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}

{John Smith, San Jose}

{John Smith, Los Angeles}

ETL

Name	City
John Smith	(SJ, 0.66), (LA, 0.33)

Name	City
John Smith	LA

Name	Sales
J. Smith	$50K

Similarity Join

City	Sales
LA	$50K

? (0.92)

Information extraction

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

Annotator

0.2

Hotels

NY Marriott
Paris Hilton

0.8

Celebrities

Britney Spears
Paris Hilton

09/09/2007
Re: system crash

This morning, my ORACLE system on LINUX exploded in a spectacular fireball ...

Text Miner

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

Data Uncertainty - Continued

Anonymization

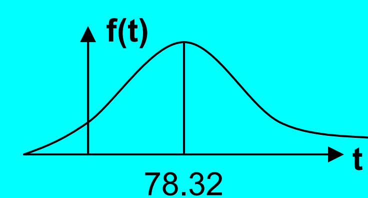
{JohnSmith, age 42} → **Privacy Filter** →

Name	Age
John Smith	Between 40 and 50

Measurement Uncertainty

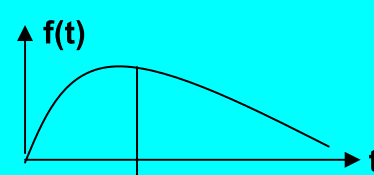
Sensor →

Sensor_ID	Temp (F)
S23	78.32



System Monitor →

Event	Time
Buffer overflow	10/17/2007:18:20:02



Extrapolation Uncertainty: Portfolio Values

Customer

CustID	OptionID	NumShares
John Smith	23	50
…		…

EuroCallOptions

OptionID	InitVal	r	a	dt	StrikeP	OVal
23	$2.35	0.8	1.01	0.0001	$4.00	?
	…					…

Option value one month from now (exercise date)

```
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:

$$dV = rV\,dt + \left(a\sqrt{V}\right)V\,dW \qquad \mathrm{OVal} = \max\left(V(t_{\mathrm{final}}) - S, 0\right)$$

Simulation approximation (Euler formula):

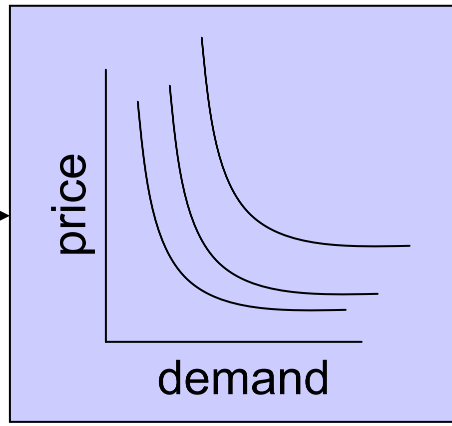
$$V(t+\Delta t) = V(t) + rV(t)\Delta t + \left(a\sqrt{V(t)}\right)V(t)\sqrt{\Delta t}\,Z_j$$

Sample from Normal dist'n

Pricing Decisions: Individual Demand Curves

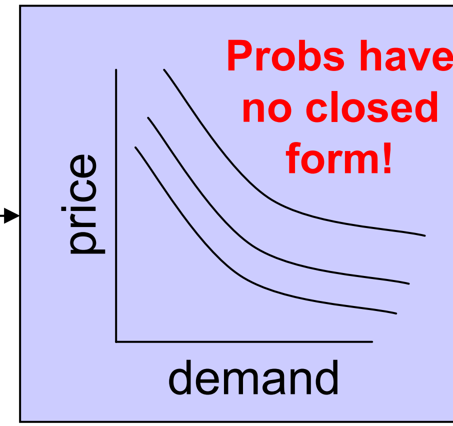
Data for all customers

CustID	Unit Price	Order Amount
J. Smith	$10.20	500
…	…	…



Global demand distribution (prior)

Data for one customer



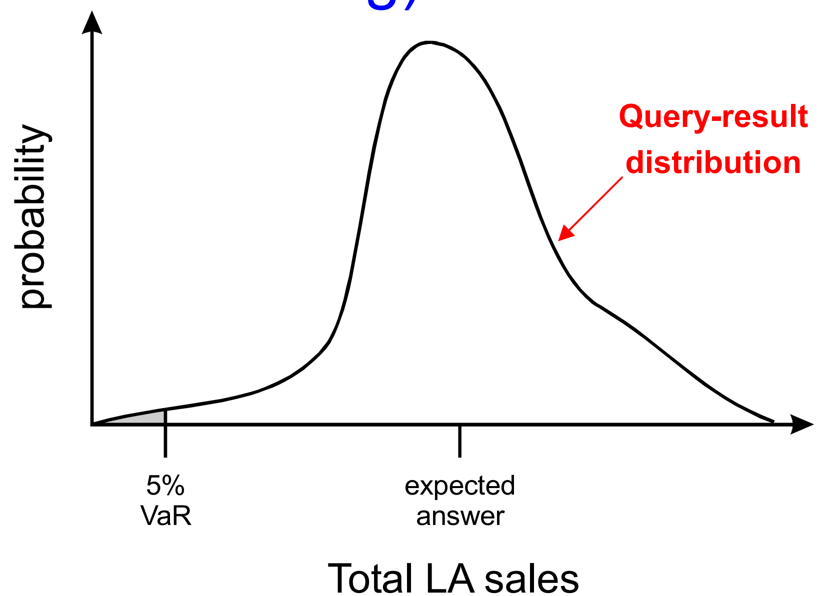
Individual demand distribution (posterior)

- 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

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

Joe	0.2
Jon	0.6
Tim	0.4
Jen	0.1

$\bowtie$

Jan	0.1
Tim	0.8
Tom	0.3
Joe	0.2

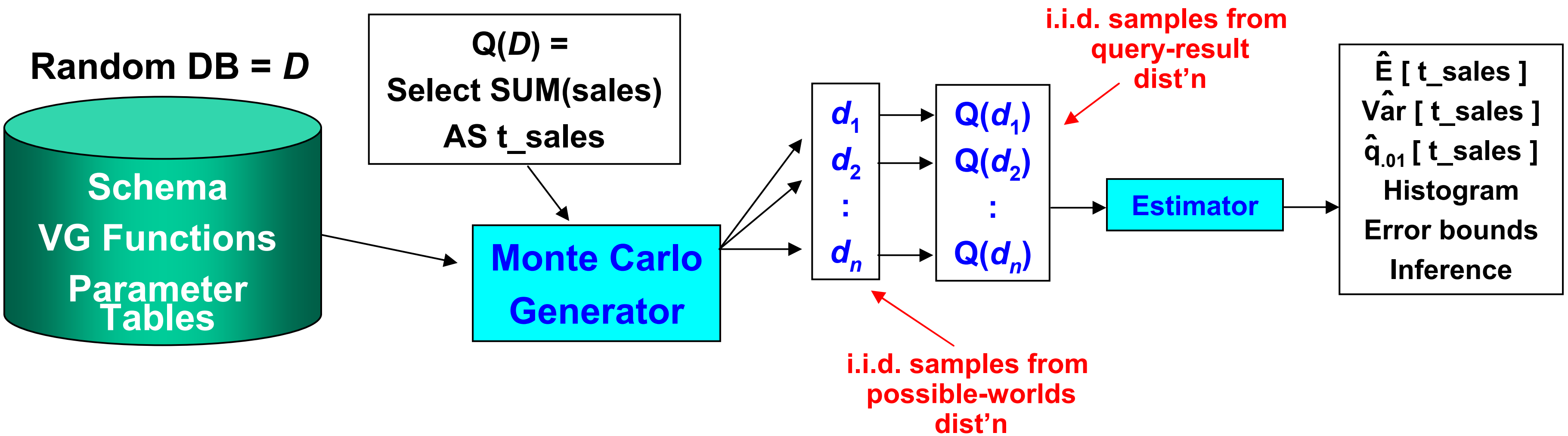
$\rightarrow$

Tim	.32
Joe	.04

Outline

- Motivation
- MCDB: Monte Carlo Database System
- MC^3
- Related projects
- Future directions

The MCDB System



MCDB Example

CID	Region
102	NewEngland
226	Midwest

CUST_ATTR

Region	Scale
NewEngland	7.0
Midwest	2.1

AMT_SCALE

CID	Shape
102	1.2
226	0.7

AMT_SHAPE

CID	Amount

SALES

CID	Shape	Scale
102	1.2	7.0
226	0.7	2.1

Q: SELECT SUM(Amount)
FROM SALES
AS t_sales

VG function

Gamma(shape, scale)

d_1

CID	Amount
102	\$120.00
226	\$60.00

d_2

CID	Amount
102	\$80.00
226	\$90.00

d_3

CID	Amount
102	\$80.00
226	\$130.00

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

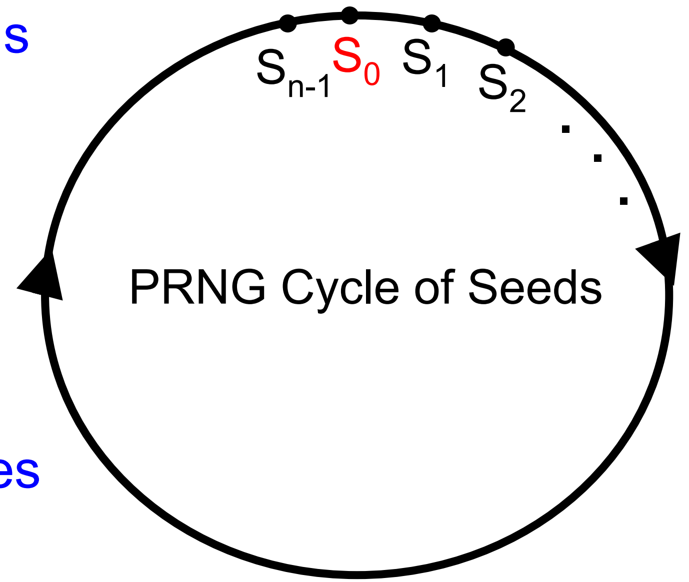
$\hat{E}[t\_sales] = \$186.67$ $\hat{STD}[t\_sales] = \$20.82$

Advantages of MCDB

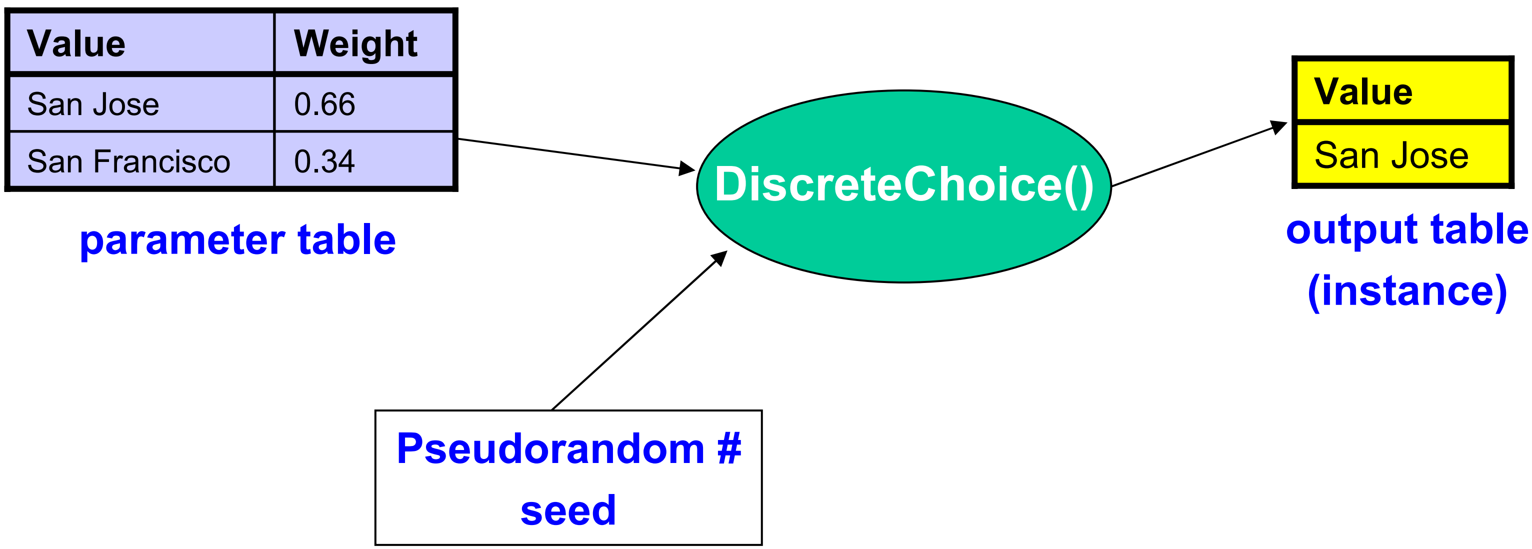
- Flexible and extensible uncertainty model
  - Can capture extended relational models (Trio, MayBMS, Mystiq,…)
  - Can capture arbitrarily complex correlations, continuous data
  - Can capture dynamic, highly parameterized distributions
  - Can bring complex stochastic models to data (no extraction 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:
  - $S_{i+1} = M * S_i \bmod m$
  - 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

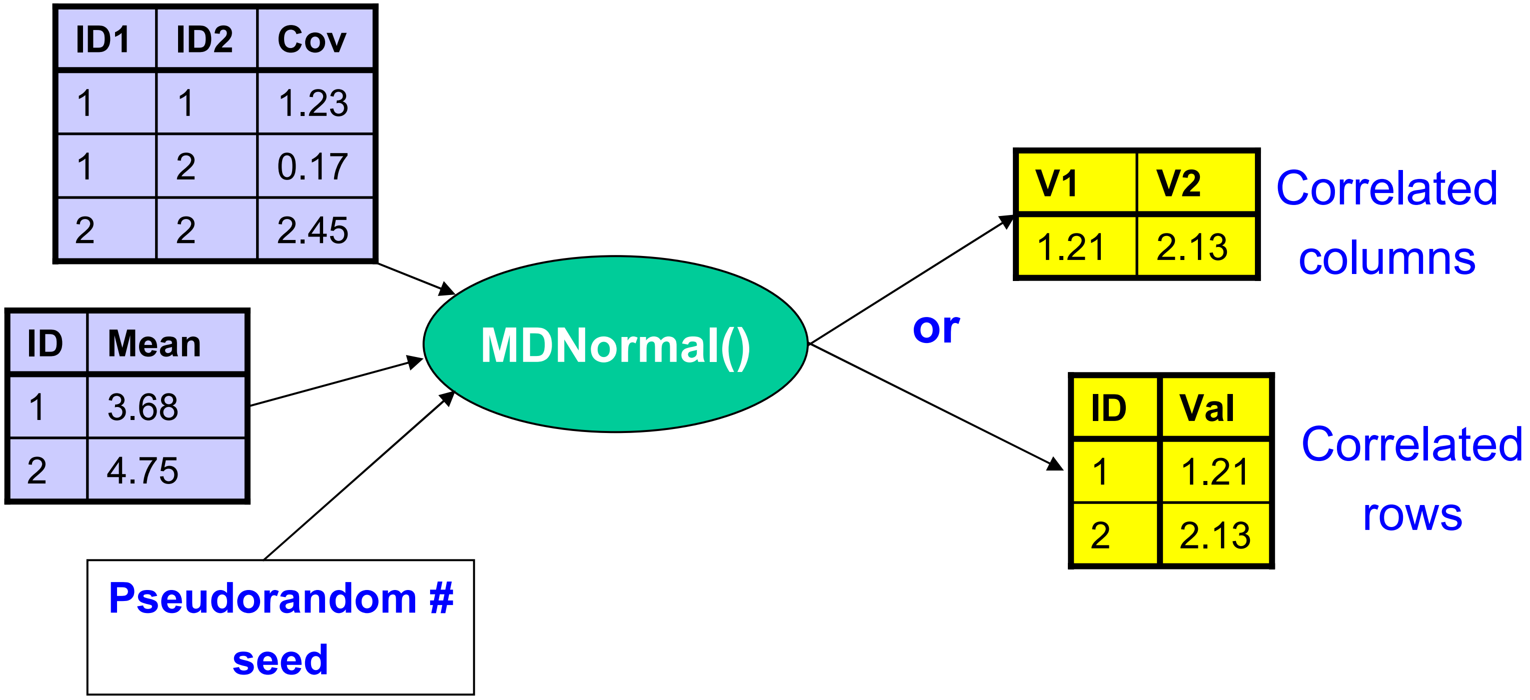


VG Functions



- **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



ID1	ID2	Cov
1	1	1.23
1	2	0.17
2	2	2.45

ID	Mean
1	3.68
2	4.75

V1	V2
1.21	2.13

ID	Val
1	1.21
2	2.13

Schema Syntax: Example

```
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)
--
(Jane, Smith, 21)

Tuple bundle

isPresent

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

Representation

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

Compressed representation

Operations on Tuple Bundles

- **Seed**:

(Jane, Smith, --, --) ⇒
    (Jane, Smith, --, --, Seed)

- **Split**:

(Jane, Smith, (20,21,20,21), (T,T,T,T), Seed) ⇒
    (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) ⇒
    (Jane, Smith, 20, 0.5),
    (Jane, Smith, 21, 0.5)

Also: Aggregate

Standard Operations

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

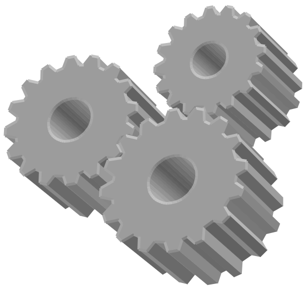
(Jane, Smith, (20,21,20,21), (F,T,T,T), Seed)
(John, Jones, (32,31,20,30), (T,T,F,T), Seed)
(Jane, Jones, (21,23,22,22), (T,T,T,T), Seed) ⇒
(Jane, Smith, (20,21,20,21), (F,F,T,F), Seed)

- **Join** (equijoin on Department #)

Uses SPLIT
+
sort-merge

(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)

Estimation and Inference



MCDB inference operator

Distinct tuple values

TotSales	Frac
20K	0.324
…	…

Frac. replications where value appears (vs bit vector)

OutputTable

```
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
```

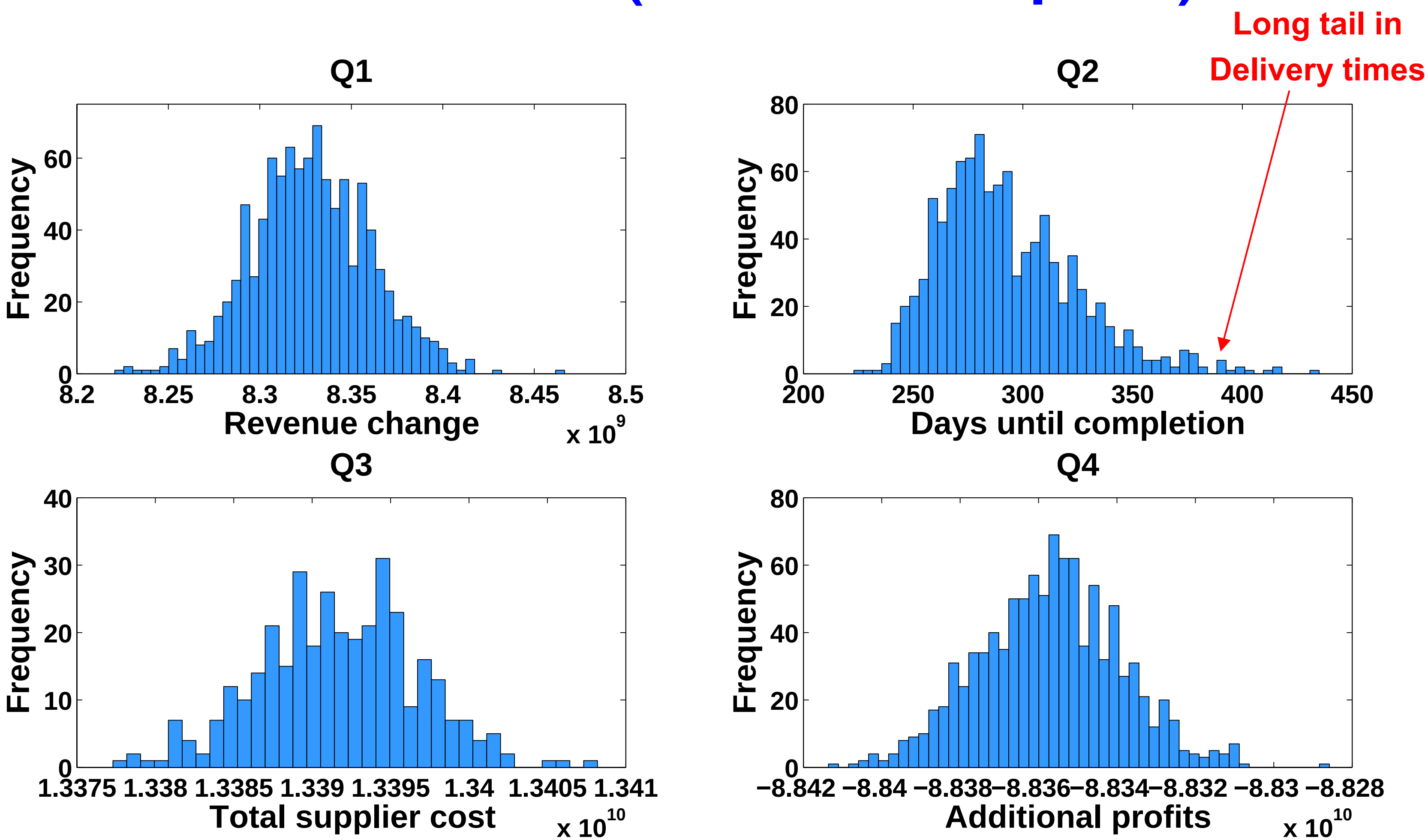
```
WITH CumDistFn(TotSales, Cum, PrevCum) AS (
 SELECT TotSales,
        SUM(Frac) OVER (ORDER BY TotSales
          ROWS BETWEEN UNBOUNDED PRECEDING
          AND CURRENT ROW),
        SUM(Frac) OVER (ORDER BY TotSales
          ROWS BETWEEN UNBOUNDED PRECEDING
          AND 1 PRECEDING)
 FROM OutputTable)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

SQL queries

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)

Query	1 rep	10 reps	100 reps	1000 reps
Q1	25	25	25	28
Q2	36	35	36	36
Q3	37	42	87	222*
Q4	42	45	60	214

vs 25000,
36000

*Based on 350 reps

- Much faster than naïve method in all cases

Outline

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

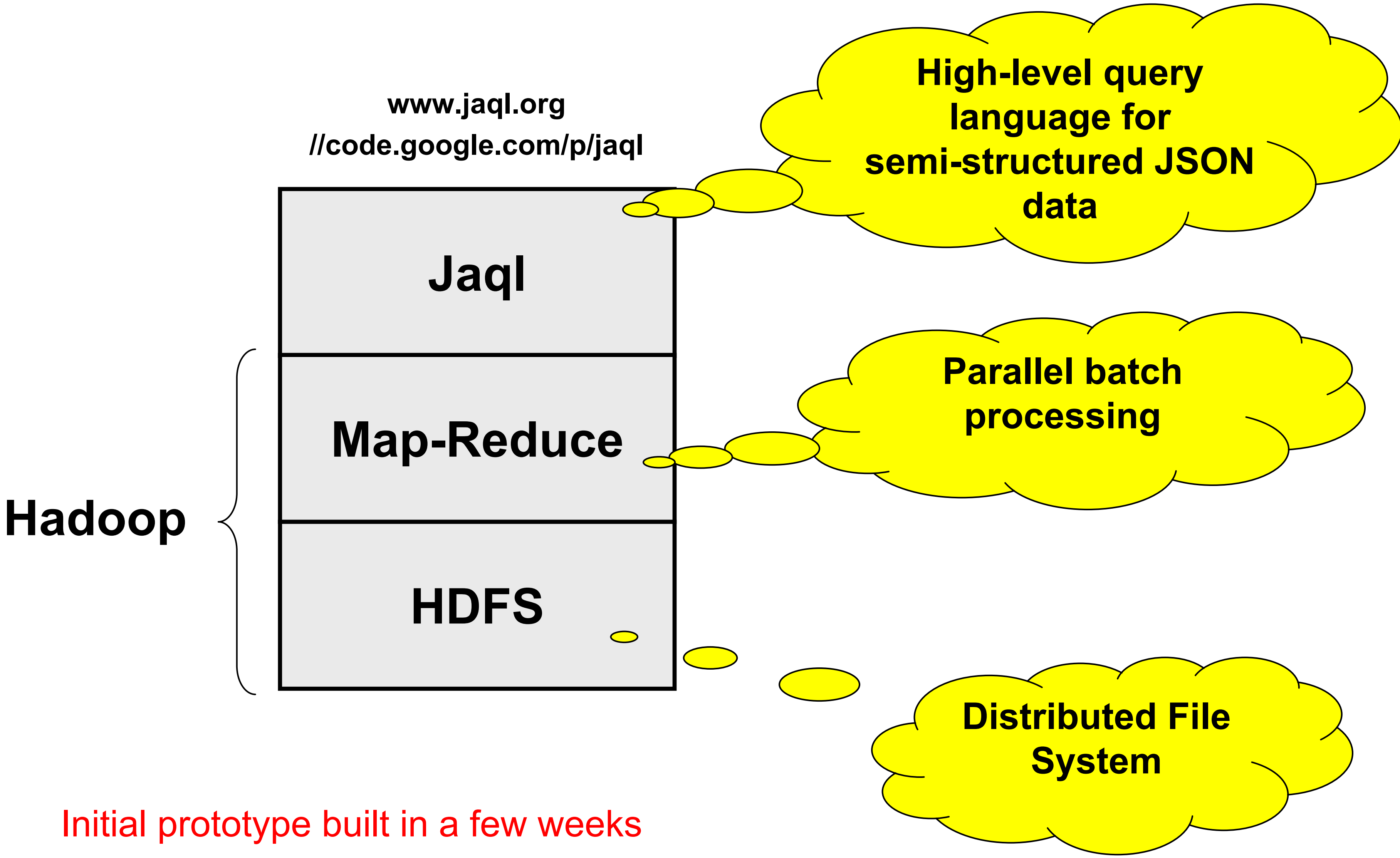
Motivation

- 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

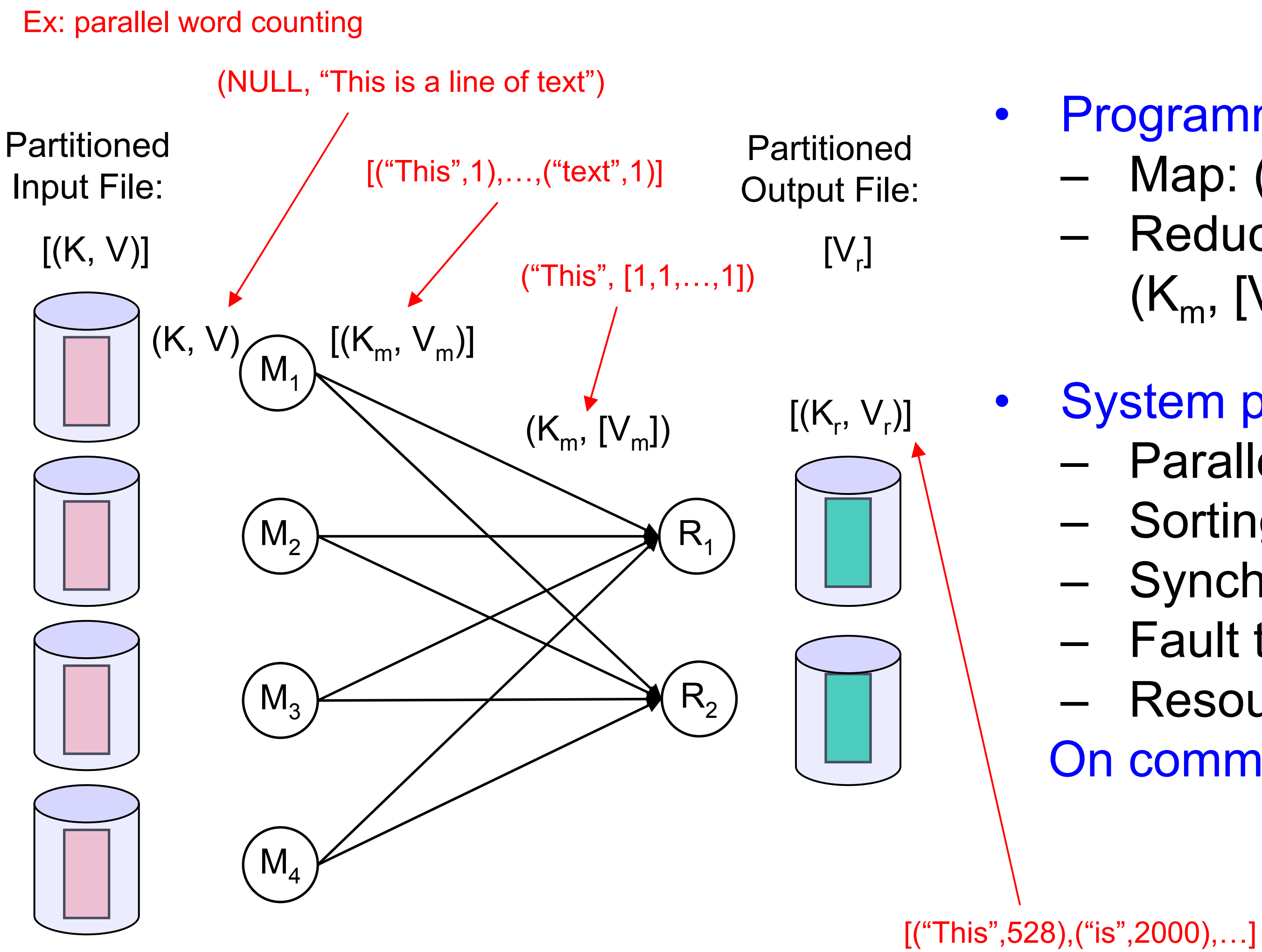
- How to represent bundles?
- How to specify map-reduce jobs?
- How to parallelize?
- How to seed tuple bundles?

A Cluster-Computing Infrastructure



Initial prototype built in a few weeks

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

MCDB Example

CID	Region
102	NewEngland
226	Midwest

CUST_ATTR

Region	Scale
NewEngland	7.0
Midwest	2.1

AMT_SCALE

CID	Shape
102	1.2
226	0.7

AMT_SHAPE

CID	Amount

SALES

CID	Shape	Scale
102	1.2	7.0
226	0.7	2.1

Q: SELECT SUM(Amount)
FROM SALES
AS t_sales

VG function

Gamma(shape, scale)

d_1

CID	Amount
102	\$120.00
226	\$60.00

d_2

CID	Amount
102	\$80.00
226	\$90.00

d_3

CID	Amount
102	\$80.00
226	\$130.00

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

$\hat{E}[t\_sales] = \$186.67$ $\hat{STD}[t\_sales] = \$20.82$

JSON and MC^3

[{cid: 102, region: NewEngland}, …]

↓

Join + Project

↓

[{cid: 102, shape: 1.2, scale: 7.0}, …]

↓

Seed

↓

[{cid: 102, shape: 1.2, scale: 7.0, seed: 306576301}, …]

↓

Instantiate

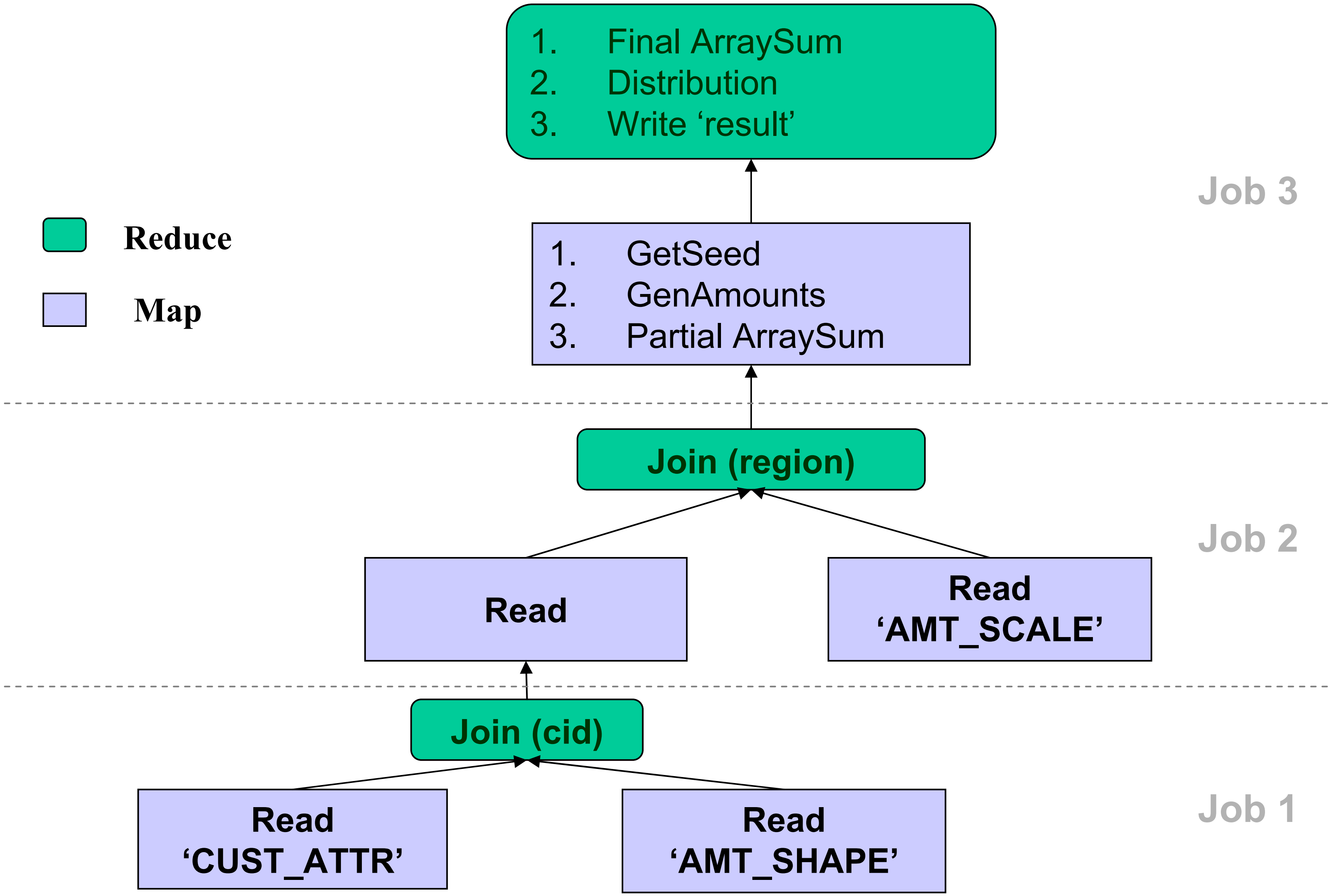
↓

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

JAQL and MC^3: 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



Parallelism Schemes

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

Tuple 1: (r1,…,r1000)

Tuple 2: (r1,…,r1000)

⋮

- 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

Tuple 1: (r1,…,r500)

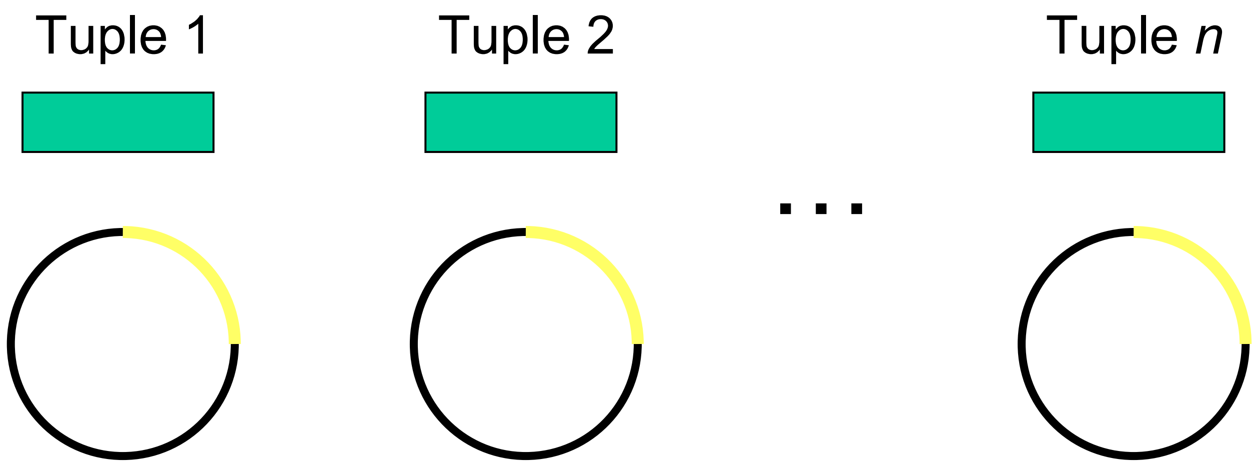
Tuple 1: (r501,…,r1000)

Tuple 2: (r1,…,r500)

Tuple 2: (r501,…,r1000)

⋮

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*)

Seeding

Tuple *j* at node *i*:

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

↓

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

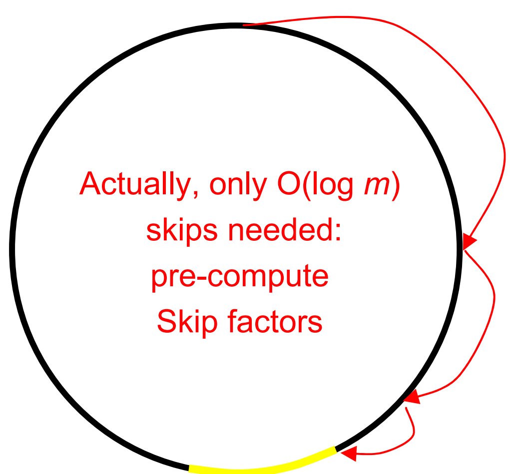
Instantiation

Tuple j at node i: Make

$m = b \times i + j$

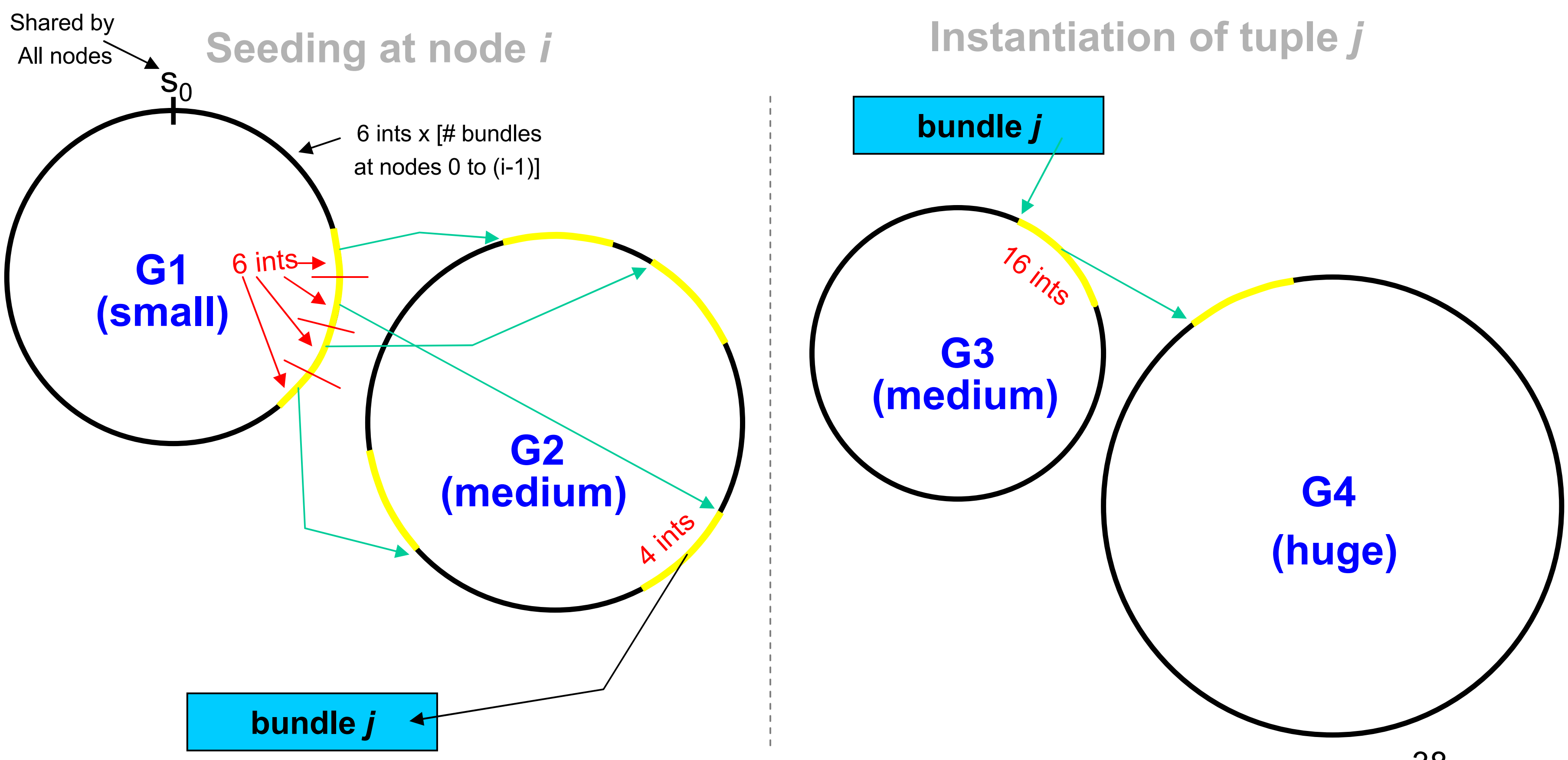
skips of length $c \times n$

to get to starting point



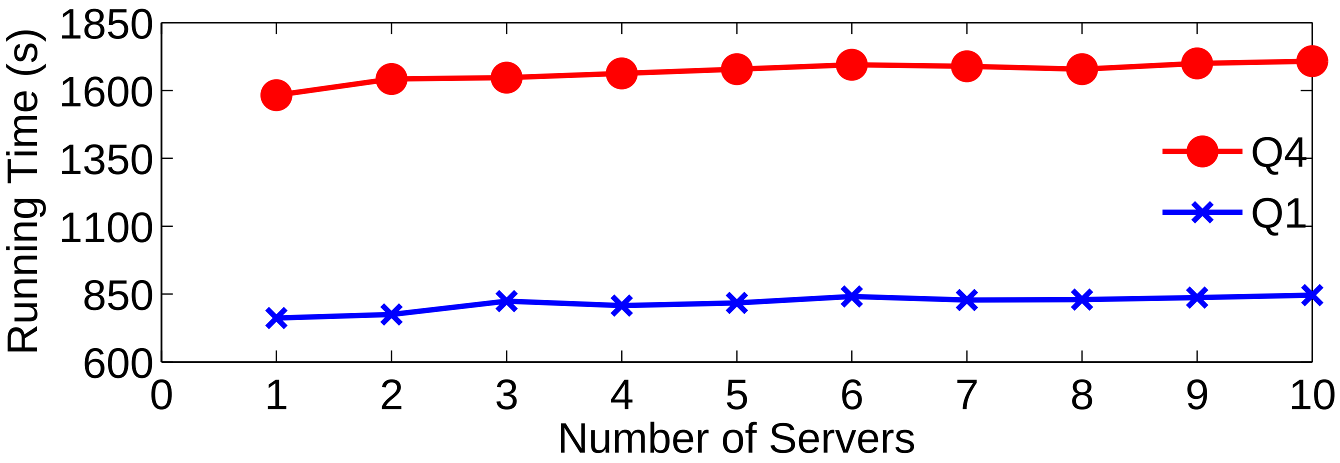
Multi-PRNG Method

- 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}



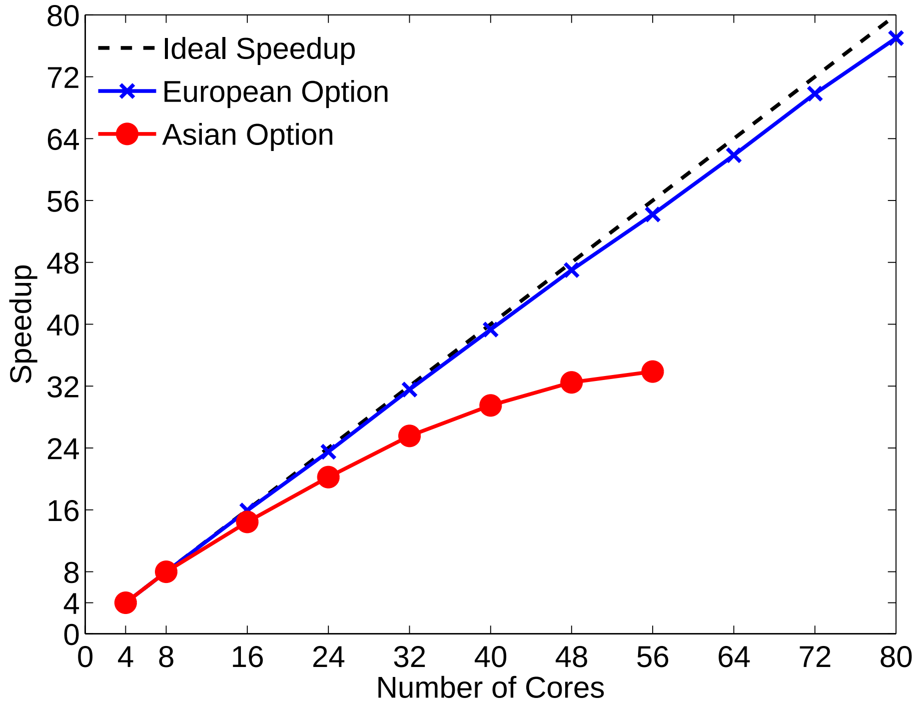
Scale-up Results: Inter-Tuple Parallelism

- 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

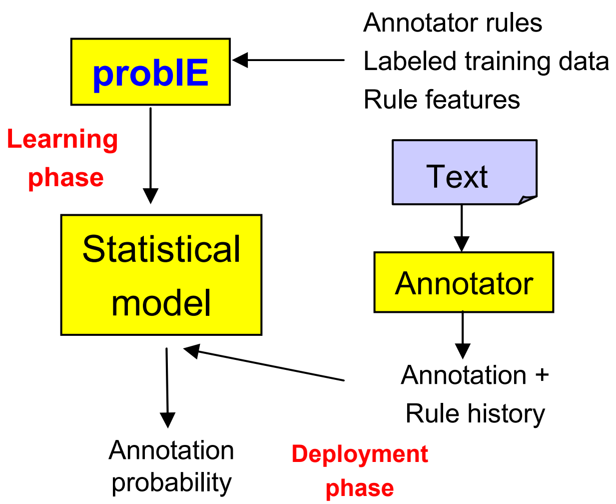
City	State	Strict range	Status
San Francisco	CA	[$30,$230]	guaranteed
San Jose	CA	[$70,$200]	non-guaranteed

Sum(Sales) group by City,State

State	Strict range	Status
CA	[$230,$230]	guaranteed

Sum(Sales) group by State

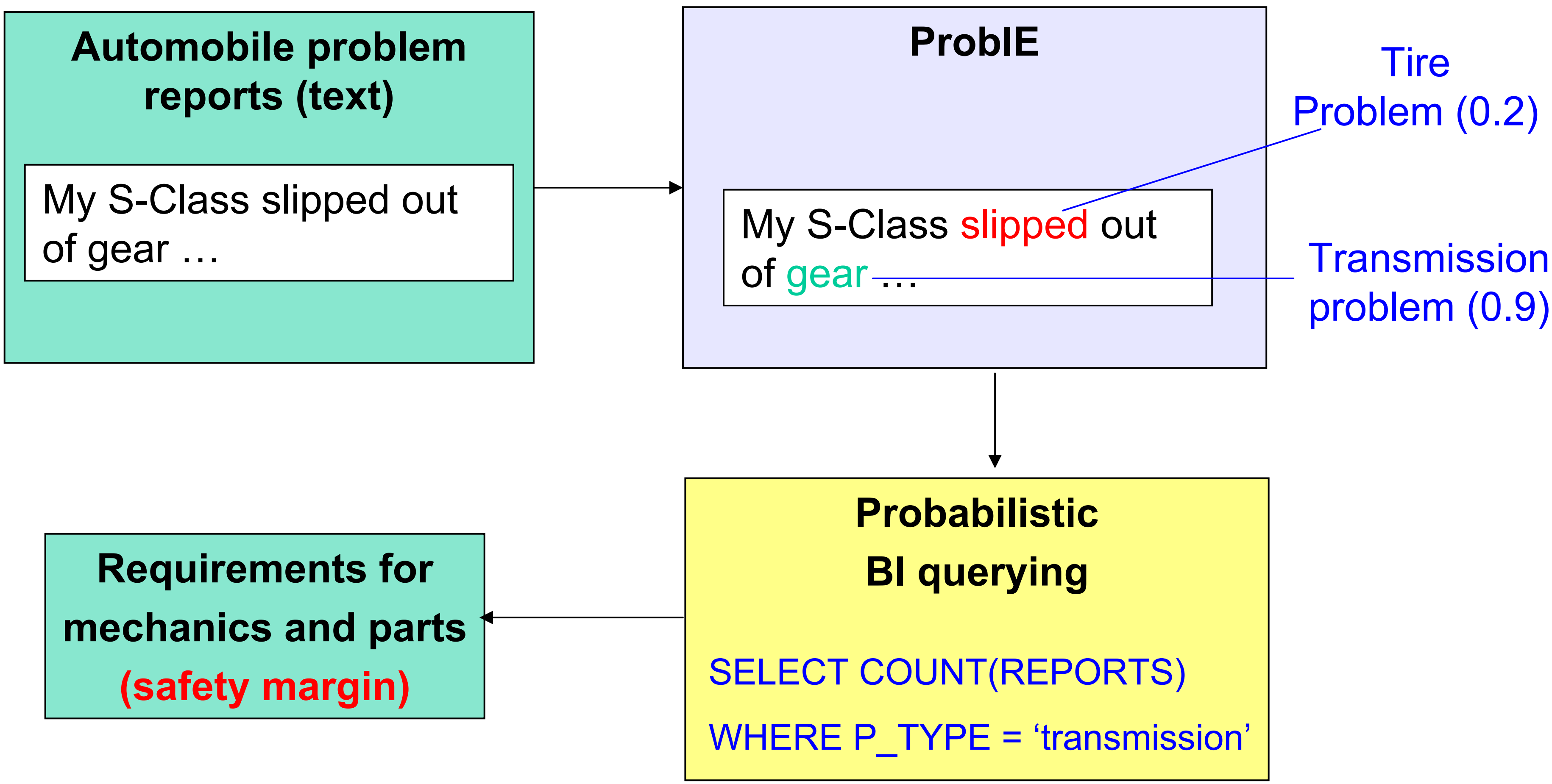
- ProbIE: Probabilistic info extraction [Michelakis et al., SIGMOD09]
  - For rule-based IE system (e.g., SystemT)
  - Provides confidence #'s for base/derived annotations
  - Based on "rule history", lower-level results
  - MaxEnt-based learning approach



Outline

- Motivation
- MCDB
- MC^3
- Related projects
- Future directions

An End-to-End ERP Scenario



Future Directions

- Performance
  - Query optimizer
    - E.g., push down inference & instantiation, choose parallelization scheme
    - Improve JAQL rewriter (MC^3 aware)?
  - Sequential and/or adaptive simulation? (MC^3)
  - 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
- MC^3: SIGMOD 2009
- ProbIE: 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

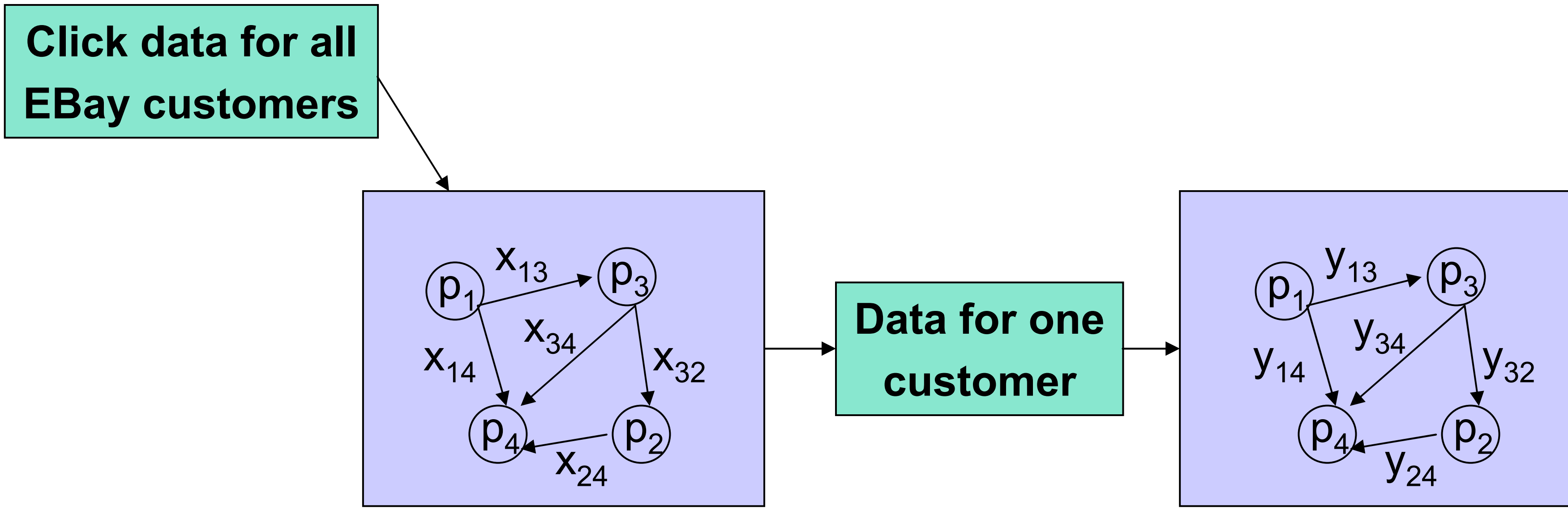
Pharmacy data for all customers

Cox hazard-rate disease model

Clinic-resource demand model

CustID	Time period	Resource needed
Jane Smith	June-Sept	?
…		…

Individual Click Behavior (EBay)



Global Markov model distribution (Dirichelet prior)

Individual Markov model distribution (posterior)

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

Logistics Under Uncertainty

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

Shipment

ITEM_ID	QUANTITY
curtains	50
…	…

In_Stock

ITEM_ID	QUANTITY
curtains	20
…	…

Current_Price

ITEM_ID	Price
curtains	$120
…	…

- Random tables

Sales_W_Ship

CUST_ID	ITEM_ID	QUANTITY
Smith	curtains	?
	…	…

Sales_WO_Ship

CUST_ID	ITEM_ID	QUANTITY
Smith	curtains	?
	…	…

- Queries:

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

```
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

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) **1 row, 1 column**
  - MONEY_SCALE (REGION, SCALE)
  - MONEY_SHAPE (CID, SHAPE)

Normalized storage

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
```

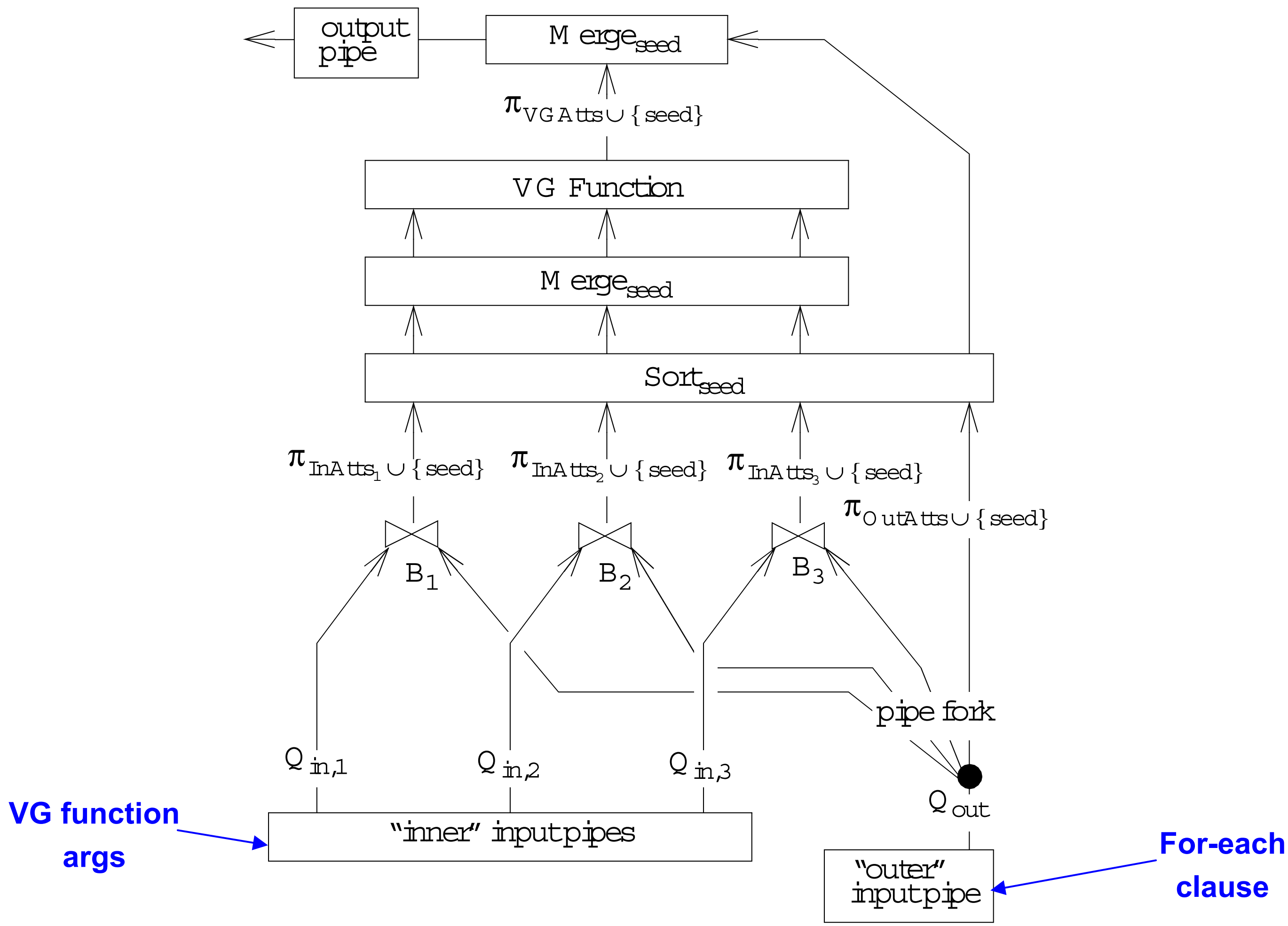
MLI has 1 row, 2 columns

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)

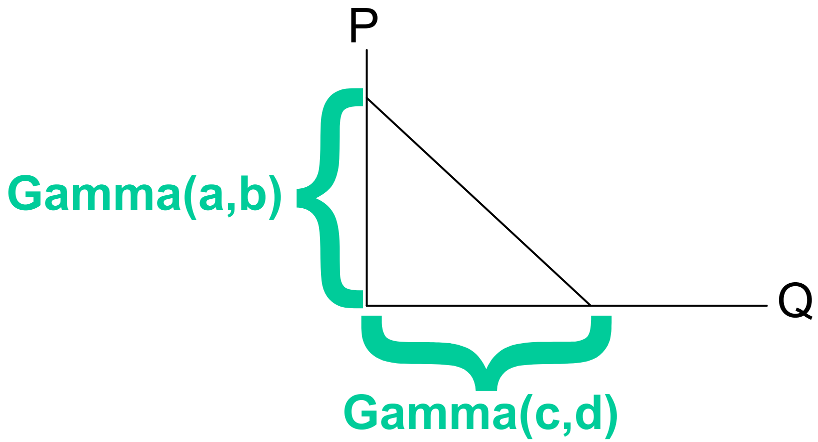
```
CREATE TABLE SENSORS (ID, LAT, LONG, TEMP) AS
 FOR EACH g in (SELECT DISTINCT GID FROM S_PARAMS)
  WITH TEMP AS MDNormal (
    (SELECT m.ID, m.MEAN FROM MEANS m S_PARAMS ss
     WHERE m.ID = ss.ID AND ss.GID = g.GID),
    (SELECT c.ID1, c.ID2, c.COV FROM COVARS c, S_PARAMS ss
     WHERE c.ID1 = ss.ID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



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



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:

Annotator	Candidate-Generation Rules	Rule Precision
Person **Base** annotator	**P1: <Salutation><CapitalizedWord><CapitalizedWord>** **P2: <First Name Dictionary><Last Name Dictionary>** **P3: <CapitalizedWord><CapitalizedWord>**	**High** **High** **Low**
PhoneNumber **Base** annotator	**Ph1: <PhoneClue><\d{3}-\d{3}-\d{4}>** **Ph2: <\d{3}-\d{3}-\d{4}>** **Ph3: <\d{5}>**	**High** **Medium** **Low**
PersonPhone **Derived** annotator	**PP1: <Person><"can be reached at"><PhoneNumber>** **PP2: <"call"><Person><0-2 tokens><PhoneNumber>** **PP3: [<Person><PhoneNumber>]$_{\text{sentence}}$**	**High** **High** **Medium**

+ 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 …

Annotator	Annotation	Rules
Person	Greg Mann	P2, P3
PhoneNumber	408-663-2817	Ph2
PersonPhone	(Greg Mann, 408-663-2817)	PP1

Document d_2

… please call Heather Choate at x33278 …

Annotator	Annotation	Rules
Person	Heather Choate	P2, P3
PhoneNumber	33278	Ph3
PersonPhone	(Heather Choate, 33278)	PP2

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

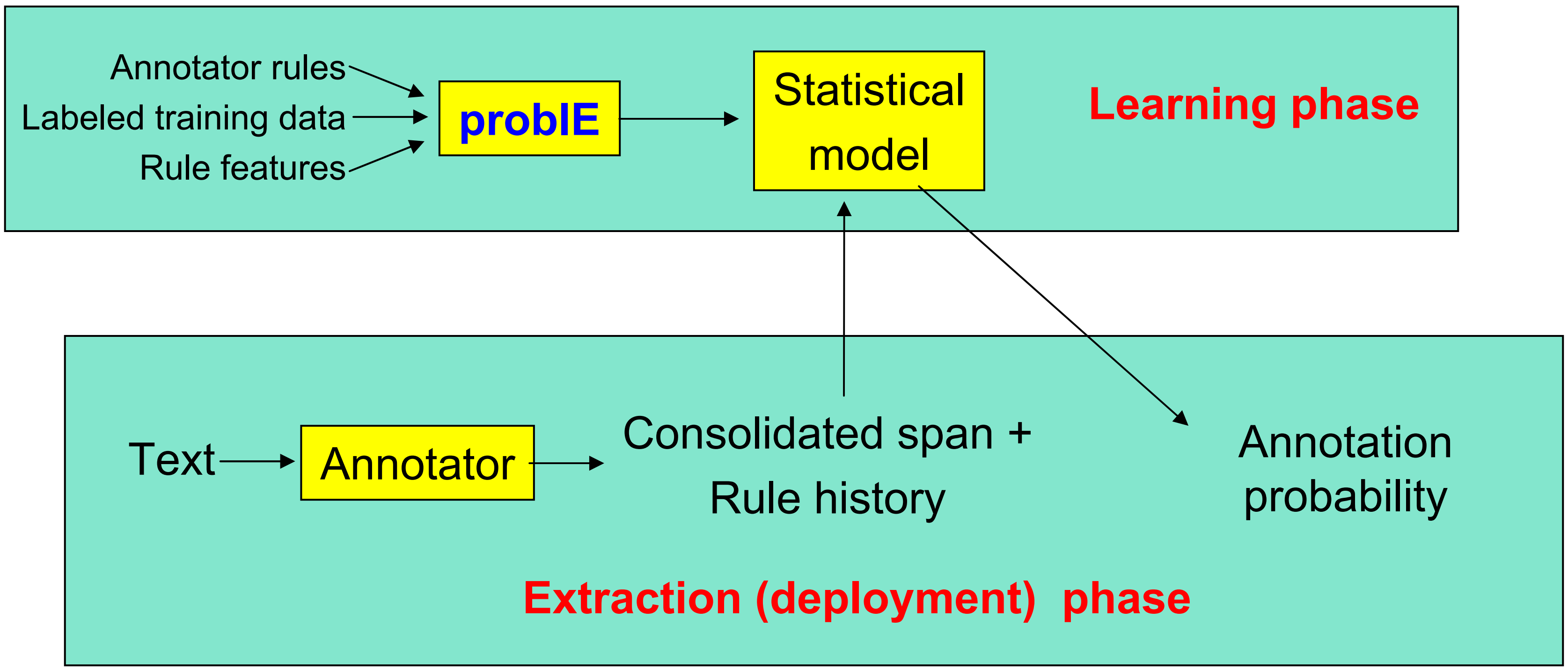
span

	P1	P2	P3
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 ith 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

$$p_0(r) = P(R(s) = r \mid A(s) = 0, K(s) = 1)$$

$$p_1(r) = P(R(s) = r \mid A(s) = 1, K(s) = 1) \quad \Rightarrow \quad q(r) = \frac{\pi p_1(r)}{\pi p_1(r) + (1-\pi) p_0(r)}$$

$$\pi = P(A(s) = 1 \mid K(s) = 1)$$

  - π 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 θ for exponential distribution

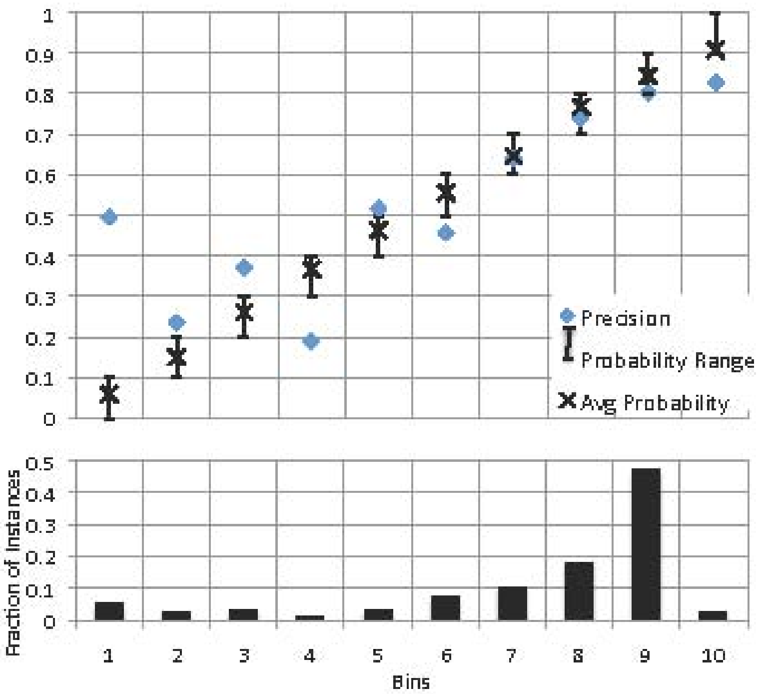
$$p_1(r;\theta) = \frac{1}{Z(\theta)} \exp\left\{\sum_{c \in C} \theta_c f_c(r)\right\}$$

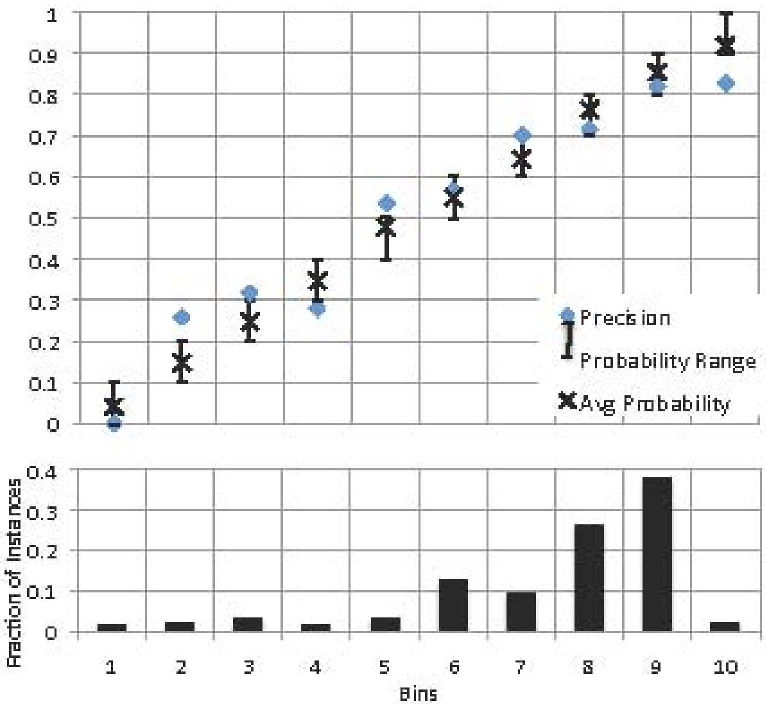
f_c = Indicator function for constraint c

  - Use improved iterative scaling (IIS) to fit θ 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)