Trio
A System for Integrated Management of Data, Accuracy, and Lineage

Jennifer Widom
Stanford University

Basic Premise

- Traditional Database Management Systems (DBMS's) are too rigid for some applications
  - Every data item is either in the database or it isn't
  - Its value is absolute
  - Its derivation history is irrelevant
- Trio relaxes these constraints by making:
  1. Data
  2. Accuracy
  3. Lineage
  all first-class interrelated concepts

Formula for a Database Research Project

- Consider traditional DBMS's – highly sensitive to their basic assumptions
- Add a twist or two
- Forced to reconsider many aspects of data management and query processing
  - Data model and algebra, query language, query optimization and execution, index structures, storage structures, application and user interfaces, …
  - Many Ph.D. theses
  - Significant prototype effort

Next in the Talk

- Trio Features – overview of the twists
  Trio introduces to a conventional DBMS
- Specific project goals and non-goals
- A quiz, to wake you up
- Several motivating applications

Trio Features: Accuracy

Data values may be inexact
- Ex: A numeric value lying somewhere in the range [3.2, 4.4]
- Ex: A record belonging in the database with probability 97%
- Ex: A relation missing about 15% of its records

Queries over inexact data return inexact answers
- Ex: This result value lies somewhere in the range [1,6]
- Ex: This record belongs in the result with probability 85%
- Ex: This answer is missing about 25% of its records
Trio Features: Lineage

Lineage is an integral part of the data model

– Suppose record $r$ was derived by query $Q$ over data $D$ at time $T$. Trio keeps track.

– Lineage captures:
  • Query-based derivations
  • Program-based derivations
  • Update-based derivations
  • Bulk data loads
  • Data import from outside sources

Trio Features: Querying Accuracy

Accuracy may be queried

Ex: Find all numeric values whose approximation is within 1%

Ex: Consider only records with ≥ 98% chance of belonging in the database

Trio Features: Querying Lineage

Lineage may be queried

Ex: Find all records whose derivation includes data from relation $R$

Ex: Determine whether record $r$ was derived from data imported on 4/1/04

Trio Features: Combining Lineage, Accuracy

Lineage and accuracy may be combined in queries

Ex: Find all records derived solely from high-confidence data

Trio Features: Enhancing Updates

• Lineage can be used to enhance updates
  • Data updates
    – Propagate updates from base to derived data (or vice-versa), similar to materialized views
  • Accuracy updates
    – When data becomes more (or less) exact, compute and propagate effect on accuracy of derived data

Trio is Not...

• A comprehensive temporal DBMS
• A DBMS for semistructured data
• The last word in approximate or uncertain data, or in data lineage
• A federated or distributed system
### Main Contributions – Trio Goals

1. Simple and usable model incorporating both accuracy and lineage
2. Query language that extends SQL to handle data, accuracy, and lineage together
3. Working system that is straightforward and efficient enough to actually get used

### Motivating Applications

No lack of them
- How similar are they really?
- Can we accommodate all of them?
- Which 2-3 “killer apps” should we focus on?

### Scientific Data Management

- Scientific experiments produce vast amounts of data
  - Often structured but inexact
- Additional data imported from outside sources
  - Sources vary in quality and reliability
- Many levels of derivation and aggregation
- Data and accuracy evolve over time
  - A perfect fit for Trio

### Sensor Data Management

- Some sensor environments permit centralized collection
  - With sufficient bandwidth and battery power
- Readings may still be unreliable
  - Missing values, incorrect values
- Many levels of derivation and aggregation
  - May want to trace to original sensors
  - Another perfect fit

### Data Cleaning

- Deduplication: Find and merge items likely to represent the same real-world entity
  - Uncertainty in data, in match, and in merge
- “Merge history” (lineage) important for:
  - Computing and propagating uncertainty
  - Unmerging
  - Yet another perfect fit!
- Related application: “Profile Assembly”
More Applications

• Storing/querying approximate values
• Hypothetical reasoning
• Online query processing
• Privacy preservation
• And others…

Running Ex: Christmas Bird Count

Bird Counters

Remainder of the Talk

• The Trio Data Model – TDM
  – A solid proposal
• The Trio Query Language – TriQL
  – Basic features
  – Underlying algebra
• The Trio System – Trio
  – Very basic architectural choices

The Trio Data Model (TDM)

**WARNING:** This model is preliminary and subject to change

1. Data: relational (+ user-defined types)
2. Accuracy: approximation, confidence, and coverage
3. Lineage

TDM: Accuracy

a) Attribute-level approximation
b) Tuple-level (or relation-level) confidence
c) Relation-level coverage
TDM: Approximation

- Broadly, an approximate value is a set of possible values along with a probability distribution.

- Specifically, each Trio attribute value is either:
  1) Exact value (default)
  2) Set of values, each with \( \text{prob} \in [0,1] \) (default)
  3) Min + Max for a range (uniform distribution)
  4) Mean + Deviation for Gaussian distribution

- Type 2 sets may include “unknown” (\( \perp \))
- Independence of approximate values within a tuple.

TDM: Confidence

Each tuple \( t \) has confidence \( c \in [0,1] \)
- Informally: chance of \( t \) correctly belonging in relation
- Default: \( \text{confidence}=1 \)
- Can also define at relation level.

TDM: Coverage

Each relation \( R \) has coverage \( \ell \in [0,1] \)
- Informally: percentage of correct \( R \) that is present
- Default: \( \text{coverage}=1 \)

True Meaning of Accuracy

- Question: What does inaccurate data really mean?
- Answer: Nothing in particular
  - We provide the mechanisms
  - Application determines interpretation
  Sort of

Accuracy in CBC

Each participant \( P \):
- \( P \)-sightings (time, latitude, longitude, species)

- Approximation
  - \( \text{time, latitude, longitude} \): range or Gaussian
  - \( \text{species} \): set of values with probabilities; may include \( \perp \)

- Confidence: based on \( P \)-s confidence or experience; tuple- or relation-level
- Coverage: fraction of activity captured by \( P \)
Subtleties in TDM Accuracy Model

• Difference between: \( \{a, b, c, d\} \)
  and: \[ a \] \( \text{conf} = 0.25 \)
  \[ b \] \( \text{conf} = 0.25 \)
  \[ c \] \( \text{conf} = 0.25 \)
  \[ d \] \( \text{conf} = 0.25 \)

• Difference between: \( \{c, d\} \)
  and: \( \text{conf} = 0.5 \)

Subtleties: CBC

• Difference between: \( \{\text{sparrow, finch, toucan, macaw}\} \)
  and: \[ \text{sparrow} \] \( \text{conf} = 0.25 \)
  \[ \text{finch} \] \( \text{conf} = 0.25 \)
  \[ \text{toucan} \] \( \text{conf} = 0.25 \)
  \[ \text{macaw} \] \( \text{conf} = 0.25 \)

• Difference between: \( \{\text{sparrow}\} \)
  and: \[ \text{sparrow} \] \( \text{conf} = 0.5 \)

TDM Accuracy – Status

• Studying the varied literature in uncertain, probabilistic, fuzzy, approximate, incomplete, and imprecise databases
• Studying several applications in detail
  – Christmas Bird Count
  – Microarray database (SMD)
  – Gene sequence database (SGD)
  – Sensor and RFID data management
  – Deduplication

TDM Accuracy – Status (cont’d)

• Decisions: what’s in and what’s out
  Out \( \Rightarrow \) user-defined types
• Accuracy model decisions closely tied to algebra of operations (TriQL)

TDM: Lineage

• Lineage (a.k.a. Provenance)
  – How data came into existence
  – How it has evolved over time
• TDM tracks lineage at tuple level

Digression: No-Overwrite Storage

• Tuples never physically updated or deleted
  – Create new tuple
  – “Expire” old one
  – Associate them using lineage
• Advantages
  – Historical lineage
  – Phantom lineage (deleted data only)
  – Versioning
Back to Lineage

- When, how, and from-what was a tuple derived?
- *When*
  - Usually “at time T”
  - Sometimes “now”

TDM: Lineage – How

- Result of a query, or part of a query-defined view
- Inserted by a program, invoked with certain parameters
- Result of a database update
- Part of a bulk data load
- Part of a data import from outside sources

TDM: Lineage – From What

- Differs for different lineage types (details omitted)
- Instance-based versus schema-based lineage
  - Data values vs. schema elements
  - Fine-grained vs. coarse-grained
  - Expensive vs. cheap!
- Forward versus backward lineage
  - What data was used to derive tuple t?
  - What data was tuple t used to derive?

Formalizing Lineage

- Every database has Lineage Relation (logical)
  - Lineage (tupleID, derivation-type, time, how-derived, lineage-data)
  - tupleID is key
- Inexact lineage using TDM accuracy model?
- Default for each relation: no lineage

Lineage in CBC

- Raw observations merged and massaged into main database each January
- Combined with previous years
  - Interesting design question: Capture evolution in data or in lineage?
- Correlations with other data: environmental, geographic, population, …

TriQL: The Trio Query Language

- Queries and updates
- Extend SQL
- Keep it simple
- Keep it closed
  - Queries on TDM data produce TDM data (e.g., no ranked results)
TriQL Steps

1. Semantics of standard SQL over TDM data
2. Extensions to SQL for queries involving explicit operations on accuracy and lineage
3. Update options, especially accuracy updates

Standard SQL Over TDM Data

• Query(data + accuracy + lineage) ⇒ Result(data + accuracy + lineage)
• Lineage computation “straightforward”
  – Based on previous work
• Accuracy computation gets very interesting very fast
  – Define Accuracy Algebra

Accuracy Algebra: Basic Questions

• Minimum, maximum, product, …
  ⇒ Support multiple join operators + UDFs

Accuracy Algebra: Observation (1)

Simple operations can turn approximation into confidence

\[ \sigma_{A=x} (A \mid (x, y)) = A \mid x \quad \text{conf} = 0.5 \]

\[ A \mid (w, x, y) \times A \mid (x, y, z) = A \mid (x, y) \quad \text{conf} = 0.9 \]

Non-uniform set-approximations, interval approximations, aggregation, …

Accuracy Algebra: Observation (2)

Simple operations can produce inexpressible results

\[ A \mid (x, y) \times A \mid x 1 = A \mid x 1 \quad \text{conf} = 0.5 \]

\[ A \mid (x, y) \times B \mid y 2 = A \mid x y \quad \text{conf} = 0.25 \]

Approximate approximations?
Accuracy Algebra: Status

- Studying simplified TDM accuracy model
  - Set approximations + “maybe” tuples
  - Various theoretical results
- Worrying about lack of closure
- TDM as user interface to more powerful (closed, complete) underlying model?

Accuracy Algebra: Status (cont’d)

- Studying applications
  - CBC, SMD, SGD, sensor, RFID, deduplication
  - Frequency of various operations
- What’s in and what’s out?
  - Out ⇒ user-defined functions

The Trio Prototype

- Usual goals
  - Rapid deployment of first version
  - Resilience to research fickleness
  - Reasonably efficient
  - Extensibility
- Need to choose among:
  1. Implement on top of a conventional DBMS
  2. Build from scratch
  3. Use extensible OR-DBMS

Trio on Conventional DBMS

- Rapid deployment
- Resilience to (small) changes
- Messy, inefficient, uninteresting?
- No customized storage structures, indexes, query optimization, ...

Trio from Scratch

- Keeps the grad students busy
- Can experiment at every level of the system, fine-tune performance
- Delayed deployment
- Not resilient to changes?
- Many DBMS functions recoded
  - Buffer management, concurrency control, recovery, ...

Trio using Extensible OO-DBMS

- Built-in Accuracy Predicates
- Built-in Lineage Functions
- Stored Procedures for Special Query Processing
Conclusion

1. Data
2. Accuracy
3. Lineage

Plenty of applications

- Data Model – Combine and distill previous work
- Query Language – Algebra, TriQL, UDFs
- System – Efficient, usable, soon

http://www-db.stanford.edu/trio
Google: "stanford trio"

Thanks

- Omar Benjelloun, Ashok Chandra, Anish Das Sarma, Alon Halevy, Evan Fanfan Zeng
- Jim Gray, Wei Hong
- David DeWitt, Dave Maier