Handling Uncertainty in Data Management

Jian Li

Tsinghua University, Beijing, China

Feb. 2012

HotDB 2012, Beijing

Uncertain Data

- Uncertain data is ubiquitous
  - Data Integration and Information Extraction
  - Sensor Networks; Information Networks

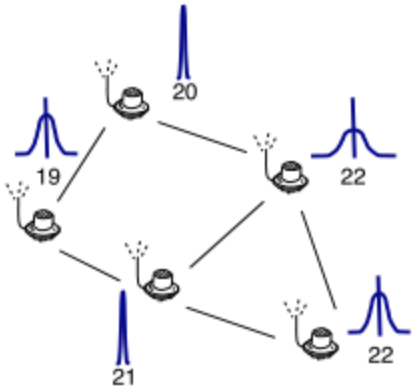
SSN	Name
208-79-4209	John Williams

SSN	Name
208-79-4209	Michael Lewin

SSN	Name	Prob
208-79-4209	John Williams	0.5
208-79-4209	Michael Lewin	0.5

Tuple uncertainty

Data integration

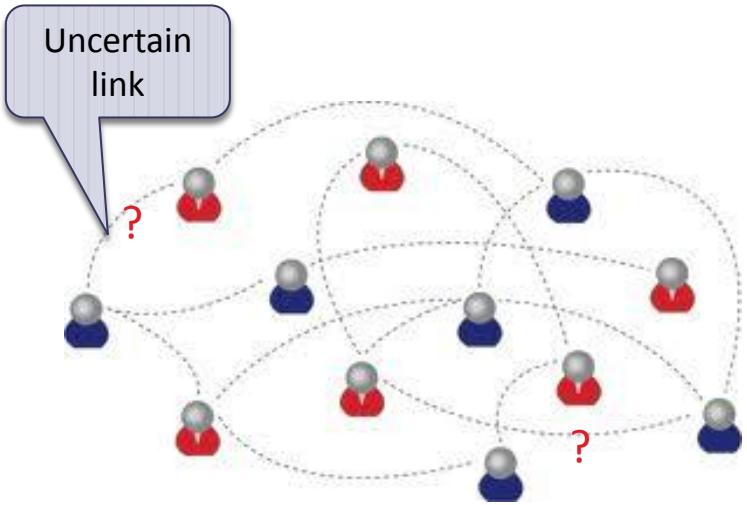


Sensor network

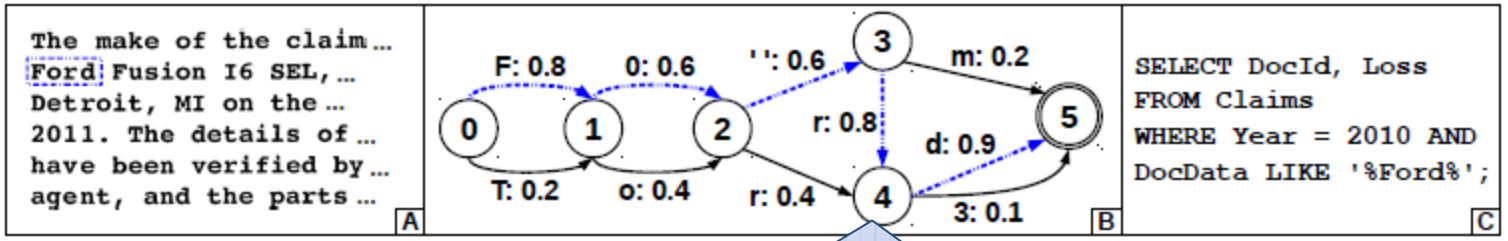
Sensor ID	Temp.
1	Gauss(40,4)
2	Gauss(50,2)
3	Gauss(20,9)
…	…

Attribute uncertainty

Uncertain Data



Social network



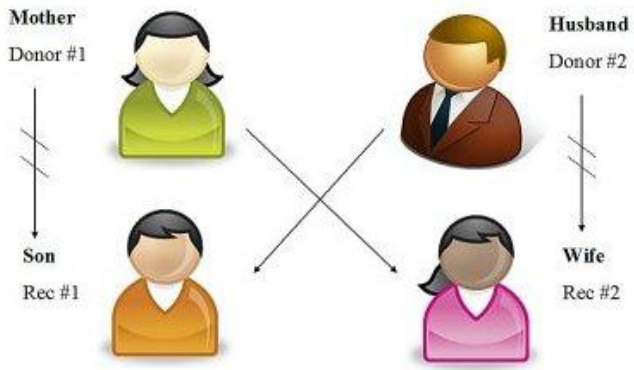
Stochastic Finite Automata

OCR (Optical Character Recognition) data. [Kumar and Re, VLDB2011]

Uncertain Data

Decision making under uncertainty

- Kidney exchange



- Estimation of the success prob. based on blood type etc.
- Need to run the crossmatch test to ensure a match (more expensive and time-consuming).

- Future data is destined to be uncertain

Dealing with Uncertainty

- There is an increasing need for analyzing and reasoning over such data
- Handling uncertainty is a very broad topic that spans multiple disciplines
  - Economics / Game Theory
  - Finance
  - Probability Theory / Statistics
  - Psychology
  - Computer Science

Outline

- Dealing with uncertainty in data management
  - Probabilistic databases
    - Possible world semantics
    - Conjunctive queries
    - Ranking and top-k queries
    - Other queries
  - Beyond expected values – expected utility theory
  - Some tools out there that may be useful (with applications)
    - Uncertainty resolution
    - Portfolio theory
    - Multi-arm bandit

Probabilistic Databases

- Probabilistic databases
  - Goal: Managing probabilistic data and support declarative (SQL) query processing
- Many probabilistic database prototypes
  - Mystiq (U. Washington)
  - Trio (Stanford)
  - Orion (Purdue)
  - MayBMS (Cornell)
  - PrDB (UMD)
  - MCDB (Rice & IBM)
  - PODS (UMass)

Probabilistic Databases

- Probabilistic data models
  - Independent tuples

ID	Score	Prob
t_1	200	0.2
t_2	150	0.8
t_3	100	0.4

Tuple uncertainty

Sensor ID	Temp.
1	Gauss(40,4)
2	Gauss(50,2)
…	…

Attribute uncertainty

  - Block independent tuple / x-tuples

ID	Score	Prob
t_1	200	0.2
t_2	150	0.8
t_3	100	0.4

Block 1

Block 2

At most one tuple exists in a block

Probabilistic Databases

- Probabilistic data models
  - Probabilistic c-table
  - Probabilistic and/xor trees
  - World set algebra
  - Graphical Models
  - ….

Possible World Semantics

View a probabilistic database as probability distribution over the set of possible worlds

ID	A	Prob
t_1	1	0.2
t_2	1	0.8
t_3	2	0.4

A probabilistic table
(assume tuple-independence)

pw1

ID	A
t_1	1
t_2	1
t_3	2

w.p. 0.064

pw2

ID	A
t_1	1
t_2	1

w.p. 0.096

pw3

ID	A
t_2	1
t_3	2

w.p. 0.256

8 worlds

Possible World Semantics

View a probabilistic database as probability distribution over the set of possible worlds

ID1	A	Prob
t_1	1	0.2
t_2	1	0.8
t_3	2	0.4

T

ID2	B	Prob
s_1	1	0.5

S

pw1

ID1	A
t_1	1
t_2	1
t_3	2

ID2	B
s_1	1

w.p. 0.032

pw2

ID1	A
t_1	1
t_2	1

ID2	B
s_1	1

w.p. 0.048

pw3

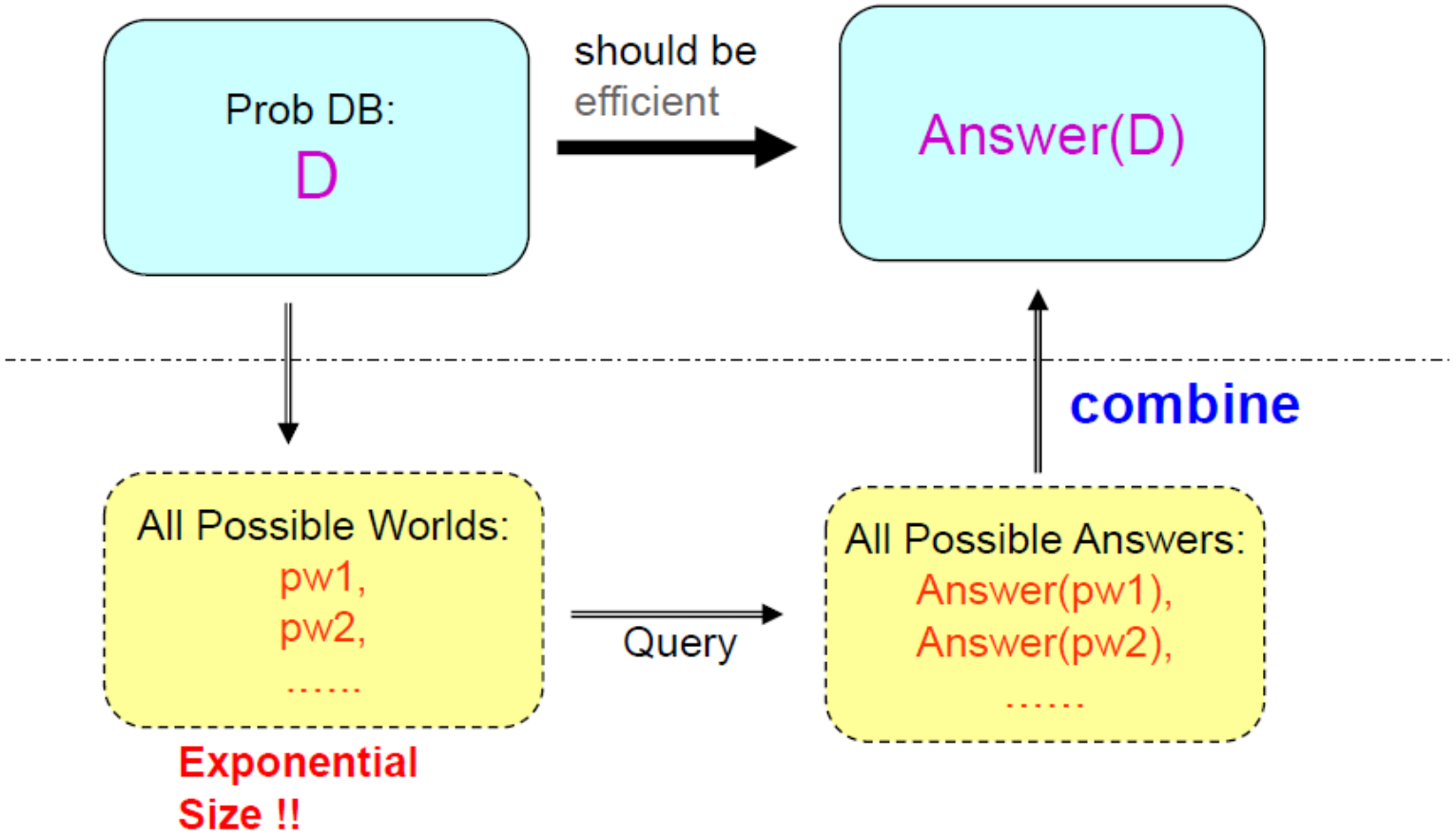
ID1	A
t_2	1
t_3	2

ID2	B

w.p. 0.128

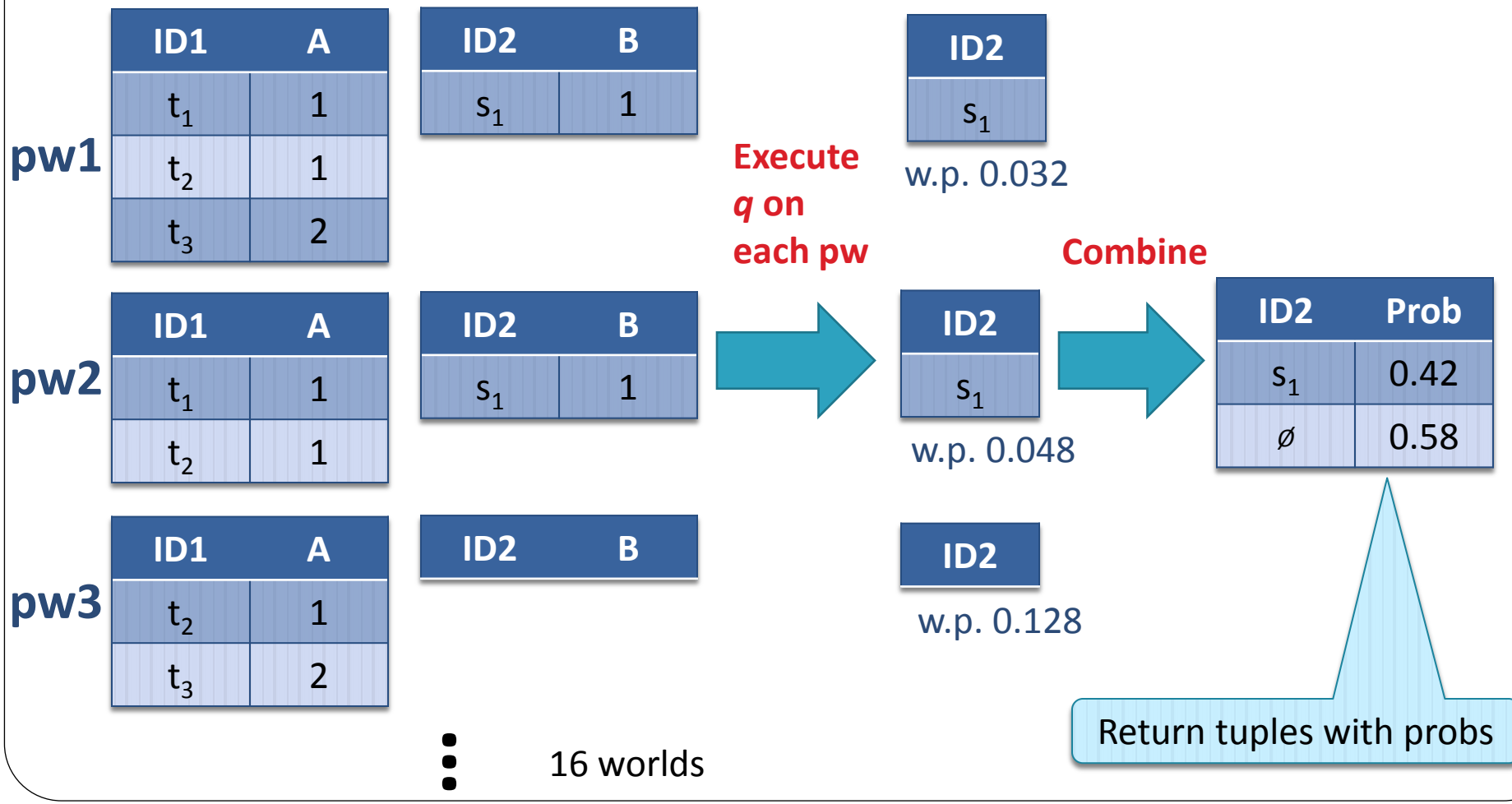
16 worlds

Possible World Semantics



Possible World Semantics

- Conjunctive query: *q(id2):= T(id1, a), S(id2,b), a=b*



Outline

- Dealing with uncertainty in data management
  - Probabilistic databases
    - Possible world semantics
    - Conjunctive queries
    - Ranking and top-k queries
    - Other queries
  - Beyond expected values – expected utility theory
  - Some tools out there that may be useful (with applications)
    - Uncertainty resolution
    - Portfolio theory
    - Multi-arm bandit

Conjunctive Query

- Safe plan

ID1	A	Prob
t_1	1	0.2
t_2	1	0.8
t_3	2	0.4

T

ID2	B	Prob
s_1	1	0.5

S

- Conjunctive query: *q(id2):= T(id1, a), S(id2,b), a=b*

ID1	A	Prob
t_1	1	0.2
t_2	1	0.8
t_3	2	0.4

$\pi_A(T)$

A	Prob
1	0.84
2	0.4

1-(1-0.2)(1-0.8)

$\pi_A(T) \bowtie_{A=B} S$

A	B	ID2	Prob
1	1	s_1	0.42

0.84×0.5

$\pi_{ID2}(\pi_A(T) \bowtie_{A=B} S)$

ID2	Prob
s_1	0.42

PLAN 1

Efficient query evaluation on probabilistic databases, Dalvi, N. and Suciu, D. VLDB Journal, 2007

Conjunctive Query

Safe plan

ID1	A	Prob
t_1	1	0.2
t_2	1	0.8
t_3	2	0.4

T

ID2	B	Prob
s_1	1	0.5

S

- Conjunctive query: *q(id2):= T(id1, a), S(id2,b), a=b*

ID1	A	Prob
t_1	1	0.2
t_2	1	0.8
t_3	2	0.4

$T \bowtie_{A=B} S$

ID1	A	B	ID2	Prob	
t_1	1	1	s_1	0.1	0.2×0.5
t_2	1	1	S_1	0.4	0.8×0.5

$\pi_{ID2}(T \bowtie_{A=B} S)$

ID2	Prob	
s_1	0.46	$1 - (1 - 0.1) \times (1 - 0.4)$

PLAN 2

Which one is correct??

Conjunctive Query

Safe plan

ID1	A	Prob
t_1	1	0.2
t_2	1	0.8
t_3	2	0.4

T

ID2	B	Prob
s_1	1	0.5

S

- Conjunctive query: *q(id2):= T(id1, a), S(id2,b), a=b*

ID1	A	Prob
t_1	1	0.2
t_2	1	0.8
t_3	2	0.4

$T \bowtie_{A=B} S$

ID1	A	B	ID2	Prob
t_1	1	1	s_1	0.1
t_2	1	1	s_1	0.4

0.2×0.5

0.8×0.5

NOT INDEPENDENT!

$\pi_{\mathrm{ID2}}(T \bowtie_{A=B} S)$

ID2	Prob
s_1	0.46

$1 - (1 - 0.1) \times (1 - 0.4)$

Wrong!

Not safe!

PLAN 2

Which one is correct?? Plan 1!

Conjunctive Query

- How to generate a safe plan (High-level):
  - Try to do all safe projections late in the query plan (safeness can be determined by functional dependencies)
  - If no safe projection is possible, try to perform a join (need to meet certain separateness condition)
- The Dichotomy Theorem

  For any conjunctive query *q* without self-join,
  - either there is a safe plan for *q*,
  - or the data complexity of *q* is #P-complete

Efficient query evaluation on probabilistic databases, Dalvi, N. and Suciu, D. VLDB J, 2007

- Similar dichotomy results also hold for conjunctive query with self-joins, and union of conjunctive queries (SPJU)

The Dichotomy of Conjunctive Queries on Probabilistic Structures , Nilesh Dalvi, Dan Suciu, PODS, 2007
Computing query probability with incidence algebras , Nilesh Dalvi, Karl Schnaitter, Dan Suciu, In PODS, 2010

Conjunctive Query

- What if the query is #P-hard??

- We can use the Karp-Luby-Madras method to approximate the probability.
  - The algorithm was developed for counting DNF solutions, but can be adopted to compute probabilities.
  - We can get a ϵ-approximation for any $\epsilon > 0$
    (i.e., our estimate$\in [(1-\epsilon),(1+\epsilon)] \times$true value)
    in Poly(n, 1/ϵ) time with high probability.

Richard M. Karp, Michael Luby, Neal Madras: Monte-Carlo Approximation Algorithms for Enumeration Problems. J. Algorithms, 1989

Monte Carlo Method

- The Basic Monte Carlo:
  - Sample N possible worlds: $pw_1, pw_2, \ldots\ldots pw_N$.
  - Execute the query *q* for all worlds: $q(pw_1), q(pw_2), \ldots\ldots q(pw_N)$.
  - *Our estimate:* $Prob(t) = \#\{i \mid t \in pw_i\} / N$

- Estimator Theorem (Chernoff Bound Essentially):

The Monte Carlo method gives

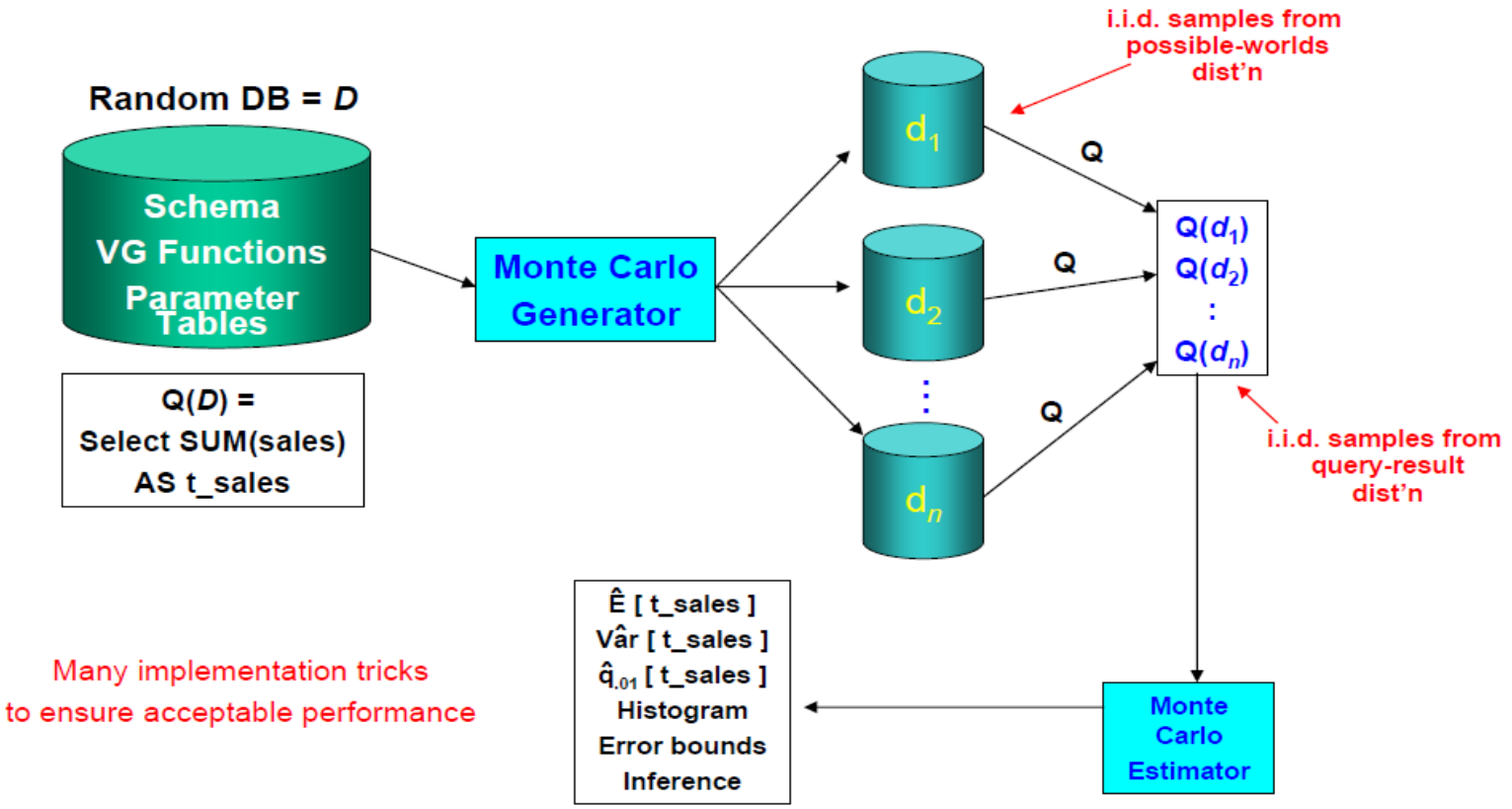
$$\text{Prob}\left[our\ estimate \in [true\ value - \epsilon, true\ value + \epsilon]\right] \geq 1 - \delta$$

if the number of samples $N \geq O(\frac{1}{\epsilon^2} \log \frac{1}{\delta})$

Monte Carlo Method

Use Monte Carlo for all queries!

We can only get approximate answers (but the prob may not be accurate anyway!)



MCDB: a monte carlo approach to managing uncertain data, Jampani, R. and Xu, F. and Wu, M. and Perez, L.L. and Jermaine, C. and Haas, P.J. SIGMOD 2008

Monte Carlo Method

- However, the above scheme is not efficient enough for estimating very small probabilities.
- Estimator Theorem (Chernoff Bound Essentially):

The Monte Carlo method gives

$$\text{Prob}\left[our\ estimate \in [true\ value - \epsilon, true\ value + \epsilon]\right] \geq 1 - \delta$$

if the number of samples $N \geq O(\frac{1}{\epsilon^2}\log\frac{1}{\delta})$

If ϵ is extremely small, we need a lot of samples

- But estimating small probabilities is very important in many applications, such as risk management. Therefore, we need more advanced techniques.
  - Karp-Luby-Madras
  - MCMC (Markov Chain Monte Carlo)
  - Gibbs sampling, e.g., MCDB-R: Risk analysis in the database, Arumugam, S. and Xu, F. and Jampani, R. and Jermaine, C. and Perez, L.L. and Haas, P.J. VLDB 2010

Outline

- Dealing with uncertainty in data management
  - Probabilistic databases
    - Possible world semantics
    - Conjunctive queries
    - Ranking and top-k queries
    - Other queries
  - Beyond expected values – expected utility theory
  - Some tools out there that may be useful (with applications)
    - Uncertainty resolution
    - Portfolio theory
    - Multi-arm bandit

Ranking over Probabilistic Databases

- Our goal: support "ranking" or "top-*k*" query processing
  - Deciding which apartments to inquire about
  - Selecting a set of sensors to "probe"
  - Choosing a set of stocks to invest in
  - …
- How? Choose tuples with large scores? Or tuples with higher probabilities?
  - A complex trade-off

Top-*k* Query Processing

Score values are used to rank the tuples in every *pw*.

ID	Score	Prob
t_1	200	0.2
t_2	150	0.8
t_3	100	0.4

A probabilistic table
(assume tuple-independence)

pw1

ID	Score
t_1	200
t_2	150
t_3	100

w.p. 0.064

pw2

ID	Score
t_1	200
t_2	150

w.p. 0.096

pw3

ID	Score
t_2	150
t_3	100

w.p. 0.256

The top-1 answer for each possible world

Top-*k* Queries: Many Prior Proposals

- Return *k* tuples *t* with the highest *score(t)Pr(t)* **[exp. score]**
- Returns the most probable top *k*-answer **[U-top-k]**

  [Soliman et al. ICDE'07]
- At rank *i*, return tuple with max. prob. of being at rank *i* **[U-rank-k]**

  [Soliman et al. ICDE'07]
- Return *k* tuples *t* with the largest $Pr\ (r(t) \leq k)$ values **[PT-k/GT-k]**

  [Hua et al. SIGMOD'08] [Zhang et al. EDBT'08]
- Return *k* tuples *t* with smallest **expected rank**: $\sum_{pw} Pr(pw)\ r_{pw}(t)$

  [Cormode et al. ICDE'09]
- Return k tuples t with expected score of best available tuple **[k-selection]** [Liu et al. DASFAA'10]

Top-*k* Queries: Many Prior Proposals

- Probabilistic Threshold (PT-k/GT-k) [Hua et al. SIGMOD'08] [Zhang et al. EDBT'08]
  - Return *k* tuples *t* with the largest $Pr(r(t) \leq k)$ values

ID	Score	Prob
t_1	200	0.2
t_2	150	0.8
t_3	100	0.4

Possible worlds	Prob
t_1, t_2, t_3	0.064
t_1, t_2	0.096
t_1, t_3	0.016
t_2, t_3	0.256
t_1	0.024
t_2	0.384
t_3	0.064
Φ	0.096

K=2

ID	Prob(r(t)≤2)
t_1	0.2
t_2	0.8
t_3	0.336

Ranking: t_2, t_3, t_1

Top-*k* Queries

- Which one should we use???
- Comparing different ranking functions

Normalized Kendall Distance between two top-k answers:

Penalizes #reversals and #mismatches

Lies in [0,1], 0: Same answers; 1: Disjoint answers

	E-Score	PT/GT	U-Rank	E-Rank	U-Top
E-Score	----	0.124	0.302	0.799	0.276
PT/GT	0.124	----	0.332	0.929	0.367
U-Rank	0.302	0.332	-----	0.929	0.204
E-Rank	0.799	0.929	0.929	----	0.945
U-Top	0.276	0.367	0.204	0.945	----

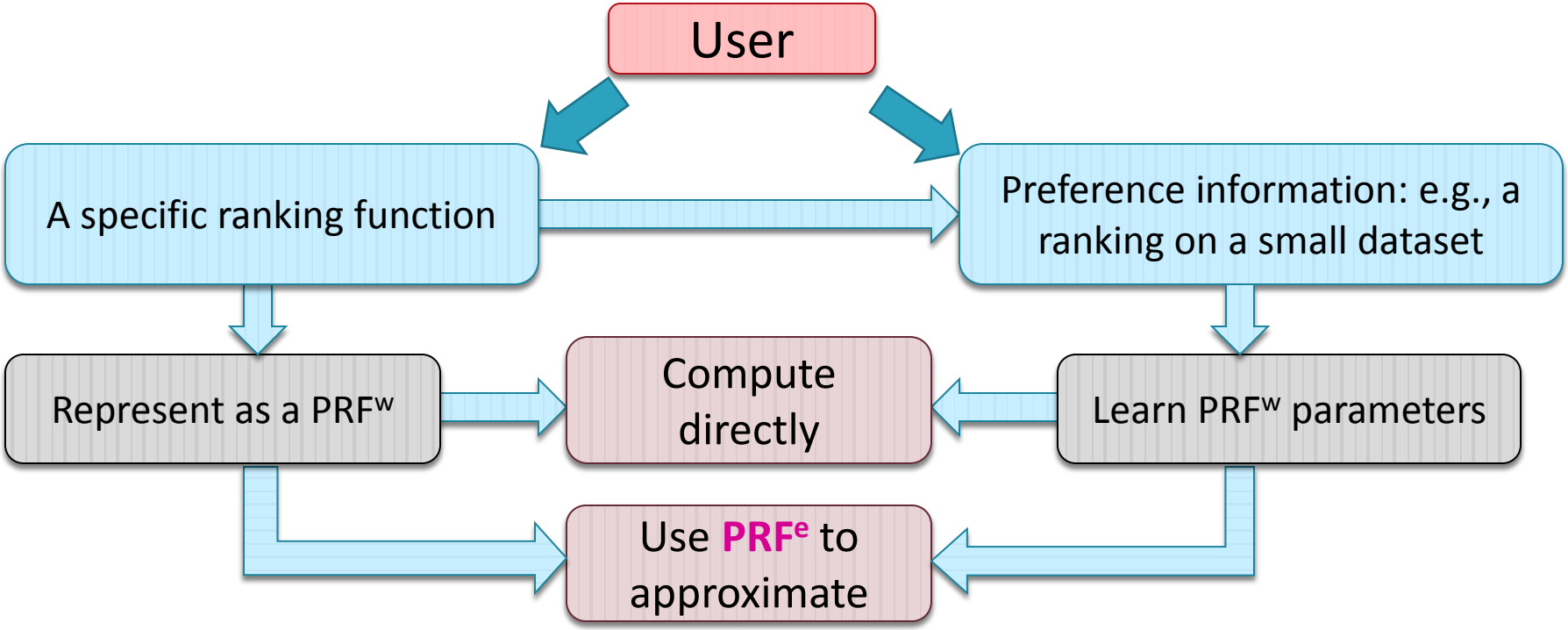
Real Data Set: 100,000 tuples, Top-100

	E-Score	PT/GT	U-Rank	E-Rank	U-Top
E-Score	----	0.864	0.890	0.004	0.925
PT/GT	0.864	----	0.395	0.864	0.579
U-Rank	0.890	0.395	-----	0.890	0.316
E-Rank	0.004	0.864	0.890	----	0.926
U-Top	0.925	0.579	0.316	0.926	----

Synthetic Dataset: 100,000 tuples, Top-100

A Unified Approach

- Define two *parameterized* ranking functions: PRF^w; PRF^e
  - .. that can simulate or approximate a variety of ranking functions
  - PRF^e much more efficient to evaluate (than PRF^w)



A unified approach to ranking in probabilistic databases, Li, J. and Saha, B. and Deshpande, A. , VLDB 2009

Parameterized Ranking Function

PRF$^{\omega}$(h): Weight Function : ω : rank$\rightarrow$ C

$$\Upsilon_\omega(t) = \sum_{i=1}^{h} \omega(i) \cdot \Pr(r(t) = i).$$

Positional probability:
Probability that t *is ranked at position* i

PRF^{e}(α): $\omega(i)=\alpha^i$ where α *can be a real or a complex*

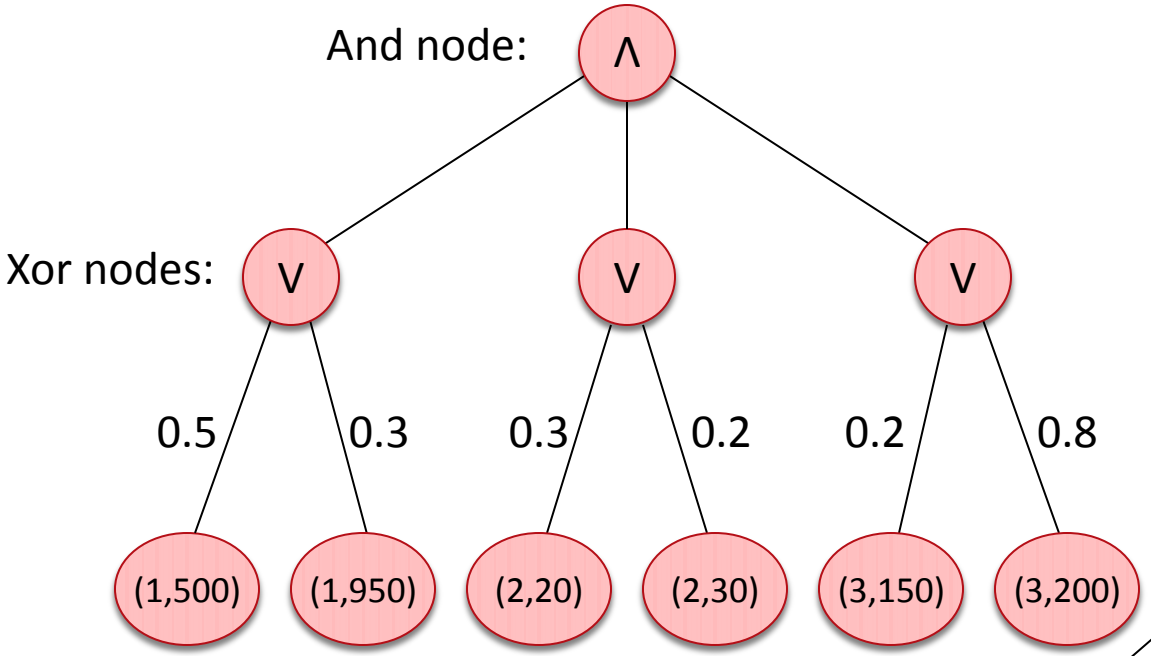
$$\Upsilon_\omega(t) = \sum_{i\geq 1} \alpha^i \cdot \Pr(r(t) = i).$$

Return *k* tuples with the highest $|\Upsilon_\omega|$ values.

- E.g., ω(i)= 1 : Rank the tuples by probabilities
- E.g., ω(i)=1 for 1≤i≤k, ω(i)=0 for i>k: PT-k (i.e., ranking by $Pr(r(t)\leq k)$)
- Generalizes PT/GT-k, *U-Rank, E-Rank*
- We can easily incorporate the score as an feature

Probabilistic And/Xor Trees

- Capture two types of correlations: mutual exclusivity and coexistence.
- Generalize x-tuples which can model only mutual exclusivity

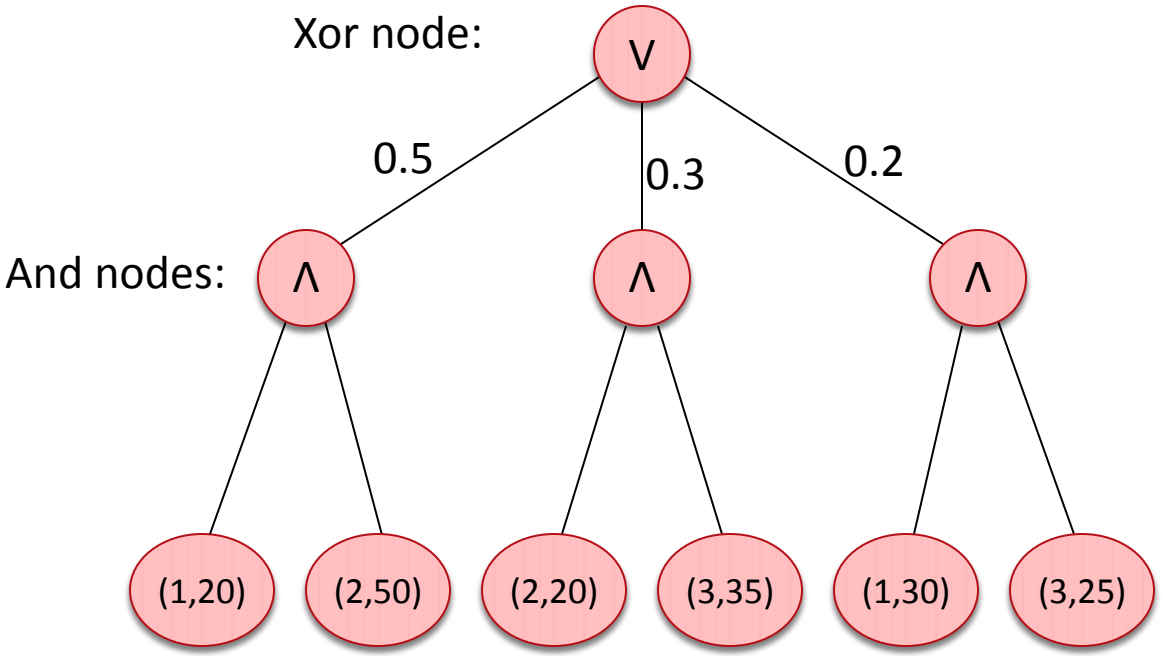


Possible Worlds	Pr
(3,150)	**0.02**
(3,200)	0.08
…….	
(1,500);(2,20);(3,150)	0.03
(1,950);(2,20);(3,150)	0.018
…….	

(1-0.5-0.3)*(1-0.3-0.2)*0.2=0.02

Probabilistic And/Xor Trees

- And/Xor trees can represent any finite set of possible worlds (not necessarily compact).



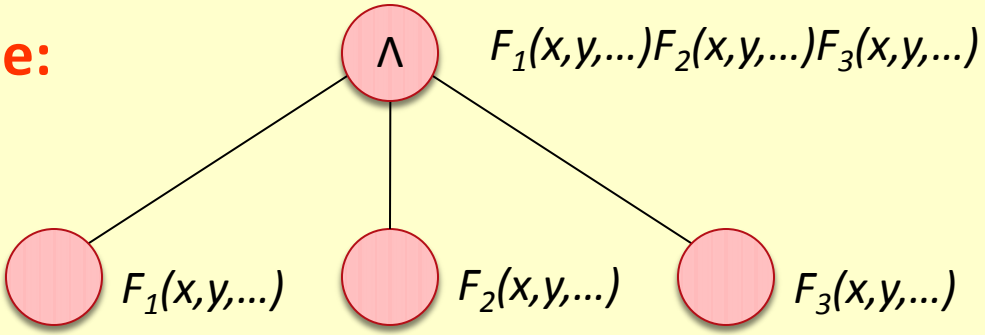
Possible Worlds	Pr
(1,20);(2,50)	0.5
(2,20);(3,35)	0.3
(1,30);(3,25)	0.2

Computing Probabilities on And/Xor Trees

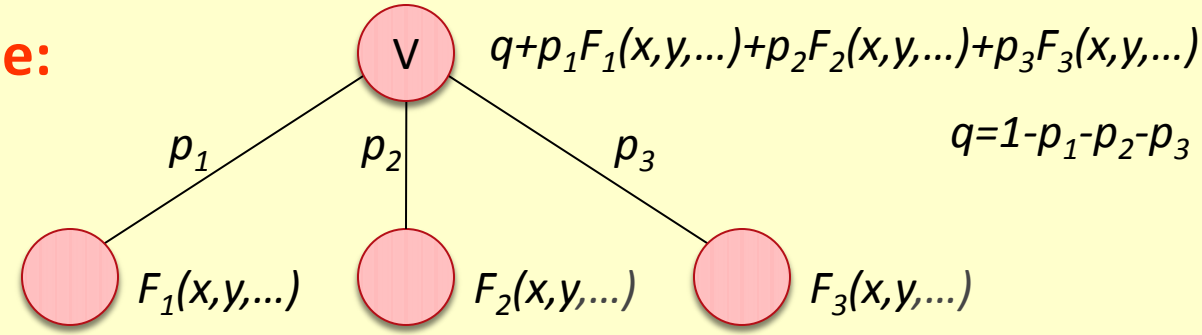
Generating Function Method:

Leaves: x y x z

And Node:



Xor Node:



Computing Probabilities on And/Xor Trees

Generating Function Method:

Root:  $F(x,y,\ldots)=\sum_{ij\ldots} c_{ij\ldots} x^i y^j \ldots$

THM: The coefficient $c_{ij\ldots}$ of the term $x^i y^j \ldots$

= total prob. of the possible worlds which contain

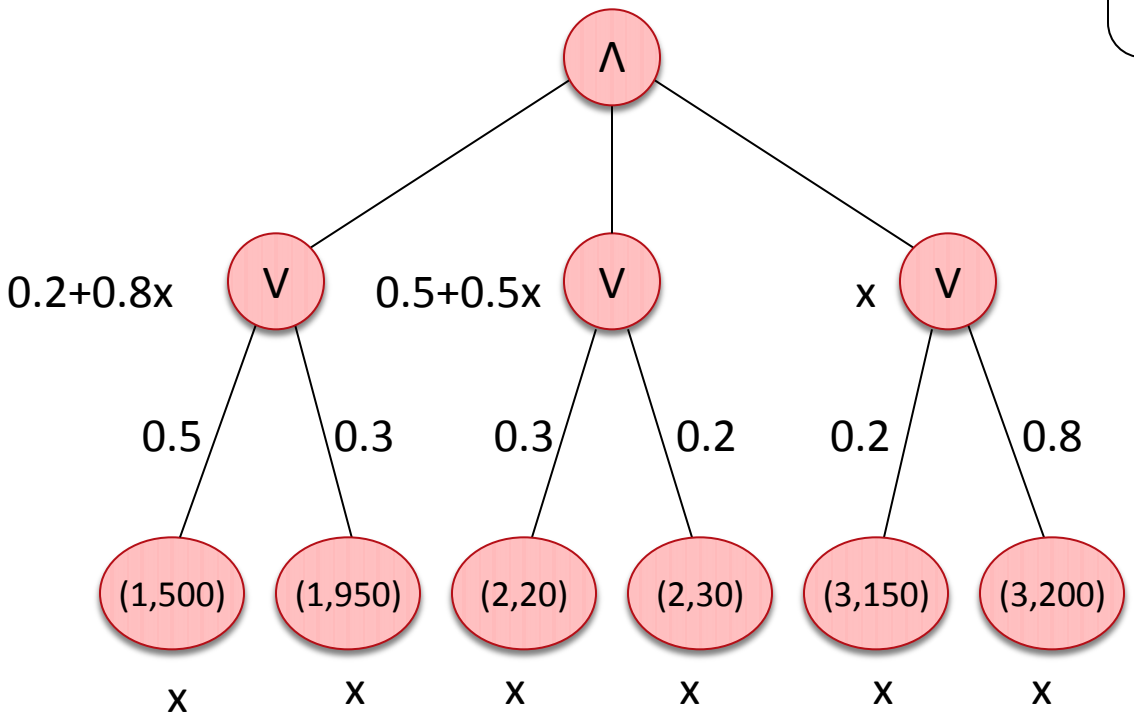
- i tuples annotated with x,
- j tuples annotated with y,……

Computing Probabilities on And/Xor Trees

Example: Computing the prob. dist. of the size of the pw

$(0.2+0.8x)(0.5+0.5x)x = 0.4\,x^3+0.5\,x^2+0.1\,x$ ⟹

Pr(|pw|=3)=0.4
Pr(|pw|=2)=0.5
Pr(|pw|=1)=0.1



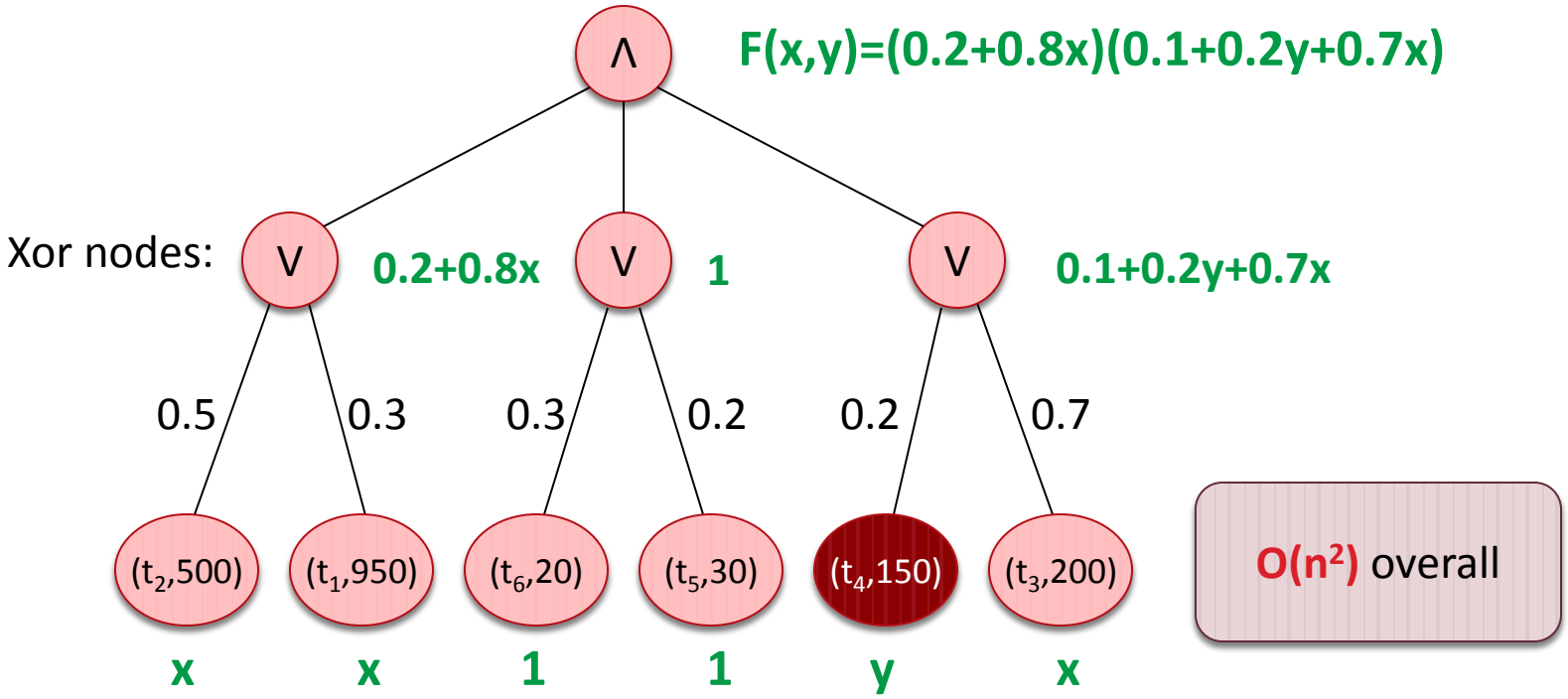
Computing PRF: And/Xor Trees

Construct generating function for t_4

$r(i)=j$ if and only if (1) $j-1$ tuples with higher scores appear

(2) tuple i appears

$Pr(r(t_4)=j)$ = coeff of $x^{j-1}y$

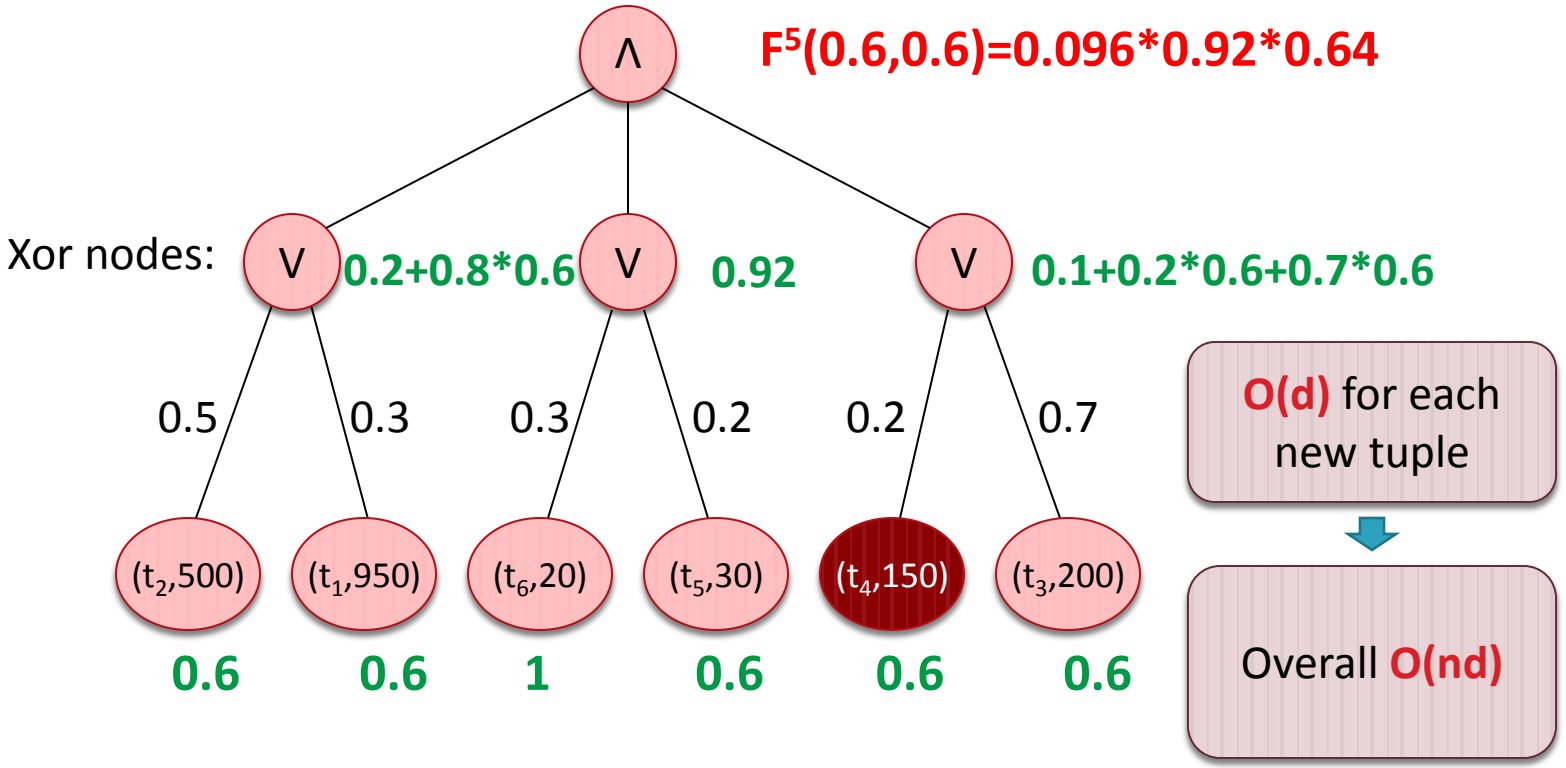


Computing PRF$^e(\alpha)$: And/Xor Trees

$$\Upsilon(t_i) = \mathcal{F}^i(\alpha, \alpha) - \mathcal{F}^i(\alpha, 0).$$

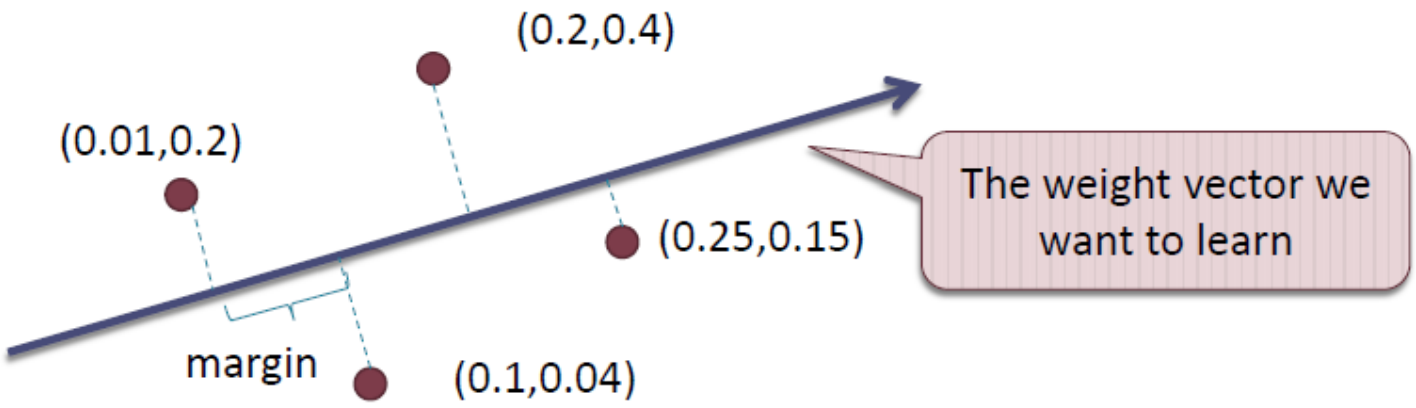
We maintain only the numerical values of $F^i(\alpha,\alpha)$ and $F^i(\alpha,0)$ at each node.

E.g., α=0.6. Now we want to compute **$F^5(0.6,0.6)$**



Learn the weight

- We can learn the weight from user feedbacks.
  - A feedback can be a total or partial ordering of the tuples.
- Use the positional probabilities as the features.
  - Feature vector: *{Pr(r(t)=1), Pr(r(t)=2), Pr(r(t)=3),…}.*
- Use Ranking-SVM to learn the weight.
  - Maximize the margin.



Summary of Results

$PRF^w(h)$:

- Independent tuples: $O(nh+n\log n)$
  - Previous results for U-Rank: $O(n^2h)$ [Soliman et al. ICDE'07], $O(nh+n\log n)$ [Yi et al. TKDE'09]
  - Previous results for PT-k: $O(nh+n\log n)$ [Hua et al. SIGMOD'08]
- And/Xor trees: $O(dnh+n\log n)$ (d is the height of the tree, d=2 for x-tuples)
  - Previous results for U-Rank over x-tuples: $O(n^2h)$ [Soliman et al. ICDE'07], $O(n^2h)$ [Yi et al. TKDE'09]
  - Previous results for PT-k over x-tuples: $O(n^2h)$ [Hua et al. SIGMOD'08]

PRF^e:

- Independent tuples: $O(n\log n)$
- And/Xor trees: $O(nd+n\log n)$

Outline

- Dealing with uncertainty in data management
  - Probabilistic databases
    - Possible world semantics
    - Conjunctive queries
    - Ranking and top-k queries
    - Other queries
  - Beyond expected values – expected utility theory
  - Some tools out there that may be useful (with applications)
    - Uncertainty resolution
    - Portfolio theory
    - Multi-arm bandit

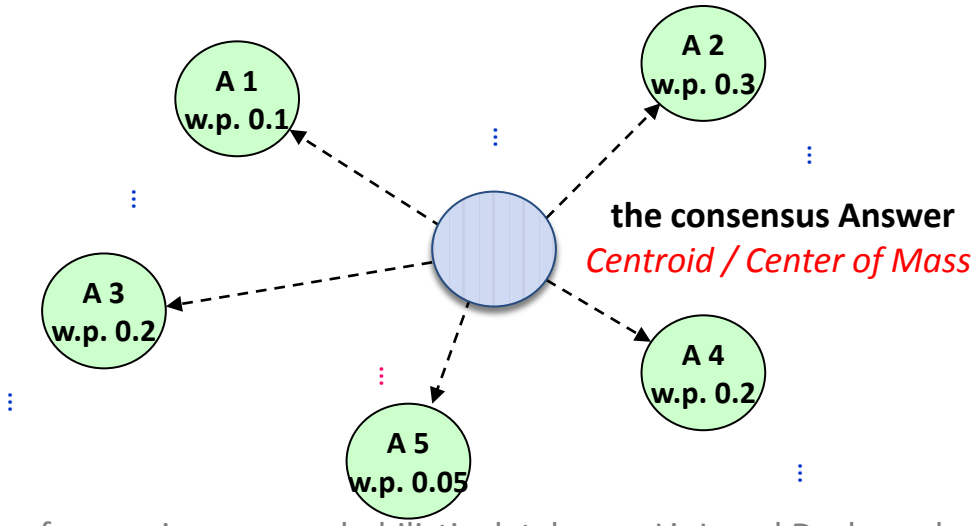
Consensus Answer

Consensus Answer:

- Think of each possible answers as a point in the space.
- Suppose d() is a distance metric between answers.
- Consensus Answer is a single deterministic answer

$$\tau = \arg\min_{\tau' \in} \{\mathbb{E}[d(\tau', \tau_{pw})]\}$$

where τ_{pw} is the answer for the possible world pw



Consensus answers for queries over probabilistic databases, Li, J. and Deshpande, A., PODS, 2009

Consensus Answer

- **Consensus Answer:**
- We show that PT-k is equivalent to Consensus-Top-k under symmetric difference $T_1 \Delta\, T_2 = (T_1 \backslash T_2) \cup (T_2 \backslash T_1)$
- More generally, PRFw is equivalent to Consensus-Top-k under weighted symmetric difference

- We can use the framework for other types of queries, such as aggregate queries, clustering

Outline

- Dealing with uncertainty in data management
  - Probabilistic databases
    - Possible world semantics
    - Conjunctive queries
    - Ranking and top-k queries
    - Other queries
  - Beyond expected values – expected utility theory
  - Some tools out there that may be useful (with applications)
    - Uncertainty resolution
    - Portfolio theory
    - Multi-arm bandit

Aggregate Queries

- *Aggregate Query:*

Item	Forecaster	Profit	*P*
Widget	Alice	$–99K	0.99
	Bob	$100M	0.01
Whatsit	Alice	$1M	1

`Profit`(Item;Forecaster,Profit;*P*)

```
SELECT SUM(PROFIT)
FROM PROFIT
WHERE ITEM='Widget'
```

(a) Expectation Style

```
SELECT ITEM
FROM PROFIT
WHERE ITEM='Widget'
HAVING SUM(PROFIT) > 0.0
```

(b) `HAVING` Style

Answer: E[profit]=19.9K

Answer: Prob[profit>0]=0.01

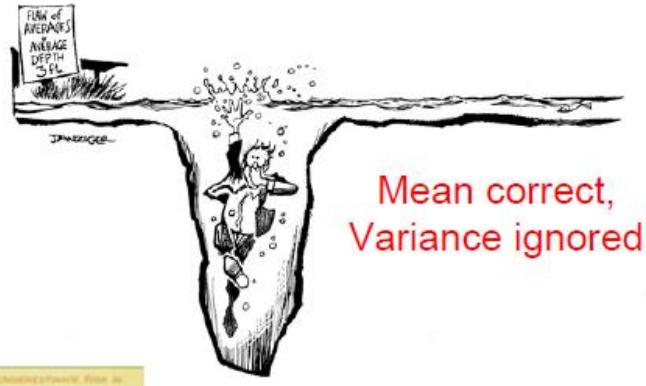
Expected value may not be sufficient!

Example taken from The trichotomy of HAVING queries on a probabilistic database, Re, C. and Suciu, D., The VLDB Journal, 2009

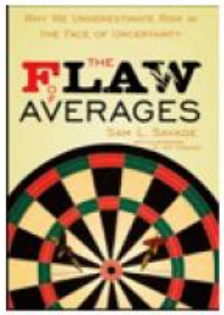
Inadequacy of Expected Value

- Be aware of risk!

Flaw of averages (weak form):



Mean correct,
Variance ignored



Flaw of averages (strong form):

Wrong value of mean:

$f(E[X]) \neq E[f(X)]$

Inadequacy of Expected Value

- Inadequacy of expected value:
  - Unable to capture risk-averse or risk-prone behaviors
    - Action 1: $100 VS Action 2: $200 w.p. 0.5; $0 w.p. 0.5
    - Risk-averse players prefer Action 1
    - Risk-prone players prefer Action 2 (e.g., a gambler spends $100 to play Double-or-Nothing)
  - St. Petersburg paradox
    - You pay x dollars to enter the game
      - Repeatedly toss a fair coin until a tail appears
      - payoff=2^k where k=#heads
    - How much should x be?
      - Expected payoff =1x(1/2)+2x(1/4)+4x(1/8)+……=
      - Few people would pay even $25 [Martin '04]

Expected Utility Maximization Principle

A : The set of valid answers

$w_{pw}(a)$: the cost of answer in pw

u: $R \rightarrow R$: the utility function

Expected Utility Maximization Principle:
The most desirable answer *a* is the answer that max. the exp. utility, i.e.,

$$a = \max_{a' \in A} \mathrm{E}_{pw}[\mu(w_{pw}(a'))]$$

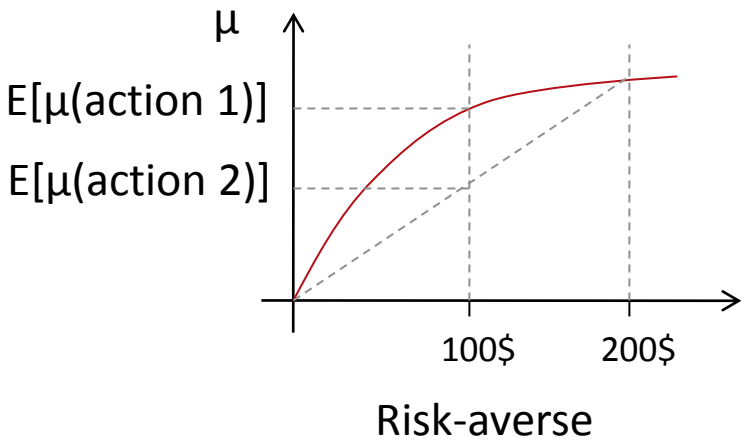
Von Neumann and Morgenstern provides an *axiomitization* of the principle (known as von Neumann-Morgenstern expected utility theorem).

Expected Utility Maximization Principle

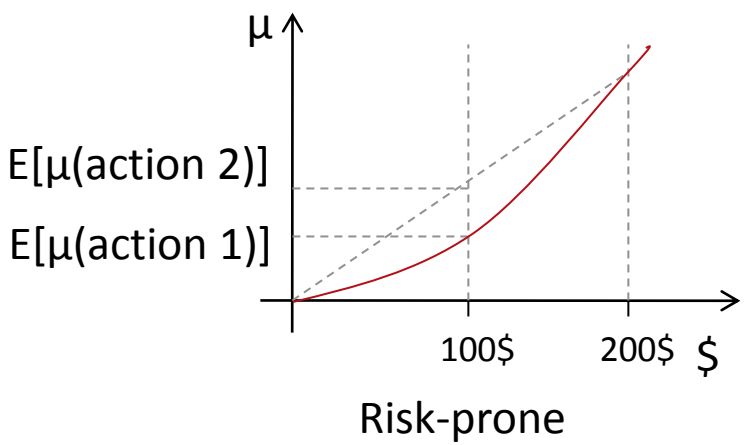
u: $R \to R$: The utility function: profit-> utility

Expected Utility Maximization Principle: the decision maker should choose the action that maximizes the **expected utility**

- Action 1: \$100
- Action 2: \$200 w.p. 0.5; \$0 w.p. 0.5



Risk-averse



Risk-prone

- Von Neumann and Morgenstern provides an *axiomitization* of the principle (known as von Neumann-Morgenstern expected utility theorem).

Expected Utility Maximization Principle

A : The set of valid answers

Think $\mathcal{A}$ as the set of tuples

$w_{pw}(a)$: the cost of answer in pw

Think $w_{pw}(a)$ as the rank for tuple a in pw

u: $R \rightarrow R$: the utility function

Expected Utility Maximization Principle:

The most desirable answer a is the answer that max. the exp. utility, i.e.,

$$a = \max_{a' \in A} \mathrm{E}_{pw}[\mu(w_{pw}(a'))]$$

This gives us PRFw!

Expected Utility Maximization Principle

A : The set of valid answers

Think -$\mu\left(w_{pw}(a)\right)$ as the distance/dissimilarity between answer *a* and the actual answer for *pw*

Expected Utility Maximization Principle:

The most desirable answer *a* is the answer that max. the exp. utility, i.e.,

$$a = \max_{a' \in A} \mathrm{E}_{pw}[\mu\left(w_{pw}(a')\right)]$$

This gives us Consensus Answer!

Prob. DB Research

- Many works on handling more general correlation – incorporating graphical models: E.g.,
  - Representing and Querying Correlated Tuples in Probabilistic Databases, Prithviraj Sen, Amol Deshpande, ICDE 2007
  - Indexing Correlated Probabilistic Databases, Bhargav Kanagal, Amol Deshpande, SIGMOD 2009
  - Scalable probabilistic databases with factor graphs and mcmc, Wick, M. and McCallum, A. and Miklau, G., VLDB, 2010
- Other works
  - Probabilistic streams: E.g.,
    - Estimating statistical aggregates on probabilistic data streams, Jayram, TS and McGregor, A. and Muthukrishnan, S. and Vee, TODS 2008
    - Sketching probabilistic data streams, Cormode, G. and Garofalakis, M., SIGMOD 2007.
  - Probabilistic graphs, E.g.,
    - K-nearest neighbors in uncertain graphs, Potamias, M. and Bonchi, F. and Gionis, A. and Kollios, G. PVLDB, 2010
    - Distance-Constraint Reachability Computation in Uncertain Graphs. Ruoming Jin, Lin Liu, Bolin Ding, Haixun Wang , PVLDB, 2011
  - Other operators, such as probabilistic skylines. E.g.,
    - Probabilistic skylines on uncertain data, Pei, J. and Jiang, B. and Lin, X. and Yuan,Y. , VLDB, 2007
  - Sensitivity analysis: E.g.,
    - Sensitivity analysis and explanations for robust query evaluation in probabilistic databases. Bhargav Kanagal, Jian Li, Amol Deshpande. SIGMOD, 2011

Prob. DB Research

- Our strength: support declarative queries, query processing and optimization techniques (indexing etc.).

- Current issues
  - Independence assumption.
  - Expressiveness/scalability trade off.
  - Different existing prototypes excels at different aspects (but not all).
  - Semantics not rich enough (need to go beyond expected values and probabilistic thresholds).

Outline

- Dealing with uncertainty in data management
  - Probabilistic databases
    - Possible world semantics
    - Conjunctive queries
    - Ranking and top-k queries
    - Other queries
  - Beyond expected values – expected utility theory
  - Some tools out there that may be useful (with applications)
    - Uncertainty resolution
    - Portfolio theory
    - Multi-arm bandit

Uncertainty Resolution

- Reduce the level of uncertainty by conducting extra experiments
  - E.g., let a human to recognize the characters to clean the uncertain OCR data
- A typical problem
  - A set of random variables $x_1, x_2, \ldots\ldots, x_n$
  - Resolved the uncertainty of each x_i costs c_i
  - We have a budget *C*
  - Goal: Estimate some function $f(x_1, x_2, \ldots\ldots, x_n)$
  - E.g. *f*= min or max or other aggregate function or even some combinatorial optimization problems

Asking the right questions: Model-driven optimization using probes, Goel, A. and Guha, S. and Munagala, K., PODS 2006

How to probe for an extreme value, Goel, A. and Guha, S. and Munagala, K., ACM Transactions on Algorithms, 2010

Adaptive Uncertainty Resolution in Bayesian Combinatorial Optimization Problems, Guha, S. and Munagala, K. ACM Transactions on Algorithms, 2012

Uncertainty Resolution

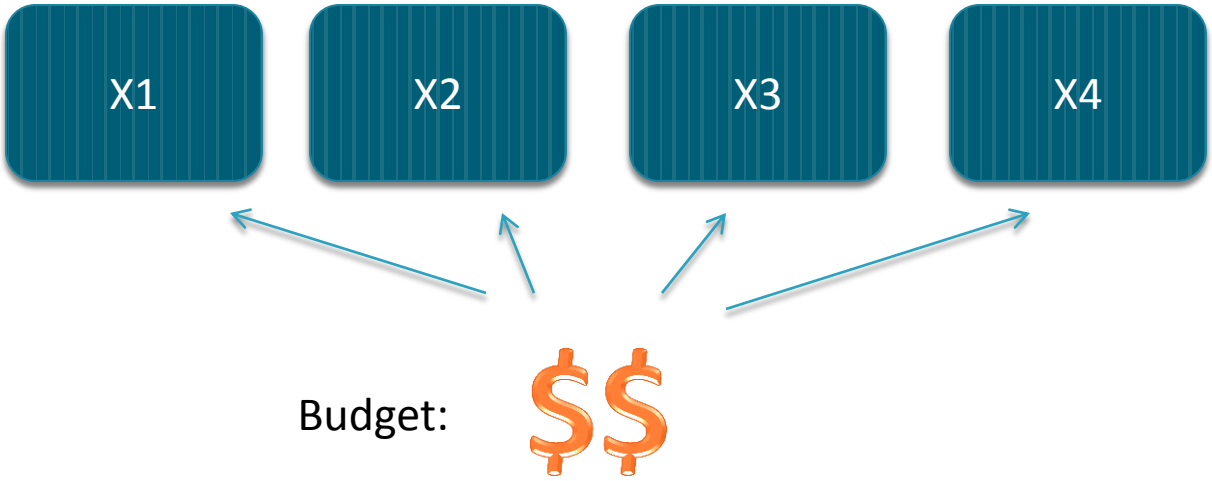
- Another typical problem
  - A set of random variables $x_1, x_2, \ldots\ldots, x_n$
  - Resolved the uncertainty of each x_i costs c_i
  - We have a budget *C*
  - Goal: (Estimate some function $f(x_1, x_2, \ldots\ldots, x_n)$?) But we don't know what function *f* we will use, or we may need to estimate a lot of functions.
  - How? Reduce "the level of uncertainty" (measured using Entropy or Variance)
    - Quite nontrivial if the random variables are correlated.
    - Such problems are connected to the area of stochastic optimization, submodular optimization, statistical experiment design.
    - A large body of literature.

Outline

- Dealing with uncertainty in data management
  - Probabilistic databases
    - Possible world semantics
    - Conjunctive queries
    - Ranking and top-k queries
    - Other queries
  - Beyond expected values – expected utility theory
  - Some tools out there that may be useful (with applications)
    - Uncertainty resolution
    - Portfolio theory
    - Multi-arm bandit

Portfolio Theory

Available securities: Rate of return: random variable!

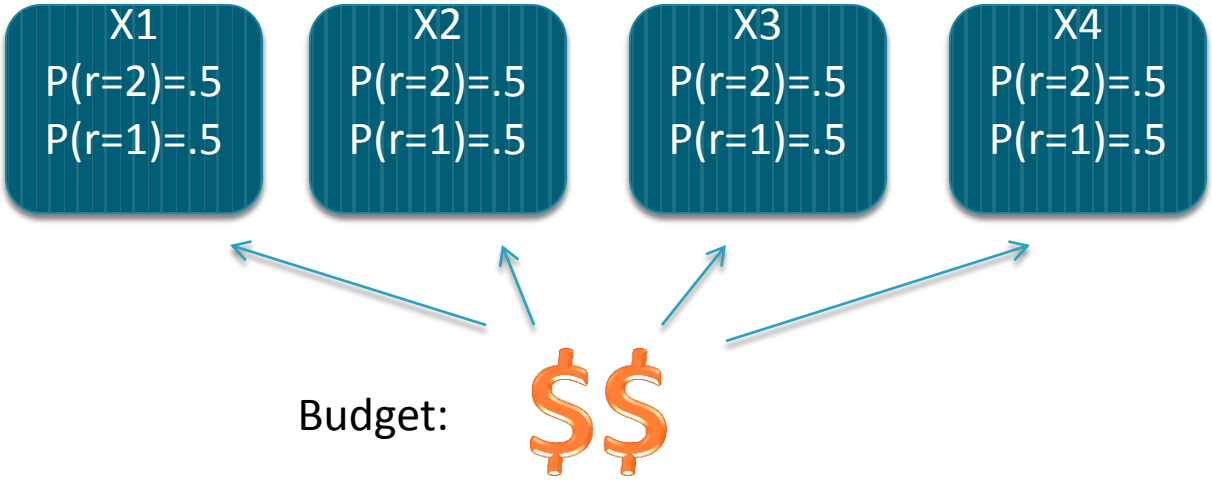


How to invest your money?

Portfolio Theory

Available securities: Rate of return: random variable!

Assume Xi are independent



Two strategy: (assume we have 1$)

(1): Invest 1$ to X1: E[return]=1.5 Var[return]=0.25

(2): Invest .25$ to each Xi E[return]=1.5 **Var[return]=0.25/4=0.0625**

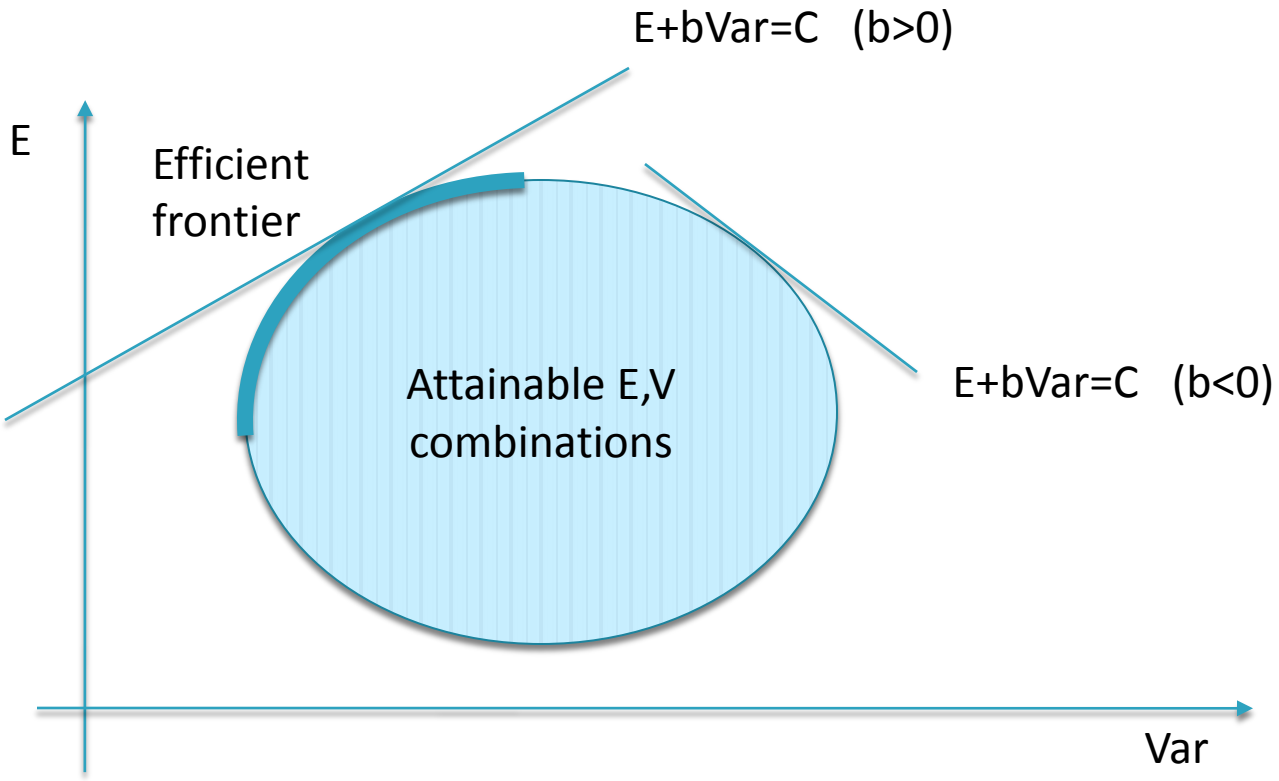
The risk is much smaller

Do not put all your eggs in one basket!

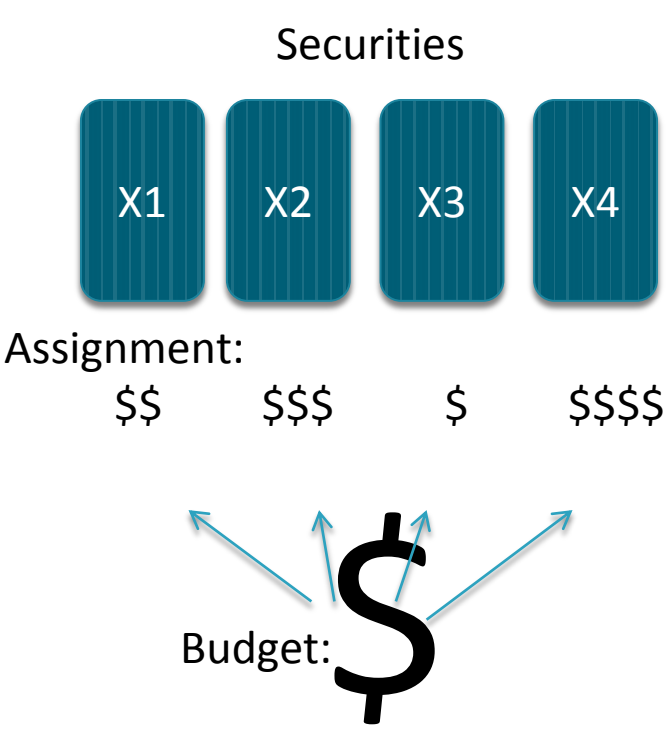
Portfolio Theory

- What to optimize?
  - Maximize E[R]
  - Minimize Var[R]
  - Minimize Var[R], subject to E[R]>=t
  - Maximize E[R], subject to Var[R]<=t
  - Maximize E[R] - b×Var[R]
    - b>0 : risk averse
    - b<0 : risk loving

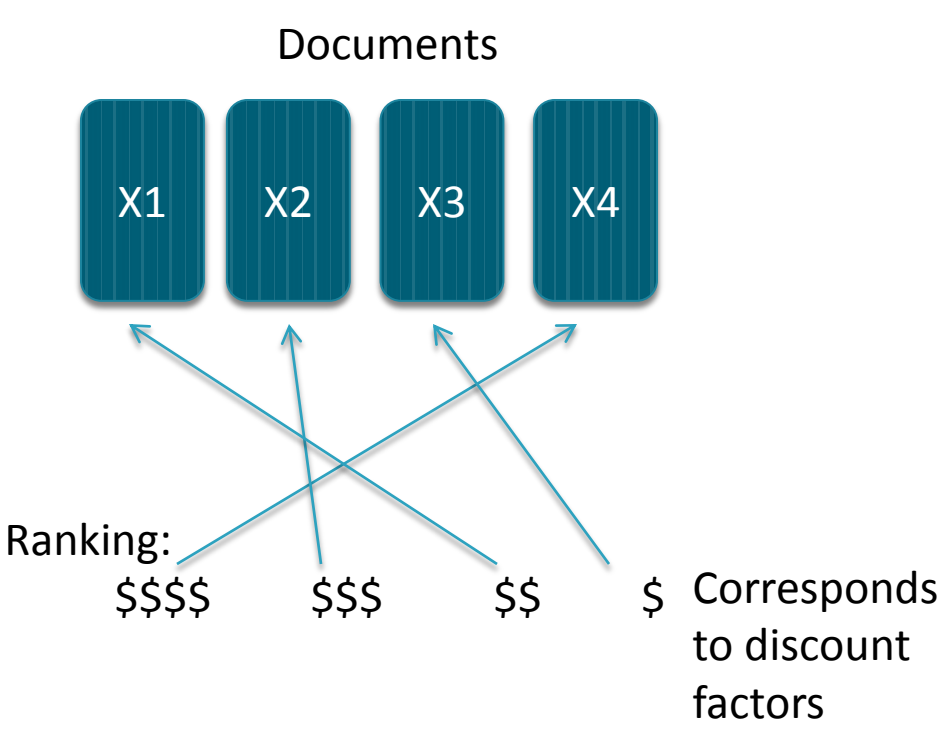
Portfolio Theory



Portfolio Theory in IR



Find an assignment.



Find a ranking.

Portfolio Theory of Information Retrival. Wang and Zhu. SIGIR 09

Outline

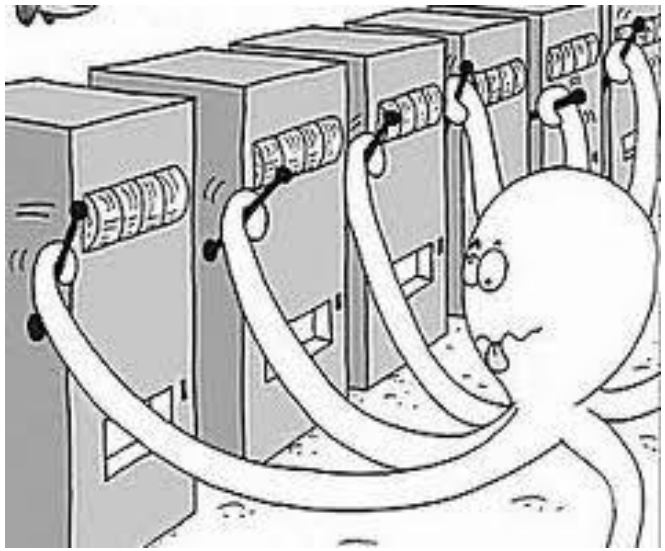
- Dealing with uncertainty in data management
  - Probabilistic databases
    - Possible world semantics
    - Conjunctive queries
    - Ranking and top-k queries
    - Other queries
  - Beyond expected values – expected utility theory
  - Some tools out there that may be useful (with applications)
    - Uncertainty resolution
    - Portfolio theory
    - Multi-arm bandit

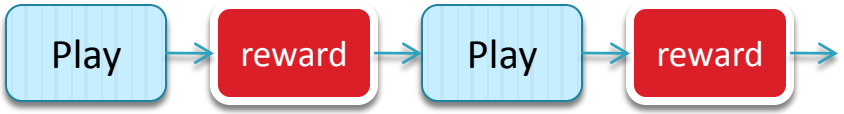
Multi-armed Bandits

- The multi-armed bandits problem

  K gambling machines. Playing machine i yield rewards x_{i1}, x_{i2}, …. which are i.i.d according to some unknown distribution.
  Find a strategy (how to play) to maximize the expected payoff.

  Models the explore and exploit trade-off





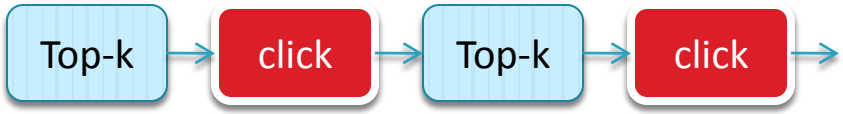
There are strategies that can achieve :

E[payoff up to time T] >= OPT – R(T)
Where R(T)=o(T).

Multi-armed Bandit

An application in ranking documents

- A set of documents $D=\{d_1,\ldots,d_n\}$.
- A population of users.
- Users of type i will click d_j with prob p_{ij}
- Each time the system presents to the user a top-k list.
- The user click the first doc she likes.
- Objective: Maximizes E[#users who click at least once]

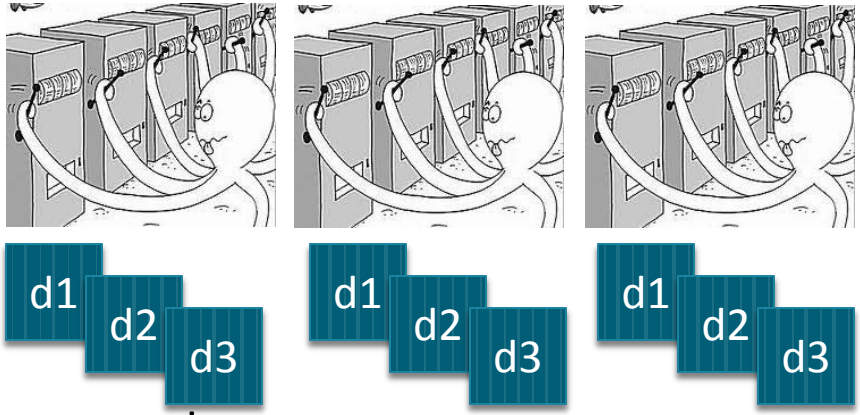


Learning diverse ranking with multi-armed bandits.
Radlinski, Kleinberg and Joachims. ICML08.

Multi-armed Bandit

- For each slot (totally k of them), we run an MAB instance.

Top-k slots:



- Each doc corresponds to an arm.
- If the user click any doc in the list, we get payoff 1.
- There exists a strategy that achieves a expect payoff

of at least $(1-1/e)OPT-O(k\times\sqrt{Tn\ logn})$

Learning diverse ranking with multi-armed bandits. Radlinski, Kleinberg and Joachims. ICML08.

A lot of other applications

e.g., data cleaning Explore or exploit?: effective strategies for disambiguating large databases, Cheng, R. and Lo, E. and Yang, X.S. and Luk, M.H. and Li, X. and Xie, X. VLDB, 2010

Thanks.

Questions/Comments, please send to lijian83@mail.tsinghua.edu.cn