πgora: An Integration System for Probabilistic Data
Dan Olteanu and Lampros Papageorgiou and Sebastiaan J. van Schaik
Department of Computer Science, University of Oxford, Oxford, OX1 3QD, UK

πgora’s Capabilities

Query Interface

pc-tables

FSTs

Bayesian networks

Finite State Transducers (FSTs)

The pc-tables are relations extended with a special column that encodes the uncertainty of the records using probabilistic events.

The pc-tables formalism supports select-project-join queries whose answers can be represented as pc-tables.

Bayesian networks support inference queries that ask for the conditional probability of an event given another event.

Bayesian networks are a natural fit for managing expert knowledge, where the probabilistic relationship between input random variables, which are observable quantities, unknown parameters, or hypotheses, exhibits conditional independence.

Bayesian networks export a relational schema without the events column.

Examples from the UCI machine learning repository at http://archive.ics.uci.edu/ml/datasets.html

Examples at http://hazy.cs.wisc.edu/hazy/staccato/

They support probabilistic processing to varying degrees:

The pc-tables formalism supports select-project-join queries whose answers can be represented as pc-tables.

Bayesian networks support inference queries that ask for the conditional probability of an event given another event.

Bayesian networks are stochastic automata used by optical character recognition programs, such as those powering Google Books, to capture probability distributions over all possible strings that could be represented in a given image.

They admit a common interpretation via the possible worlds semantics:

pc-tables represent finite probability distributions over sets of possible tables.

Bayesian networks represent finite probability distributions over sets of correlated observations.

FSTs represent finite probability distributions over sets of possible strings represented in an image.

Declarative representation of conditional independence:

\[
\begin{align*}
\text{P} & \text{H.tumor = true \land B.breast = left \land B.age = x}, \\
\text{P} & \text{B.age = x}, \\
\end{align*}
\]

Query: probability of a pregnant woman suffering from a left breast tumour, given that she also suffers from hypothyroidism.

\[
\begin{align*}
\text{SELECT} & \text{conf() FROM Hypothyroid H, Breast B} \\
\text{WHERE} & \text{B.tumor='true' AND B.breast='left' AND H.tumor='true' AND H.pregnant='true'} \\
\text{GIVEN} & \text{B.age=x AND H.hypothyroid='primary'}
\end{align*}
\]

Data sources: Bayesian networks Hypothyroid and Breast_cancer.

Strategy 1: purely Bayesian evaluation.

Phrase the SQL query as a sum of inference queries:

\[
\begin{align*}
\sum & \text{P(H.tumor = true \land B.breast = left \land B.age = x \land H.pregnant = true),} \\
& \text{(B.age = x AND H.age = x \land H.hypothyroid = primary))}
\end{align*}
\]

For a given value x for age, we have the inference query:

\[
\begin{align*}
\text{P(B.tumor = true \land B.breast = left \land H.tumor = true \land H.pregnant = true | (B.age = x \land H.age = x \land H.hypothyroid = primary))}
\end{align*}
\]

Since the two Bayesian networks are independent, we can regroup as follows:

\[
\begin{align*}
\text{P(B.tumor = true \land B.breast = left | B.age = x),} \\
\text{P(H.tumor = true \land H.pregnant = true | (H.age = x \land H.hypothyroid = primary))}
\end{align*}
\]


Resolve the GIVEN clause using the conditional probability formula:

\[
P(A | B) = \frac{P(A \land B)}{P(B)}
\]

Strategy 3: hybrid evaluation assuming Breast_cancer is a pc-table.

Split the query into the subqueries over each of Hypothyroid and Breast_cancer.

For each value of x for age we have the inference query over Hypothyroid:

\[
\forall x \land P_{H}(x) = P(H.tumor = true \land H.pregnant = true | H.age = x \land H.hypothyroid = primary)
\]

Rewrite the subquery over Breast_cancer by resolving the GIVEN clause:

\[
\begin{align*}
\text{CREATE TABLE T1 AS SELECT B.age, conf1 AS p1 FROM Breast_cancer B} \\
\text{WHERE B.tumor='true' AND B.breast='left' \text{GROUP BY B.age}} \\
\text{CREATE TABLE T2 AS SELECT B.age, conf2 AS p2 FROM Breast_cancer B \text{GROUP BY B.age}} \\
\text{CREATE TABLE T3 AS SELECT T1.age, p1/p2 AS P_B FROM T1, T2 \text{WHERE T1.age = T2.age}}
\end{align*}
\]

The query answer is obtained by joining the independent intermediate results:

\[
\sum_{	ext{age}} P_{B}(\text{age}) \times P_{H}(\text{age})
\]

where \(P_{B}(\text{age})\) denotes \(P_{B}\) for the tuple (age, \(P_{B}\)) in \(T_{3}\).