Modelling Retrieval Models in a Probabilistic Relational Algebra with a new Operator: The Relational Bayes

CWI, Amsterdam, December 2007
Thomas Roelleke
Queen Mary University of London
Outline

- Introduction
- Background: ProbDB, IR-on-DB, Retrieval Models, PRA
- The Relational Bayes
  - DB+IR Toy Database: `person(Name, City, Nationality); coll(Term, DocId);`
  - Motivation
  - City.Nationality and Nationality.City
  - Syntax and Examples
- Modelling Retrieval Models: TF-IDF, BIR, LM
- Implementation
- Summary and Outlook
Data and information independence

Application

User Interface

Business Logic

"all-in-one"

SQL

Data

all-in-one

three-tier

four-tier

Data and information independence
Generation of probabilistic databases
Layers
Generation of probabilistic databases

Non-probabilistic database  Bayes  Probabilistic database
## Layers

<table>
<thead>
<tr>
<th>Layer</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>External layer</td>
<td>Information sorted by relevance</td>
</tr>
<tr>
<td>Logical layer</td>
<td>Probabilistic relations</td>
</tr>
<tr>
<td></td>
<td>Probability estimation: Bayes<a href=""></a></td>
</tr>
<tr>
<td>Physical layer</td>
<td>Relational model/algebra; SQL</td>
</tr>
</tbody>
</table>
Probabilistic Databases

[Cavallo and Pittarelli, 1987]: Relational and probabilistic databases, information content (Shannon), probabilistic data dependencies

[Fuhr and Roelleke, 1997]: Probabilistic relational algebra for the integration of IR and DB: intensional vs extensional semantics, event expressions, solve Norbert’s ”db AND NOT ir OR ir AND NOT db”

[Dalvi and Suciu, 2004]: Efficient query evaluation: intensional semantics and possible worlds semantics, safe-plan optimisation algorithm
IR-on-DB

[Schek and Pistor, 1982]: Data Structures for an Integrated Database Management and Information Retrieval System

[Agrawal et al., 2002, Hristidis and Papakonstantinou, 2002]: DBXplorer, DISCOVER: keyword search

[Schefe, 1983]: Natuerlichsprachiger Zugang zu Datenbanken?

[Chaudhuri et al., 2004, Chaudhuri et al., 2006]: Probabilistic ranking of database query results (based on BIR model)

[Ercegovac et al., 2005]: TEXTURE benchmark (automatic scaling of benchmark; three competing systems)

[Corlacchia and de Vries, 2007]: A parametrised search system
(Probabilistic) Retrieval Models

[Robertson and Sparck Jones, 1976, Croft and Harper, 1979]:
BIR, “the probabilistic model”

[Wong and Yao, 1995, Roelleke et al., 2006]: $P(d \rightarrow q)$ and matrix framework


“probability of being informative”, on theoretical arguments for idf, on event spaces, a parallel derivation of probabilistic models


[Robert Ross, 2002]: probabilistic aggregates: the aggregates underline the difference between “normal” attributes and tuple probabilities

[Grossman and Frieder, 2004]: implement TF-IDF VSM in SQL

[Roelleke and Fuhr, 1996, Fuhr et al., 1998, Lalmas et al., 2002]: POOL SIGIR 06, “Dolores” SIGIR 98, POOL (in Intelligent Exploration of the Web 02), Logic in IR.
### Example

#### Coll

<table>
<thead>
<tr>
<th>Term</th>
<th>DocId</th>
</tr>
</thead>
<tbody>
<tr>
<td>sailing</td>
<td>doc1</td>
</tr>
<tr>
<td>boats</td>
<td>doc1</td>
</tr>
<tr>
<td>sailing</td>
<td>doc2</td>
</tr>
<tr>
<td>sailing</td>
<td>doc2</td>
</tr>
<tr>
<td>boats</td>
<td>doc2</td>
</tr>
<tr>
<td>sailing</td>
<td>doc3</td>
</tr>
<tr>
<td>east</td>
<td>doc3</td>
</tr>
<tr>
<td>coast</td>
<td>doc3</td>
</tr>
<tr>
<td>sailing</td>
<td>doc4</td>
</tr>
<tr>
<td>boats</td>
<td>doc5</td>
</tr>
</tbody>
</table>

#### probColl: “tf”

<table>
<thead>
<tr>
<th>Prob</th>
<th>Term</th>
<th>DocId</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>sailing</td>
<td>doc1</td>
</tr>
<tr>
<td>0.5</td>
<td>boats</td>
<td>doc1</td>
</tr>
<tr>
<td>0.66</td>
<td>sailing</td>
<td>doc2</td>
</tr>
<tr>
<td>0.33</td>
<td>boats</td>
<td>doc2</td>
</tr>
<tr>
<td>0.33</td>
<td>sailing</td>
<td>doc3</td>
</tr>
<tr>
<td>0.33</td>
<td>east</td>
<td>doc3</td>
</tr>
<tr>
<td>0.33</td>
<td>coast</td>
<td>doc3</td>
</tr>
<tr>
<td>1.0</td>
<td>sailing</td>
<td>doc4</td>
</tr>
<tr>
<td>1.0</td>
<td>boats</td>
<td>doc5</td>
</tr>
</tbody>
</table>
-- PSQL
INSERT INTO probQuery VALUES
  0.4 ('sailing', 'q2'),
  0.6 ('boats', 'q2');

-- Query
CREATE VIEW retrieved AS
  SELECT DocId
  FROM probQuery, probColl
  WHERE probQuery.Term = probColl.Term;
### Non-distinct

<table>
<thead>
<tr>
<th>Prob</th>
<th>DocId</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.4 · 0.5</td>
<td>doc1</td>
</tr>
<tr>
<td>0.6 · 0.5</td>
<td>doc1</td>
</tr>
<tr>
<td>0.4 · 0.66</td>
<td>doc2</td>
</tr>
<tr>
<td>0.6 · 0.33</td>
<td>doc2</td>
</tr>
<tr>
<td>0.4 · 0.33</td>
<td>doc3</td>
</tr>
<tr>
<td>0.4 · 1.0</td>
<td>doc4</td>
</tr>
<tr>
<td>0.6 · 1.0</td>
<td>doc5</td>
</tr>
</tbody>
</table>

### Distinct (aggregated)

<table>
<thead>
<tr>
<th>Prob</th>
<th>DocId</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>doc1</td>
</tr>
<tr>
<td>1.0</td>
<td>doc2</td>
</tr>
<tr>
<td>0.4 · 0.33</td>
<td>doc3</td>
</tr>
<tr>
<td>0.4</td>
<td>doc4</td>
</tr>
<tr>
<td>0.6</td>
<td>doc5</td>
</tr>
</tbody>
</table>

Aggregation of non-distinct tuples:

```
SELECT DISJOINT DocId FROM retrieved;
```
- Where do the probabilities in \(\text{probColl} \) ("tf") and \(\text{probQuery} \) come from?
- How can the estimation be described algebraically?
### Motivation

A `Toy Database` can be defined as a set of relations and their associated data. In this case, we have a `person` relation with attributes for `Name`, `City`, and `Nationality`. Additionally, there is a `coll` relation with attributes for `Term` and `DocId`.

#### person

<table>
<thead>
<tr>
<th>Name</th>
<th>City</th>
<th>Nationality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Peter</td>
<td>London</td>
<td>German</td>
</tr>
<tr>
<td>Paul</td>
<td>London</td>
<td>Irish</td>
</tr>
<tr>
<td>Mary</td>
<td>London</td>
<td>Irish</td>
</tr>
<tr>
<td>Thomas</td>
<td>London</td>
<td>German</td>
</tr>
<tr>
<td>Thomas</td>
<td>Dortmund</td>
<td>German</td>
</tr>
<tr>
<td>Thomas</td>
<td>Hamburg</td>
<td>German</td>
</tr>
<tr>
<td>Hany</td>
<td>London</td>
<td>Egyptian</td>
</tr>
<tr>
<td>Hany</td>
<td>London</td>
<td>Polish</td>
</tr>
<tr>
<td>Jun</td>
<td>London</td>
<td>Chinese</td>
</tr>
<tr>
<td>Zhi</td>
<td>London</td>
<td>Chinese</td>
</tr>
</tbody>
</table>

#### coll

<table>
<thead>
<tr>
<th>Term</th>
<th>DocId</th>
</tr>
</thead>
<tbody>
<tr>
<td>sailing</td>
<td>doc1</td>
</tr>
<tr>
<td>boats</td>
<td>doc1</td>
</tr>
<tr>
<td>sailing</td>
<td>doc2</td>
</tr>
<tr>
<td>sailing</td>
<td>doc2</td>
</tr>
<tr>
<td>boats</td>
<td>doc2</td>
</tr>
<tr>
<td>sailing</td>
<td>doc3</td>
</tr>
<tr>
<td>east</td>
<td>doc3</td>
</tr>
<tr>
<td>coast</td>
<td>doc3</td>
</tr>
<tr>
<td>sailing</td>
<td>doc4</td>
</tr>
<tr>
<td>boats</td>
<td>doc5</td>
</tr>
</tbody>
</table>
The Relational Bayes Modelling Retrieval Models

CREATE VIEW City_Nationality AS
SELECT DISJOINT City, Nationality
FROM person | DISJOINT(Nationality);

<table>
<thead>
<tr>
<th>Prob</th>
<th>City</th>
<th>Nationality</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.500000</td>
<td>&quot;London&quot;</td>
<td>&quot;German&quot;</td>
</tr>
<tr>
<td>1.000000</td>
<td>&quot;London&quot;</td>
<td>&quot;Irish&quot;</td>
</tr>
<tr>
<td>0.250000</td>
<td>&quot;Dortmund&quot;</td>
<td>&quot;German&quot;</td>
</tr>
<tr>
<td>0.250000</td>
<td>&quot;Hamburg&quot;</td>
<td>&quot;German&quot;</td>
</tr>
<tr>
<td>1.000000</td>
<td>&quot;London&quot;</td>
<td>&quot;Egyptian&quot;</td>
</tr>
<tr>
<td>1.000000</td>
<td>&quot;London&quot;</td>
<td>&quot;Polish&quot;</td>
</tr>
<tr>
<td>1.000000</td>
<td>&quot;London&quot;</td>
<td>&quot;Chinese&quot;</td>
</tr>
</tbody>
</table>
## Nationality_City

<table>
<thead>
<tr>
<th></th>
<th>City</th>
<th>Nationality</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.250000</td>
<td>&quot;London&quot;</td>
<td>&quot;German&quot;</td>
</tr>
<tr>
<td>0.250000</td>
<td>&quot;London&quot;</td>
<td>&quot;Irish&quot;</td>
</tr>
<tr>
<td>1.000000</td>
<td>&quot;Dortmund&quot;</td>
<td>&quot;German&quot;</td>
</tr>
<tr>
<td>1.000000</td>
<td>&quot;Hamburg&quot;</td>
<td>&quot;German&quot;</td>
</tr>
<tr>
<td>0.125000</td>
<td>&quot;London&quot;</td>
<td>&quot;Egyptian&quot;</td>
</tr>
<tr>
<td>0.125000</td>
<td>&quot;London&quot;</td>
<td>&quot;Polish&quot;</td>
</tr>
<tr>
<td>0.250000</td>
<td>&quot;London&quot;</td>
<td>&quot;Chinese&quot;</td>
</tr>
</tbody>
</table>

-- PSQL

```sql
CREATE VIEW Nationality_City AS
    SELECT DISJOINT City, Nationality
    FROM person | DISJOINT(City);
```

# PRA

```plaintext
Nationality_City =
    Project DISJOINT[$2,$3] (Bayes[$2](person))
```
Bayes \langle \text{estAssumption}\rangle \quad \text{-- PSQL}

\text{SELECT} \ldots \text{FROM} \ldots \text{WHERE}
\langle \text{evidenceKey}\rangle \langle \text{prae}\rangle
\text{ASSUMPTION} \langle \text{estAssumption}\rangle
\text{EVIDENCE KEY} \langle \text{evidenceKey}\rangle

evidence = \text{Project estAssumption[i1..in](a)}

T := T_a

P(τ) = \begin{cases} 
\frac{P_a(τ)}{\log P_a(τ)} & \frac{P_{\text{evidence}(τ[i1..in])}}{\log P_{\text{evidence}(τ[i1..in])}} 
\end{cases}

In 2003 (early Bayes), Bayes result was distinct; in 2004, revision led to \( T := T_a \).
Motivation: “Seamlessly” integrated DB+IR technology

- Support document/context and tuple retrieval
- Support free-text and semantic/data retrieval
- Support the flexible modelling of all retrieval models
- Support the high-level (abstract) modelling of general and specific retrieval tasks (ad-hoc retrieval, classification, summarisation, structured document retrieval, hypertext retrieval, multimedia retrieval, ...)
- Support text, XML, and SQL
Retrieval Models: Notation

Notation ([Roelleke et al., 2006]):

\[
\begin{align*}
\n_D(t, x) & \quad \text{number of Documents} \\
_L(t, x) & \quad \text{number of Locations} \\
P_D(t, x) & := \frac{n_D(t, x)}{N_D(x)} \quad \text{document-based term probability} \\
P_L(t, x) & := \frac{n_L(t, x)}{N_L(x)} \quad \text{location-based term probability}
\end{align*}
\]
Retrieval Models: RSV’s

\[ RSV_{\text{TF-IDF}}(d, q) := \sum_{t \in q} tf(t, d) \cdot idf(t, c) \]  

(1)

\[ RSV_{\text{BIR}}(d, q) := \sum_{t \in d \cap q} \log \frac{P(t|r)}{P(t|\bar{r})} \]  

(2)

\[ P(q|d) := \prod_{t \in q} [\delta \cdot P(t|d) + (1 - \delta) \cdot P(t|c)] \]  

(3)
Retrieval Models: Relationships, Rewritings

\[ RSV_{\text{BIR}}(d, q) := \sum_{t \in d \cap q} \text{idf}(t, \bar{r}) - \text{idf}(t, r) \]  \hspace{1cm} (4)

\[ RSV_{\text{LM}}(d, q) := \log \frac{P(q|d)}{\prod_{t \in q}(1 - \delta) \cdot P(t|c)} \]  \hspace{1cm} (5)

\[ = \sum_{t \in q} \log \left[ 1 + \frac{\delta \cdot P(t|d)}{(1 - \delta) \cdot P(t|c)} \right] \]  \hspace{1cm} (6)

eqn 4: BIR and IDF:
[Robertson, 2004, de Vries and Roelleke, 2005]
eqn 6: LM:
[Hiemstra, 2000]
Investigating the relationships important for “good” design of probabilistic relational modelling.
-- inverse document frequency
CREATE VIEW idf AS
  SELECT Term FROM coll
  ASSUMPTION MAX INFORMATIVE
  EVIDENCE KEY ();

-- query term weighting
CREATE VIEW wQuery AS
  SELECT Term, QueryId FROM Query, idf
  WHERE Query.Term = idf.Term;

-- normalisation
CREATE VIEW norm_wQuery AS
  SELECT Term, QueryId FROM wQuery
  EVIDENCE KEY (QueryId);
-- retrieve documents
CREATE VIEW std_tf_idf_retrieve AS
  SELECT DISJOINT DocId, QueryId
  FROM norm_wQuery, tf
  WHERE norm_wQuery.Term = tf.Term;
The Relational Bayes

Modelling Retrieval Models

Implementation

Summary and Outlook

Retrieval Models: Notation

Retrieval Models: RSV's

Retrieval Models: Relationships, Rewritings

Modelling TF-IDF

Modelling BIR

Modelling LM

# tf := P(t|d)

--- P(t occurs | d)

CREATE VIEW tf AS

SELECT DISJOINT Term, DocId
FROM coll |

DISJOINT (DocId);

# idf(t,c) :=

--- P(t informs | c)

CREATE VIEW idf AS

SELECT Term
FROM coll

ASSUMPTION MAX_IDF

EVIDENCE KEY ();

# tf =

Project DISJOINT(

Bayes DISJOINT[](coll));

# idf(t,c) :=

# -log P(t|c) /

# max_idf(c)

idf =

Bayes MAX_IDF[]()

Project[](coll));
-- idf in collection / non-relevant:
CREATE VIEW idf_c AS
    SELECT Term FROM coll
ASSUMPTION MAX_IDF
EVIDENCE KEY ();

-- idf in relevant:
CREATE VIEW idf_r AS
    SELECT Term FROM relColl
ASSUMPTION MAX_IDF
EVIDENCE KEY ();
CREATE VIEW docModel AS
  SELECT Term, DocId FROM lambda1, pt_d;

CREATE VIEW collModel AS
  SELECT Term, DocId FROM lambda2, pt_c, retrieved;

-- combine document and collection models
CREATE VIEW lm1_mix AS
  docModel UNION DISJOINT collModel;

-- retrieve documents
CREATE VIEW lm1_retrieve AS
  SELECT SUM_LOG DocId, QueryId
  FROM Query, lm1_mix
  WHERE Query.Term = lm1_mix.Term;
Implementation

- Retrieval of $P(t|c)$ probabilities in $O(1)$ (special index)
- Incremental update facility
- $P(t|d)$ probabilities/views are materialised off-line; future research
- Difficulty: System A runs SQL, System B runs PSQL: How to compare?
- top-k and early-response processing
Summary

- Probabilistic DB, IR-on-DB, PRA, Retrieval Models
- The magnificent five (Select, Project, Join/Multiply, Unite, Subtract): describe probability *AGGREGATION*
- The relational Bayes: describe probability *ESTIMATION*
- DB+IR requires “relaxed” probability theory
  - *idf*-based ("informativeness") probabilities
  - here and there, relax the boundaries of probabilistic modelling and rethink the genuine formulation of IR models
- Scalability: $O(1)$ retrieval of Bayes tuples for $P(t|c)$, given Bayes-oriented indexing structure
Outlook

- Database/tuple ranking: modelling and application of retrieval models to SELECT-FROM-WHERE; entropy-based ranking
- Design and verification of probabilistic logical programs
- Optimisation: Semantic, algebraic, and processing (Hengzhi Wu)
- High-level languages: RDF-SPARQL → PSQL/PRA (Hany Azzam)
- POLIS: Probabilistic Object-oriented logic for information summarisation (Fred Forst)
- POLAR: Probabilistic Object-oriented annotation-based retrieval (Ingo Frommholz)
Dbxplorer: A system for keyword-based search over relational databases.
In ICDE, pages 5–16.

The management of probabilistic data.

Fuzzy queries and relational databases.

The theory of probabilistic databases.

Probabilistic ranking of database query results.
In VLDB, pages 888–899.

Probabilistic information retrieval approach for ranking of database query results.

A parameterised search system.
In ECIR, pages 4–15.
Using probabilistic models of document retrieval without relevance information.

*Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, New York. ACM.

Efficient query evaluation on probabilistic databases.

Relevance information: A loss of entropy but a gain for idf?
In *ACM SIGIR*, Salvador, Brazil.

The texture benchmark: Measuring performance of text queries on a relational dbms.
In *VLDB*, pages 313–324.

A probabilistic framework for vague queries and imprecise information in databases.

Dolores: A system for logic-based retrieval of multimedia objects.

Thomas Roelleke  The Relational Bayes
Introduction

Background

The Relational Bayes

Modelling Retrieval Models

Implementation

Summary and Outlook

In [Croft et al., 1998], pages 257–265.

**Fuhr, N. and Roelleke, T. (1996).**
A probabilistic NF2 relational algebra for integrated information retrieval and database systems.

**Fuhr, N. and Roelleke, T. (1997).**
A probabilistic relational algebra for the integration of information retrieval and database systems.


**Hiemstra, D. (2000).**
A probabilistic justification for using tf.idf term weighting in information retrieval.

**Hristidis, V. and Papakonstantinou, Y. (2002).**
Discover: Keyword search in relational databases.

**Lafferty, J. and Zhai, C. (2003).**
Intelligent hypermedia retrieval.
Springer-Verlag Group (Physica-Verlag).

An extended relational database model for uncertain and imprecise information.

Text retrieval and the relational model.

Vague: A user interface to relational databases that permits vague queries.

Accommodating imprecision in database systems: Issues and solutions.
Sigmod record, 19(4):69.

A straightforward NF2 relational interface with applications in information retrieval.

A language modeling approach to information retrieval.
In [Croft et al., 1998], pages 275–281.
Probabilistic aggregates.

Understanding inverse document frequency: On theoretical arguments for idf.

On event spaces and probabilistic models in information retrieval.

Relevance weighting of search terms.

A frequency-based and a Poisson-based probability of being informative.

Retrieval of complex objects using a four-valued logic.

A general matrix framework for modelling information retrieval.

*Journal on Information Processing & Management (IP&M), Special Issue on Theory in Information Retrieval, 42(1).*

A parallel derivation of probabilistic information retrieval models.
In *ACM SIGIR*, pages 107–114, Seattle, USA.

Natürlichsprachiger zugang zu datenbanken?

Data structures for an integrated database management and information retrieval system.
In *Proceedings of the 8th International Conference on Very Large Data Bases*, pages 197–207, Los Altos, California. Morgan Kaufman.

The relational model with relation-valued attributes.

On modeling information retrieval with probabilistic inference.