Stochastic Analytics in the Database

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

August, 2010
Outline

• Motivation via examples
• MCDB: Monte Carlo Database System
• MC$^3$: MCDB + map-reduce
• Future directions
Problem Setting

• Large data sets
• Missing or uncertain data
• Stochastic models used to “guess” values
  – Model gives probability distribution on data values
• Want to run BI queries over guessed values
• Want to assess uncertainty in query answers
  – Risk assessment
  – Decisionmaking
Ex. 1: Portfolio Values

<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>

**EuroCallOptions**

<table>
<thead>
<tr>
<th>OptionID</th>
<th>InitVal</th>
<th>StrikeP</th>
<th>OVal</th>
</tr>
</thead>
<tbody>
<tr>
<td>23</td>
<td>$2.35</td>
<td>$4.00</td>
<td>?</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

SELECT SUM (c.NumShares * o.Val)
FROM Customer c, EuroCallOptions o
WHERE  c.OptionID = o.OptionID
AND c.CustType = ‘Institutional’

Modified Black-Scholes model for European call option:

\[ dV = rV dt + (\sigma \sqrt{V}) V dW \]

\[ OVal = \max (V(t_{final}) - S, 0) \]

Simulation approximation (Euler formula):

\[ V(t + \Delta t) = V(t) + rV(t)\Delta t + (\sigma \sqrt{V(t)}) V(t)\sqrt{\Delta t}Z_j \]

Option value one month from now (exercise date)

Sample from Normal dist’n
Ex. 2: Pricing Decisions

- Can analyze arbitrary dynamic customer segments when determining effect of price increase
- Similar approach for web-click behavior (EBay, Websphere portal)
- Issues: Complex model, huge number of dynamic parameters

<table>
<thead>
<tr>
<th>CustID</th>
<th>Unit Price</th>
<th>Order Amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>J. Smith</td>
<td>$10.20</td>
<td>500</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Global demand distribution (prior) → Data for one customer → Individual demand distribution (posterior)
Ex. 3: Data-Warehouse 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>

<table>
<thead>
<tr>
<th>Name</th>
<th>City</th>
</tr>
</thead>
<tbody>
<tr>
<td>John Smith</td>
<td>LA</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>City</th>
<th>Sales</th>
</tr>
</thead>
<tbody>
<tr>
<td>LA</td>
<td>$50K</td>
</tr>
</tbody>
</table>

Information extraction

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

System T Hotel Annotator → NY Marriott

<table>
<thead>
<tr>
<th>Hotels</th>
</tr>
</thead>
<tbody>
<tr>
<td>NY Marriott</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Hotels</th>
</tr>
</thead>
<tbody>
<tr>
<td>Paris Hilton</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Source</th>
<th>Problem type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cust0385</td>
<td>(DBMS, 0.8), (OS, 0.2)</td>
</tr>
</tbody>
</table>

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

Text Miner → Source

<table>
<thead>
<tr>
<th>Source</th>
<th>Problem type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cust0385</td>
<td>(DBMS, 0.8), (OS, 0.2)</td>
</tr>
</tbody>
</table>

System T Hotel Annotator → Paris Hilton

<table>
<thead>
<tr>
<th>Hotels</th>
</tr>
</thead>
<tbody>
<tr>
<td>Paris Hilton</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Hotels</th>
</tr>
</thead>
<tbody>
<tr>
<td>NY Marriott</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Source</th>
<th>Problem type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cust0385</td>
<td>(DBMS, 0.8), (OS, 0.2)</td>
</tr>
</tbody>
</table>

System T Hotel Annotator → Paris Hilton

<table>
<thead>
<tr>
<th>Hotels</th>
</tr>
</thead>
<tbody>
<tr>
<td>Paris Hilton</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Hotels</th>
</tr>
</thead>
<tbody>
<tr>
<td>NY Marriott</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Source</th>
<th>Problem type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cust0385</td>
<td>(DBMS, 0.8), (OS, 0.2)</td>
</tr>
</tbody>
</table>

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

Text Miner → Source

<table>
<thead>
<tr>
<th>Source</th>
<th>Problem type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cust0385</td>
<td>(DBMS, 0.8), (OS, 0.2)</td>
</tr>
</tbody>
</table>
Risk Due to Data Uncertainty

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

  ```sql
  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

August, 2010
Traditional Workflow

- Data extraction slow and bug-prone
- Only coarse-grained modeling
- No encapsulation for user

- Hard to re-link model results to DB
- Hard to deal with data updates
- Sensitivity, what-if analysis are hard

Goal: Integrate model with Database

August, 2010
Prior Work

• **MauveDB (Deshpande & Madden 06)**
  – Continuous deterministic models only

• **Probabilistic databases**
  (Trio, MayBMS, ORION, MystiQ, K-relations, et al.)
  – For data-warehouse uncertainty
  – Hard-wired, limited uncertainty model (deterministic skeleton)
  – Limited queries (top-k)
  – Complexity issues
  – Independence assumptions
  – What-if analysis is hard

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Joe</td>
<td>0.2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Jon</td>
<td>0.6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tim</td>
<td>0.4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Jen</td>
<td>0.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Jan</td>
<td>0.1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tim</td>
<td>0.8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tom</td>
<td>0.3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Joe</td>
<td>0.2</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tim</td>
<td>0.32</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Joe</td>
<td>0.04</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Outline

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

Random DB = D

- Schema
- VG Functions
- Parameter Tables

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

Monte Carlo Generator

d1

Q

d2

Q

... Q(d_n)

Monte Carlo Estimator

- \( \hat{E}[t_{sales}] \)
- \( \hat{V}[t_{sales}] \)
- \( \hat{q}_{01}[t_{sales}] \)
- Histogram
- Error bounds
- Inference

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

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

August, 2010
### MCDB Example

#### SQL Query

```
Q: SELECT SUM(Amount)
    FROM SALES
    AS t_sales
```

#### Data Tables

<table>
<thead>
<tr>
<th>CID</th>
<th>Region</th>
<th>Shape</th>
<th>Scale</th>
</tr>
</thead>
<tbody>
<tr>
<td>102</td>
<td>NewEngland</td>
<td>1.2</td>
<td>7.0</td>
</tr>
<tr>
<td>226</td>
<td>Midwest</td>
<td>0.7</td>
<td>2.1</td>
</tr>
</tbody>
</table>

```
<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>
```

**VG function**

```
Q(d_1) = $180
Q(d_2) = $170
Q(d_3) = $210
```

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

August, 2010
Advantages of MCDB

- **Flexible and extensible uncertainty model**
  - Can capture extended relational models (Trio, MayBMS, Mystiq, …)
  - Can capture huge range of stochastic models

- 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
VG Functions

- 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)
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:
  - $S_{i+1} = M \cdot S_i \mod m$
  - 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 and Correlation

<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>

Correlated columns

<table>
<thead>
<tr>
<th>ID</th>
<th>Val</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.21</td>
</tr>
<tr>
<td>2</td>
<td>2.13</td>
</tr>
</tbody>
</table>

Correlated rows

<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()
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)
    • Permits a variety of performance optimizations
Tuple Bundles (4 MC Repetitions)

---

**Basic ideas:**
(a) Keep bundles in *compressed* form whenever possible
(b) Apply selections early to compressed bundles
(c) Amortize I/O, network costs, etc. over multiple reps
Operations on Tuple Bundles

• **Seed:**

\[
\text{(Jane, Smith, --, --) } \Rightarrow \\
\text{(Jane, Smith, --, --, Seed)}
\]

• **Split:**

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

• **Inference:**

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

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) \[\Rightarrow\]
    - (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) \[\Rightarrow\]
      - (Smith, D2, Jones, D2, (F,T,F,F), Seed1, Seed2)

Uses SPLIT + sort-merge
Estimation and Inference

MCDB inference operator

**OutputTable**

<table>
<thead>
<tr>
<th>TotSales</th>
<th>Frac</th>
</tr>
</thead>
<tbody>
<tr>
<td>20K</td>
<td>0.324</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
</tr>
</tbody>
</table>

**SQL queries**

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

**CumDistFn**

```sql
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
```
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
• 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

August, 2010
Map-Reduce Overview

- **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

Ex: parallel word counting

- \((K, V)\):
  - \([(\text{"This"}, 1), \ldots, (\text{"text"}, 1)]\)

- Partitioned Input File:
  - \([(K, V)]\)

- Partitioned Output File:
  - \([(K_m, V_m)]\)

- \((K_m, [V_m])\):
  - \([(\text{"This"}, [1, 1, \ldots, 1])]\)

- \([V_r]\):
  - \([(\text{"This"}, 528), (\text{"is"}, 2000), \ldots]\)

- \((K_r, V_r)\):
  - \([(\text{"text"}, 1), \ldots, (\text{"text"}, 1)]\)
MCDB Example

Q: SELECT SUM(Amount)
FROM SALES
AS t_sales

<table>
<thead>
<tr>
<th>CID</th>
<th>Region</th>
<th>Scale</th>
</tr>
</thead>
<tbody>
<tr>
<td>102</td>
<td>NewEngland</td>
<td>7.0</td>
</tr>
<tr>
<td>226</td>
<td>Midwest</td>
<td>2.1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CID</th>
<th>Shape</th>
<th>Scale</th>
</tr>
</thead>
<tbody>
<tr>
<td>102</td>
<td>1.2</td>
<td>7.0</td>
</tr>
<tr>
<td>226</td>
<td>0.7</td>
<td>2.1</td>
</tr>
</tbody>
</table>

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

\[
\begin{align*}
\text{Q}(d_1) &= 180 \\
\text{Q}(d_2) &= 170 \\
\text{Q}(d_3) &= 210 \\
\end{align*}
\]

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

August, 2010
JSON and MC³

\[
\text{[\{cid: 102, region: NewEngland\}, \ldots\}]
\]

\[
\downarrow
\]

Join + Project

\[
\text{[\{cid: 102, shape: 1.2, scale: 7.0\}, \ldots\}]
\]

\[
\downarrow
\]

Seed

\[
\text{[\{cid: 102, shape: 1.2, scale: 7.0, seed: 306576301\}, \ldots\}]
\]

\[
\downarrow
\]

Instantiate

\[
\text{[\{cid: 102, shape: 1.2, scale: 7.0, seed: 306576301, amount: \{ seed: 306576301, samples: [$120.30, $65.00, \ldots $] \}, isPresent: [T, T, \ldots ] \}, \ldots\]}\]

August, 2010
JAQL and MC³: Example

```javascript
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’

1. GetSeed
2. GenAmounts
3. Partial ArraySum

Join (region)

Join (cid)

Read ‘CUST_ATTR’

Read ‘AMT_SHAPE’

Read ‘AMT_SCALE’

Reduce

Map

Job 1

Job 2

Job 3
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 $i$:

$\{\text{cid: 102, shape: 1.2, scale: 7.0}\}$

$\{\text{cid: 102, shape: 1.2, scale: 7.0, seed: [i, j]}\}$

Instantiation

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

Actually, only $O(\log m)$ skips needed: pre-compute Skip factors
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\(^3\)
• Future directions
An End-to-End ERP Scenario

Automobile problem reports (text)
My S-Class slipped out of gear …

ProbIE
My S-Class slipped out of gear …

Requirements for mechanics and parts (safety margin)

Probabilistic BI querying
SELECT COUNT(REPORTS)
WHERE P_TYPE = 'transmission'

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

- **Tail Sampling**
  - Extreme-quantile (VAR) estimation and more
  - "Gibbs cloner" approach

- **Performance**
  - Query optimization
    - E.g., push down inference & instantiation, choose parallelization scheme
    - Improve JAQL rewriter (MC³ aware)?
    - Re-use of partial results, multi-query optimizations?
  - Sequential and/or adaptive simulation? (MC³)
  - Combine with exact methods? Sampling?
  - Indexing, etc.

- **Functionality**
  - User-defined precision
  - Semi- and unstructured data
  - Robust, full-featured re-implementation (underway)

- **Possible Applications**
  - Automotive ERP
  - Health records

August, 2010
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

- **Other Monte Carlo/Statistics analysis in Hadoop (XAP)**
  - Operational risk calculations
  - RICARDO: synthesis of R and Hadoop [Das et al., SIGMOD10]
  - Recommender systems

### Table: Results

<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>

### Diagram: ProbIE Architecture

- **Text**
  - Labeled training data
  - Rule features

- **Statistical model**
  - Annotator rules
  - Annotation + Rule history

- **Annotator**
  - Annotation probability

- **Deployment phase**
  - Learning phase

---

August, 2010
Further Details:

- MCDB: SIGMOD 2008
- MC³: SIGMOD 2009
- ProblE: SIGMOD 2009
- MCDB-R: VLDB 2010

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

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:

```sql
SELECT SUM (c.price * s.quantity)
FROM SALES_W_SHIP s,
    CUR_PRICE c
WHERE c.ITEM_ID = s.ITEM_ID
```

```sql
SELECT SUM (c.price * s.quantity)
FROM SALES_WO_SHIP s,
    CUR_PRICE c
WHERE c.ITEM_ID = s.ITEM_ID
```

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

{John Smith, 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>

August, 2010
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

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)

```sql
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

$$\pi_{\text{VG Atts}} \cup \{\text{seed}\}$$

$$\pi_{\text{In Atts}_1} \cup \{\text{seed}\}$$

$$\pi_{\text{In Atts}_2} \cup \{\text{seed}\}$$

$$\pi_{\text{In Atts}_3} \cup \{\text{seed}\}$$

$$\pi_{\text{Out Atts}} \cup \{\text{seed}\}$$

For-each clause

VG function args

"inner" input pipes

"outer" input pipe

output pipe

Merge seed

Merge seed

Sort seed

pipe fork

Q_{in,1} Q_{in,2} Q_{in,3} Q_{out}
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

\[
\begin{align*}
\text{Gamma}(a,b) & \quad \text{Gamma}(c,d)
\end{align*}
\]
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 $i$

- $G_1$ (small) shared by all nodes
- $G_2$ (medium)
- $G_3$ (medium)
- $G_4$ (huge)

Instantiation of tuple $j$

- $6$ ints x [number of bundles at nodes 0 to (i-1)]
- $16$ ints

August, 2010
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 Base annotator</td>
<td>P1: &lt;Salutation&gt;&lt;CapitalizedWord&gt;&lt;CapitalizedWord&gt;</td>
<td>High</td>
</tr>
<tr>
<td></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 Base annotator</td>
<td>Ph1: &lt;PhoneClue&gt;&lt;\d{3}-\d{3}-\d{4}&gt;</td>
<td>High</td>
</tr>
<tr>
<td></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 Derived annotator</td>
<td>PP1: &lt;Person&gt;&lt;“can be reached at”&gt;&lt;PhoneNumber&gt;</td>
<td>High</td>
</tr>
<tr>
<td></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

**Goal:** Attach probabilities to annotations in a principled, scalable manner

<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>

<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>

---

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

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

span

P1  P2  P3
r = ( 0, 1, 1)

• Rule features

  – Qualitative correlations and anti-correlations
  – Ex: “Rules P1 and P2 tend to occur together”
ProbIE Framework
(Base Annotator)

Annotator rules
Labeled training data
Rule features

problE

Statistical model

Learning phase

Text

Annotator

Consolidated span +
Rule history

Annotation probability

Extraction (deployment) phase
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
    $$p_0(r) = P(R(s) = r \mid A(s) = 0, K(s) = 1)$$
    $$p_1(r) = P(R(s) = r \mid A(s) = 1, K(s) = 1)$$
  $$q(r) = \frac{\pi p_1(r)}{\pi p_1(r) + (1 - \pi) p_0(r)}$$
  $$\pi = P(A(s) = 1 \mid K(s) = 1)$$
  - $\pi$ 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: $p_1(r) = P(R(s) = r \mid A(s) = 1, K(s) = 1)$ with $R(s) = (R_1(s), \ldots, R_k(s))$
  – From features to constraints
    \[ P(R_3(s) = 1 \mid A(s) = 1, K(s) = 1) = a_3 \] (one marginal constraint per rule)
    \[ P(R_2(s) = 1 \text{ and } R_7(s) = 1 \mid A(s) = 1, K(s) = 1) = a_{2,7} \] (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)

August, 2010