

A Bayesian Method for Guessing the Extreme Values in a Data Set

Mingxi Wu

University of Florida

May, 2008



- Problem Definition
- Example Applications
- A Bayesian Method
- Results
- Related Work
- Conclusion



The Problem

Given a finite set of real values, can we take a sample (without replacement) from the set, and use the sample to predict the k^{th} largest value in the entire set?



Example

- D contains 100 numbers
- Interested in 2^{nd} largest value
- Take a sample $S = \{1, 3, 79\}$

Then, the problem:

Can we use S to guess the 2^{nd} largest value in D ?



Example

- D contains 100 numbers
- Interested in 2^{nd} largest value
- Take a sample $S = \{1, 3, 79\}$

Then, the problem:

Can we use S to guess the 2^{nd} largest value in D ?

Very Difficult!

Imagine D is the result set of an arbitrary query:

```
SELECT 3*R.a+R.b  
FROM R  
WHERE R.c>20
```



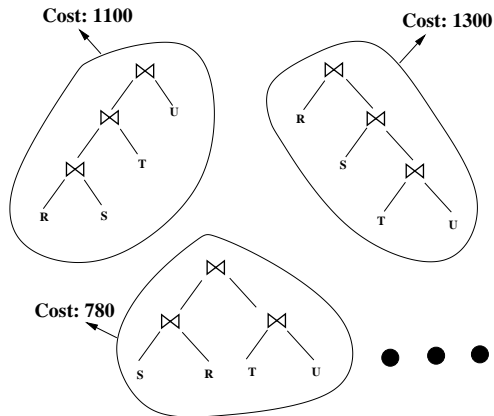
Any research with key words **top/ kth/ max/ min/...**

- Database management
 - **Min/max** online aggregation
 - Probabilistic QO (**min** cost)
 - **Top-k** query processing
 - Distance join (**top** closest pairs)
- Data mining
 - Outlier detection (**kth** NN)
 - Spatial anomaly detection (**top-k** regions)



How Useful?

Probabilistic Query Optimization

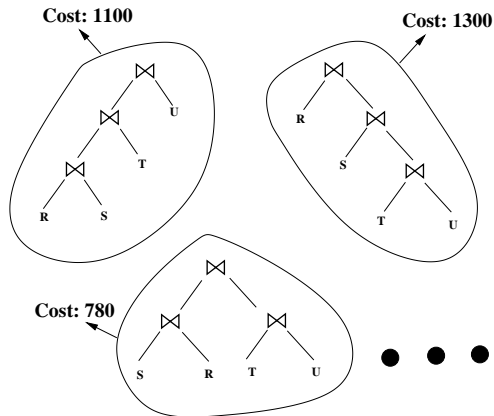


Sub-optimal??



How Useful?

Probabilistic Query Optimization



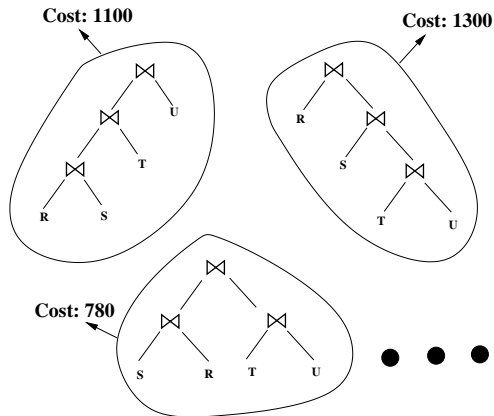
1 Sample 3 plans

Sub-optimal??



How Useful?

Probabilistic Query Optimization

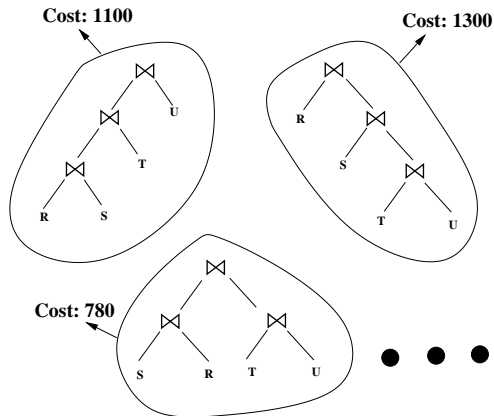


- 1 Sample 3 plans
- 2 Estimate each cost

Sub-optimal??



Probabilistic Query Optimization

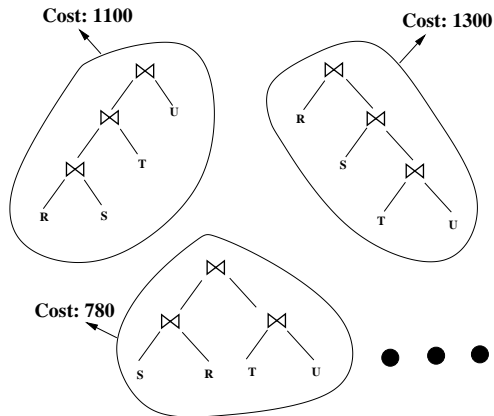


Sub-optimal??

- 1 Sample 3 plans
- 2 Estimate each cost
- 3 Predict the min cost



Probabilistic Query Optimization



Sub-optimal??

- 1 Sample 3 plans
- 2 Estimate each cost
- 3 Predict the min cost
- 4 Any good plan already?



- Problem Definition
- Example Applications
- **A Bayesian Method**
- Results
- Related Work
- Conclusion



A Bird's-eye View of Our Approach

- Propose a natural estimator
- Characterize error distribution of estimator (Bayesian)
 - 1 Learn a **prior** model from the past query workload
 - 2 Update the prior model using a sample
 - 3 Sample an error distribution from the **posterior** model

With estimator and its error distribution, we can confidence bound the k^{th} largest

- **Example:** the 5th largest value in the set is in [500.46, 506.8] with 95% probability.



A Natural Estimator

- Data set size N
- Sample size n
- Estimator is the $(k')^{th}$ largest in sample
 - $k' = \lceil \frac{n}{N} \times k \rceil$, since $\frac{k'}{n} = \frac{k}{N}$ captures the relative rank

Example

- $N = 100$, $n = 3$, $k = 2$
- Sample is $S = \{1, 3, 79\}$
- Then, estimator is **79**, since $k' = \lceil \frac{3}{100} \times 2 \rceil = 1$



A Natural Estimator

- Data set size N
- Sample size n
- Estimator is the $(k')^{th}$ largest in sample
 - $k' = \lceil \frac{n}{N} \times k \rceil$, since $\frac{k'}{n} = \frac{k}{N}$ captures the relative rank

Example

- $N = 100$, $n = 3$, $k = 2$
- Sample is $S = \{1, 3, 79\}$
- Then, estimator is **79**, since $k' = \lceil \frac{3}{100} \times 2 \rceil = 1$

How Accurate?

- Imagine $D = \{1, 2, 3 \dots 79 \dots 10^5, 10^7\}$
- Imagine $D = \{-1, 1, 1.1 \dots 3 \dots 79, 80, 81\}$



How to determine the estimator's error?

Study the relationship (estimator versus answer)



How to determine the estimator's error?

Study the relationship (estimator versus answer)

- **Option 1:** Take their difference, and find the difference's distribution



How to determine the estimator's error?

Study the relationship (estimator versus answer)

- **Option 1:** Take their difference, and find the difference's distribution
 - Not good, limited by scale



How to determine the estimator's error?

Study the relationship (estimator versus answer)

- **Option 1:** Take their difference, and find the difference's distribution
 - Not good, limited by scale
- **Option 2:** Take their ratio, and find the ratio distribution



How to determine the estimator's error?

Study the relationship (estimator versus answer)

- **Option 1:** Take their difference, and find the difference's distribution
 - Not good, limited by scale
- **Option 2:** Take their ratio, and find the ratio distribution
 - Good, not constrained by scale



How to determine the estimator's error?

Study the relationship (estimator versus answer)

- **Option 1:** Take their difference, and find the difference's distribution
 - Not good, limited by scale
- **Option 2:** Take their ratio, and find the ratio distribution
 - Good, not constrained by scale

Our Choice: Study the distribution of $\frac{k^{th}}{(k')^{th}}$.



Bounding with ratio distribution

Given the distribution of $\frac{k^{th}}{(k')^{th}}$, we can confidence bound the answer:

- **If** there is 95% chance $l < \frac{k^{th}}{(k')^{th}} < h$, **then**

there is 95% chance $l \cdot (k')^{th} < k^{th} < h \cdot (k')^{th}$



Why Bayesian?

Running Example

- $D = \{1, 3 \dots 79 \dots 98, 10^5, 10^6\}$; total 100 numbers
- Interested in 2^{nd} largest value
- Take a sample $S = \{1, 3, 79\}$

Discussion

- Impossible to predict the ratio ($\frac{10^5}{79}$) by looking at sample only
- Without knowledge about D , cannot bias 79 towards larger value
- With *domain knowledge* and *sample*, can guess behavior of D



Why Bayesian?

Running Example

- $D = \{1, 3 \dots 79 \dots 98, 10^5, 10^6\}$; total 100 numbers
- Interested in 2^{nd} largest value
- Take a sample $S = \{1, 3, 79\}$

Discussion

- Impossible to predict the ratio ($\frac{10^5}{79}$) by looking at sample only
- Without knowledge about D , cannot bias 79 towards larger value
- With *domain knowledge* and *sample*, can guess behavior of D
- **Bayesian can combine both**



Next question

What domain knowledge should be modeled to help solving this problem?



Next question

What domain knowledge should be modeled to help solving this problem?

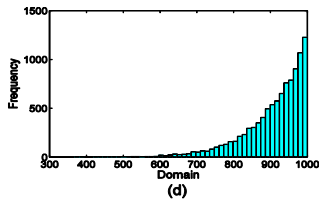
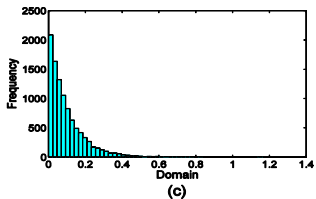
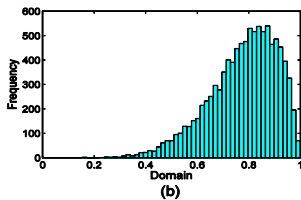
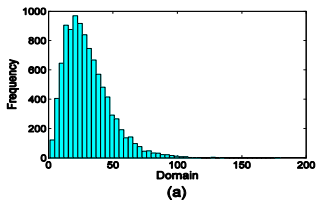
Some experiments might help...



Characterize Error Distribution of Estimator

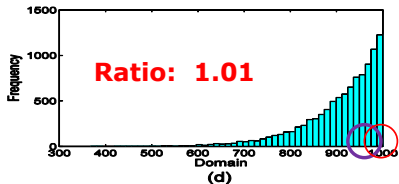
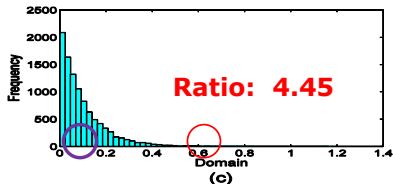
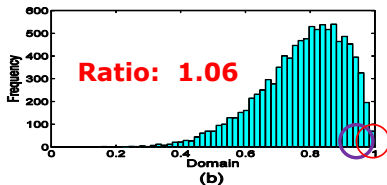
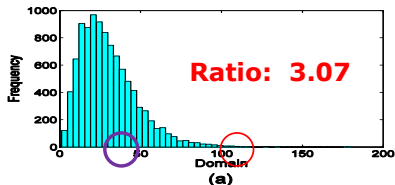
Setup: four data sets with different histogram shapes; each has 10,000 values; I am looking for the largest value.

Experiment: take a 100-element sample, record the obtained ratio $\frac{k^{th}}{(k')^{th}}$. Do this 500 times.



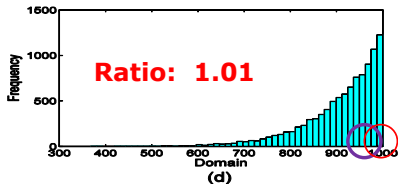
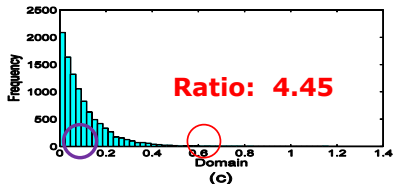
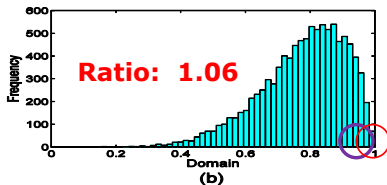
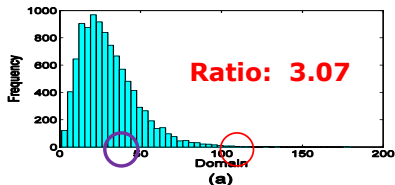
Characterize Error Distribution of Estimator

Observation: The histogram shape affects the ratio $\frac{k^{th}}{(k')^{th}}$.



Characterize Error Distribution of Estimator

Observation: The histogram shape affects the ratio $\frac{k^{th}}{(k')^{th}}$.

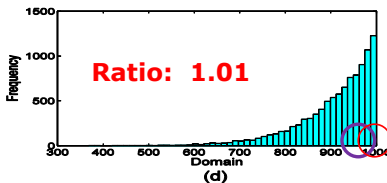
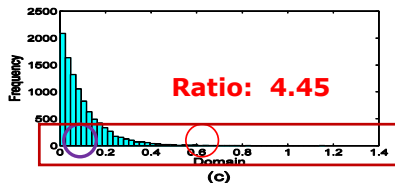
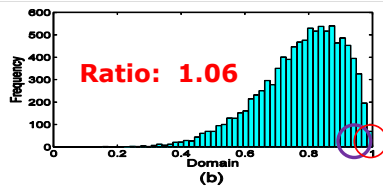
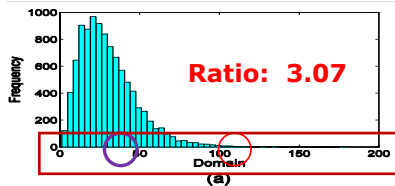


Model **Shape!**



Characterize Error Distribution of Estimator

Side result: Verified the scale does not matter.



First step in modeling the shape

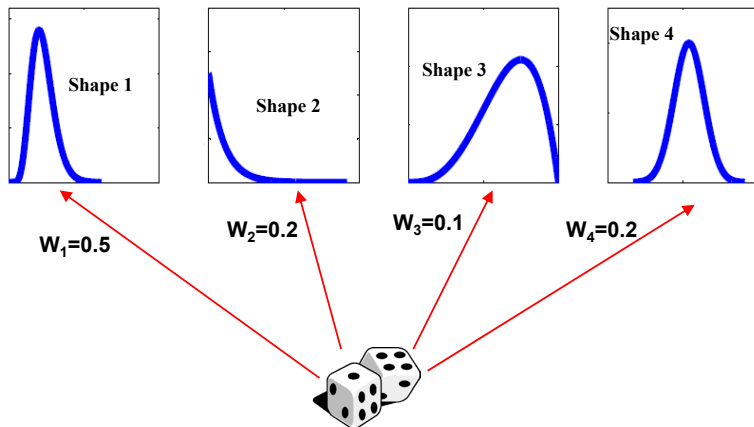
Imagine how a realistic model would have produced the data set.



Characterize Error Distribution of Estimator

The Generative Model

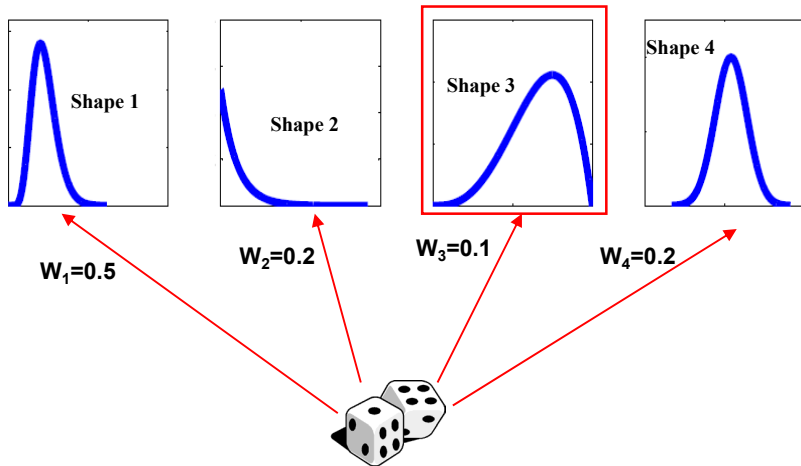
- Assume the existence of a set of possible shape patterns
- Each shape has a weight, specifying how likely it matches with a new data set's histogram shape



Characterize Error Distribution of Estimator

To generate a new data set...

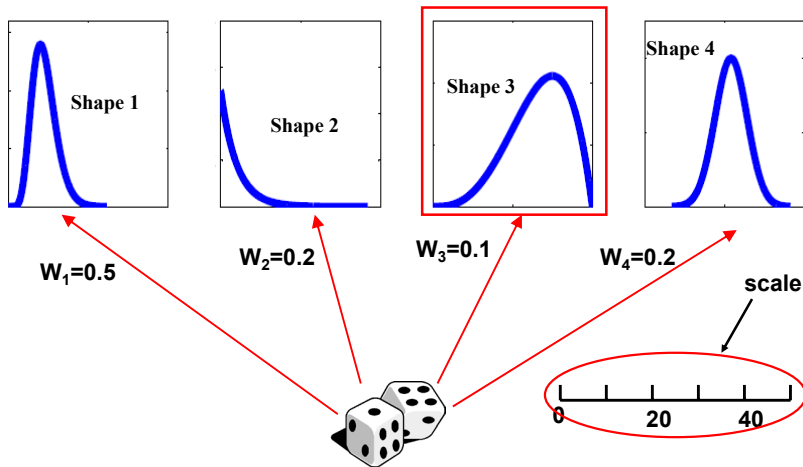
Step 1: Roll a biased die



Characterize Error Distribution of Estimator

To generate a new data set...

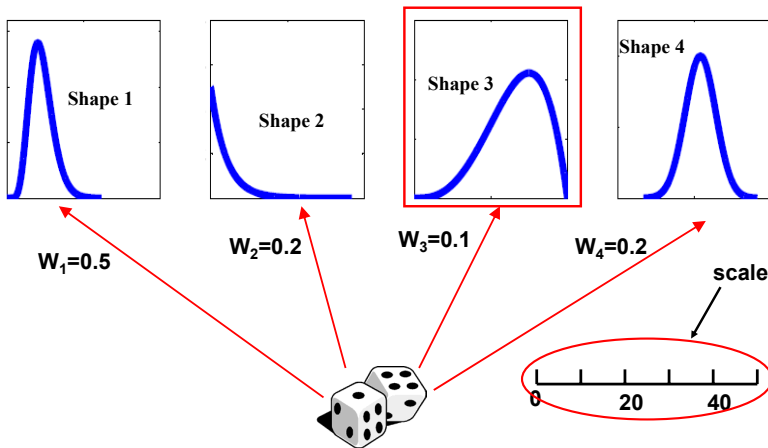
Step 2: Randomly select a scale



Characterize Error Distribution of Estimator

To generate a new data set...

Step 3: Instantiate a parametric distribution $f(\mathbf{x}|\text{shape}, \text{scale})$; this distribution is repeatedly sampled from to generate the new data set.



Next, formalize and learn the model from domain data(workload)

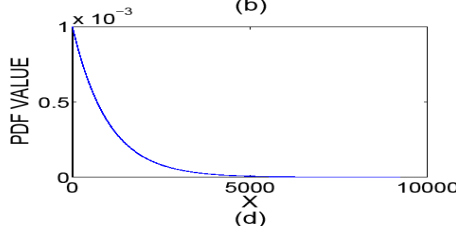
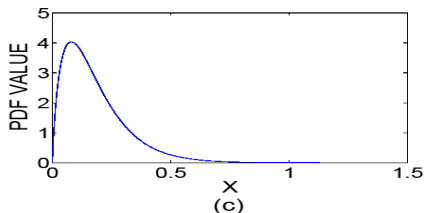
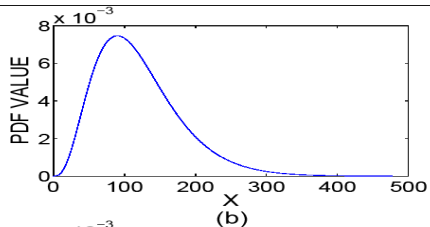
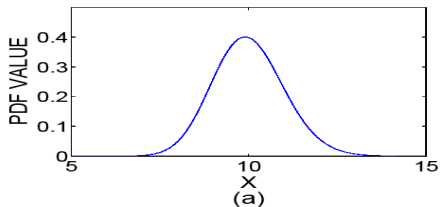
- Devised a close-form model: a variant of Gamma mixture model
- Employed an EM algorithm to learn the model from historical data



Define a Prior Shape Model

Choose an Appropriate Parametric Model

Gamma distribution can produce data with arbitrary right leaning skew



Deriving the Likelihood Model...

- Gamma distribution pdf is:

$$p_{\text{Gamma}}(x|\alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x}, \quad x > 0$$

where α is a shape parameter, β is a scale parameter

- Since scale does not matter, we treat β as an unknown random variable, and integrate it.
- The resulting likelihood of a given data set D is in the form:

$$L(D|\alpha)$$



Deriving the Likelihood Model...

- The likelihood of D given one shape α is:

$$L(D|\alpha)$$

- Our model assumes a set of c weighted shapes. Thus, the complete likelihood model of observing D is:

$$L^*(D|\Theta) = \sum_{j=1}^c w_j L(D|\alpha_j)$$

where w_j 's are each non-negative weights and $\sum_j w_j = 1$.



Learning the Parameters

- Given a set of independent domain data sets $\mathbf{D} = \{D_1, \dots, D_r\}$, the likelihood of observing them is:

$$L(\Theta|\mathbf{D}) = \prod_{i=1}^r L^*(D_i|\Theta)$$

- We use EM algorithm to learn the most likely Θ^* so that:

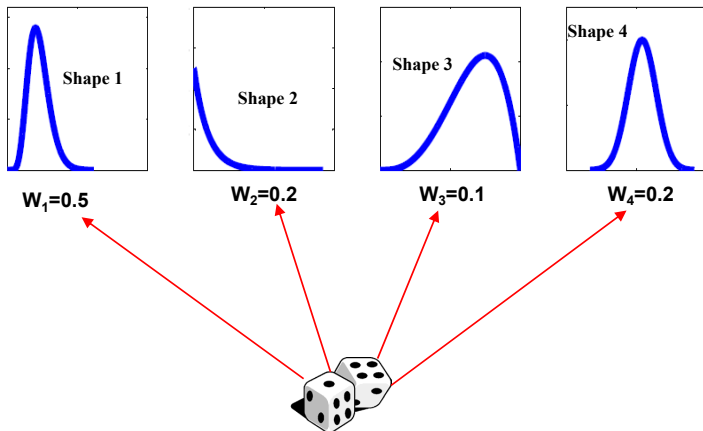
$$\Theta^* = \underset{\Theta}{\operatorname{argmax}} \log L(\Theta|\mathbf{D})$$

where $\Theta = \{\theta_1, \dots, \theta_c\}$ and $\theta_j = \{\mathbf{w}_j, \alpha_j\}$.



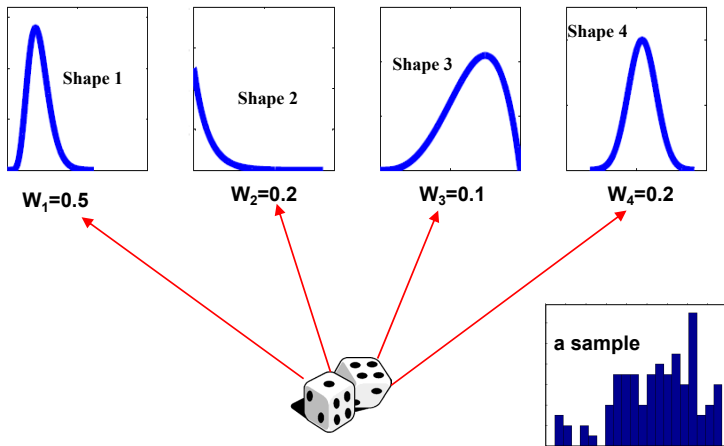
Bayesian Update

At this point, we have learned a prior shape model



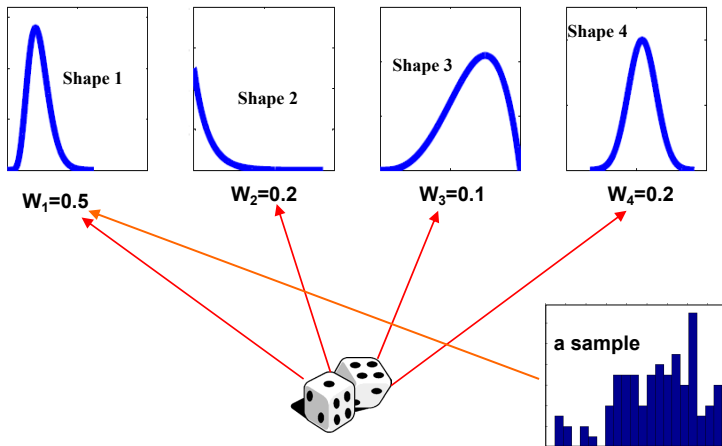
Bayesian Update

Now, take a sample from the current data set



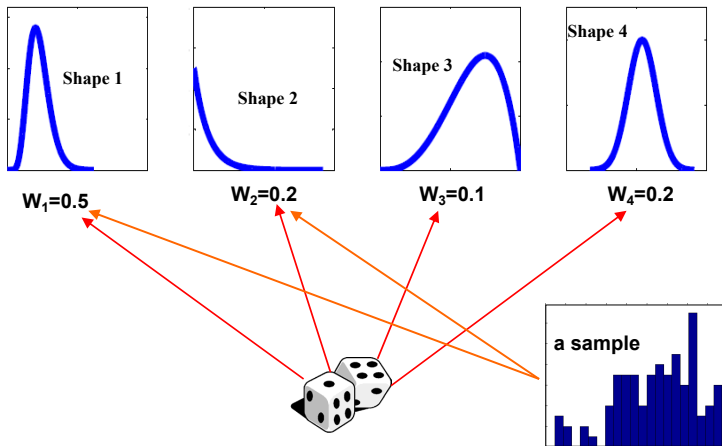
Bayesian Update

Use the sample to update prior weight of each shape pattern



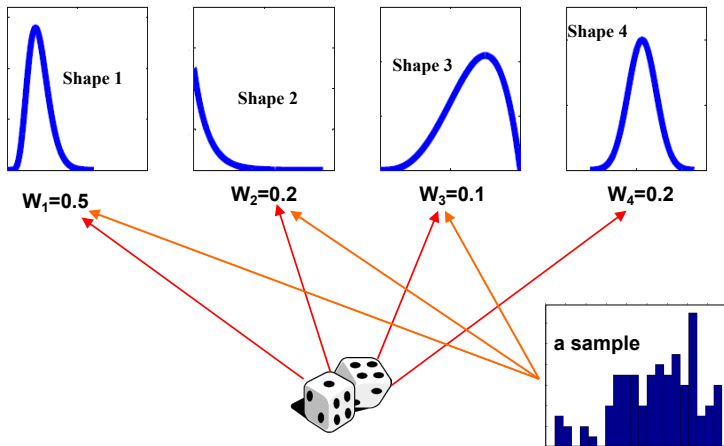
Bayesian Update

Use the sample to update prior weight of each shape pattern



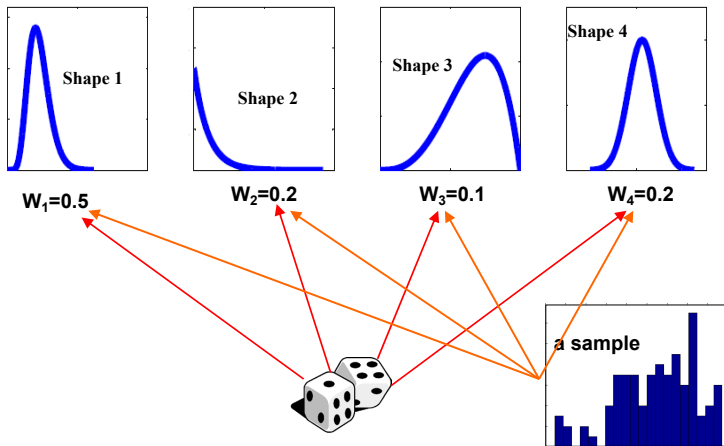
Bayesian Update

Use the sample to update prior weight of each shape pattern



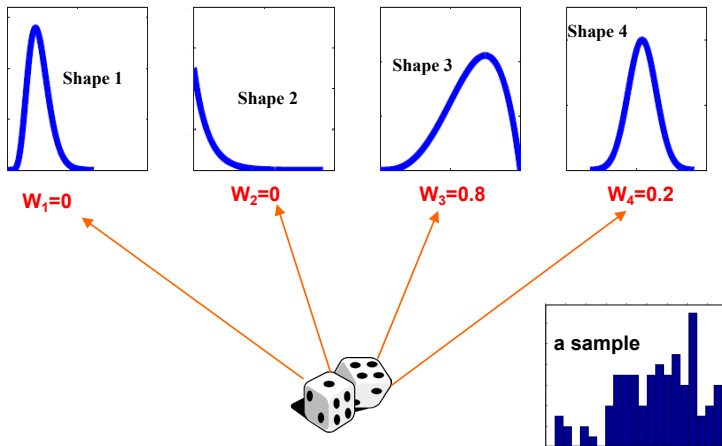
Bayesian Update

Use the sample to update prior weight of each shape pattern



Bayesian Update

The result is a posterior model with updated shape weights



Classic Bayesian Fashion

- Let S be our sample, applying Bayes' rule, the posterior weight of shape pattern j is:

$$w'_j = \frac{w_j L(S|\alpha_j)}{\sum_{k=1}^c w_k L(S|\alpha_k)}$$

- The resulting posterior shape model:

$$L^*(D|\Theta) = \sum_{j=1}^c w'_j L(D|\alpha_j)$$



Classic Bayesian Fashion

- Let S be our sample, applying Bayes' rule, the posterior weight of shape pattern j is:

$$w'_j = \frac{w_j L(S|\alpha_j)}{\sum_{k=1}^c w_k L(S|\alpha_k)}$$

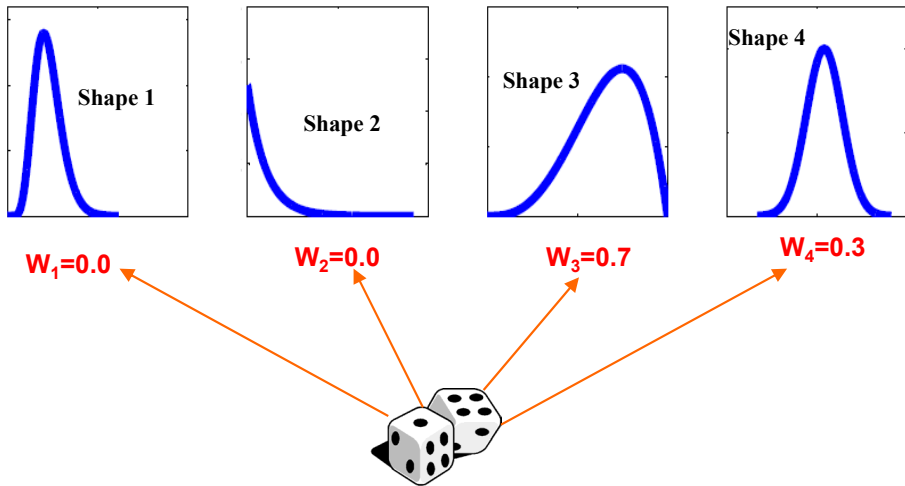
- The resulting posterior shape model:

$$L^*(D|\Theta) = \sum_{j=1}^c w'_j L(D|\alpha_j)$$



Produce an Error Distribution from Posterior Model

Recall the Posterior Shape model



Produce an Error Distribution from Posterior Model

Each shape characterizes an error distribution of $\frac{k^{th}}{(k')^{th}}$

To find the error distribution for a shape α , do:

- Pick a scale β
- Instantiate a Gamma distribution $f(x|\alpha, \beta)$
- Sample a ratio distribution from f (see TKD method in paper):



Produce an Error Distribution from Posterior Model

Each shape characterizes an error distribution of $\frac{k^{th}}{(k')^{th}}$

To find the error distribution for a shape α , do:

- Pick a scale β
- Instantiate a Gamma distribution $f(x|\alpha, \beta)$
- Sample a ratio distribution from f (see TKD method in paper):
 - 1 Generate a database instance from $f(x|\alpha, \beta)$, then find k



Produce an Error Distribution from Posterior Model

Each shape characterizes an error distribution of $\frac{k^{th}}{(k')^{th}}$

To find the error distribution for a shape α , do:

- Pick a scale β
- Instantiate a Gamma distribution $f(x|\alpha, \beta)$
- Sample a ratio distribution from f (see TKD method in paper):
 - 1 Generate a database instance from $f(x|\alpha, \beta)$, then find k
 - 2 Take a sample from the database instance, then find k'



Produce an Error Distribution from Posterior Model

Each shape characterizes an error distribution of $\frac{k^{th}}{(k')^{th}}$

To find the error distribution for a shape α , do:

- Pick a scale β
- Instantiate a Gamma distribution $f(x|\alpha, \beta)$
- Sample a ratio distribution from f (see TKD method in paper):
 - 1 Generate a database instance from $f(x|\alpha, \beta)$, then find k
 - 2 Take a sample from the database instance, then find k'
 - 3 Take the ratio $\frac{k^{th}}{(k')^{th}}$



Produce an Error Distribution from Posterior Model

Each shape characterizes an error distribution of $\frac{k^{th}}{(k')^{th}}$

To find the error distribution for a shape α , do:

- Pick a scale β
- Instantiate a Gamma distribution $f(x|\alpha, \beta)$
- Sample a ratio distribution from f (see TKD method in paper):
 - 1 Generate a database instance from $f(x|\alpha, \beta)$, then find k
 - 2 Take a sample from the database instance, then find k'
 - 3 Take the ratio $\frac{k^{th}}{(k')^{th}}$
 - 4 Repeat step 1 to 3 many times

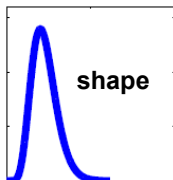


Produce an Error Distribution from Posterior Model

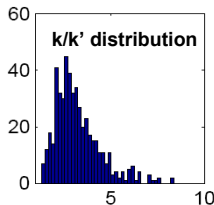
Each shape characterizes an error distribution of $\frac{k^{th}}{(k')^{th}}$

To find the error distribution for a shape α , do:

- Pick a scale β
- Instantiate a Gamma distribution $f(x|\alpha, \beta)$
- Sample a ratio distribution from f (see TKD method in paper):
 - 1 Generate a database instance from $f(x|\alpha, \beta)$, then find k
 - 2 Take a sample from the database instance, then find k'
 - 3 Take the ratio $\frac{k^{th}}{(k')^{th}}$
 - 4 Repeat step 1 to 3 many times



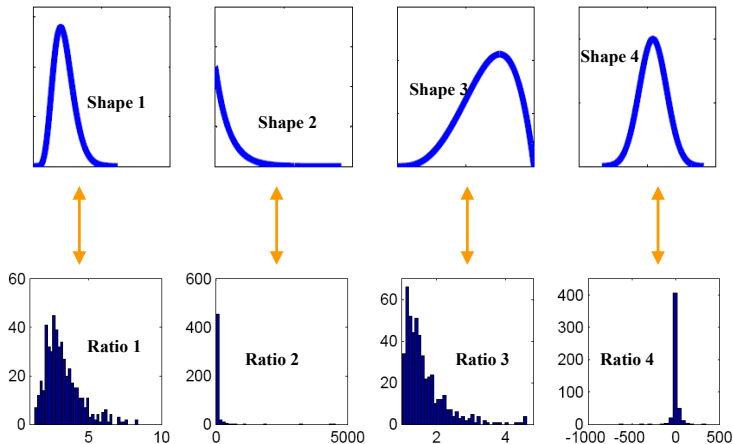
characterize



Produce an Error Distribution from Posterior Model

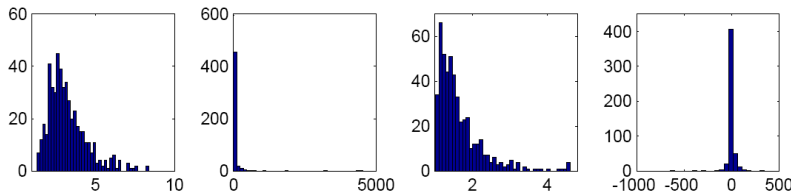
Each shape characterizes an error distribution $\frac{k^{th}}{(k')^{th}}$

- The prob. we will see a shape is the prob. we will see its error distribution.



Produce an Error Distribution from Posterior Model

Attaching each shape's posterior weight to its error distribution, we get a posterior error distribution



$W_1=0.0$

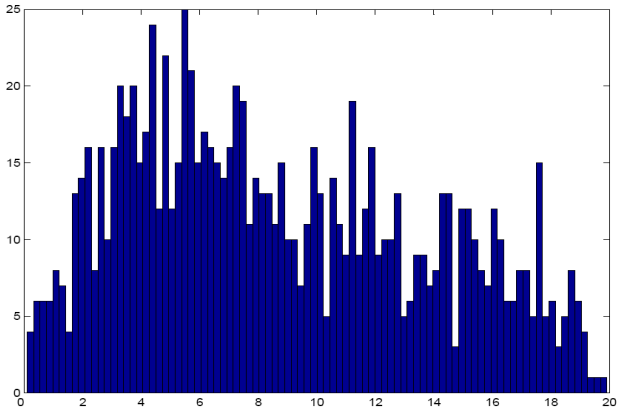
$W_2=0.0$

$W_3=0.7$

$W_4=0.3$



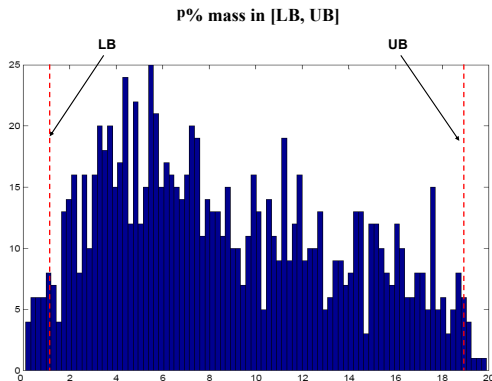
The final mixture error distribution:



Final Bounding of the k^{th} Largest Value

Given the distribution of $\frac{k^{\text{th}}}{(k')^{\text{th}}}$, we can confidence bound the answer:

- 1 Choose a pair $(\mathbf{LB}, \mathbf{UB})$, s.t. $p\%$ probability is covered
- 2 Bound k^{th} by $[(k')^{\text{th}} \times \mathbf{LB}, (k')^{\text{th}} \times \mathbf{UB}]$ with $p\%$ probability
 - since with $p\%$ prob. $\frac{k^{\text{th}}}{(k')^{\text{th}}} \in [\mathbf{LB}, \mathbf{UB}]$



Summary

- 1 Learn a prior shape model from historical queries
 - Devised a close-form model: a variant of Gamma mixture model
 - Employed an EM algorithm to learn the model from historical data
- 2 Update prior shape model with a sample
 - Applied Baye's rule to update shape pattern's weight
- 3 Produce an error distribution from the posterior model
 - Posterior weight attached to each shape's error distribution

With our estimator and its error distribution, we can bound answer.



- Problem Definition
- Example Applications
- A Bayesian Method
- **Results**
- Related Work
- Conclusion



Estimate the k^{th} Largest Value

Setup:

- A query result set is created by:
 - 1 Randomly select a point p
 - 2 Selectivity s is randomly picked from 5% to 20%
 - 3 $s \times |D|$ nearest neighbors (NN) of p are picked out.
 - 4 A random function is applied to the picked NNs.
- 500 training queries and 500 testing queries
- Confidence level is set to 95%



Estimate the k^{th} Largest Value

Real Data Sets:

Data set	Continuous/Feature	Size
Letter	16/17	20,000
CAHouse	7/9	20,640
El Nino	7/7	93,935
Cover Type	10/55	581,012
KDDCup99	34/41	4,898,430
Person90	12/13	5,000,000
Household90	7/11	5,523,522

Table: Data description. These data sets consist of both categorical and continuous features.



Estimate the k^{th} Largest Value

Bounding Reliability:

Data set	$k = 1$	$k = 5$	$k = 10$	$k = 20$
Letter	1.00	0.98	0.97	0.94
CAHouse	0.97	0.99	0.97	0.96
El Nino	1.00	1.00	0.99	0.99
Cover Type	1.00	1.00	0.99	0.99
KDDCup99	0.92	0.91	0.92	0.93
Person90	0.92	0.94	0.90	0.91
Household90	0.98	0.97	0.97	0.97

Table: Coverage rates for 95% confidence bounds with various k values using a 10% sample.

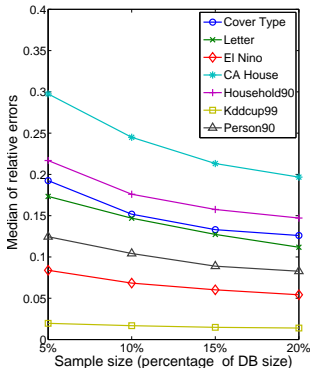


Estimate the k^{th} Largest Value

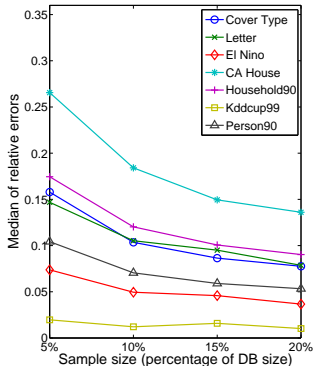
Bounding Effectiveness:

Median relative error of 500 test queries;

Y-axis is $(0.5 \text{ c.f. interval})/\text{answer}$.



(a) $k=1$



(b) $k=10$



More results...

- **Distance Join:** improve the performance of a nested-loop algorithm by a factor of **3.5** over two real multimedia data sets.
- **Outlier Detection:** improve the performance of state-of-the-art algorithm on an average factor of **4** over seven large data sets.
- **Spatial Anomaly Detection:** improve the performance of a naive algorithm on a factor of **50** over a real data set placed on a 128×128 grid.



- Survey Sampling [**Sarndal 92**]
 - My problem in this domain.
- Extreme Value Distribution Theory [**Leadbetter 83**]
 - Not suitable for modeling different scale; Require samples of extreme values to fit a model.
- Random sampling from databases [**Olken 93**]
 - Relevant, but different problems
- Online Aggregation [**Hellerstein 97**]
 - My work is the **first step** in attacking one of the open problems in this regime.



Conclusion

- Defined the problem of estimating the k^{th} largest value in a real set
- Proposed an estimator
- Characterized the ratio error distribution by a Bayesian framework
- Applied (succesfully) the proposed method to four research problems

