Variety-aware analysis of document-oriented databases

Stefano Rizzi
DISI - University of Bologna, Italy
A joint work with Enrico Gallinucci and Matteo Golfarelli
Summary

Introduction
◦ NoSQL databases
◦ OLAP

Related work & contribution
(Explain: Schema profiling)
Analyze: Approximate OLAP

Conclusion
Introduction

In recent years, NoSQL databases have been progressively eroding the predominance of relational databases.

Market growth, 2019 (Gartner)

- RDBMSs: +15.2%
- NoSQL: +51.7%
Introduction

A NoSQL database provides a mechanism for storage and retrieval of data that is modeled differently from the tabular relations used in relational databases

- key-value store
- columnar
- graph-based
- document-based

The particular suitability of a given NoSQL database depends on the problem it must solve
Introduction

SQL

NoSQL

- Column-Family
- Graph
- Document
- Key-Value
Introduction

Why NoSQL?

- Better scaling
- No ACID transactions
- No need for a unique schema

Most new models adopt a schemaless representation for data

- Schema is a “soft” concept and the instances referring to the same concept can be stored using different local schemas
- Schemaless databases are preferred for storing heterogeneous data with variable schemas, such as those located in data lakes
Introduction

Typical schema variants that can be found within a NoSQL database consist in

- missing or additional attributes
- different names or types for an attribute
- different nesting

The absence of a unique schema gives flexibility to operational applications, but...

- ...some extra effort is required to understand the rules that drove the use of alternative schemas
- ...there is more complexity in analytical applications and OLAP, where queries often involve instances with different (possibly conflicting) schemas
Introduction

OLAP (On-Line Analytical Processing)

- Dynamic, multidimensional analyses that need to read a huge quantity of data to compute a set of numbers summing up the performance of a company

<table>
<thead>
<tr>
<th>Quarter</th>
<th>Metrics</th>
<th>Customer Region</th>
<th>Dollar Sales</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>North-East</td>
<td></td>
</tr>
<tr>
<td>Q1 1997</td>
<td>$ 1.526</td>
<td>$ 1.249</td>
<td>$ 978</td>
</tr>
<tr>
<td></td>
<td></td>
<td>South-East</td>
<td></td>
</tr>
<tr>
<td>Q2 1997</td>
<td>$ 2.879</td>
<td>$ 1.415</td>
<td>$ 2.644</td>
</tr>
<tr>
<td></td>
<td></td>
<td>North-West</td>
<td></td>
</tr>
<tr>
<td>Q3 1997</td>
<td>$ 3.010</td>
<td>$ 1.772</td>
<td>$ 5.076</td>
</tr>
<tr>
<td></td>
<td></td>
<td>South-West</td>
<td></td>
</tr>
<tr>
<td>Q4 1997</td>
<td>$ 2.711</td>
<td>$ 5.030</td>
<td>$ 2.025</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Central</td>
<td></td>
</tr>
<tr>
<td>Q1 1998</td>
<td>$ 5.288</td>
<td>$ 3.399</td>
<td>$ 4.935</td>
</tr>
<tr>
<td></td>
<td></td>
<td>South</td>
<td></td>
</tr>
<tr>
<td>Q2 1998</td>
<td>$ 1.773</td>
<td>$ 3.698</td>
<td>$ 1.802</td>
</tr>
<tr>
<td></td>
<td></td>
<td>North-West</td>
<td></td>
</tr>
<tr>
<td>Q3 1998</td>
<td>$ 1.618</td>
<td>$ 5.402</td>
<td>$ 1.709</td>
</tr>
<tr>
<td></td>
<td></td>
<td>South-West</td>
<td></td>
</tr>
<tr>
<td>Q4 1998</td>
<td>$ 2.051</td>
<td>$ 2.986</td>
<td>$ 4.654</td>
</tr>
</tbody>
</table>
Introduction

Data Warehouse

- a repository of information that collects and integrates data coming from different, heterogeneous sources making them available for analyses aimed at planning and decision making

In a data warehouse, data are stored in multidimensional form
Related work

Schema discovery from XML/JSON documents
  ◦ dealing with heterogeneity, data quality, versioning, similarity, comprehensiveness...
  ◦ ...to produce unified schemas, schema matches, skeletons...
  ◦ ...to be used for querying, integration, validation

Schema matching for XML/JSON documents
  ◦ using clustering, machine learning...
  ◦ ...possible considering a context

OLAP analysis of document collections
  ◦ schema-on-write, schema-on-read...
  ◦ but no (or limited) management of variety
Our approach

Stop fighting against schema variety and welcome data heterogeneity as an inherent source of information wealth in schemaless sources

◦ focus on collections of documents in document-oriented databases

1. Schema profiling, to explain the schema variants within a collection by capturing the hidden rules explaining the use of these variants\(^1\)

2. Approximate OLAP, to enable multidimensional querying of collections with variable schemas\(^2\)
Schema profiling

*Explain the schema variants within a collection by capturing the hidden rules explaining the use of these variants*

Useful to...

◦ ...decode the behavior of an undocumented application that manages a document-base
◦ ...carry out a data quality project on schemaless data
◦ ...enable a schema-on-read approach to query a document-oriented database
◦ ...design a data warehouse on top of a schemaless data source
Approximate OLAF

Enable multidimensional querying of collections with variable schemas

OLAP querying on a “soft” schema where each source attribute is present to some extent

- First variety-aware approach for approximate OLAP on document-oriented databases
- Querying is directly carried out on the data source (no cube materialization)
- Inclusive solution to integration
- Deal with both inter-schema and intra-schema variety
- Query reformulation on heterogeneous documents builds on a formal approach, which ensures its correctness and completeness
Overview

Collection

Schema integration

FD enrichment

Querying

Dependency graph

Global schema, Mappings

Local schemas

LEGEND
- data
- metadata
- control
Schema extraction
Goal: find the (local) schema of a document

```json
[  
  {  
    "_id": ObjectId("54a4332f44cfc02424f961d4"),
    "User":
    {  
      "FullName": "John Smith",
      "Age": 42  
    },
    "StartedOn": ISODate("2017-06-15T10:20:44.000Z"),
    "Facility":
    {  
      "Name": "PureGym Piccadilly",
      "Chain": "PureGym"  
    },
    "SessionType": "RunningProgram",
    "DurationMins": 90,
    "Exercises":
    [  
      {  
        "Type": "Leg press",
        "ExCalories": 28,
        "Sets":
        [  
          {  
            "Reps": 14,
            "Weight": 60  
          },
          
            
            
          
        ],
        
        "Type": "Tapis roulant"  
      },
      
      "Type": "Tapis roulant"  
    ]
  }  
...  
]}
```
Schema integration
Goal: integrate the local schemas to obtain a single schema

Integration through mappings

Local 1
- _id
- User.FullName
- User.Age
- StartedOn
- Facility.Name
- Facility.Chain
- SessionType
- DurationMins

Exercises
- Exercises_id
- Type
- ExCalories

Sets
- Sets_id
- Reps
- Weight

GLOBAL
- _id
- User.FullName
- User.FirstName
- User.LastName
- StartedOn
- Facility.Name
- Facility.Chain
- Facility.City
- SessionType
- DurationMins

Exercises
- Exercises_id
- Type
- ExCalories

Sets
- Sets_id
- Reps
- Weight

Series
- Series_id
- ExType
- Reps
- Weight

Local 2
- _id
- FirstName
- LastName
- Date
- Gym.Name
- Gym.City
- SessionType
- DurationSecs
Schema integration

Goal: integrate the local schemas to obtain a single schema

Integration through mappings
  - **Primitive mappings**
    - Only *exact* mappings
    - Transcoding functions required

INRIA Webinaire - Jan. 29, 2021
Schema integration
Goal: integrate the local schemas to obtain a single schema

Integration through mappings
- **Primitive mappings**
  - Only *exact* mappings
  - Transcoding functions required
- **Array mappings**
  - Define the context of primitive mappings
Schema integration
Goal: integrate the local schemas to obtain a single schema

1. Build a preliminary global schema as the name-based union of all local schemas (automated)

2. Refine the preliminary global schema by iteratively merging matching (sets of) fields (semi-automated)
FD enrichment

Goal: give an MD view of the global schema to enable OLAP analyses

To build MD hierarchies we search for functional dependencies (FDs)

FDs can be identified
- From the schema (intensional)
FD enrichment
Goal: give an MD view of the global schema to enable OLAP analyses

To build MD hierarchies we search for functional dependencies (FDs)

FDs can be identified
- From the schema (intensional)
- From the data (approximate FDs)
FD enrichment
Goal: give an MD view of the global schema to enable OLAP analyses

To build MD hierarchies we search for functional dependencies (FDs)

FDs can be identified

- From the schema (intensional)
- From the data (approximate FDs)

With FDs we define a dependency graph

- each field has a support, i.e., an indicator of how frequently it appears

high-support field

low-support field

INRIA Webinaire - Jan. 29, 2021
FD enrichment

Goal: give an MD view of the global schema to enable OLAP analyses

To build MD hierarchies we search for functional dependencies (FDs)

FDs can be identified
- From the schema (intensional)
- From the data (*approximate FDs*)

With FDs we define a *dependency graph*
- each field has a *support*, i.e., an indicator of how frequently it appears
- each FD has an *accuracy*, i.e., an indicator of how frequently it holds
Querying

Goal: **formulate**, execute, evaluate

**OLAP query**

- Group-by set (non-empty)
- Selection predicate (optional)
- Measure and aggregation operator

\[
\text{avg}\left(\text{ExCalories} \geq 60\right)
\]
For reformulating queries from the global schema to the local schemas we rely on the BIN framework\(^1\)

- an approach to enable OLAP on a P2P data warehousing architecture

Compliance with the BIN framework guarantees the correctness of the approach

---

Execution requires to

- Translate each local query to MongoDB according to its query language
- Execute each local query
- Collect and aggregate the results

```
{ $unwind: "$Exercises" },
{ $unwind: "$Exercises_SETS" },
{ $match: { "User.Age": { $gte: 60 } } }
{ $project: {
  "Facility.City": { $ifNull:
    ["$FacilityCity","$FacilityName"]
  },
  "Exercises.Type": 1,
  "Exercises_SETS.Weight": 1,
  "balanced": {
    $cond: ["$FacilityCity",false,true]
  }
}},
{ $group: {
  "id": {
    "Facility.City","$FacilityCity",
    "Exercises.Type","$Exercises.Type",
    "balanced","$balanced"
  },
  "Exercises_SETS.Weight": {
    $avg: "$Exercises_SETS.Weight"
  },
  "count": { $sum: 1 },
  "count-m": { $sum: {
    $cond: ["$Exercises_SETS.Weight",1,0]
  } }
} }
```
Querying

Goal: formulate, execute, evaluate

We introduce indicators to evaluate the quality of an OLAP query in terms of coverage and reliability

- **Completeness** takes into account missing values of hierarchies
- **Precision** takes into account missing values of measures

To estimate these indicators *before* query execution, we resort to schema profiling
Experimental results

Efficiency

◦ extraction

<table>
<thead>
<tr>
<th># records</th>
<th>DB size</th>
<th>Time (standalone)</th>
<th>Time (cluster)</th>
</tr>
</thead>
<tbody>
<tr>
<td>5 K</td>
<td>2 MB</td>
<td>4 sec</td>
<td>3 sec</td>
</tr>
<tr>
<td>50 K</td>
<td>20 MB</td>
<td>33 sec</td>
<td>19 sec</td>
</tr>
<tr>
<td>500 K</td>
<td>197 MB</td>
<td>6 min</td>
<td>3 min</td>
</tr>
<tr>
<td>5 M</td>
<td>1.7 GB</td>
<td>60 min</td>
<td>32 min</td>
</tr>
</tbody>
</table>

◦ enrichment (overall, 32 minutes)
Experimental results

Effectiveness

- query completeness and precision during a progressive integration of local schemas
Conclusion

Dealing with heterogeneity and schema variety intrinsic to document-oriented DBs is a challenge.

We claim variety should be considered as a source of information wealth and shown to users together with an assessment of its impact in terms of completeness and precision.

To this end, we enable OLAP queries over the collection and make users aware of the impact of schema variety through a set of indicators related to query completeness and precision.
Thanks