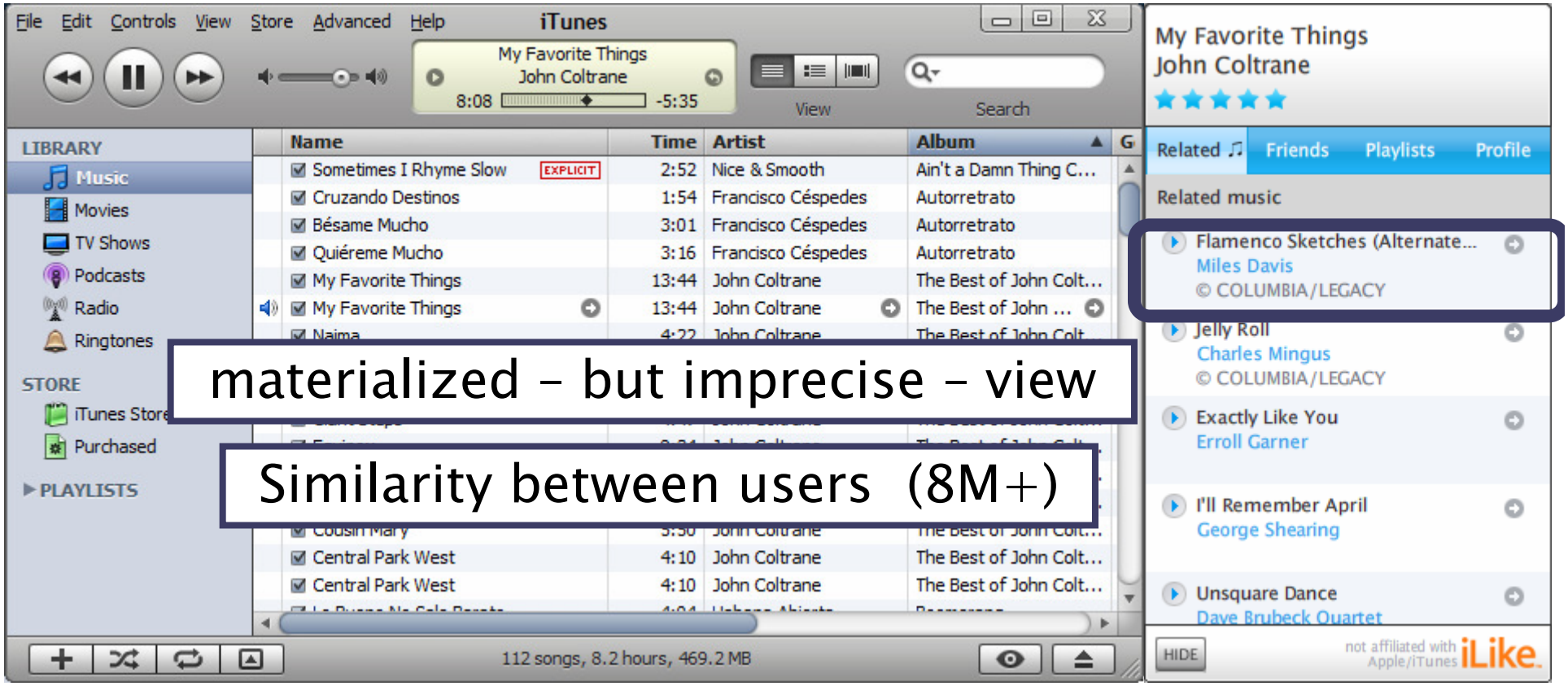
Materialized Views in Probabilistic Databases
for Information Exchange and Query Optimization

Christopher Re and Dan Suciu
University of Washington

Motivating Example: Optimization



materialized – but imprecise – view

Similarity between users (8M+)

Single Slide Summary

- Renewed interest in probabilistic data
  - Trio, MayBMS, Maryland, Purdue, UW
  - *Classical* : Integration, record linkage, etc.
  - *Emerging:* iLike "Similarity Scores"
- Tod… DBs
  - F…
  - Better maintainability
- The Catch: Every view using lineage, but…
  - Correlations cause lineage to become large

When can we get the benefits of materialized views in prob DBs?

Overview

- Motivation and Background
- Technical Meat
- Experiments
- Conclusion

Probabilistic DBs
Restaurant Example

- *Block Independent Disjoint* (BID)
- Popular: Barbara92, Trio, Mystiq, Green *et al.*
- Query Evaluation
  - *Safe* Queries
  - *Multisimulation*

Chef	Dish	Rate	P
TD	Crab	High	0.8
		Med	0.1
		Low	0.1
TD	Lamb	High	0.3
		Low	0.7

Rating(Chef,Dish; Rating)

Possible Worlds Key

Value Attributes

Restaurant Example

Chef	Restaurant	P	
TD	D. Lounge	0.9	p1
TD	P .Kitchen	0.7	p2

W(Chef,Restaurant) *WorksAt*

Chef	Dish	Rate	P	
TD	Crab	High	0.8	q1
TD	Lamb	High	0.3	q2

Lineage could be large

Reprocessing lineage is expensive

Restaurant	Dish
D. Lounge	Crab
P. Kitchen	Crab
P. Kitchen	Lamb

S(Restaurant,Dish) *Serves*

"Chefs who serve a highly rated dish"

V(c) :- W(c,r),S(r,d),R(c,d,"High")

Chef	Restaurant	P	
TD	D. Lounge	0.72	p1*q1
TD	P.Kitchen	0.602	p2*(1 - (1-q1)(1-q2))

Understand w.o. "lineage"?

Views and Query Semantic

Views: Conjunctive, Constants $V(H) :- g_1, \ldots, g_n$

DB Semantics: Possible Worlds

$$\mathcal{W} = \{W_1, \ldots, W_n\} \quad \mu : \mathcal{W} \rightarrow [0,1] \quad \sum_{W \in \mathcal{W}} \mu(W) = 1$$

View Semantics

$$\mu(V(t)) \stackrel{\text{def}}{=} \sum_{W : W \models V(t)} \mu(W)$$

Add worlds, if V is true

$$\mathcal{O}(V) = \{(t, p) \mid \mu(V(t)) = p > 0\}$$

Output of V

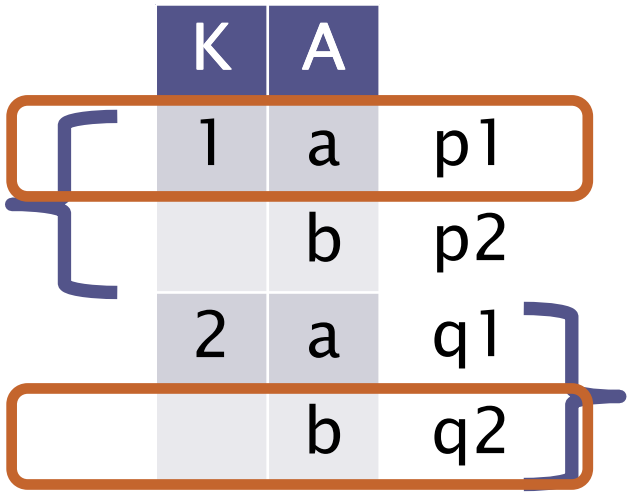
Overview

- Motivation and Background
- Technical Meat
- Experiments
- Conclusion

Technical Question: Representation

- Is output of V(H) on any BID database a BID table?
  - Represent with Schema + marginal probs.
- Yes, if there is $K \subseteq H$ s.t.
  - V is K–"block independent" ← this talk
  - V is K–"disjoint in blocks"

K-"block Independence"



K	A	
1	a	p1
	b	p2
2	a	q1
	b	q2

All tuples from distinct "blocks"

Multiply probs p1 * q2

Intuition: Fails if tuples in different blocks *depend* on same tuple

$$I \subseteq O(V) \text{ s.t. } s,t \in I \; s[K] = t[K] \implies s = t$$

$$\mu(\bigwedge_{s \in I} V(s[H])) = \prod_{s \in I} s[P]$$

Critical tuples

all tuples are disjoint critical

- Preliminary notion
  - Def: t is a *disjoint critical tuple* for a Boolean view V() if exists W

$$W \models V(), \text{ but } W - \{t\} \not\models V()$$

a world, W

V() :- W('TD','DL'),S('DL',d),R('TD',d,'High')

Chef	Rest
TD	DL

W(Chef, Restaurant)

Rest	Dish
D. L	Crab

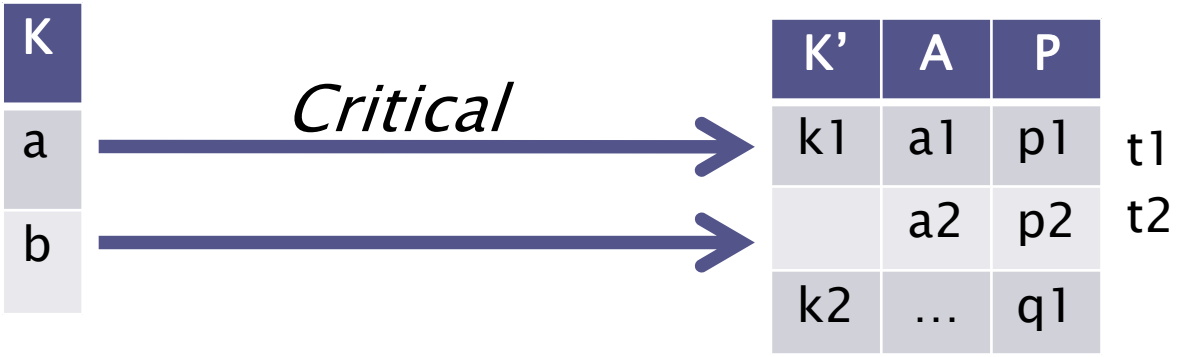
S(Restaurant,Dish)

Chef	[illegible]	Rate
TD	[illegible]	High

R(Chef,Dish,Rate)

Doubly Critical tuples

- *property of view V on any DB*
- *Exists* t1 critical for V(a) & t2 critical for V(b)
  - t1 and t2 in same block in a prob. relation



Thm: A conjunctive view V is K-Block independent iff no K-doubly critical tuples

Complexity…and a Practical test

- Thm: Deciding if a view is block independent is decidable and $\Pi_2^P - \text{Complete}$
- In wild, practical test almost always works

V(c,r) :- W(c,r),S(r,d),R(c,d,'High') K={c,r}

V(c) :- W(c,r),S(r,d),R(c,d,'High') K={c}

  - Test: "Can a prob tuple unify with different heads?"
    - If so, *not* block independent
- Thm: If view has no self-joins, test is complete.

Additional Results

- How to pick K in the view
- Dealing with disjointness
  - "Disjoint in blocks"
- Partial representability.
  - Some views not representable,
    - But a query on a view is still correct
  - In general, hard, but practical test
- Sets of Views

Overview

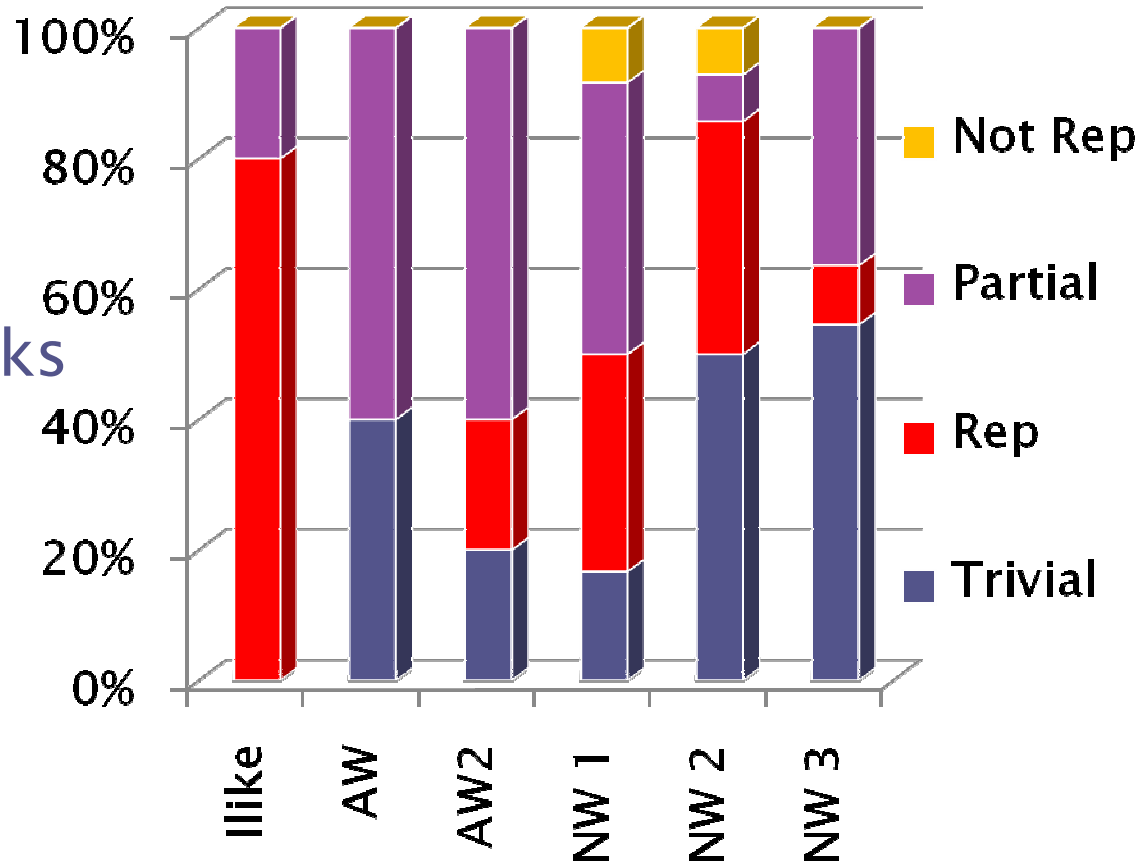
- Motivation and Background
- Technical Meat
- Experiments
- Conclusion

Experiments: Wild Queries, % rep.

- Three Datasets
  - iLike
  - SQL Server
    - Adventure works
    - Northwinds

96% partially
63% representable

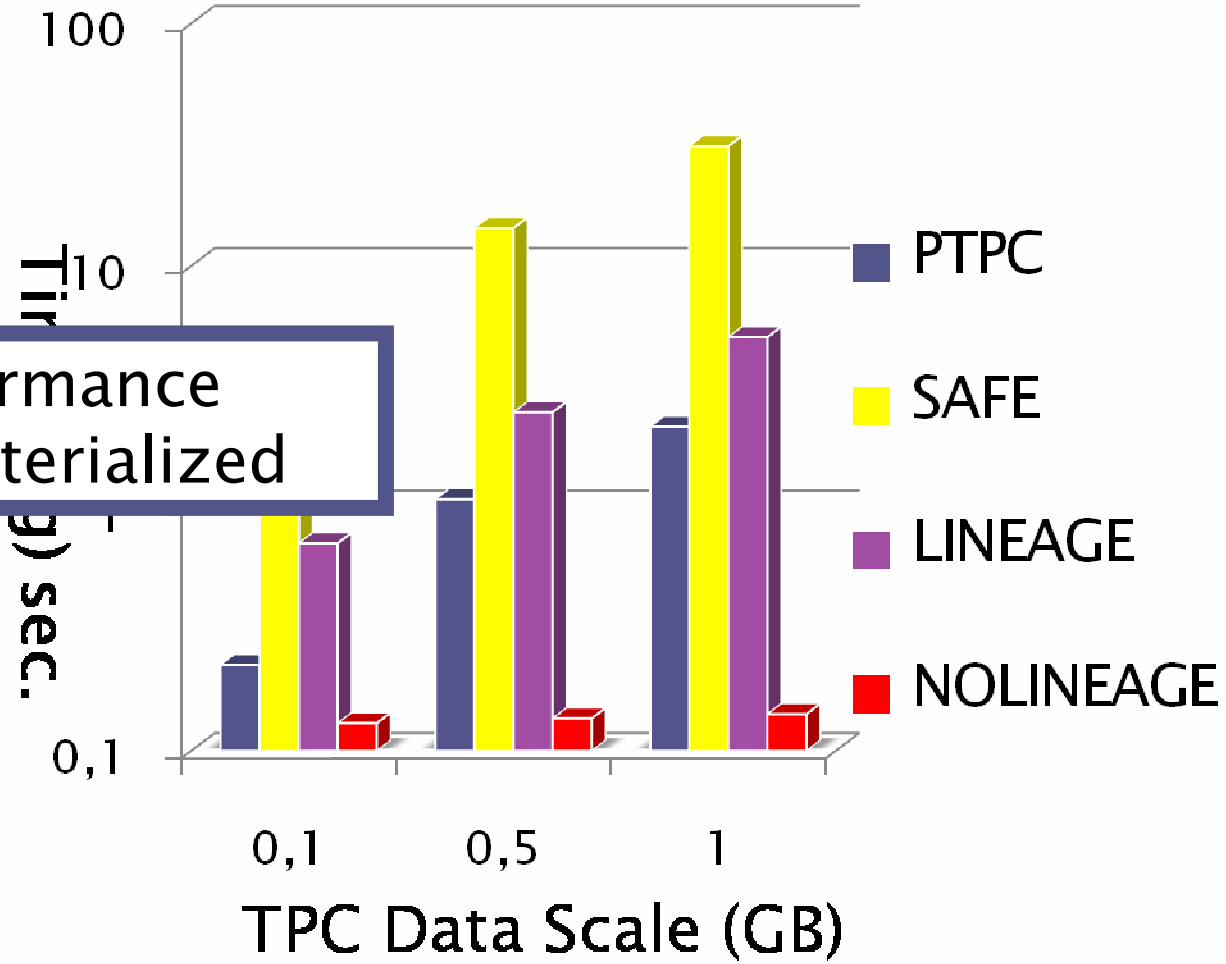
99.5% of iLike workload use representable views



Experiments

- TPC-H data
- Q10

Expected performance increase from materialized views



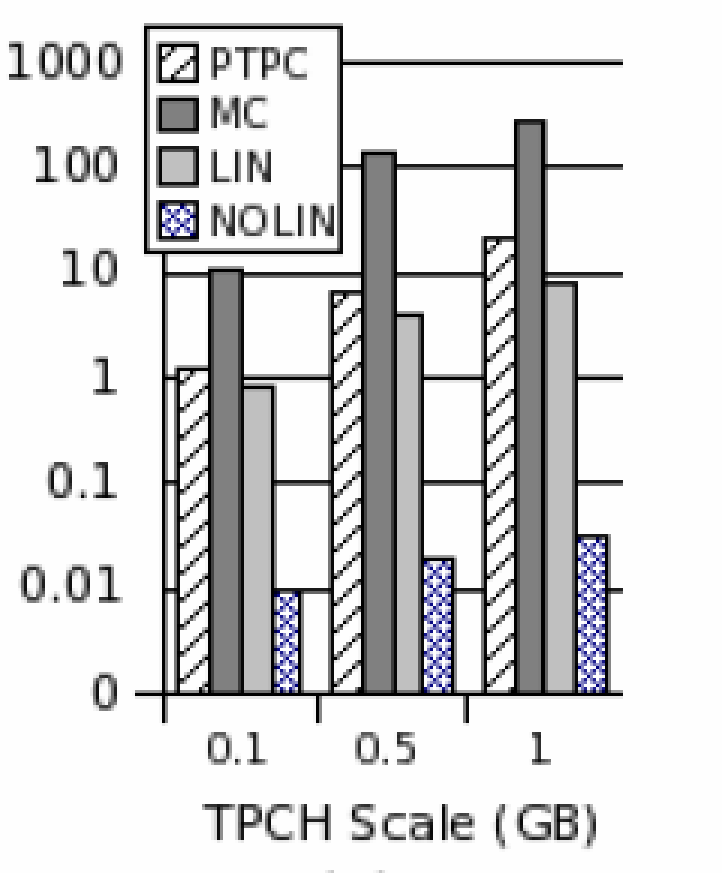
Conclusion

- Materialized views for probabilistic data
  - Problem: Retain classical benefits of views
- Contributions
  - A complete theoretical solution
  - Practical solutions
- Verified Experimentally
  - Views exist in practice
  - Query processing benefits, as expected

Experiments

- TPC-H data
- Q5 unsafe query.
- Key
  - PTPC: w.o prob
  - MC: Monte Carlo
  - LIN: w. lineage
  - NOLIN: Our technique

NB: LIN not an End-to-End running time. So needs another ~ MC additional seconds!



Information Exchange

Chef	Restaurant	P
TD	D. Lounge	0.9
TD	P.Kitchen	0.7
MS	C.Bistro	0.8

W(Chef,Restaurant) *WorksAt*

Restuarant	Dish
D. Lounge	Crab
P. Kitchen	Crab
P. Kitchen	Lamb
C. Bistro	Fish

S(Restaurant,Dish) *Serves*

Chef	Dish	Rate	P
TD	Crab	High	0.8
		Med	0.1
		Low	0.1
TD	Lamb	High	0.3
		Low	0.7
MS	Fish	High	0.6
		Low	0.3

R(Chef,Dish,Rate) *Rated*

V(c,r) :- W(c,r),S(r,d),R(c,d,'High')

Technical Question 2: Partially representable

- Question 2: Given a BID database, a view V and a query Q, can we answer the result of V(D) from Q?
- Show a query that is partially representable and one that correctly uses it, and one that does not.
- Does not define a unique probability distribution