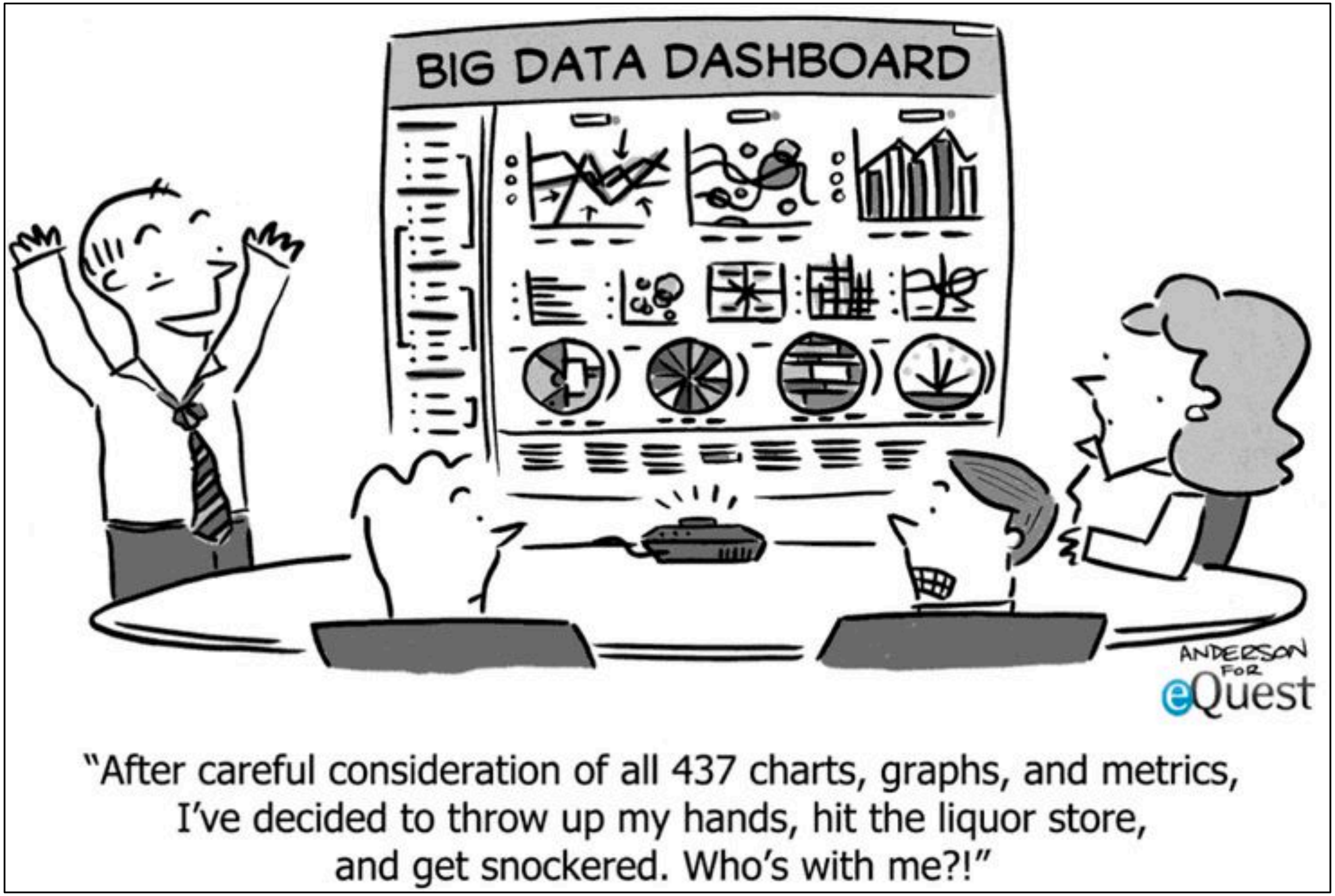
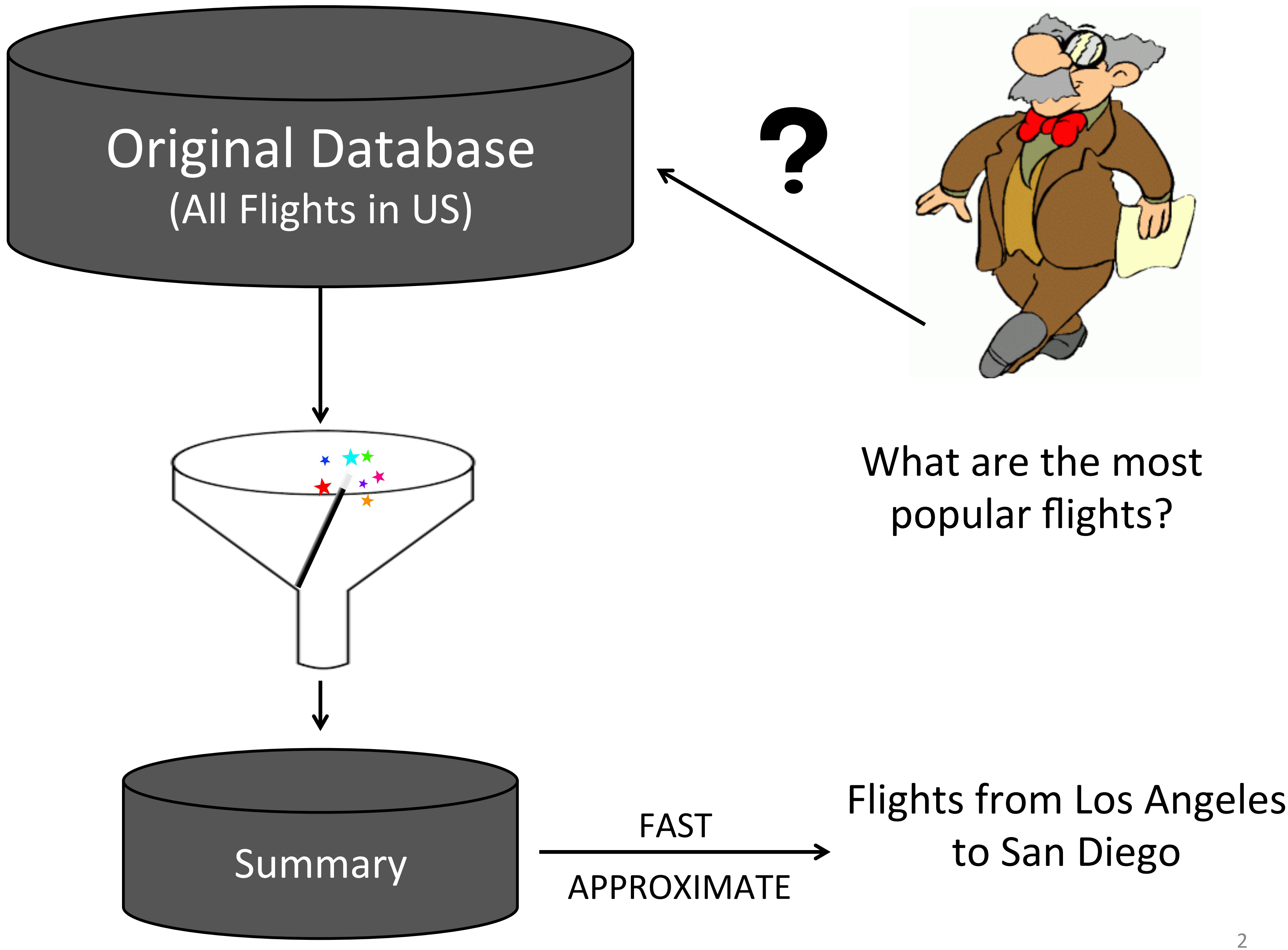


A Probabilistic Approach to Data Summarization

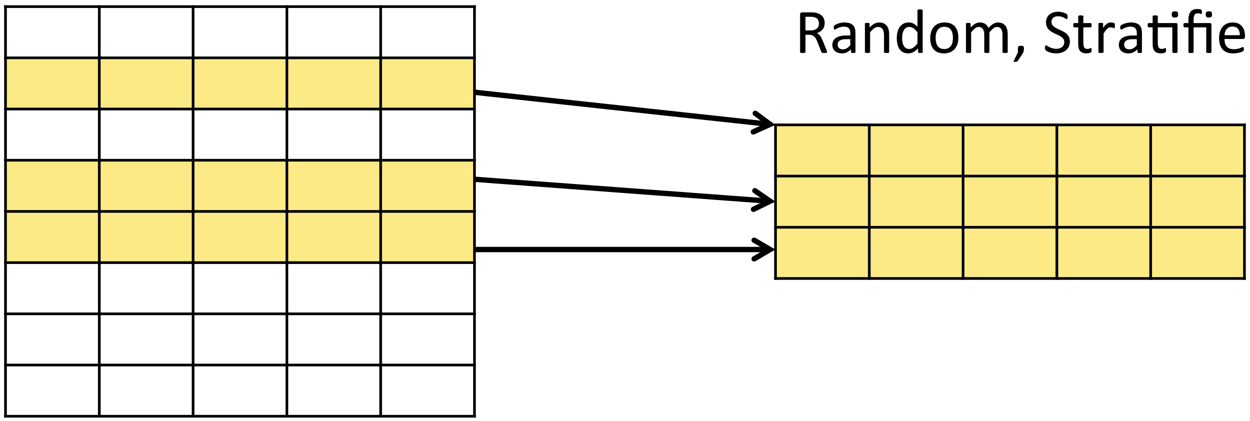
Laurel Orr, Magdalena Balazinska, and Dan Suciu
DB Research Day 2015


Original Database
(All Flights in US)
?
What are the most popular flights?
Summary
FAST
APPROXIMATE
Flights from Los Angeles to San Diego

Existing Summarization Techniques

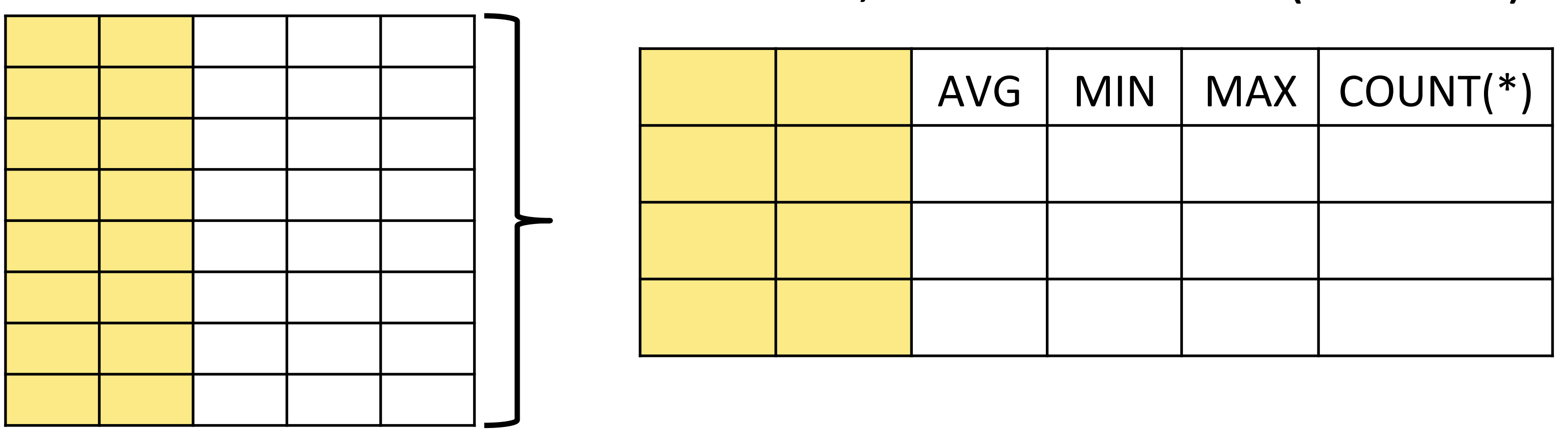
Sampling

Random, Stratified, Weighted,...



Aggregation

Online, Error Bounded (BlinkDB)



Flights (origin, destination, fl_time, …) ~ 2.6 GB

Sampling **Aggregation**

```
SELECT origin, COUNT(*)
FROM Flights
GROUP BY origin;
```

Full Query Time: 20 sec

✓ ✓

```
SELECT *
FROM Flights
WHERE origin=‘SEATTLE, WA’
LIMIT 10;
```

Full Query Time: 0.4 sec

✓ ✗

```
SELECT origin, COUNT(*)
FROM Flights
WHERE dest = ‘LAUREL, MS’
AND fl_time < 120
GROUP BY origin;
```

Full Query Time: 30 sec

✗ ?

IDEA

Find a compact, *probabilistic* representation of our database

Flights with high probability of existence = Popular

By knowing the probability of relations and tuples, we can answer queries probabilistically

The Simplest Summary

Assume there is some concrete relation R(A, B), and you summarized R by its active domain and cardinality.

Given this summary alone, what are the possible relations R could have been (possible worlds of R)?

Possible World Semantics

active domain

A
a_1
a_2

B
b_1
b_2

n = 2

slotted instance

	A	B
id1		
id2		

4

A
a_1
a_1

A
a_1
a_2

A
a_2
a_1

A
a_2
a_2

X

4

B
b_1
b_1

B
b_1
b_2

B
b_2
b_1

B
b_2
b_2

= 16 Possible Instances

$$\sum_{I \in PWD} \Pr(I) = 1 \longrightarrow \Pr(I) = \frac{1}{16}$$

set of all possible instances
(stand for Possible WorlDs)

Possible World Semantics

active domain

A
a_1
a_2

B
b_1
b_2

n = 2

slotted instance

	A	B
id1		
id2		

4

A
a_1
a_1

A
a_1
a_2

A
a_2
a_1

A
a_2
a_2

X

4

B
b_1
b_1

B
b_1
b_2

B
b_2
b_1

B
b_2
b_2

= 16 Possible Instances

$$\Pr((a_1, b_1)) = \sum_{\substack{I \in PWD \\ (a_1, b_1) \in I}} \frac{1}{16} = \frac{7}{16}$$

Tuple Probability

Adding Constraints

active domain

A
a_1
a_2

B
b_1
b_2

n = 100

$|\sigma_{R.A\,=\,a1}(R)| = 70$

$|\sigma_{R.A\,=\,a2}(R)| = 30$

...

$|\sigma_{R.A\,=\,a1\ \wedge\ R.B\,=\,b1}(R)| = 40$

probabilistic instance

$$E[|\sigma_{I.A=a_1}(I)|] = 70$$
$$E[|\sigma_{I.A=a_2}(I)|] = 30$$
$$\ldots$$
$$E[|\sigma_{I.A=a_1 \wedge I.B=b_1}(I)|] = 40$$

$$\sum_{I \in PWD} |\sigma_{I.A=a_1}(I)| \Pr(I) = 70$$
$$\sum_{I \in PWD} |\sigma_{I.A=a_2}(I)| \Pr(I) = 30$$
$$\ldots$$
$$\sum_{I \in PWD} |\sigma_{I.A=a_1 \wedge I.B=b_1}(I)| \Pr(I) = 40$$

How can we solve for Pr(I)?

Principle of Maximum Entropy

The Principle of Maximum Entropy states that subject to prior data, the probability distribution which best represents the state of knowledge is the one that has the largest entropy

In other words, you want to maximize

$$-\sum_{I \in PWD} Pr(I) * \log(Pr(I))$$

over all possible worlds

More Formally

$R(A_1, \ldots, A_m)$, $|R| = n$

D_i = distinct domain of $A_{i,}$

$Tup = \{D_1 \times D_2 \times \ldots \times D_m\}$,

Φ = set of equality predicates ϕ

$$Pr(I) = P^{-n} \prod_{\phi \in \Phi} \alpha_\phi^{|\sigma_{\phi(R)}|}$$

$$P = \sum_{t \in Tup} \prod_{\phi \in \Phi | \phi(t) = true} \alpha_\phi$$

all possible tuples in our active domain

To include constraints on each ϕ

$$s_R(\phi) = |\sigma_\phi(R)| = E[|\sigma_{\phi(I)}|]$$

We can show

$$s_R(\phi) = \frac{n\alpha_\phi P_{\alpha_\phi}}{P}$$

— derivative of P with respect to α_ϕ

To solve, maximize the potential function by gradient descent

$$\Psi = \sum_{\phi \in \Phi} \ln(\alpha_\phi) s_R(\phi) - \ln(P^n)$$

Query Transformation

Aggregates: take expected value

```
SELECT origin, COUNT(*)
FROM Flights
GROUP BY origin;
```

```
SELECT origin, E[|σ_origin(Flights)|]
FROM Flights, alpha_origin,...
WHERE origin=alphas.origin
GROUP BY origin;
```

GROUP BY + COUNT(*)

For each origin o

E[|$\sigma_{\underbrace{\text{origin=o}}_{\phi}}$(Flights)|]

$$E[|\sigma_\phi(I)|] = \frac{n\alpha_\phi P_{\alpha_\phi}}{P}$$

An equation in terms of the α's we have calculated and stored

Optimizations

$$P = \sum_{t \in Tup} \prod_{\phi \in \Phi | \phi(t) = true} \alpha_\phi$$

all possible tuples in
our active domain

1. Factorize P (solve 1D predicates independently)
2. Add relevant 2+D predicates (ex: [A = a1 ^ B = b1])
3. Remove tuples that don't exist

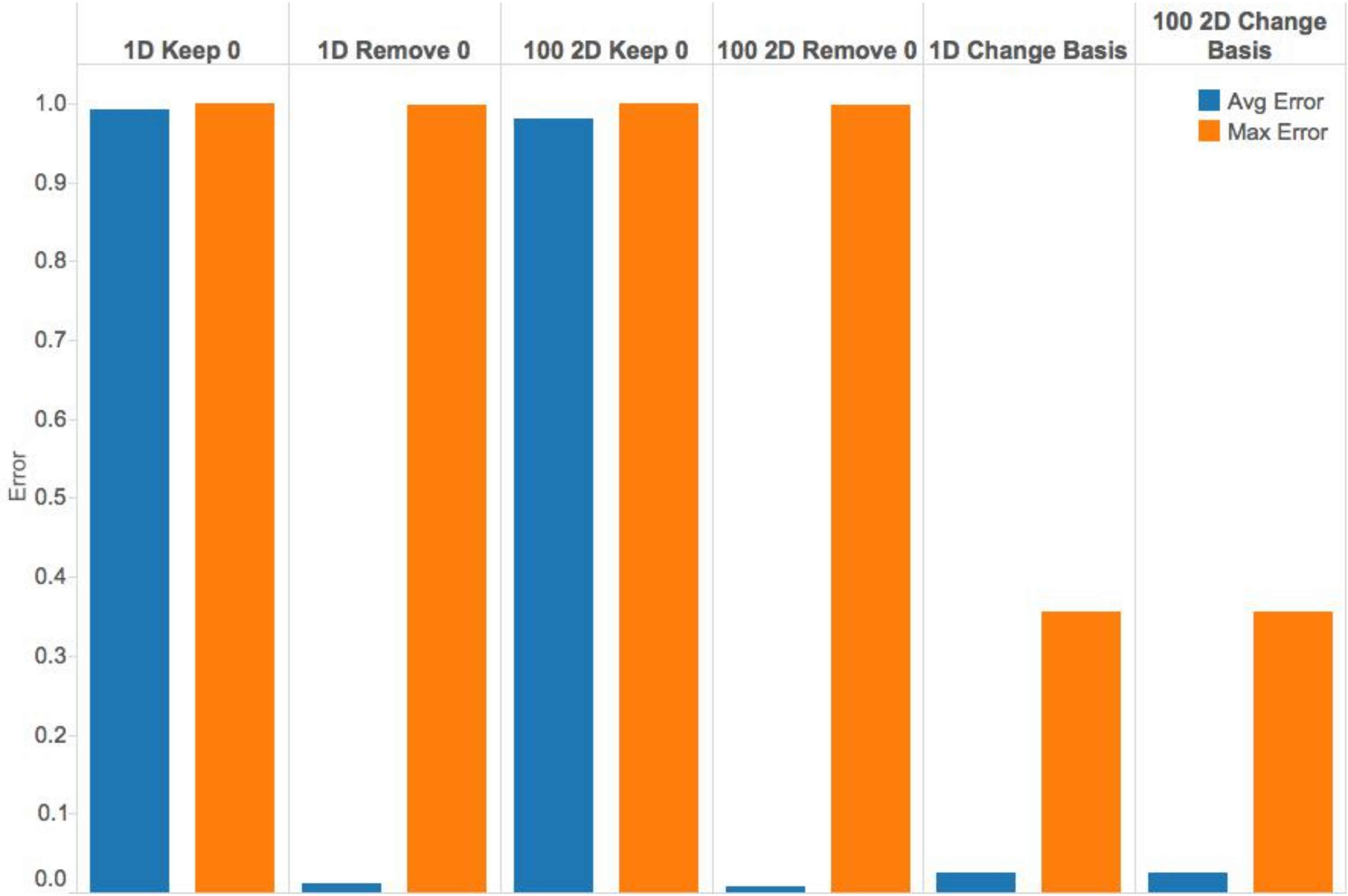
$$P* = P - \sum_{t \in (Tup - R)} \prod_{\phi \in \Phi | \phi(t) = true} \alpha_\phi$$

4. Change Basis (for correlations)
   new attribute AB = f(A, B)
   (ex: AB = A – B)

Experiment with TPC-H

```
SELECT order_date, ship_date, COUNT(*)
FROM orders JOIN lineitem
GROUP BY order_date, ship_date;
```

$$Error = \frac{|Est - True|}{Est + True}$$



Change Basis: order_date – ship_date

Conclusion

- Introduced new way to summarize and approximately query massive datasets
  - Complements sampling and approximate aggregation
- Allows fine grained control over which attributes and values get summarized
- Encouraging preliminary results
- Still need to better address scalability and expand query language
- Need to understand how best to choose statistics