>
<th>x</th>
<th>¬x ∨ z</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asma</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Alon</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Courses
<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>

**I:**

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

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

**J = \mu_4(I):**

<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></td>
<td></td>
<td>lecturer</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DB</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PL</td>
</tr>
</tbody>
</table>
### Example

<table>
<thead>
<tr>
<th>(\mathcal{I}):</th>
<th>(\mathcal{ObsLects})</th>
<th>(\mathcal{ObsTAs})</th>
<th>(\mathcal{Courses})</th>
</tr>
</thead>
<tbody>
<tr>
<td>lecturer</td>
<td>(x)</td>
<td>ta</td>
<td>course</td>
</tr>
<tr>
<td>Ahuva</td>
<td>(\neg x \lor y)</td>
<td>Asma</td>
<td>DB</td>
</tr>
<tr>
<td>Avia</td>
<td></td>
<td>Alon</td>
<td>PL</td>
</tr>
<tr>
<td>(\pi(x, \text{true}) = 0.5)</td>
<td>(\pi(x, \text{false}) = 0.5)</td>
<td>(\pi(z, \text{true}) = 0.4)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>lecturer</th>
<th>ta</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\pi(y, \text{true}) = 0.3)</td>
<td>Ahuva</td>
<td>Alon</td>
</tr>
<tr>
<td>(\pi(y, \text{false}) = 0.7)</td>
<td>Avia</td>
<td>Asma</td>
</tr>
<tr>
<td>(\pi(z, \text{false}) = 0.6)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[p(J) = \pi(\mu_1) + \pi(\mu_2) + \pi(\mu_3) + \pi(\mu_4)\]
In a *Block-Independent-Disjoint Database*, or *BID database* for short, each relation is divided into *blocks* of tuples.
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*
A BID database can be represented as a special case of a pc-instance.
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:
  
  1. Each $\varphi(f)$ is an atomic condition (of the form $x = a$).
  2. No two facts use the same atomic condition.
  3. That is, if $f \neq f'$, then $\varphi(f)$ and $\varphi(f')$ use either different $x$s or different $a$s.
  4. Different relations use disjoint sets of variables.
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 $x = a$)
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
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>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>Ahuva</td>
</tr>
<tr>
<td>Aviv</td>
<td>Avner</td>
<td>Avia</td>
</tr>
<tr>
<td>Adi</td>
<td>Alma</td>
<td>Asma</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>x = 0</th>
<th>0.5</th>
<th>0.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>x = 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>y = 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>y = 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>z = 0</td>
<td>0.3</td>
<td>0.5</td>
</tr>
<tr>
<td>z = 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>z = 2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>w = 0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Due to the special form of a BID database, we can use a more intuitive representation, by assigning a primary key to each relation.
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:
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.
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 $\mathcal{R}$.
  - $\pi : I \to [0, 1]$ 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>Avia</td>
<td>Alon</td>
<td>Ahuva</td>
</tr>
<tr>
<td>Aviv</td>
<td>Avner</td>
<td>Avia</td>
</tr>
<tr>
<td>Adi</td>
<td>Alma</td>
<td>Asma</td>
</tr>
</tbody>
</table>

\[
\pi(x, 0) = 0.5 \quad \pi(x, 1) = 0.5
\]
\[
\pi(y, 0) = 0.3 \quad \pi(y, 1) = 0.5 \quad \pi(y, 2) = 0.2
\]
\[
\pi(z, 0) = 0.4 \quad \pi(z, 1) = 0.3 \quad \pi(z, 2) = 0.3
\]
\[
\pi(w, 0) = 0.9 \quad \pi(w, 1) = 0.1
\]
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>Ahuva</td>
</tr>
<tr>
<td>Aviv</td>
<td>Avner</td>
<td>Avia</td>
</tr>
<tr>
<td>Adi</td>
<td>Alma</td>
<td>Asma</td>
</tr>
</tbody>
</table>

\[
\begin{align*}
\pi(x, 0) &= 0.5 & \pi(x, 1) &= 0.5 \\
\pi(y, 0) &= 0.3 & \pi(y, 1) &= 0.5 & \pi(y, 2) &= 0.2 \\
\pi(z, 0) &= 0.4 & \pi(z, 1) &= 0.3 & \pi(z, 2) &= 0.3 \\
\pi(w, 0) &= 0.9 & \pi(w, 1) &= 0.1
\end{align*}
\]

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

| key      | ta     | course  |
| a        | Asma   | DB      |
| a        | Alon   | Ahuva   |
| a        | Avner  | Avia    |
| b        | Alma   | Asma    |

π(x, 0) = 0.5 \quad π(x, 1) = 0.5
π(y, 0) = 0.3 \quad π(y, 1) = 0.5 \quad π(y, 2) = 0.2
π(z, 0) = 0.4 \quad π(z, 1) = 0.3 \quad π(z, 2) = 0.3
π(w, 0) = 0.9 \quad π(w, 1) = 0.1
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.
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>ObsLects</th>
<th>ObsTAs</th>
<th>Courses</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ahuva</td>
<td>$x = 0$</td>
<td>Asma</td>
<td>DB</td>
</tr>
<tr>
<td>Avia</td>
<td>$y = 0$</td>
<td>Alon</td>
<td>PL</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$\pi(x, 0)$</th>
<th>$\pi(x, 1)$</th>
<th>$\pi(y, 0)$</th>
<th>$\pi(y, 1)$</th>
<th>$\pi(z, 0)$</th>
<th>$\pi(z, 1)$</th>
<th>$\pi(z, 2)$</th>
<th>$\pi(w, 0)$</th>
<th>$\pi(w, 1)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>0.5</td>
<td>0.3</td>
<td>0.7</td>
<td>0.4</td>
<td>0.3</td>
<td>0.3</td>
<td>0.9</td>
<td>0.1</td>
</tr>
</tbody>
</table>
Due to the special form of a tuple-independent database, we can again use a more intuitive representation.
Due to the special form of a tuple-independent database, we can again use a more intuitive representation.

A tuple-independent database $P$ over a signature $\mathcal{R}$ is a pair $(I, \pi)$, where:

- $I$ is an instance over $\mathcal{R}$
- $\pi : I \rightarrow [0, 1]$
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 $\mathcal{R}$
- $\pi : I \to [0, 1]$

The probability space $[\mathcal{P}]$ 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>lecturer</th>
<th>$x = 0$</th>
<th>$y = 0$</th>
<th>ta</th>
<th>$z = 0$</th>
<th>$w = 0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ahuva</td>
<td></td>
<td></td>
<td>Asma</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Avia</td>
<td></td>
<td></td>
<td>Alon</td>
<td></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>

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

<table>
<thead>
<tr>
<th>lecturer</th>
<th>$x = 0$</th>
<th>$y = 0$</th>
<th>ta</th>
<th>$z = 0$</th>
<th>$w = 0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ahuva</td>
<td></td>
<td></td>
<td>Asma</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Avia</td>
<td></td>
<td></td>
<td>Alon</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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


References II


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


References IV


End of lecture 9

Probabilistic Databases