Some Tools and Techniques for Managing Uncertain Data

Peter J. Haas
Kevin Beyer
Vuk Ercegovac
Bo Shekita

Chris Jermaine*
Ravi Jampani
Luis Perez
Mingxi Wu
Fei Xu

IBM Almaden Research Ctr.
U. Florida Gainesville
*Rice University

NEDS, September 2009
Outline

• Motivation via examples
• MCDB: Monte Carlo Database System
• MC$^3$: MCDB + map-reduce
• Related projects
• Future directions
Sources of Data Uncertainty

Data Integration

{John Smith, San Jose} → ETL
{John Smith, San Jose} → ETL
{John Smith, Los Angeles} → ETL

<table>
<thead>
<tr>
<th>Name</th>
<th>City</th>
</tr>
</thead>
<tbody>
<tr>
<td>John Smith</td>
<td>(SJ, 0.66), (LA, 0.33)</td>
</tr>
</tbody>
</table>

Name City

Text Miner Source Problem type
Cust0385 (DBMS, 0.8), (OS, 0.2)

09/09/2007
Re: system crash
This morning, my ORACLE system on LINUX exploded in a spectacular fireball ...

A lovely thing to behold is Paris Hilton in the Springtime ...

Annotator

NY Marriott
Paris Hilton

0.2

Celebrities

Britney Spears
Paris Hilton

0.8

Source Problem type
Cust0385 (DBMS, 0.8), (OS, 0.2)

Information extraction

<table>
<thead>
<tr>
<th>Name</th>
<th>Sales</th>
</tr>
</thead>
<tbody>
<tr>
<td>J. Smith</td>
<td>$50K</td>
</tr>
</tbody>
</table>

Join

City Sales
LA $50K

Similarity

0.92

NEDS, September 2009
Data Uncertainty - Continued

**Anonymization**

{JohnSmith, age 42} → Privacy Filter →

<table>
<thead>
<tr>
<th>Name</th>
<th>Age</th>
</tr>
</thead>
<tbody>
<tr>
<td>John Smith</td>
<td>Between 40 and 50</td>
</tr>
</tbody>
</table>

**Measurement Uncertainty**

<table>
<thead>
<tr>
<th>Sensor_ID</th>
<th>Temp (F)</th>
</tr>
</thead>
<tbody>
<tr>
<td>S23</td>
<td>78.32</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Event</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Buffer overflow</td>
<td>10/17/2007:18:20:02</td>
</tr>
</tbody>
</table>

NEDS, September 2009
## Extrapolation Uncertainty: Portfolio Values

### Table: Customer and EuroCallOptions

<table>
<thead>
<tr>
<th>CustID</th>
<th>OptionID</th>
<th>NumShares</th>
</tr>
</thead>
<tbody>
<tr>
<td>John Smith</td>
<td>23</td>
<td>50</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>OptionID</th>
<th>InitVal</th>
<th>r</th>
<th>a</th>
<th>dt</th>
<th>StrikeP</th>
<th>OVal</th>
</tr>
</thead>
<tbody>
<tr>
<td>23</td>
<td>$2.35</td>
<td>0.8</td>
<td>1.01</td>
<td>0.0001</td>
<td>$4.00</td>
<td>?</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**SQL Query**

```sql
SELECT SUM (c.NumShares * o.Val) 
FROM Customer c, EuroCallOptions o 
WHERE c.OptionID = o.OptionID 
AND c.CustID = 'John Smith'
```

### Modified Black-Scholes Model for European Call Option

**Equation:**

\[
dV = rV dt + \left( a \sqrt{V} \right) V dW
\]

**Option Value:**

\[
OV_{\text{Val}} = \max \left( V(t_{\text{final}}) - S, 0 \right)
\]

### Simulation Approximation (Euler Formula)

**Equation:**

\[
V(t + \Delta t) = V(t) + rV(t)\Delta t + \left( a \sqrt{V(t)} \right) V(t)\sqrt{\Delta t}Z_j
\]

*Sample from Normal dist’n*
Pricing Decisions: Individual Demand Curves

Can analyze arbitrary dynamic customer segments when determining effect of price increase

Similar approach for web-click behavior (EBay)

Issues
- Complex model, huge number of dynamic parameters
- Can we integrate into database?
Risk Due to Data Uncertainty

• Ex: Value of assets (for financial reporting, compliance, business-process monitoring)

```
SELECT SUM (s.amount)
FROM SALES s, CUST c
WHERE s.ID = c.ID
AND c.city = 'Los Angeles'
```

• Ex: ERP
  – # OS experts needed for help desk
  – Based on (uncertain) extracted text data from last year
  – Provide principled safety factor
Motivation: Summary

• Customer needs: translate data uncertainty into query uncertainty
  – Risk assessment
  – Decision-making under uncertainty

• Uncertainty models
  – Both warehouse and extrapolation uncertainty
  – Highly heterogeneous and complex
  – Often depend dynamically on huge # of parameters
    • Correlation matrices for multivariate distributions
    • Customer purchase histories
    • Probabilities perpetually changing

• BI Queries
  – Complicated SQL aggregation queries
  – Subqueries, DISTINCT clauses, …

• What-if and sensitivity analysis are crucial
  – Input probabilities are not precise (so check sensitivity)
  – Want to try many different policies

NEDS, September 2009
Prior Work: Extended Relational Models (ERM)

- **ERM:**
  - Basis of “probabilistic databases”
  - Relational model augmented with probabilities, correlations, etc.
  - Relational operations modified accordingly
  - Trio, MayBMS, ORION, MystiQ, K-relations, et al.
  - Emphasis on “top-k” queries

- **Drawbacks**
  - Hard-wired uncertainty model
  - Hard to fit data into tuples
  - Hard to change probabilities
  - What-if analysis is hard
  - Exact analysis (PTIME) only for very simple queries, data, output stats
  - Exact methods have trouble with aggregation queries
Outline

• Motivation
• MCDB: Monte Carlo Database System
• MC$^3$
• Related projects
• Future directions
The MCDB System

Random DB = D

Schema
VG Functions
Parameter Tables

Monte Carlo Generator

Q(D) =
Select SUM(sales)
AS t_sales

Q(d_1)
Q(d_2)
... 
Q(d_n)

Estimator

\[ \hat{E}[t_{sales}] \]
\[ \hat{\text{Var}}[t_{sales}] \]
\[ \hat{q}_{0.01}[t_{sales}] \]
Histogram
Error bounds
Inference

i.i.d. samples from possible-worlds dist’n

i.i.d. samples from query-result dist’n
MCDB Example

Q: SELECT SUM(Amount)
FROM SALES
AS t_sales

<table>
<thead>
<tr>
<th>CID</th>
<th>Amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>102</td>
<td>$120.00</td>
</tr>
<tr>
<td>226</td>
<td>$60.00</td>
</tr>
</tbody>
</table>

\[
\hat{E}[t_{sales}] = $186.67 \quad \text{STD}[t_{sales}] = $20.82
\]

NEDS, September 2009
Advantages of MCDB

• Flexible and extensible uncertainty model
  – Can capture extended relational models (Trio, MayBMS, Mystiq,…)
  – Can capture arbitrarily complex correlations, continuous data
  – Can capture dynamic, highly parameterized distributions
  – Can bring complex stochastic models to data (no extraction needed)

• Encapsulates complexity
  – Once expert has written VG function, naïve user can run queries

• Can handle arbitrary SQL queries

• What-if analysis, sensitivity analysis, data updates are easy
Pseudorandom Number Generators (PRNG)

- Needed by VG function
  - E.g., to generate “random” sales values
- Produces a deterministic sequence of seeds
  - Appears random
  - Cycles around
- Typical PRNG recurrence:
  - \( S_{i+1} = M \times S_i \mod m \)
  - Seed \( S = \) vector of \( k \) unsigned integers
  - \( M \) is a matrix
- Transform seeds to desired random samples
- Cycle usually “split” into disjoint segments
  - Skip factor
- Keeping only initial seed, \( S_0 \), is sufficient to regenerate sequence
VG Functions

<table>
<thead>
<tr>
<th>Value</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>San Jose</td>
<td>0.66</td>
</tr>
<tr>
<td>San Francisco</td>
<td>0.34</td>
</tr>
</tbody>
</table>

- **Used to generate instances of values in random tables**
  - Parameter tables are standard relational tables (can index, etc.)
  - Library of standard functions: DiscreteChoice, Normal, Poisson, …
  - Can define custom functions (similar to UDFs)
## VG Functions and Correlation

The image illustrates the concept of VG Functions and Correlation. It shows a network diagram with data points and tables that represent the relationships between variables.

### Tables

<table>
<thead>
<tr>
<th>ID1</th>
<th>ID2</th>
<th>Cov</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1.23</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>0.17</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2.45</td>
</tr>
</tbody>
</table>

### Table: ID and Mean

<table>
<thead>
<tr>
<th>ID</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3.68</td>
</tr>
<tr>
<td>2</td>
<td>4.75</td>
</tr>
</tbody>
</table>

### Pseudorandom # seed

### MDNormal() Function

- **Correlated columns**
  - V1: 1.21
  - V2: 2.13

- **Correlated rows**
  - ID: 1
    - Val: 1.21
  - ID: 2
    - Val: 2.13
CREATE TABLE RAND_CUST (CID, GENDER, MONEY, LIVES_IN) AS
FOR EACH d in CUST
WITH MONEY AS Gamma(
    (SELECT n.SHAPE FROM MONEY_SHAPE n WHERE n.CID = d.CID),
    (SELECT sc.SCALE FROM MONEY_SCALE sc WHERE sc.REGION = d.REGION),
    (SELECT SHIFT FROM MONEY_SHIFT)
) WITH LIVES_IN AS DiscreteChoice (
    (SELECT c.NAME, c.PROB
     FROM CITIES c
     WHERE c.REGION = d.REGION)
)
SELECT d.CID, d.GENDER, m.VALUE, l.VALUE
FROM MONEY m, LIVES_IN l
Query Processing

• Naïve approach
  – Repeatedly instantiate DB and run query
  – Has horrible performance

• MCDB approach
  – Execute query plan once
  – Process tuple bundles instead of tuples
    • Represents tuple in all simulated possible worlds (MC reps)
  – Keep bundles in compressed form whenever possible
    • Use pseudorandom seeds for compression
    • Apply selections early to compressed bundles
Tuple Bundles (4 MC Repetitions)

(Jane, Smith, 20)
(Jane, Smith, 21)

<Tuple bundle>

(Jane, Smith, (20,21,x,21), (T,T,F,T), Seed)

<Representation>

(Jane, Smith, (T,T,F,T), Seed)

<Compressed representation>

isPresent
Operations on Tuple Bundles

• **Seed:**

\[(Jane, Smith, --, --) \Rightarrow (Jane, Smith, --, --, Seed)\]

• **Split:**

\[(Jane, Smith, (20,21,20,21), (T,T,T,T), Seed) \Rightarrow (Jane, Smith, 20, (T,F,T,F), Seed), (Jane, Smith, 21, (F,T,F,T), Seed)\]

• **Inference:**

\[(Jane, Smith, (20,21,20,21), (T,T,T,T), Seed) \Rightarrow (Jane, Smith, 20, 0.5), (Jane, Smith, 21, 0.5)\]

Also: **Aggregate**
Standard Operations

• **Select** (FNAME = ‘Jane’ AND AGE = 20)

(Jane, Smith, (20,21,20,21), (F,T,T,T), Seed)
(John, Jones, (32,31,20,30), (T,T,F,T), Seed)
(Jane, Jones, (21,23,22,22), (T,T,T,T), Seed) ⇒
  (Jane, Smith, (20,21,20,21), (F,F,T,F), Seed)

• **Join** (equijoin on Department #)

(Smith, (D1,D2,D2,D1), (F,T,T,T), Seed1)
  (Jones, (D1,D2,D2,D2), (T,T,F,T), Seed2) ⇒
    (Smith, D2, Jones, D2, (F,T,F,F), Seed1, Seed2)

Uses SPLIT + sort-merge
Estimation and Inference

MCDB inference operator

WITH Stats(Mu, Var) AS (  
SELECT SUM(Val1*Frac),  
    SUM(Val*Val1*Frac)  
    - SUM(Val1*Frac)*SUM(Val1*Frac)  
FROM OutputTable)  
SELECT Mu AS Mean, SQRT(Var) AS Stdev,  
    1.96*SQRT(Var)/SQRT(1000) AS CIHW  
FROM Stats

WITH CumDistFn(TotSales, Cum, PrevCum) AS (  
SELECT TotSales,  
    SUM(Frac) OVER (ORDER BY TotSales  
    ROWS BETWEEN UNBOUNDED PRECEDING  
    AND CURRENT ROW),  
    SUM(Frac) OVER (ORDER BY TotSales  
    ROWS BETWEEN UNBOUNDED PRECEDING  
    AND 1 PRECEDING)  
FROM OutputTable)  
SELECT Val FROM CumDistFn  
WHERE Cum >= 0.5 AND PrevCum < 0.5

Distinct tuple values

OutputTable

Frac. replications where value appears (vs bit vector)

WITH Stats(Mu, Var) AS (  
SELECT SUM(Val1*Frac),  
    SUM(Val*Val1*Frac)  
    - SUM(Val1*Frac)*SUM(Val1*Frac)  
FROM OutputTable)  
SELECT Mu AS Mean, SQRT(Var) AS Stdev,  
    1.96*SQRT(Var)/SQRT(1000) AS CIHW  
FROM Stats

WITH CumDistFn(TotSales, Cum, PrevCum) AS (  
SELECT TotSales,  
    SUM(Frac) OVER (ORDER BY TotSales  
    ROWS BETWEEN UNBOUNDED PRECEDING  
    AND CURRENT ROW),  
    SUM(Frac) OVER (ORDER BY TotSales  
    ROWS BETWEEN UNBOUNDED PRECEDING  
    AND 1 PRECEDING)  
FROM OutputTable)  
SELECT Val FROM CumDistFn  
WHERE Cum >= 0.5 AND PrevCum < 0.5

Distinct tuple values

OutputTable

Frac. replications where value appears (vs bit vector)
Experimental Queries

• Q1: Next year’s revenue gain from Japanese products
  – Assuming current trends hold
  – Each order duplicated Poisson # of times
  – Poisson mean = (this year)/(last year) for customer

• Q2: Order Delays
  – From placement to delivery
  – Fitted Gamma distribution for each delay type (for each part)

• Q3: What if we had used cheapest supplier?
  – TPC-H only has current prices
  – Prior prices generated by backwards random walk with drift

• Q4: Change in profits with 5% price increase
  – Bayesian model of customer demand
  – Based on all customers orders at current price
Results 1 (1000 Reps*)

*Q3 histogram based on 350 reps
Results 2: Execution Times (Min)

<table>
<thead>
<tr>
<th>Query</th>
<th>1 rep</th>
<th>10 reps</th>
<th>100 reps</th>
<th>1000 reps</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1</td>
<td>25</td>
<td>25</td>
<td>25</td>
<td>28</td>
</tr>
<tr>
<td>Q2</td>
<td>36</td>
<td>35</td>
<td>36</td>
<td>36</td>
</tr>
<tr>
<td>Q3</td>
<td>37</td>
<td>42</td>
<td>87</td>
<td>222*</td>
</tr>
<tr>
<td>Q4</td>
<td>42</td>
<td>45</td>
<td>60</td>
<td>214</td>
</tr>
</tbody>
</table>

*Based on 350 reps

- Much faster than naïve method in all cases

vs 25000, 36000
Outline

• Motivation
• MCDB
• MC$^3$: MCDB + map-reduce
• Related projects
• Future directions
Motivation

• Exploit massive parallelism of MCDB computations
  – Extend domain of applicability
• Faster path to market?
  – Forward-looking architecture
• Handle semi-structured, nested data
  – E.g., web-click example: Petabytes of log file data
• Local expertise/interest in map-reduce
  – Learning experience for interesting analytical problem
  – MCDB computations often CPU-intensive
  – Ease of prototyping
Technical Issues

• How to represent bundles?
• How to specify map-reduce jobs?
• How to parallelize?
• How to seed tuple bundles?
A Cluster-Computing Infrastructure

- Jaql
  - High-level query language for semi-structured JSON data
- Map-Reduce
  - Parallel batch processing
- HDFS
  - Distributed File System

Initial prototype built in a few weeks

www.jaql.org
//code.google.com/p/jaql

NEDS, September 2009
Map-Reduce Overview

Ex: parallel word counting

Partitioned Input File:

\[ [(K, V)] \]

\( M_1 \)

\( [(K, V)] \)

\( (K, V) \)

\( [((K_m, V_m))] \)

\( (K_m, V_m) \)

\( (K_m, [V_m]) \)

\( [((K_r, V_r))] \)

\( (K_r, V_r) \)

Partitioned Output File:

\[ [V_r] \]

\( [V_r] \)

\( NULL, \text{"This is a line of text"} \)

\[ \text{(["This",1],…,("text",1)]} \]

\( [("This", [1,1,…,1]) \)

\([("This",528),("is",2000),…]\)

\( R_1 \)

\( R_2 \)

\( M_1 \)

\( M_2 \)

\( M_3 \)

\( M_4 \)

• Programmer focus:
  – Map: \((K, V) \rightarrow [((K_m, V_m))]\)
  – Reduce:
    \((K_m, [V_m]) \rightarrow [((K_r, V_r))]\)

• System provides:
  – Parallelism
  – Sorting
  – Synchronization
  – Fault tolerance
  – Resource allocation

On commodity hardware

NEDS, September 2009
MCDB Example

Q: SELECT SUM(Amount) FROM SALES AS t_sales

\[
\begin{array}{|c|c|c|}
\hline
\text{CID} & \text{Shape} & \text{Scale} \\
\hline
102 & 1.2 & 7.0 \\
226 & 0.7 & 2.1 \\
\hline
\end{array}
\]

\[\Gamma(\text{shape}, \text{scale})\]

\[
\begin{array}{|c|c|c|}
\hline
\text{CID} & \text{Amount} \\
\hline
102 & $120.00 \\
226 & $60.00 \\
\hline
102 & $80.00 \\
226 & $90.00 \\
\hline
102 & $80.00 \\
226 & $130.00 \\
\hline
\end{array}
\]

Q\(d_1\) = $180

Q\(d_2\) = $170

Q\(d_3\) = $210

\[\hat{E}[t_{\text{sales}}] = $186.67 \quad \text{STD}[t_{\text{sales}}] = $20.82\]

NEDS, September 2009
JSON and MC³

\[
\begin{align*}
&\text{[\{ cid: 102, region: NewEngland\}, \ldots \}] \\
&\text{Join + Project} \\
&\text{[\{ cid: 102, shape: 1.2, scale: 7.0\}, \ldots \}] \\
&\text{Seed} \\
&\text{[\{ cid: 102, shape: 1.2, scale: 7.0, seed: 306576301\}, \ldots \}] \\
&\text{Instantiate} \\
&\text{[\{ cid: 102, shape: 1.2, scale: 7.0, amount: \{ seed: 306576301, samples: \{120.30, 65.00, \ldots \} \}, isPresent: [T, T, \ldots ]\}, \ldots \}] 
\end{align*}
\]
JAQL and MC³: Example

1 $cust = READ(hdfs('cust_attr'));
   $shape = READ(hdfs('amt_shape'));
   $scale = READ(hdfs('amt_scale'));
2 JOIN $shape, $cust, $scale
   WHERE $shape.cid == $cust.cid
       AND $cust.region == $scale.region
   INTO {$shape, $scale}
   //Seed
3 → TRANSFORM { $.*, seed: GetSeed() }
   //Instantiate: generate array of 1000 samples
4 → TRANSFORM GenAmounts($.seed, $.shape, $.scale, 1000)
   // Sum all sales tuple bundles
6 → GROUP INTO ArraySum($)
   // Compute the distribution
7 → TRANSFORM Distribution($)
8 → WRITE(hdfs('result'));
Example of a Query Plan

1. Final ArraySum
2. Distribution
3. Write ‘result’

Map
1. GetSeed
2. GenAmounts
3. Partial ArraySum

Reduce

Join (region)

Job 1
Read
‘CUST_ATTR’

Join (cid)

Job 2
Read
‘AMT_SCALE’

Job 3

NEDS, September 2009
Parallelism Schemes

- **Inter-tuple parallelism**
  - Partition tuple bundles among nodes
  - Natural fit with Map-Reduce
  - Good when many bundles or cheap VG functions

- **Intra-tuple parallelism**
  - Split up tuple bundles
    - Break Monte Carlo replications into chunks
  - Apply inter-tuple parallelism methods to chunks
  - Good when few tuples with
    - Expensive VG functions and/or
    - Many MC replications
Distributed Seeding

- Must avoid overlapping seed sequences
- Maximize parallelization (tuples on different processors)
- Minimize seed size stored in each tuple
Skip-Ahead Method

- Well512a generator: period = $2^{512}$
- Assume inter-tuple parallelism (for simplicity)
- Assume that we know (or have good upper bound for)
  - # of bundles seeded per node (= $b$)
  - # of seeds per VG function call (= $c$)
  - # MC reps (= $n$)

Tuple $j$ at node $i$: Make $m = b \times i + j$ skips of length $c \times n$ to get to starting point

Seeding

{cid: 102, shape: 1.2, scale: 7.0}

Instantiation

{cid: 102, shape: 1.2, scale: 7.0, seed: [i, j] }
Multi-PRNG Method

- When # of seeds per VG function call is unknown
- When skip-ahead for huge PRNG is hard to implement
- Collisions possible, but probability < 10^{-17}

Seeding at node $i$

\[ S_0 \]

Instantiation of tuple $j$

\[ \text{bundle } j \]

$G_1$ (small)

$G_2$ (medium)

$G_3$ (medium)

$G_4$ (huge)

6 ints x [# bundles at nodes 0 to (i-1)]

4 ints

6 ints

16 ints

Shared by All nodes
Scale-up Results: Inter-Tuple Parallelism

- Implemented two nontrivial queries from MCDB paper
  - Jaql: Map-Reduce plan = original MCDB plan
  - Good scalability with inter-tuple parallelism
Speed-up Results: Intra-Tuple Parallelism

- Implemented two call-option queries (Euro and Asian)
  - Euro option: expensive VG function, good speed-up
  - Asian option: cheap VG function, speed-up curve flattens
    - Sequential merging of chunks starts to dominate
  - Moral: choose appropriate parallelization scheme
Outline

• Motivation
• MCDB
• MC$^3$
• Related projects
• Future directions
Related Projects

- **RAQA: Resolution-aware query answering for Business Intelligence** [Sismanis et al., ICDE09]
  - Uncertainty due to entity resolution
  - OLAP querying (roll-up, drill-down)
  - Bounds on query answers
  - Implemented via SQL queries
  - Conservative approach

- **ProbIE: Probabilistic info extraction** [Michelakis et al., SIGMOD09]
  - For rule-based IE system (e.g., SystemT)
  - Provides confidence #’s for base/derived annotations
  - Based on “rule history”, lower-level results
  - MaxEnt-based learning approach

```
<table>
<thead>
<tr>
<th>City</th>
<th>State</th>
<th>Strict range</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>San Francisco</td>
<td>CA</td>
<td>[$30,$230]</td>
<td>guaranteed</td>
</tr>
<tr>
<td>San Jose</td>
<td>CA</td>
<td>[$70,$200]</td>
<td>non-guaranteed</td>
</tr>
</tbody>
</table>

Sum(Sales) group by City,State

<table>
<thead>
<tr>
<th>State</th>
<th>Strict range</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>CA</td>
<td>[$230,$230]</td>
<td>guaranteed</td>
</tr>
</tbody>
</table>

Sum(Sales) group by State

```

---

**probIE**

Learning phase: Annotator rules
Labeled training data
Rule features

Text

Statistical model

Annotation probability

Annotation + Rule history

Deployment phase
Outline

- Motivation
- MCDB
- MC³
- Related projects
- Future directions
An End-to-End ERP Scenario

Automobile problem reports (text)

My S-Class slipped out of gear ...

Requirements for mechanics and parts (safety margin)

ProblE

My S-Class slipped out of gear ...

Probabilistic BI querying

SELECT COUNT(REPORTS)
WHERE P_TYPE = 'transmission'

Tire Problem (0.2)
Transmission problem (0.9)
Future Directions

• Performance
  – Query optimizer
    • E.g., push down inference & instantiation, choose parallelization scheme
    • Improve JAQL rewriter (MC³ aware)?
  – Sequential and/or adaptive simulation? (MC³)
  – Combine with exact methods? Sampling?
  – Other architectures?

• Functionality
  – Correlated tables
  – Specification and provision of desired precision
  – General uncertainty model for semi- and unstructured data

• Extreme-quantile estimation (value-at-risk)
  – Black-box methods
Further Details:

- MCDB: SIGMOD 2008
- RAQA: ICDE 2009
- MC³: SIGMOD 2009
- ProblE: SIGMOD 2009

www.almaden.ibm.com/cs/people/peterh
peterh@almaden.ibm.com

Thank You!
Backup Slides
Clinic-Capacity Risk

Medical data for all customers

Stochastic dosage model

Pharmacy data for all customers

Cox hazard-rate disease model

Clinic-resource demand model

<table>
<thead>
<tr>
<th>CustID</th>
<th>Time period</th>
<th>Resource needed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jane Smith</td>
<td>June-Sept</td>
<td>?</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td>...</td>
</tr>
</tbody>
</table>
Individual Click Behavior (EBay)

- Can analyze arbitrary dynamic customer segments when determining effect of changing EBay pages

Click data for all EBay customers

Global Markov model distribution (Dirichelet prior)

Data for one customer

Individual Markov model distribution (posterior)
### Logistics Under Uncertainty

- **Retailer:** ship from warehouses to outlets today or tomorrow?
- **Deterministic tables**

<table>
<thead>
<tr>
<th>Shipment</th>
<th>In_Stock</th>
<th>Current_Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>ITEM_ID</td>
<td>QUANTITY</td>
<td>ITEM_ID</td>
</tr>
<tr>
<td>curtains</td>
<td>50</td>
<td>curtains</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

- **Random tables**

<table>
<thead>
<tr>
<th>Sales_W_Ship</th>
<th>Sales_WO_Ship</th>
</tr>
</thead>
<tbody>
<tr>
<td>CUST_ID</td>
<td>ITEM_ID</td>
</tr>
<tr>
<td>Smith</td>
<td>curtains</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

- **Queries:**

\[
\text{SELECT} \ \text{SUM} (\text{c.price} \times \text{s.quantity})
\text{FROM} \ \text{SALES_W_SHIP} \ s,
\text{CUR_PRICE} \ c
\text{WHERE} \ c.\text{ITEM_ID} = s.\text{ITEM_ID}
\]

\[
\text{SELECT} \ \text{SUM} (\text{c.price} \times \text{s.quantity})
\text{FROM} \ \text{SALES_WO_SHIP} \ s,
\text{CUR_PRICE} \ c
\text{WHERE} \ c.\text{ITEM_ID} = s.\text{ITEM_ID}
\]

- **Issues:**
  - Complicated statistical models for purchase quantity (how to integrate in DB?)
  - Uncertainty (random tables) depend dynamically on huge number of parameters
VG Function Implementation

- C++ class with four public methods
  - Initialize: set up data structures, seed RNG
  - TakeParams: read in “parameter vector”
  - OutputVals: return random value(s) for possible world
    - Return NULL when done
  - Finalize: clean up

```plaintext
If newRep:
newRep = false
uniform = myRanDGen()
probSum = i = 0
while (uniform >= probSum)
    i++
    probSum += L[i].wt / totWeight
return L[i].val
Else
newRep = true
return NULL
```

OutputVals method
For DiscreteChoice()
Schema Syntax: Example 1

- **Goal:** generate random customer table
  - MONEY, LIVES_IN are uncertain attributes
  - MONEY has Gamma dist’n
    - shift, shape, scale parameters
  - Use DiscreteChoice for LIVES_IN value
  - Customers are mutually independent, given region
- **Parameter table schemas**
  - CUST (CID, GENDER, REGION)
  - CITIES (NAME, REGION, PROB)
    - Probabilities sum to 1 in each region
  - MONEY_SHIFT (SHIFT)
  - MONEY_SCALE (REGION, SCALE)
  - MONEY_SHAPE (CID, SHAPE)
Schema Syntax: Example 2

- Suppose MONEY and LIVES_IN are correlated

```
CREATE TABLE RAND_CUST (CID, GENDER, MONEY, LIVES_IN) AS
FOR EACH d in CUST
  WITH MLI AS MyJointDistribution(…)
SELECT d.CID, d.GENDER, MLI.V1, MLI.V2
FROM MLI
```

MLI has 1 row, 2 columns
Schema Syntax: Example 3

- Correlated sensors
  - Sensors in same “sensor group” are correlated (multivariate normal)
- Parameter table schemas
  - S_PARAMS (ID, LAT, LONG, GID)
  - MEANS (ID, MEAN)
  - COVARS (ID1, ID2, COV)

CREATE TABLE SENSORS (ID, LAT, LONG, TEMP) AS
FOR EACH g in (SELECT DISTINCT GID FROM S_PARAMS)
  WITH TEMP AS MDNormal (  
    (SELECT m.ID, m.MEAN FROM MEANS m S_PARAMS ss  
      WHERE m.ID = ss.ID AND ss.GID = g.GID),  
    (SELECT c.ID1, c.ID2, c.COV FROM COVARS c, S_PARAMS ss  
      WHERE c.ID1 = ss.ID AND ss.GID = g.GID)  
  )
SELECT s.ID, s.LAT, s.LONG, t.VALUE
FROM S_PARAMS s, TEMP t
WHERE s.ID = t.ID
Instantiate Operation

```
<table>
<thead>
<tr>
<th>output pipe</th>
<th>Merge_{seed}</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>\pi_{VG\text{atts}} \cup {\text{seed}}</td>
</tr>
<tr>
<td></td>
<td>VG Function</td>
</tr>
<tr>
<td></td>
<td>Merge_{seed}</td>
</tr>
<tr>
<td></td>
<td>Sort_{seed}</td>
</tr>
<tr>
<td></td>
<td>\pi_{\text{Inatts}_1} \cup {\text{seed}}</td>
</tr>
<tr>
<td></td>
<td>\pi_{\text{Inatts}_2} \cup {\text{seed}}</td>
</tr>
<tr>
<td></td>
<td>\pi_{\text{Inatts}_3} \cup {\text{seed}}</td>
</tr>
<tr>
<td></td>
<td>B_1</td>
</tr>
<tr>
<td></td>
<td>B_2</td>
</tr>
<tr>
<td></td>
<td>B_3</td>
</tr>
<tr>
<td></td>
<td>pipe fork</td>
</tr>
<tr>
<td></td>
<td>Q_{in,1}</td>
</tr>
<tr>
<td></td>
<td>Q_{in,2}</td>
</tr>
<tr>
<td></td>
<td>Q_{in,3}</td>
</tr>
<tr>
<td></td>
<td>Q_{out}</td>
</tr>
<tr>
<td></td>
<td>&quot;inner&quot; input pipes</td>
</tr>
<tr>
<td></td>
<td>&quot;outer&quot; input pipe</td>
</tr>
</tbody>
</table>
```

For-each clause

NEDS, September 2009
Q4 Details

- **Effect on profits of 5% price increase**
  - Want more accuracy than usual aggregated demand functions
    - E.g, exploit detailed point-of-sale data
  - For each part
    - Fit “prior” demand-function distribution to all customers (MLE)
    - Determine “posterior” distribution for each cust. (Bayes Thm)
    - Generate random demand for each customer at new price
    - Use rejection algorithm to sample from posterior

\[
\text{Gamma}(a,b) \quad \text{Gamma}(c,d)
\]
Nested-Data Experiments

- TPC-H schema is used
- Two different ways to nest data
  - Nest `lineitem` table under `orders` table
  - Nest `lineitem` table under `partsupp` table
- Modified version of Q4 from MCDB paper
  - Compare MC$^3$ execution time to flat scheme
  - First nesting scheme: running time is slower
  - Second nesting scheme: running time is faster
- Only uncertain “leaf attributes” are supported
Probabilistic Information Extraction in a Rule-Based System

Motivation: System T
Hand-crafted rules for specific domain:

<table>
<thead>
<tr>
<th>Annotator</th>
<th>Candidate-Generation Rules</th>
<th>Rule Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>Person</td>
<td>P1: &lt;Salutation&gt;&lt;CapitalizedWord&gt;&lt;CapitalizedWord&gt;</td>
<td>High</td>
</tr>
<tr>
<td>Base annotator</td>
<td>P2: &lt;First Name Dictionary&gt;&lt;Last Name Dictionary&gt;</td>
<td>High</td>
</tr>
<tr>
<td></td>
<td>P3: &lt;CapitalizedWord&gt;&lt;CapitalizedWord&gt;</td>
<td>Low</td>
</tr>
<tr>
<td>PhoneNumber</td>
<td>Ph1: &lt;PhoneClue&gt;&lt;\d{3}-\d{3}-\d{4}&gt;</td>
<td>High</td>
</tr>
<tr>
<td>Base annotator</td>
<td>Ph2: &lt;\d{3}-\d{3}-\d{4}&gt;</td>
<td>Medium</td>
</tr>
<tr>
<td></td>
<td>Ph3: &lt;\d{5}&gt;</td>
<td>Low</td>
</tr>
<tr>
<td>PersonPhone</td>
<td>PP1: &lt;Person&gt;“can be reached at”&lt;PhoneNumber&gt;</td>
<td>High</td>
</tr>
<tr>
<td>Derived annotator</td>
<td>PP2: &lt;“call”&gt;&lt;Person&gt;&lt;0-2 tokens&gt;&lt;PhoneNumber&gt;</td>
<td>High</td>
</tr>
<tr>
<td></td>
<td>PP3: [&lt;Person&gt;&lt;PhoneNumber&gt;]_{sentence}</td>
<td>Medium</td>
</tr>
</tbody>
</table>

+ Consolidation rule
Consolidate(“Joe Smith”, “Mr. Joe Smith”) = “Mr. Joe Smith”
Annotations

Document $d_1$

...Greg Mann can be reached at 403-663-2817 in my absence ...

<table>
<thead>
<tr>
<th>Annotator</th>
<th>Annotation</th>
<th>Rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>Person</td>
<td>Greg Mann</td>
<td>P2, P3</td>
</tr>
<tr>
<td>PhoneNumber</td>
<td>408-663-2817</td>
<td>Ph2</td>
</tr>
<tr>
<td>PersonPhone</td>
<td>(Greg Mann, 408-663-2817)</td>
<td>PP1</td>
</tr>
</tbody>
</table>

Document $d_2$

... please call Heather Choate at x33278 ...

<table>
<thead>
<tr>
<th>Annotator</th>
<th>Annotation</th>
<th>Rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>Person</td>
<td>Heather Choate</td>
<td>P2, P3</td>
</tr>
<tr>
<td>PhoneNumber</td>
<td>33278</td>
<td>Ph3</td>
</tr>
<tr>
<td>PersonPhone</td>
<td>(Heather Choate, 33278)</td>
<td>PP2</td>
</tr>
</tbody>
</table>

Goal: Attach probabilities to annotations in a principled, scalable manner
Quantifying this uncertainty is critical as

• Extracted facts can then be queried using probabilistic databases
• Confidence numbers can be used by information integration and search applications
• It helps in improving the recall of annotators!!
Our approach

• Propose a probabilistic framework for handling uncertainty in rule-based IE
  – Each annotation is associated with a confidence
    • the probability that the annotation is correct
  – Probability is obtained by augmenting each annotator with a statistical model

• Design considerations
  – Applicable to grammar and declarative rule-based IE systems
  – Scale to annotators with a large number of (correlated) rules
  – Support incremental improvements in accuracy of probability estimates
    • as rules, data, or constraints are added
Rule Histories and Features

• Rule history

P1: <Salutation><CapitalizedWord><CapitalizedWord>
P2: <First Name Dictionary><Last Name Dictionary>
P3: <CapitalizedWord><CapitalizedWord>

Please call Heather Choate at

\[ r = (0, 1, 1) \]

Rule history

• Rule features
  – Qualitative correlations and anti-correlations
  – Ex: “Rules P1 and P2 tend to occur together”
ProbiE Framework (Base Annotator)
Probability Model of Uncertainty

• **Binary random variables associated with text and annotator**
  - $A(s) = 1$ iff span $s$ is actually a Person
  - $K(s) = 1$ iff span $s$ is annotated as a Person by consolidator
  - $R(s) = (R_1(s), R_2(s), \ldots, R_k(s))$ is stochastic rule history on span $s$
    - $R_i(s) = 1$ iff $i$th rule holds at least once on span $s$

• **Annotation probability:**
  $$q(r) = P(A(s) = 1 \mid R(s) = r, K(s) = 1)$$

• **Indirect approach (estimate a prob dist’n rather than many small probs)**
  - Estimate
    $$p_0(r) = P(R(s) = r \mid A(s) = 0, K(s) = 1)$$
    $$p_1(r) = P(R(s) = r \mid A(s) = 1, K(s) = 1)$$

  $$\Rightarrow$$
  $$q(r) = \frac{\pi p_1(r)}{\pi p_1(r) + (1 - \pi)p_0(r)}$$

  $$\pi = P(A(s) = 1 \mid K(s) = 1)$$
  - $\pi$ is easy to estimate empirically
  - Serious **data-sparsity** problem for $p_0$ and $p_1$: $2^k$ possible histories, little training data
  - Solution: Fit a **parametric model**
A Parametric Model

- Parametric exponential model for $p_1$ (model for $p_0$ is similar):
  - Recall: $p_1(r) = P(R(s) = r | A(s) = 1, K(s) = 1)$ with $R(s) = (R_1(s),\ldots,R_k(s))$
  - From features to constraints
    
    $$P(R_3(s) = 1 | A(s) = 1, K(s) = 1) = a_3$$  
    (one marginal constraint per rule)

    $$P(R_2(s) = 1 \text{ and } R_7(s) = 1 | A(s) = 1, K(s) = 1) = a_{2,7}$$  
    (important correlations)

    where constants $a_3$, $a_{2,7}$, etc. computed from training data

  - Approximate $p_1$ by “simplest” (maximum entropy) distribution satisfying constraints
  - Equivalent to maximum-likelihood fit of parameter vector $\theta$ for exponential distribution

    $$p_1(r; \theta) = \frac{1}{Z(\theta)} \exp \left\{ \sum_{c \in C} \theta_c f_c (r) \right\}$$  
    $f_c = \text{Indicator function for constraint } c$

  - Use improved iterative scaling (IIS) to fit $\theta$ from training data

- Model-decomposition methods for IIS scalability to many rules and constraints

- Augment training data to handle constraints with 0 right-hand side

- Methodology extends to derived annotators such as PersonPhone
Some Experimental Results (Pay-As-You-Go)

Person annotator (No inter-rule constraints)

Person annotator (4 inter-rule constraints)