Efficient Computation of Statistical Significance of Query Results in Databases

Vishwakarma Singh, Arnab Bhattacharya, Ambuj K. Singh

University of California, Santa Barbara

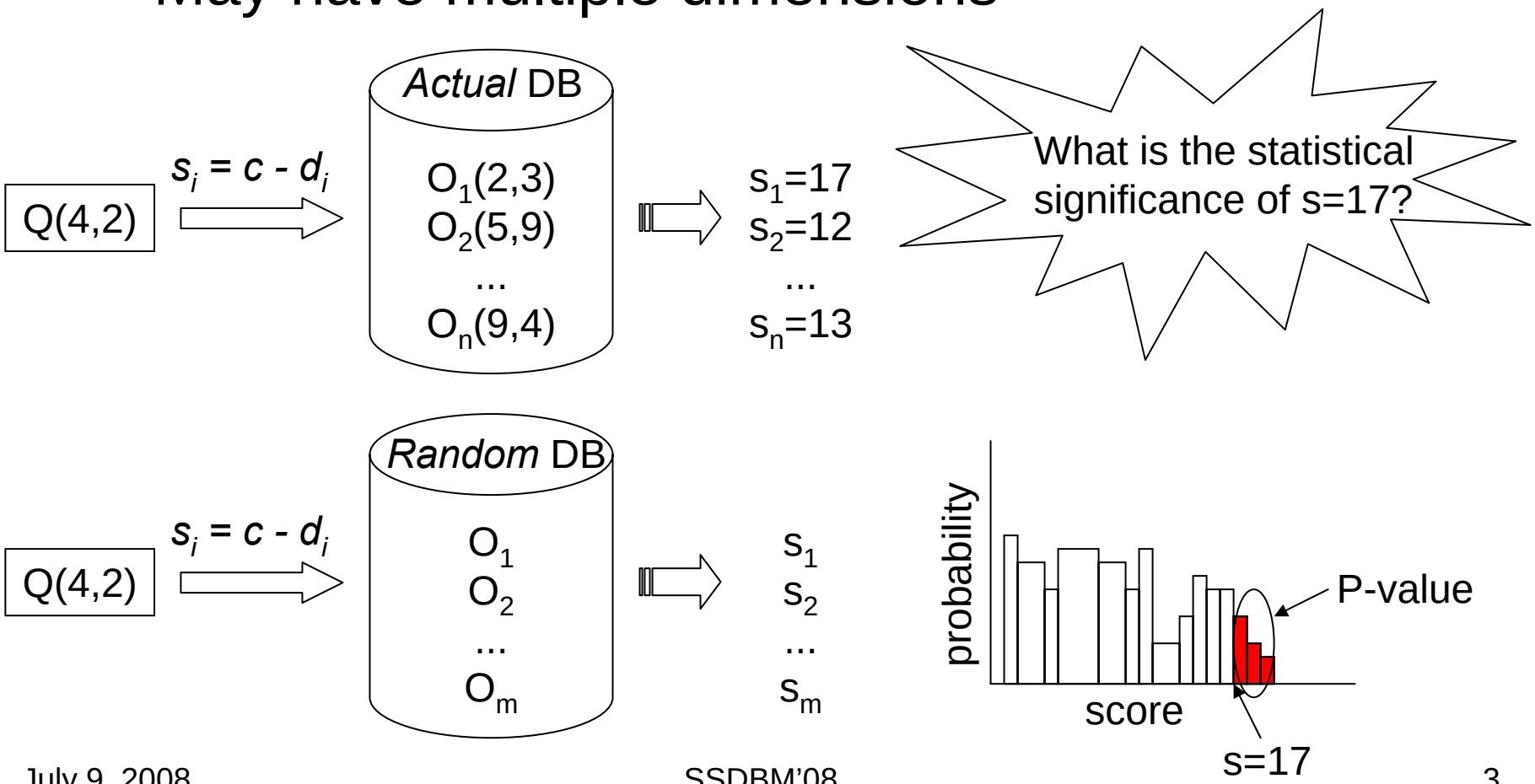
Indian Institute of Technology (IIT), Kanpur

Motivation

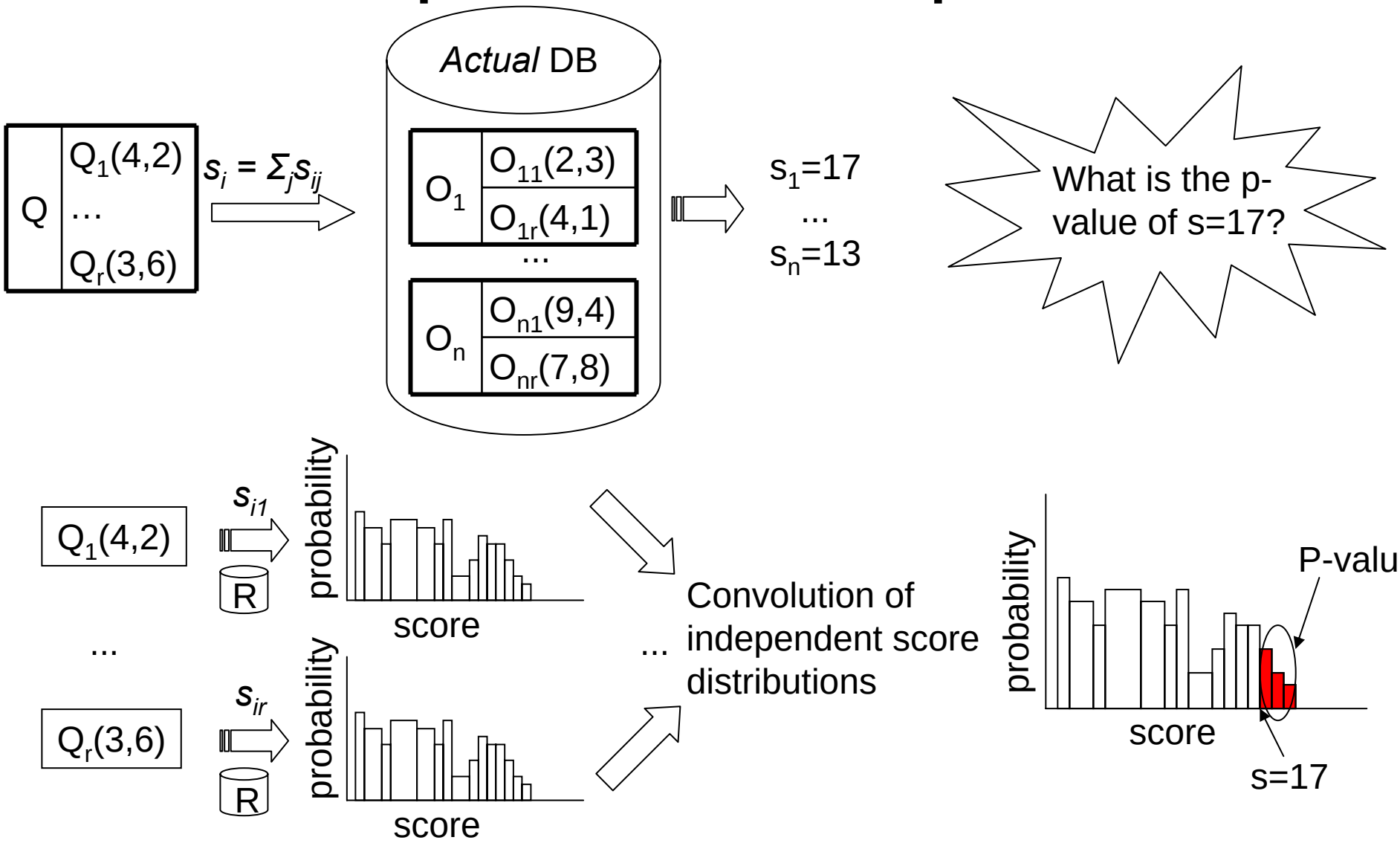
- Query on a database retrieves objects
- Score of result shows similarity with query
  - Useful for ranking
- Not directly useful for
  - Analysis of query and database properties
  - Is the result very likely in the database?
  - Or, is the score *s* an extremely rare occasion?
- Statistical significance
  - Queries with different attributes
  - Score is an aggregate function of attributes

Statistical significance of query results

- Single-attribute query
  - May have multiple dimensions



Multiple-attribute queries



How to *efficiently* compute p-value?

- Generating score distributions for each query component is expensive
- Random DB model is expensive
  - All possible combinations of all individual components in the database
- Convolution has *quadratic* complexity

Score	*Prob.*
1	0.1
2	0.4
3	0.3
4	0.2

$\oplus$

Score	*Prob.*
2	0.3
4	0.1
5	0.6

$=$

Score	*Prob.*
3	0.03
4	0.12
5	0.10
...	...
9	0.12

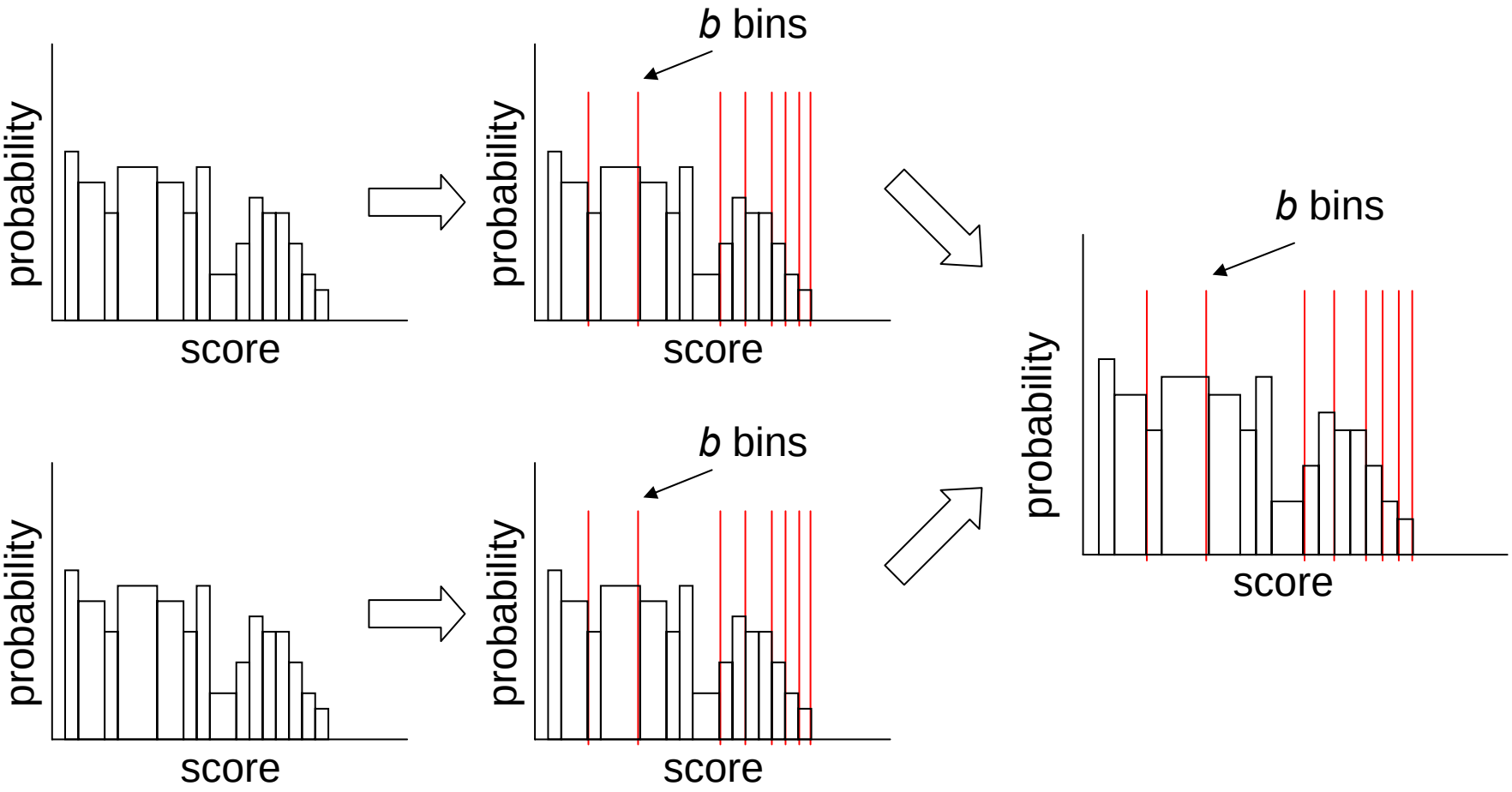
Efficient computation

- Maintain histograms instead of score distributions
  - Score pdf has too much detail
  - Number of bins, *b,* determine accuracy
- Perform cascaded convolutions

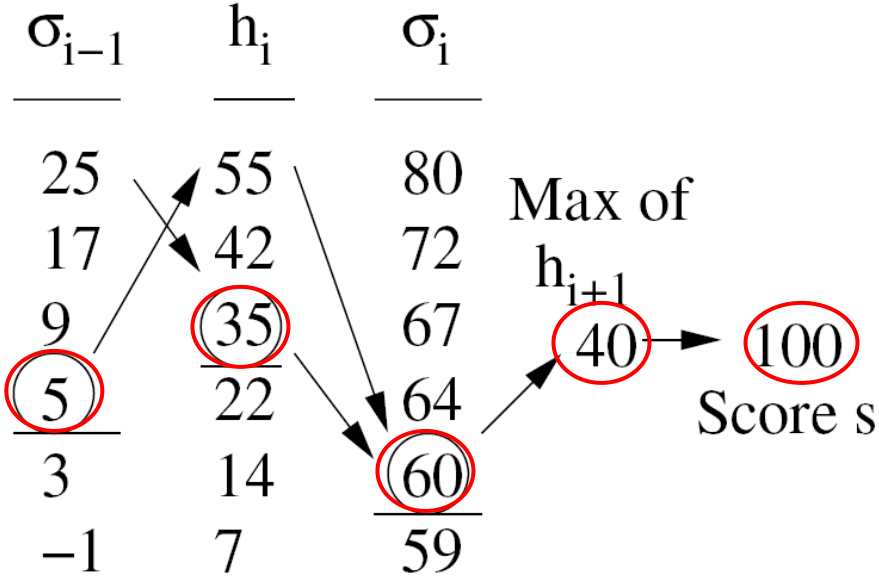
$$\sigma_i = \sigma_{i-1} \oplus h_i$$

  - Each intermediate σ_i contains *b* bins
  - $O(b^2 r)$ operations
- Use bounds to convolute histograms

So far ...


probability
score
b bins
probability
score
b bins
probability
score
probability
score
b bins
probability
score

Pruning of scores using bounds



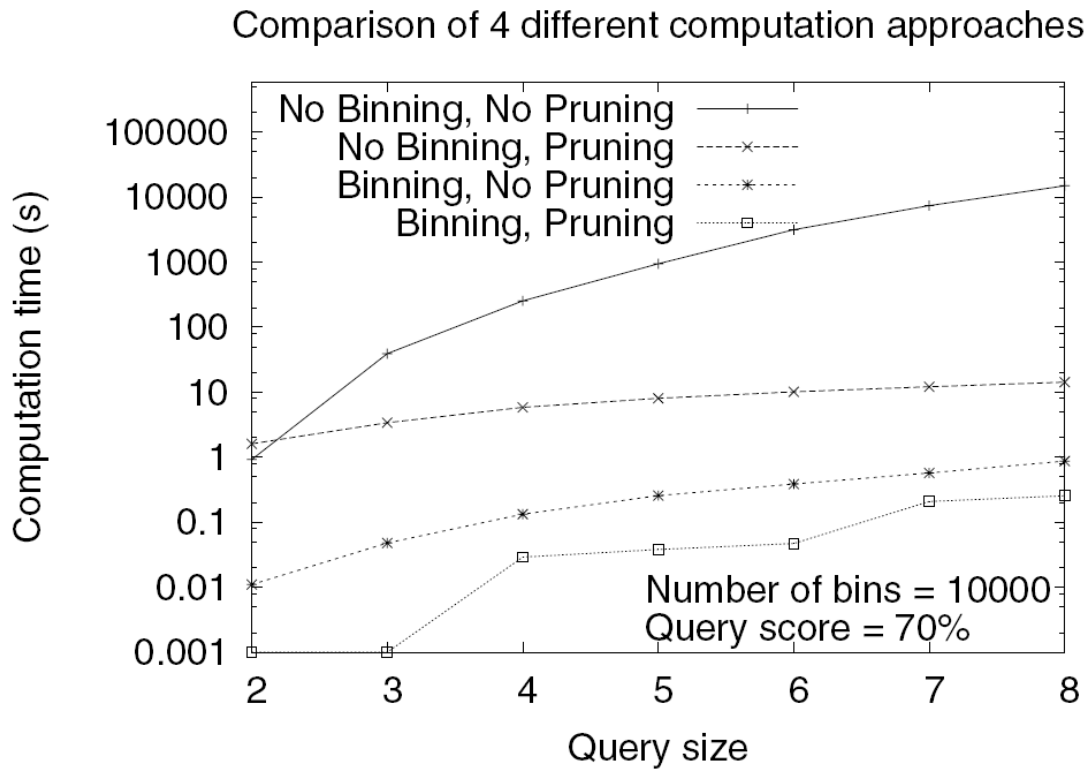
Bound on $\sigma_i = 100 - 40 = 60$

Bound on $h_i = 60 - 25 = 35$

Bound on $\sigma_{i-1} = 60 - 55 = 5$

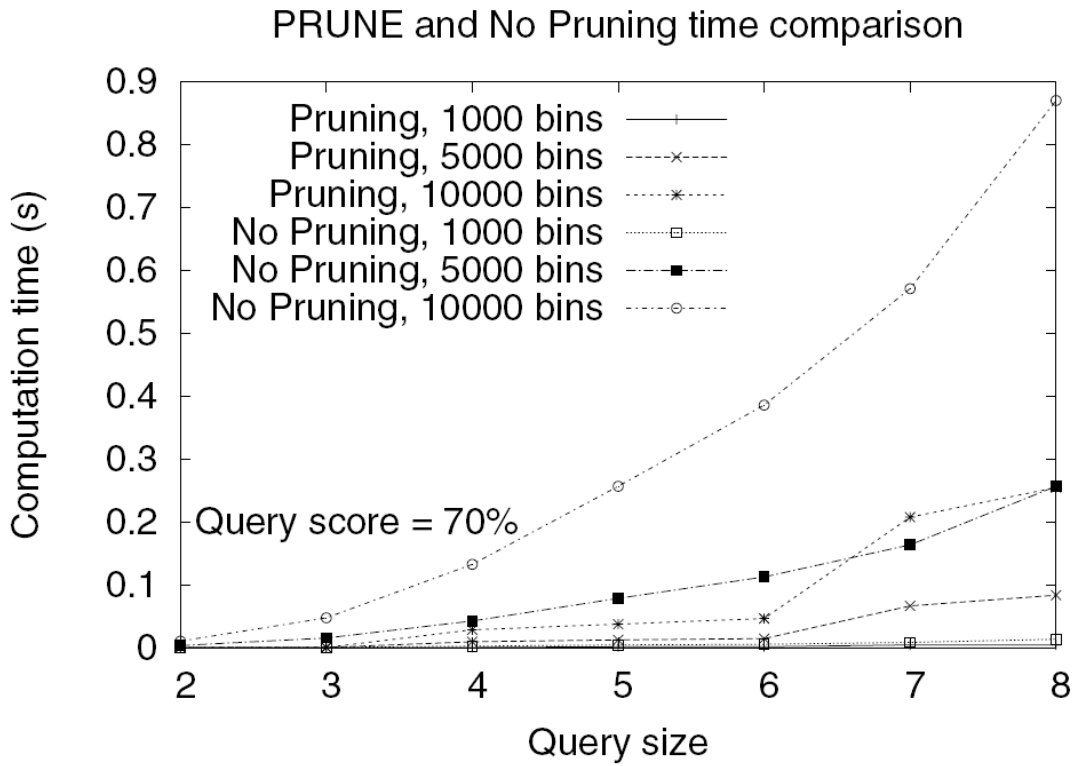
- Threshold score
  - Cannot add up to final score
  - Bins below are not required
- Only *4* and *3* bins required in σ_{i-1} and h_i respectively
  - Convolution of σ_{i-1} with h_i costs *12* operations instead of *36*

Effect of binning and pruning



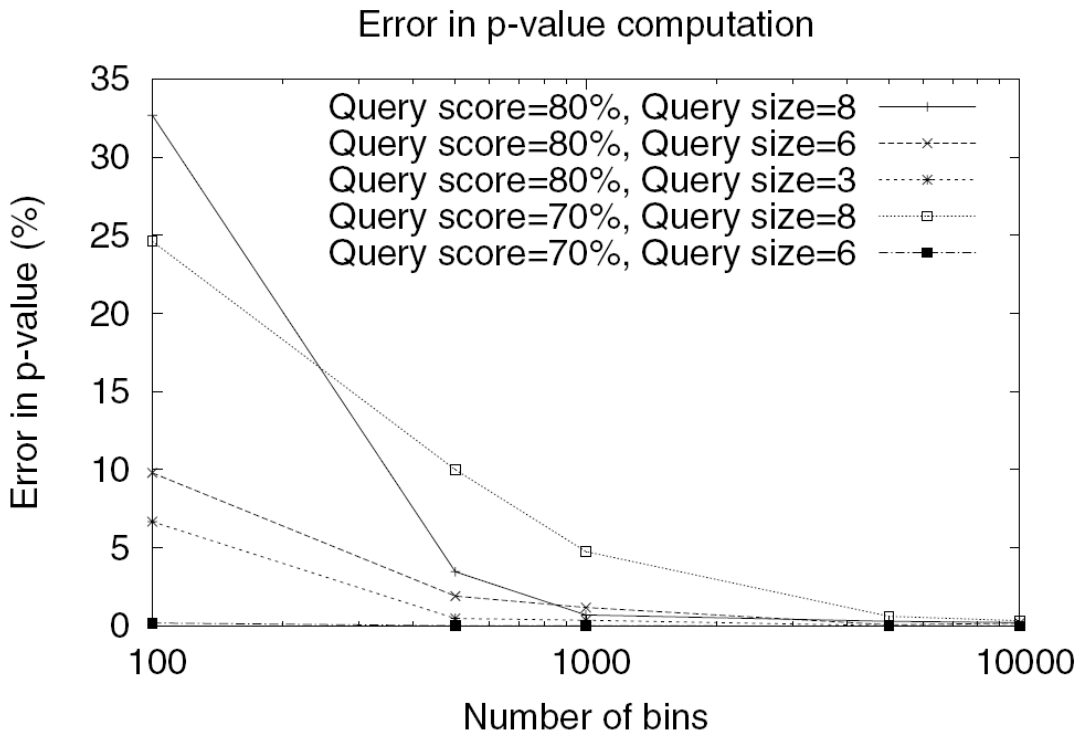
- Binning and pruning saves orders of magnitude time

Effect of query size



- Pruning makes algorithm scalable with query size

Error in p-value due to binning



- For practical number of bins, error in p-value is negligible

Conclusions

- How to efficiently compute p-value for multi-object query results for non-parametric distributions?
- Binning, cascading, bounding
  - Faster by 5 orders of magnitude
  - Error less than 5%
- Future work
  - Examining order of convolution
  - Sampling
  - Other aggregate functions like max