Data Profiling
Helena Galhardas
DEI/IST

References

- Slides “Data Profiling” course, Felix Naumann, Trento, July 2015
- T. Papenbrock and others, “Data Profiling with Metanome”, demo paper, VLDB 2015
Definition Data Profiling

- **Data profiling** is the process of examining the data available in an existing data source [...] and collecting statistics and information about that data.
  
  *Wikipedia 09/2013*

- **Data profiling** refers to the activity of creating small but informative summaries of a database.
  
  Ted Johnson, Encyclopedia of Database Systems

- **Data profiling** is the set of activities and processes to determine the metadata about a given dataset.
<table>
<thead>
<tr>
<th>Column labels</th>
<th>Number of rows</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>50</td>
</tr>
<tr>
<td>Age</td>
<td>50</td>
</tr>
<tr>
<td>Gender</td>
<td>50</td>
</tr>
<tr>
<td>Occupation</td>
<td>50</td>
</tr>
<tr>
<td>City</td>
<td>50</td>
</tr>
<tr>
<td>State</td>
<td>50</td>
</tr>
<tr>
<td>Zip</td>
<td>50</td>
</tr>
<tr>
<td>Phone</td>
<td>50</td>
</tr>
<tr>
<td>Email</td>
<td>50</td>
</tr>
<tr>
<td>Address</td>
<td>50</td>
</tr>
<tr>
<td>Income</td>
<td>50</td>
</tr>
</tbody>
</table>

### Example Data

<table>
<thead>
<tr>
<th>Name</th>
<th>Age</th>
<th>Gender</th>
<th>Occupation</th>
<th>City</th>
<th>State</th>
<th>Zip</th>
<th>Phone</th>
<th>Email</th>
<th>Address</th>
<th>Income</th>
</tr>
</thead>
<tbody>
<tr>
<td>John Smith</td>
<td>30</td>
<td>Male</td>
<td>Engineer</td>
<td>New York</td>
<td>NY</td>
<td>10001</td>
<td>555-123-4567</td>
<td><a href="mailto:john.smith@gmail.com">john.smith@gmail.com</a></td>
<td>50000.00</td>
<td></td>
</tr>
<tr>
<td>Jane Doe</td>
<td>25</td>
<td>Female</td>
<td>Nurse</td>
<td>Los Angeles</td>
<td>CA</td>
<td>90001</td>
<td>444-888-7777</td>
<td><a href="mailto:jane.doe@gmail.com">jane.doe@gmail.com</a></td>
<td>40000.00</td>
<td></td>
</tr>
<tr>
<td>Michael</td>
<td>40</td>
<td>Male</td>
<td>Lawyer</td>
<td>Chicago</td>
<td>IL</td>
<td>60601</td>
<td>333-999-8888</td>
<td><a href="mailto:michael.lawyer@gmail.com">michael.lawyer@gmail.com</a></td>
<td>60000.00</td>
<td></td>
</tr>
</tbody>
</table>

---

*Note: This is a partial example to illustrate the structure of the data.*
Many interesting questions remain

• What are the possible primary keys and foreign keys?
  – Phone
  – firstname, lastname, street

• Are there any functional dependencies?
  – zip -> city
  – race -> voting behavior

• Which columns correlate?
  – Date-of-Birth and first name
  – State and last name

• What are frequent patterns in a column?
  – ddddd
  – dd aaaa St

### Results of data profiling

- Encompasses several methods to examine datasets and produce metadata
- **Simple results** to compute:
  - Number of null and distinct values in a column
  - Data type of a column
  - Most frequent patterns of data values in a column
- **More difficult results** to compute involve several columns:
  - Inclusion dependencies
  - Functional dependencies, etc
Challenges

• Managing the input
  – Decide which profiling tasks to execute on which parts of the data

• Performing the computation
  – Computational complexity depends on the number of rows, and the number of columns; sorting is a typical operation

• Managing the output
  – Meaningfully interpret the profiling results; usually performed by database and domain experts

Existing technology

• SQL queries and spreadsheet browsing
• Dedicated tools or components
  – E.g., IBM Information Analyzer, Microsoft SQL Server Integration Services, Informatica Data Explorer
• Innovative ways to handle the challenges
  – E.g., using indexes, parallel processing
• Methods to deliver approximate results
  – E.g., by profiling samples
• Narrowing the discovery process to certain columns or tables
  – E.g., verifying inclusion dependencies on user-suggested pairs of columns
Typical data profiling procedure

1. User specifies data to be profiled and chooses type of metadata to be generated
2. Tool computes the metadata in batch mode (using SQL queries or specialized algorithms)
   - Can last minutes or hours
3. Tool displays results in a vast collection of tabs, tables, charts, and other visualizations
   - Discovered results can be translated into rules or constraints to be enforced in a subsequent data cleaning step

Use Cases for Data Profiling

- **Data cleaning**
  - Data profiling results can be used to measure/monitor the quality of a dataset
- **Data exploration**
  - To have an insight of new datasets: simple ad-hoc SQL queries return simple statistics (e.g., nb distinct values)
  - Automated data profiling is required
- **Database management**
  - Basic statistics gathered by a DBMS: number of values, number of non-null values, etc
  - Optimizer uses these statistics to estimate selectivity of operators and perform query optimization
- **Database reverse engineering**
  - To identify relations and attributes, domain semantics, foreign keys and cardinalities
  - Result: ER model or logical schema to assist experts in maintaining, integrating and querying the DB
- **Data integration**
  - For finding semantically correct correspondences between elements of two schemata (schema matching)
  - Cross-DB inclusion dependencies suggest which tables may be combined with a join operation
- **Big Data analytics**
  - Profiling as preparation and for initial insights
  - Important to determine which data to mine, how to import it into various tools and how to interpret the results
- **Data profiling as preparation for any other data management task**
Types of storage of input data

• **Relational database**
  – So data profiling methods make use of SQL queries and indexes

• **CSV file**
  – Data profiling methods need to create its own data structures in memory or disk

• **Mixed approach**
  – Data originally in the database are read once and processed further outside the database

• The type of storage for input data has an impact on the performance of the data profiling algorithms and tools

---

Data profiling vs. data mining

• **Data profiling** gathers technical metadata to support data management

• **Data mining and data analytics** discovers non-obvious results to support business management with new insights

• **Data profiling results**: information about columns and column sets

• **Data mining results**: information about rows or row sets
  – clustering, summarization, association rules, …
  – Recommendation or classification are not related to data profiling
Outline

• Data profiling tasks
• Data profiling tools
• Visualization

Outline

➢ **Data profiling tasks**
• Data profiling tools
• Visualization
Classification of Traditional Data Profiling Tasks

Data profiling tasks and their primary uses

<table>
<thead>
<tr>
<th>Dependencies</th>
<th>Single-column</th>
<th>Multi-column</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cardinalities</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Patterns &amp; data types</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Value distributions</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Domain classification</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Correlations</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Association rules</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Clustering</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Outliers</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Summaries &amp; sketches</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Unique column combinations</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Inclusion dependencies</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Conditional inclusion dep.</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Functional dependencies</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Conditional functional dep.</td>
<td>✔</td>
<td>✔</td>
</tr>
</tbody>
</table>
Single column profiling

• Analysis of individual columns in a given table
  – Most basic form of data profiling
  – Assumption: All values are of same type
  – Assumption: All values have some common properties to be discovered
  – Discover data types
  – Often part of the basic statistics gathered by DBMS
  – Complexity: Number of values/rows

Cardinalities

• Number of values (nb of rows)
• Length of values in terms of characters
• Number of distinct values
• Number of NULLs
• MIN and MAX value

• Useful for
  – Query optimization
  – Categorization of attribute
  – Relevance of attribute
Data completeness

• Finding disguised missing values (e.g., when using web forms including fields whose values must be chosen from pull-down lists)
  – 9999-999 for the zip code
  – “Alabama” for the USA state

• Methods: determine the distribution of values and find out that disguised missing values are occurring much more often

Data types and value patterns

• Discovering the basic type of a column:
  – String vs. number
  – String vs. number vs. date
  – SQL data types (CHAR, INT, DECIMAL,...)

• Extracting frequent patterns observed in the data of a column:
  – Regular expressions \d{3}-\d{3}-\d{4}-\d+

• Finding the meaning of a column (semantic domain)
  – Address, phone, email, first name
**Value distributions**

- Probability distribution for numeric values
- Detect whether data follows some **well-known distribution**
  - Determine that distribution function for data values
- If no specific/useful function detectable: histograms

**Normal distributions**

**Laplace distributions**

**Histograms**

- Determine (and display) value frequencies for value intervals or for individual values
  - Estimation of probability distribution for continuous variables

**Grade distribution**

**Population (UK)**

- **Useful for**
  - Query optimization
  - Outlier detection
  - Visualize distribution
Multi-column data profiling

- **Covers multiple columns simultaneously**
  - Identifies inter-value dependencies and column similarities
  - Identifies correlations between values through frequent patterns or association rules
  - **Complexity:** Number of columns and number of values

Correlations and association rules

- **Correlation analysis** reveals related numeric columns (e.g., salary and age in relation Employees)
- Naïve method: compute pairwise correlations among all pairs of columns
- **Association rules**: denote relationships or patterns between attribute values among columns
  - Ex: Employees{emp-nb, dept, position, allowance}
  - {dept=finance, position=manager} -> {allowance= $1000}
- **Algorithms**: Apriori, FP-growth
Clustering

• To segment the records into homogeneous groups using a clustering algorithm
• Records that do not fit any cluster flagged as outliers
  – May indicate data quality problems
• Algorithms: K-means, for example

Dependencies

• Metadata that describe relationships among columns
  – Discovery of primary keys with the help of unique column combinations
  – Discovery of foreign keys with the help of inclusion dependencies
  – Functional dependencies
  – Complexity: Number of columns and number of values
  – Several algorithms for detecting dependencies
Uniqueness and keys

- Set of columns $R.X$ that contain only unique value combinations
- (Primary) key candidate
  - No null values
  - Uniqueness and non-null in one instance do not imply key: Only human can specify keys
- Algorithms: Gordian, DUCC, SWAN
- Useful for
  - Schema design, data integration, indexing, optimization
  - Inverse: non-uniques are duplicates

Inclusion dependencies (IND) and foreign keys (FKs)

$R.A \subseteq S.B$
All values in $R.A$ are also present in $S.B$

$R.A_1, \ldots, R.A_i \subseteq S.B_1, \ldots, S.B_i$:
All value combinations in $R.A_1, \ldots, R.A_i$ are also present in $S.B_1, \ldots, S.B_i$

- Prerequisite for foreign key:
  - Used across relations
  - Use across databases
  - But again: Discovery on a given instance, only user can specify for schema
- Algorithms for IND detection: Spider, BINDER
- INDs useful for
  - suggesting how to join two relations
Functional dependencies

\[ X \rightarrow A \]
whenever two records have the same X values, they also have the same A values, where X is a set of attributes

- E.g., street, number \( \rightarrow \) zip-code

- Algorithms for detecting FDs: TANE, FUN, FD-Mine, etc

- Useful for
  - Schema design
    - Normalization
    - Keys
  - Data cleansing

Partial dependencies

- Real datasets contain exceptions to the rule so dependencies can be relaxed
- Aka "approximate dependencies": hold for a subset of records
- INDs and FDs that do not perfectly hold
  - For all but 10 of the tuples
  - Only for 80% of the tuples
  - Only for 1% of the tuples

- Also for patterns, types, uniques, and other constraints

- Useful for
  - Data cleansing
Conditional dependencies

• Given a partial IND or FD: For which part do the hold?
• Example: conditional unique column combination street is unique for all records with city = ‘Lisbon’
• Expressed as a condition over the attributes of the relation
• Problems:
  – Infinite possibilities of conditions
  – Interestingness:
    • Many distinct values: less interesting
    • Few distinct values: surprising condition – high coverage

• Useful for
  – Integration: cross-source cINDs

Outline

• Data profiling tasks
  ➢ Data profiling tools
Research data profiling tools

- **Bellman**: Column statistics, column similarity, candidate key discovery
- **Potter’s Wheel**: Column statistics (including value patterns)
- **Data Auditor**: CFD and CIND discovery
- **RuleMiner**: Denial constraint discovery
- **MADlib**: Simple column statistics
- **Profiler**: visual data profiler tool
- **Metanome**: in a few slides

Commercial data profiling tools

- IBM InfoSphere Information Analyzer
- Oracle Enterprise Data Quality
- Talend Data Quality
- Ataccama DQ Analyzer
- SAP BusinessObjects Data Insight and SAP BusinessObjects Information Steward
- Informatica Data Explorer
- Microsoft SQL Server Integration Services Data Profiling Task and Viewer
- Trillium Software Data Profiling
- CloverETL Data Profiler
- OpenRefine
  - [http://www.openrefine.org](http://www.openrefine.org)
- and many more...

Often packaged with data quality / data cleansing software
Very long feature lists

- Num rows
- Min value length
- Median value length
- Max value length
- Avg value length
- Precision of numeric values
- Scale of numeric values
- Quartiles
- Basic data types
- Num distinct values ("cardinality")
- Percentage null values
- Data class and data type
- Uniqueness and constancy
- Single-column frequency histogram
- Multi-column frequency histogram
- Pattern discovery (Aa9)
- Soundex frequencies
- Benford Law Frequency

- Single column primary key discovery
- Multi-column primary key discovery
- Single column IND discovery
- Inclusion percentage
- Single-column FK discovery
- Multi-column IND discovery
- Multi-column FK discovery
- Value overlap (cross domain analysis)
- Single-column FD discovery
- Multi-column FD discovery
- Text profiling

Screenshots from Talend Data Quality
- Simple Statistics

<table>
<thead>
<tr>
<th>Label</th>
<th>Count</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Null Count</td>
<td>10241.30</td>
<td>100.00%</td>
</tr>
<tr>
<td>Distinct Count</td>
<td>10273.30</td>
<td>N/A</td>
</tr>
<tr>
<td>Unique Count</td>
<td>10273.30</td>
<td>N/A</td>
</tr>
<tr>
<td>Duplicate Count</td>
<td>2.00</td>
<td>N/A</td>
</tr>
<tr>
<td>Blank Count</td>
<td>0.00</td>
<td>N/A</td>
</tr>
</tbody>
</table>

- Frequency Statistics

<table>
<thead>
<tr>
<th>Value</th>
<th>count</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Null Field</td>
<td>10241.30</td>
<td>100.00%</td>
</tr>
<tr>
<td>Each Field</td>
<td>2.00</td>
<td>N/A</td>
</tr>
<tr>
<td>JamesLavry@Oak Bay.org</td>
<td>1.00</td>
<td>N/A</td>
</tr>
<tr>
<td>AnDeCor@Strewn River.org</td>
<td>1.00</td>
<td>N/A</td>
</tr>
<tr>
<td>Nancy@Petite caisse</td>
<td>1.00</td>
<td>N/A</td>
</tr>
<tr>
<td>Richard@Strewn River.org</td>
<td>1.00</td>
<td>N/A</td>
</tr>
<tr>
<td>Jocelyn@Strewn River.org</td>
<td>1.00</td>
<td>N/A</td>
</tr>
<tr>
<td><a href="mailto:AntalBaron@tbrbank.org">AntalBaron@tbrbank.org</a></td>
<td>1.00</td>
<td>N/A</td>
</tr>
<tr>
<td>ElsaTapp@Santa Anti.org</td>
<td>1.00</td>
<td>N/A</td>
</tr>
</tbody>
</table>

- Enhanced data/cloneName

- Pattern Matching

- Simple Statistics

- Soundex Frequency Table

<table>
<thead>
<tr>
<th>Value</th>
<th>count</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Null Field</td>
<td>10241.30</td>
<td>100.00%</td>
</tr>
<tr>
<td>Each Field</td>
<td>2.00</td>
<td>N/A</td>
</tr>
<tr>
<td>JamesLavry@Oak Bay.org</td>
<td>1.00</td>
<td>N/A</td>
</tr>
<tr>
<td>AnDeCor@Strewn River.org</td>
<td>1.00</td>
<td>N/A</td>
</tr>
<tr>
<td>Nancy@Petite caisse</td>
<td>1.00</td>
<td>N/A</td>
</tr>
<tr>
<td>Richard@Strewn River.org</td>
<td>1.00</td>
<td>N/A</td>
</tr>
<tr>
<td>Jocelyn@Strewn River.org</td>
<td>1.00</td>
<td>N/A</td>
</tr>
<tr>
<td><a href="mailto:AntalBaron@tbrbank.org">AntalBaron@tbrbank.org</a></td>
<td>1.00</td>
<td>N/A</td>
</tr>
<tr>
<td>ElsaTapp@Santa Anti.org</td>
<td>1.00</td>
<td>N/A</td>
</tr>
</tbody>
</table>
Screenshots for IBM Information Analyzer

Screenshots for IBM Information Analyzer
Typical Shortcomings of Tools (and research methods)

- **Usability**
  - Complex to configure
  - Results complex to view and interpret
- **Scalability**
  - Main-memory based
  - SQL based DBMS
- **Efficiency**
  - Coffee, Lunch, Overnight
- **Functionality**
  - Restricted to simplest tasks
  - Restricted to individual columns or small column sets
    - “Realistic” key candidates vs. further use-cases
    - SAP R3 schema has many tables with up to 16 columns as key
- **Interpretation of profiling results**
  - That’s the big one

Metanome

- Extensible profiling platform that incorporates several state-of-the-art metadata discovery algorithms
- **Goals:**
  - To provide novel profiling algorithms from research
  - To perform comparative evaluations
  - To support developers in building/testing new algorithms
- **Typical users:**
  - Database administrators and IT professionals
  - Developers and researchers
- **See in:** https://hpi.de/naumann/projects/data-profiling-and-analytics/metanome-data-profiling.html
Design Goals

- **Simplicity**
  - Should be easy to setup and use

- **Extensibility**
  - New algorithms and datasets should be easily addable to the system

- **Standardization**
  - All common tasks, tooling, input parsing, result handling should be provided

- **Flexibility**
  - Make as few restrictions as possible to the algorithms

Metanome architecture
Most important tasks

- **Input parsing**
  - Build an abstraction around input sources; specific formats are irrelevant to profiling algos
  - Handles relational databases/files/tables, JSON/RDF/XML files
- **Output processing**
  - Standardize the output formats depending on the type of metadata the algorithm discovers
  - Most important metadata supported: unique column combinations, INDs, FDs, order dependencies, basic statistics
- **Parameterization handling**
  - Defines the parameterization of algorithms through the configuration variables exposed by the profiling algorithms (set by the user)
- **Temporary data management**
  - Provides dedicated temp-files for storing temporary data written by profiling algorithms

Profiling algorithms

- A profiling algorithm needs to implement a given set of light-weight interfaces
- Work autonomously: they are treated as foreign code modules that manage themselves providing maximum flexibility for their design
- **Algorithms supported:**
  - UCCs: DUCC
  - INDs: MIND, SPIDER, BINDER
  - FDs: TANE, FUN, FD_MINE, etc
  - ODs: ORDER
Snapshot – visualization of results

Figure 3: Visualization of inclusion dependency clusters (left) and the table join graph of one cluster (right).

Snapshot – different visualization techniques

Figure 4: Sunburst visualization of a result set of FDs.
Outline

• Data profiling tasks
• Data profiling tools
  ➢ Visualization

Motivation

• Human in the loop for data profiling and data cleansing.

• Advanced visualization techniques
  – Beyond bar-charts and pie-charts

• Interactive visualization
  – Support users in visualizing data, profiling results
  – Support any action taken upon the results
    • Cleansing, sorting, ...
    • Re-profile and visualize immediately
Profiler: Integrated Statistical Analysis and Visualization for Data Quality

Figure 1: The Profiler User Interface. The UI contains (a) linked summary visualizations, and (d) anomaly browser. (b) Taxonomy of Data Quality Issues. We list classes, we investigate possible causes of missing MPA movie ratinf methods for detecting each issue, example routines used in we select it to highlight matching records. The Release Date Profiler, and visualizations for assessing their output.

Figure 6: Taxonomy of Data Quality Issues. We list classes, we investigate possible causes of missing MPA movie rating methods for detecting each issue, example routines used in we select it to highlight matching records. The Release Date Profiler, and visualizations for assessing their output.
Next Lecture

- Introduction to Data Warehouse