CXHist: An On-line Classification-Based Histogram for XML String Selectivity Estimation

Min Wang
IBM T. J. Watson Research Center

Joint work with
Lipyeow Lim (IBM T. J. Watson)
Jeffrey S. Vitter (Purdue University)

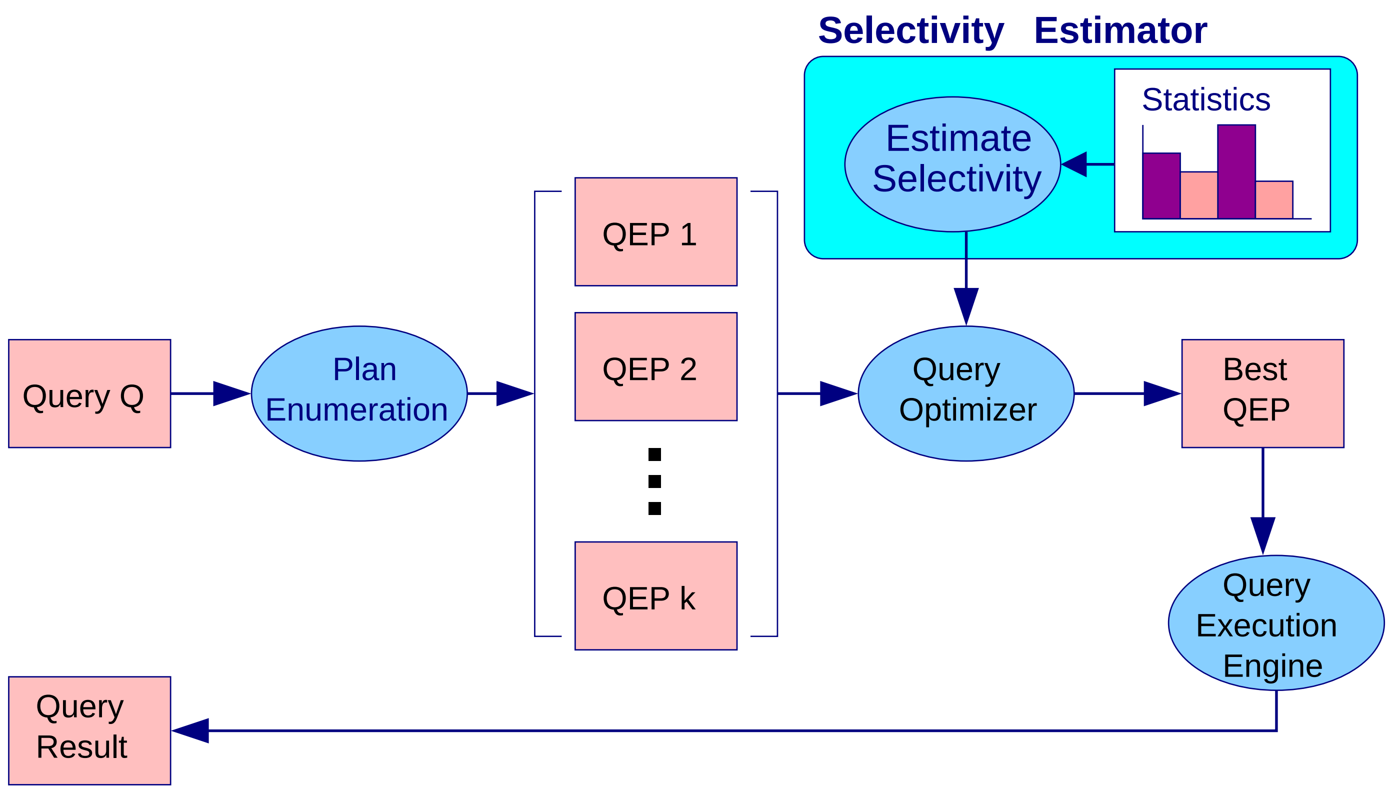
September 1, 2005

Outline

1. Motivation: Selectivity Esimation for XML Data
2. Related Work
3. Intuition for Classification-based Histograms
4. CXHist: the Method
5. Experiments
6. Conclusions

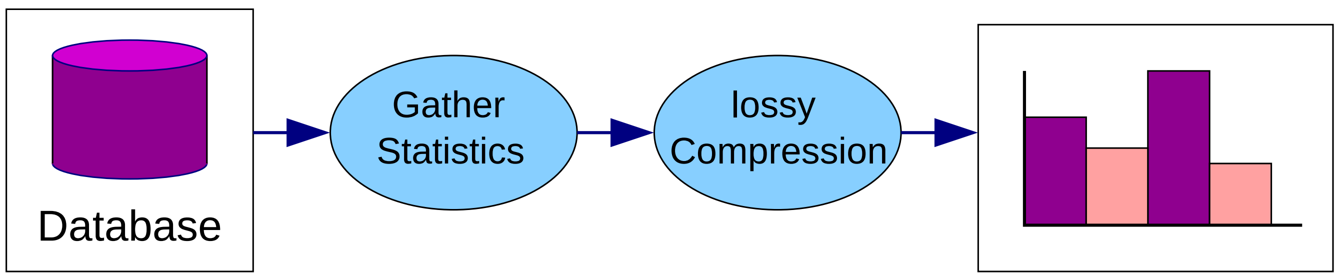
Query Optimization in Database Systems

Overview of cost-based query optimization.

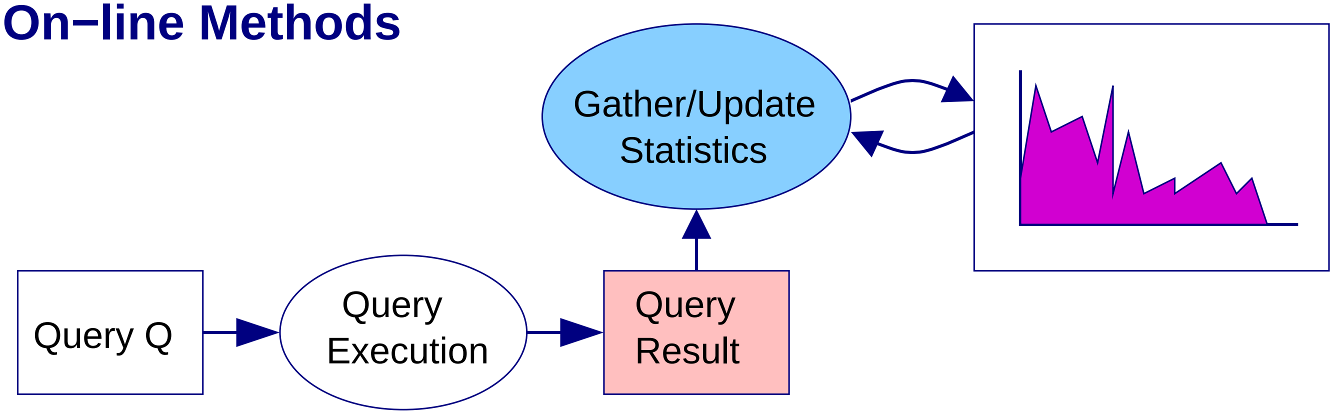


Gathering Statistics

Off–line Methods



On–line Methods



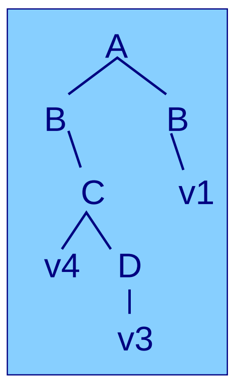
On-line Statistics Gathering

On-line methods for gathering statistics are especially attractive, because they

1. Avoid off-line scans of the data,
2. Adapt to dynamically changing data, and
3. Adapt to changing or non-uniform query workload characteristics.

Selectivity estimation for XML data

```
<A>
  <B>
    <C>
      v4
      <D>
        v3
      </D>
    </C>
  </B>
  <B>
    v1
  </B>
</A>
```



XML doc – XML tree.

- XML data are conceptually trees
- Queries are path expressions, eg.,
  - simple: `//B/C/D`
  - single-value: `//B/C/D=v3`
  - multi-value: `//B/C=v4/D=v3`
  - subtree: `/A[/B=v1]/B/C=v4`
- Query processing via
  - index (maps path to nodes)
  - tree traversal
  - combination
- Cost evaluation of QEP requires estimating number of nodes that match a path expression.

Related Work

Method	Query	Leaf Values	On/Off-line
Cor. Subpath Tree (ICDE'01)	subtree	(sub)string	Off-line
Markov Table (VLDB'01)	linear	–	Off-line
XPathLearner (VLDB'02)	**linear**	**string**	**On-line**
XSketch (SIGMOD'02)	subtree	numeric	Off-line
Statix (SIGMOD'02)	subtree	numeric	Off-line
Position Hist. (EDBT'02)	ancestor	string	Off-line

XML String Selectivity Estimation

- **The problem we solve:** How to estimate selectivity of string predicates on the value part of a path-value pair? Substring & exact match predicates.
- Number of distinct root-to-node paths is relatively small ($\sim 10^2$).
- Number of distinct path-value pairs is huge ($\sim 10^6$).
- XPathLearner does not support substring predicates
- Suffix tree based methods are too costly and tend to under-estimate for string equality predicates.

Intuition for Classification-based Histograms

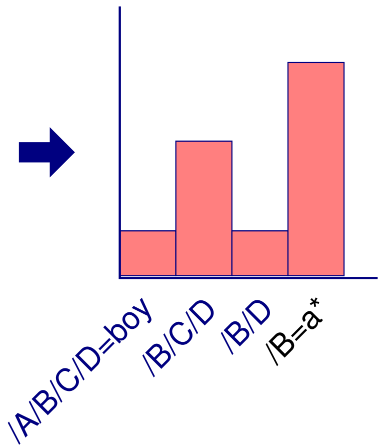
Query	sel
/A/B/C/D=boy	1
/B/C/D	3
/B/D	1
/B=a*	5

Query Workload

Intuition for Classification-based Histograms

Query	sel
/A/B/C/D=boy	1
/B/C/D	3
/B/D	1
/B=a*	5

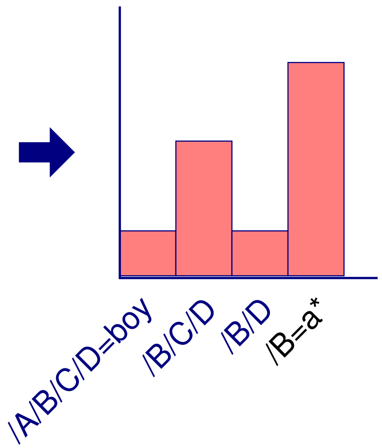
Query Workload

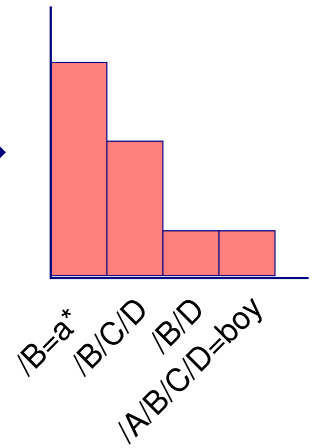


Intuition for Classification-based Histograms

Query	sel
/A/B/C/D=boy	1
/B/C/D	3
/B/D	1
/B=a*	5

Query Workload

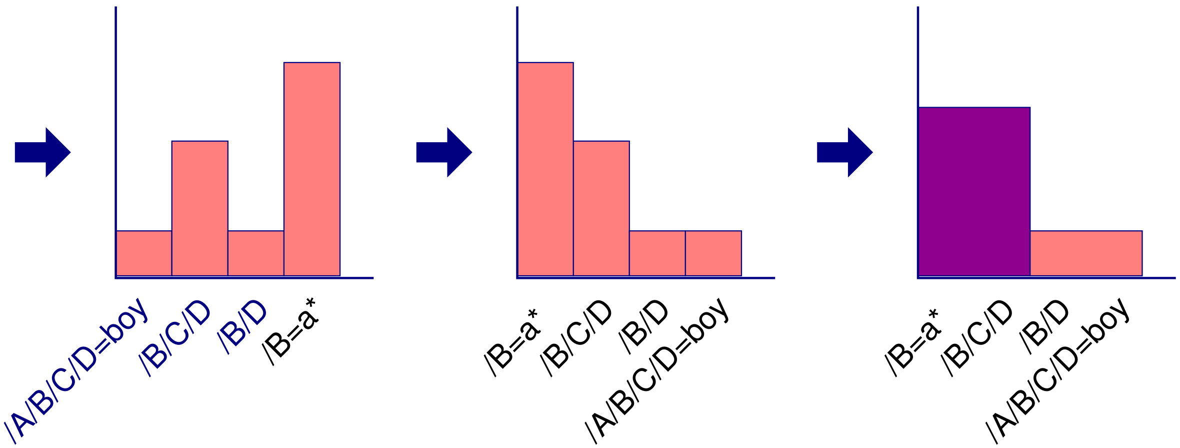




Intuition for Classification-based Histograms

Query	sel
/A/B/C/D=boy	1
/B/C/D	3
/B/D	1
/B=a*	5

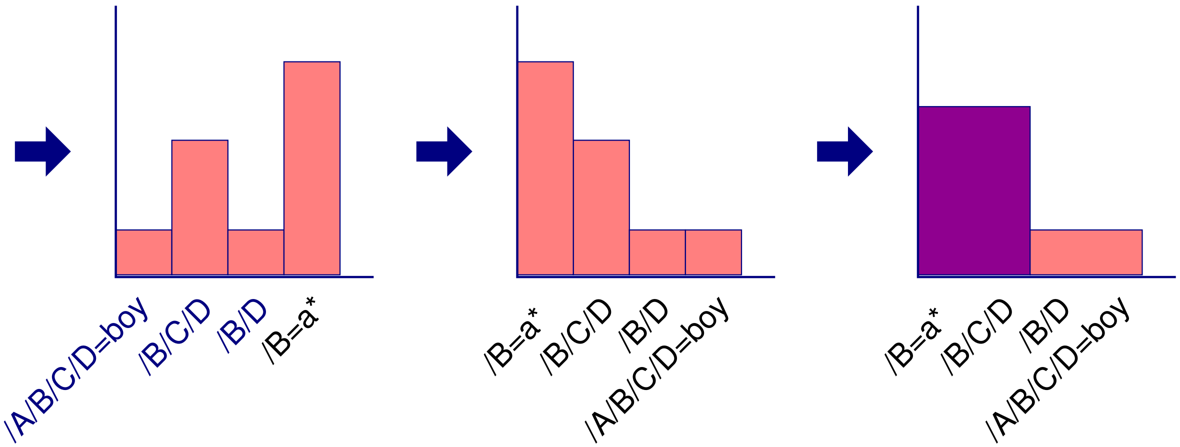
Query Workload

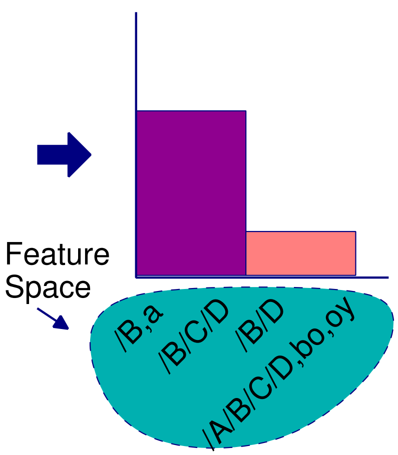


Intuition for Classification-based Histograms

Query	sel
/A/B/C/D=boy	1
/B/C/D	3
/B/D	1
/B=a*	5

Query Workload

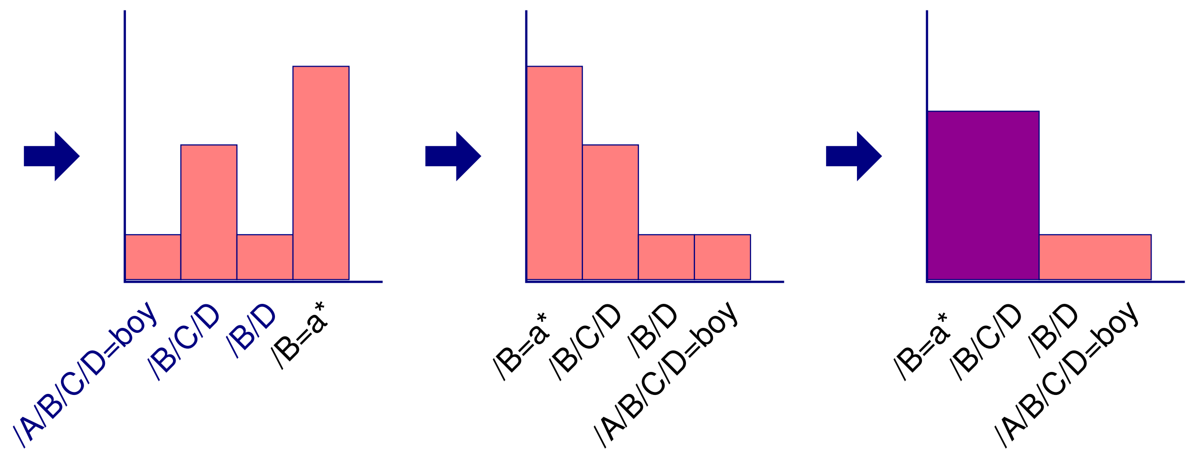


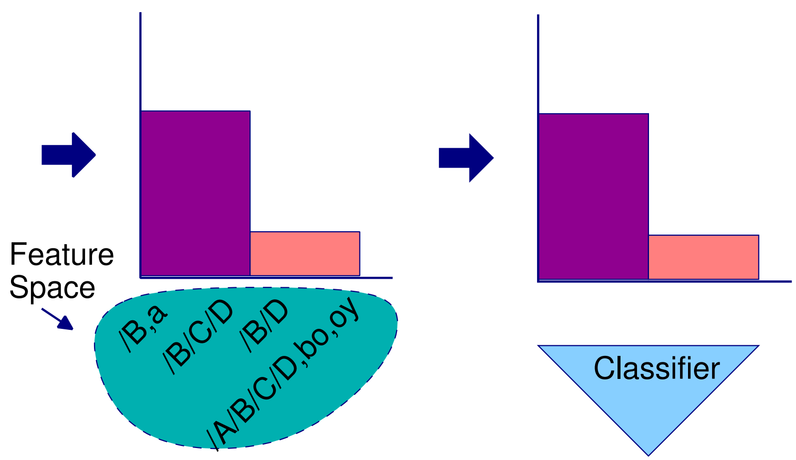


Intuition for Classification-based Histograms

Query	sel
/A/B/C/D=boy	1
/B/C/D	3
/B/D	1
/B=a*	5

Query Workload

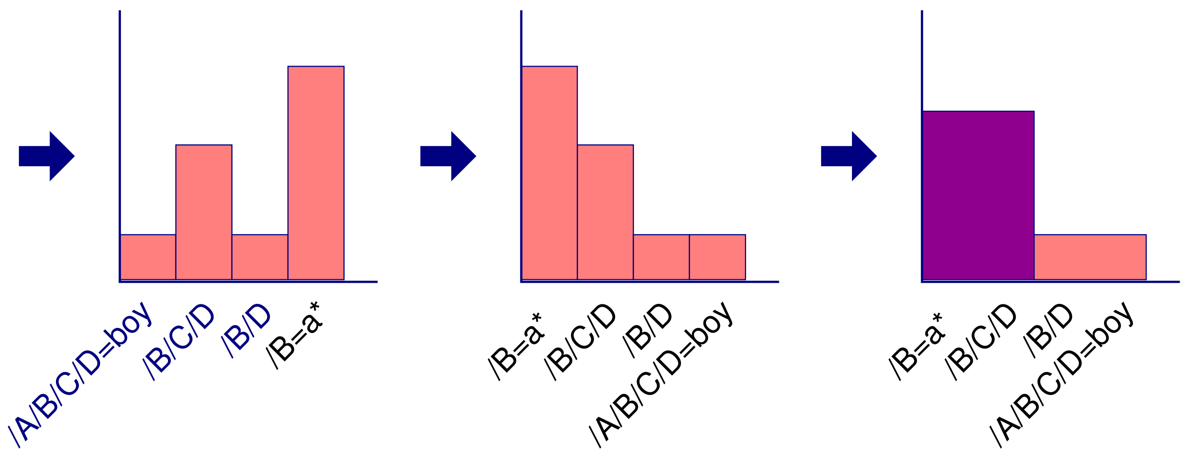


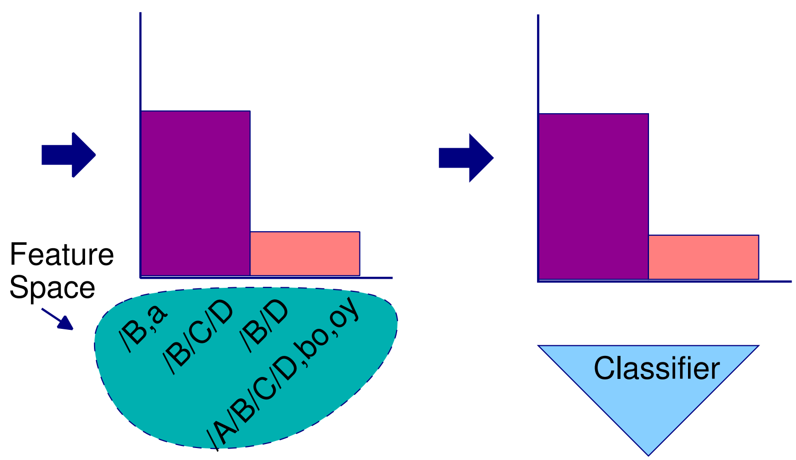


Intuition for Classification-based Histograms

Query	sel
/A/B/C/D=boy	1
/B/C/D	3
/B/D	1
/B=a*	5

Query Workload





- Use Naive Bayesian Classifiers (NBC)
- Learn NBC in an on-line manner

Bayesian Classifiers (BC)

- **Goal:** learn the mapping from feature vectors to bucket IDs.
- Model features as r.v.'s $\vec{X} = X_1, \ldots, X_k$, bucket ID as r.v. B, and the mapping as a joint probability distribution.
- Given any feature vector $\vec{x}$, the bucket is computed as

$$\begin{aligned}\widehat{b} &= \arg\max_{b \in \mathcal{B}} P(B{=}b \mid \vec{X}{=}\vec{x}) \\ &= \arg\max_{b \in \mathcal{B}} P(B{=}b) P(\vec{X}{=}\vec{x} \mid B{=}b).\end{aligned}$$

- Naive BC assumes independence of features X_i given B.

A CXHist Histogram

consists of a set of buckets and each bucket b stores:

1. $sum(b)$, the sum of the selectivities of all the query feedback that is associated with bucket b,
2. $cnt(b)$, one plus the number of query feedback seen so far that is associated with bucket b,
3. $\{P(X_i|B{=}b) : i = 1, \ldots, k\}$, a set of query feature probability distributions. One distribution is stored for each feature random variable X_i.

Modeling Queries

- Query type: exact match & substring predicates on values reachable by given path ID.

- A query is modeled as a set of features. E.g. The exact match query (5,@LIM$) can be modeled using a pathID feature with value 5 and a series of 2-gram features with values @L, LI, IM, M$.

- Each feature is associated with a random variable (X_i). E.g. T for the pathID, and G_i for the 2-gram features.

- We assume stationarity for G_i, so that we only need to store one distribution for all the G_i.

Estimating Selectivity using CXHist

- Map the given query to its feature vector $\vec{x} = \langle x_1, \ldots, x_k \rangle$
- Models features and bucket as a joint probability distribution.
- Find the bucket for $\widehat{b}$ for $\vec{x}$ using the naive bayesian classifier,

$$\widehat{b} = \mathrm{arg}\max_{b \in \mathcal{B}} \left\{ P(B{=}b) \prod_{i=1}^{n} P(X_i{=}x_i | B{=}b) \right\}.$$

- Compute the selectivity as

$$est(\widehat{b}) = \frac{sum(\widehat{b})}{cnt(\widehat{b})}.$$

Example

B	sum	cnt
0	3	3
1	20	2

B	T	N(T,B)
0	5	2
1	5	1

B	G	N(G,B)
0	@L	2
0	LI	2
0	IM	2
0	M$	2
1	@L	1
1	LI	1
1	IM	1

- Query: $(5, \text{@LIM}) \rightarrow \langle 5, \text{@L}, \text{LI}, \text{IM} \rangle$
- Compute associated bucket,

$$P(B{=}0|\vec{X}) \propto \frac{2}{3} \times \frac{2}{2} \times \frac{2}{8} \times \frac{2}{8} \times \frac{2}{8} = \frac{2}{192}$$

$$P(B{=}1|\vec{X}) \propto \frac{1}{3} \times \frac{1}{1} \times \frac{1}{3} \times \frac{1}{3} \times \frac{1}{3} = \frac{1}{81}$$

- Compute selectivity as

$$est(1) = \frac{20}{2} = 10.$$

Initializing CXHist

1. *Clustering.* If a sample query workload is available, use the MaxDiff or the Lloyd-Max quantization algorithm.

2. *Uniform intervals.* E.g. $\{0, 10, 20, 30, \ldots\}$.

3. *Exponential intervals.* E.g. $\{1, 2, 4, 8, 16, \ldots\}$.

4. *Uniform-exponential hybrid.* E.g., for 10 buckets and 5 exponential values in the interval $[1, 66]$, we could use $\{1, 2, 4, 8, 16, 26, 36, 46, 56, 66\}$.

Updating CXHist with Query Feedback

- If no classification error, increment feature distributions, bucket sum and count associated with the query feature vector $\vec{x}$.

- Otherwise the classification is wrong, $\hat{b} \neq b*$, which implies that posterior probability $P(B{=}b*|\vec{X})$ is not the maximum.

- Perform some number of iterations of gradient descent so that $P(B{=}b*|\vec{X})$ becomes the maximum.

- Each step of gradient descent, we update the count $w_i = N(X_i{=}x_i, B{=}b*)$ using,

$$w_i^{(t+1)} \leftarrow w_i^{(t)} - \gamma \frac{\partial E(\vec{x})}{\partial w_i},$$

Worked Example

B	sum	cnt
1	1	1
2	2	1
3	4	1
4	8	1
5	16	1

B	T	N(T,B)

B	G	N(G,B)

Consider a 5-bucket CXHist (left) and the following query workload:

No.	Path ID	String	Selectivity
1	0	@LIM$	2
2	1	@MIN	20
3	0	@LIM	10
4	0	@LIM$	2
5	0	IM	18

Worked Example : Query 1

B	sum	cnt
1	1	1
2	2	1
3	4	1
4	8	1
5	16	1

B	T	N(T,B)

B	G	N(G,B)

- Query (0,@LIM$),$\sigma = 2$
- Feature distributions are empty
  - $\rightarrow$ posterior is flat
  - $\rightarrow$ defaults to $B = 1$
  - $\rightarrow$ $\hat{\sigma} = 1$ (50% rel. err.).
- **Update**
  - $\rightarrow$ closest bucket is $B = 2$.
  - $\rightarrow$ update bucket counts.
  - $\rightarrow$ increment feature counts.
- Posterior is max at $B = 2$.

Worked Example : Query 1

B	sum	cnt
1	1	1
2	4	2
3	4	1
4	8	1
5	16	1

B	T	N(T,B)
2	0	1

B	G	N(G,B)
2	@L	1
2	LI	1
2	IM	1
2	M$	1

- Query (0,@LIM$),$\sigma = 2$
- Feature distributions are empty
  - $\rightarrow$ posterior is flat
  - $\rightarrow$ defaults to $B = 1$
  - $\rightarrow$ $\hat{\sigma} = 1$ (50% rel. err.).
- **Update**
  - $\rightarrow$ closest bucket is $B = 2$.
  - $\rightarrow$ update bucket counts.
  - $\rightarrow$ increment feature counts.
- Posterior is max at $B = 2$.

Worked Example : Query 2

B	sum	cnt
1	1	1
2	4	2
3	4	1
4	8	1
5	16	1

B	T	N(T,B)
2	0	1

B	G	N(G,B)
2	@L	1
2	LI	1
2	IM	1
2	M$	1

- Query (1,@MIN),$\sigma = 20$
- Posterior is flat
  - $\rightarrow$ defaults to $B = 1$
  - $\rightarrow$ $\hat{\sigma} = 1$ (95% rel. err.).
- **Update**:
  - $\rightarrow$ closest bucket is $B = 5$.
  - $\rightarrow$ update bucket counts.
  - $\rightarrow$ increment feature counts.
- Posterior is max at $B = 5$.

Worked Example : Query 2

B	sum	cnt
1	1	1
2	4	2
3	4	1
4	8	1
5	36	2

B	T	N(T,B)
2	0	1
5	1	1

B	G	N(G,B)
2	@L	1
2	LI	1
2	IM	1
2	M$	1
5	@M	1
5	MI	1
5	IN	1

- Query (1,@MIN),$\sigma = 20$
- Posterior is flat
  - $\rightarrow$ defaults to $B = 1$
  - $\rightarrow$ $\hat{\sigma} = 1$ (95% rel. err.).
- **Update**:
  - $\rightarrow$ closest bucket is $B = 5$.
  - $\rightarrow$ update bucket counts.
  - $\rightarrow$ increment feature counts.
- Posterior is max at $B = 5$.

Worked Example : Query 3

B	sum	cnt
1	1	1
2	4	2
3	4	1
4	8	1
5	36	2

B	T	N(T,B)
2	0	1
5	1	1

B	G	N(G,B)
2	@L	1
2	LI	1
2	IM	1
2	M$	1
5	@M	1
5	MI	1
5	IN	1

- Query (0,@LIM),$\sigma = 10$
- $\rightarrow$ Posterior is max at $B = 2$
  $\rightarrow$ $\hat{\sigma} = 2$ (80% rel. err.).
- **Update**:
  $\rightarrow$ closest bucket is $B = 4$.
  $\rightarrow$ update bucket counts.
  $\rightarrow$ increment feature counts.
- Posterior is max at $B = 4$.

Worked Example : Query 3

B	sum	cnt
1	1	1
2	4	2
3	4	1
4	18	2
5	36	2

B	T	N(T,B)
2	0	1
5	1	1
4	0	1

B	G	N(G,B)
2	@L	1
2	LI	1
2	IM	1
2	M$	1
5	@M	1
5	MI	1
5	IN	1
4	@L	1
4	LI	1
4	IM	1

- Query (0,@LIM),$\sigma = 10$
- $\rightarrow$ Posterior is max at $B = 2$
  $\rightarrow$ $\hat{\sigma} = 2$ (80% rel. err.).
- **Update**:
  $\rightarrow$ closest bucket is $B = 4$.
  $\rightarrow$ update bucket counts.
  $\rightarrow$ increment feature counts.
- Posterior is max at $B = 4$.

Worked Example : Query 4

B	sum	cnt
1	1	1
2	4	2
3	4	1
4	18	2
5	36	2

B	T	N(T,B)
2	0	1
5	1	1
4	0	1

B	G	N(G,B)
2	@L	1
2	LI	1
2	IM	1
2	M$	1
5	@M	1
5	MI	1
5	IN	1
4	@L	1
4	LI	1
4	IM	1

- Query (0,@LIM$),$\sigma = 2$
- Posterior is max at $B = 2$
  $\rightarrow \hat{\sigma} = 2$ (0 error).
- **Update**:
  $\rightarrow$ no classification error.
  $\rightarrow$ update bucket counts.
  $\rightarrow$ increment feature counts.
- No further updates needed.

Worked Example : Query 4

B	sum	cnt
1	1	1
2	6	3
3	4	1
4	18	2
5	36	2

B	T	N(T,B)
2	0	2
5	1	1
4	0	1

B	G	N(G,B)
2	@L	2
2	LI	2
2	IM	2
2	M$	2
5	@M	1
5	MI	1
5	IN	1
4	@L	1
4	LI	1
4	IM	1

- Query (0,@LIM$),$\sigma = 2$
- Posterior is max at $B = 2$
  $\rightarrow \hat{\sigma} = 2$ (0 error).
- **Update**:
  $\rightarrow$ no classification error.
  $\rightarrow$ update bucket counts.
  $\rightarrow$ increment feature counts.
- No further updates needed.

Worked Example : Query 5

B	sum	cnt
1	1	1
2	6	3
3	4	1
4	18	2
5	36	2

B	T	N(T,B)
2	0	2
5	1	1
4	0	1

B	G	N(G,B)
2	@L	2
2	LI	2
2	IM	2
2	M$	2
5	@M	1
5	MI	1
5	IN	1
4	@L	1
4	LI	1
4	IM	1

- Query (0,IM),$\sigma = 18$
- Posterior is max at $B = 2$
  $\rightarrow \hat{\sigma} = 2$ (89% rel. err.).
- **Update**:
  $\rightarrow$ closest bucket is $B = 5$.
  $\rightarrow$ update bucket counts.
  $\rightarrow$ since posterior is 0 at $B = 5$,
  increment feature counts.
- But max is still not $B = 5$!
- Do gradient descent.

Worked Example : Query 5

B	sum	cnt
1	1	1
2	6	3
3	4	1
4	18	2
5	54	3

B	T	N(T,B)
2	0	2
5	1	1
4	0	1
5	0	1

B	G	N(G,B)
2	@L	2
2	LI	2
2	IM	2
2	M$	2
5	@M	1
5	MI	1
5	IN	1
4	@L	1
4	LI	1
4	IM	1
5	IM	1

- Query (0,IM),$\sigma = 18$
- Posterior is max at $B = 2$
  $\rightarrow$ $\hat{\sigma} = 2$ (89% rel. err.).
- **Update**:
  $\rightarrow$ closest bucket is $B = 5$.
  $\rightarrow$ update bucket counts.
  $\rightarrow$ since posterior is 0 at $B = 5$,
  increment feature counts.
- But max is still not $B = 5$!
- Do gradient descent.

Worked Example : Query 5

B	sum	cnt
1	1	1
2	6	3
3	4	1
4	18	2
5	54	3

B	T	N(T,B)
2	0	2
5	1	1
4	0	1
5	0	1

B	G	N(G,B)
2	@L	2
2	LI	2
2	IM	2
2	M$	2
5	@M	1
5	MI	1
5	IN	1
4	@L	1
4	LI	1
4	IM	1
5	IM	1

- Query (0,IM),$\sigma = 18$
- **Gradient descent**:
  - Let $w_t{=}N(T{=}0, B{=}5)$ and $w_q{=}N(G{=}\text{IM}, B{=}5)$
  - Compute the deltas,

$$\Delta w_t = -0.034$$
$$\Delta w_q = -0.052.$$

  - Normalizing, we have $\Delta w_t = 1, \Delta w_q = 1.5$
- Max is now $B = 5$!

Worked Example : Query 5

B	sum	cnt
1	1	1
2	6	3
3	4	1
4	18	2
5	54	3

B	T	N(T,B)
2	0	2
5	1	1
4	0	1
5	0	**2**

B	G	N(G,B)
2	@L	2
2	LI	2
2	IM	2
2	M$	2
5	@M	1
5	MI	1
5	IN	1
4	@L	1
4	LI	1
4	IM	1
5	IM	**2.5**

- Query (0,IM),$\sigma = 18$
- **Gradient descent**:
  - Let $w_t{=}N(T{=}0, B{=}5)$ and $w_q{=}N(G{=}\text{IM}, B{=}5)$
  - Compute the deltas,

$$\Delta w_t = -0.034$$
$$\Delta w_q = -0.052.$$

  - Normalizing, we have $\Delta w_t = 1, \Delta w_q = 1.5$
- Max is now $B = 5$!

Pruning CXHist

- When histogram size reaches $triggersize$ bytes, the histogram is pruned down to $targetsize$ bytes.

- Small counts in the feature distribution are discarded/pruned.
  - Small counts suggest less frequent use.
  - Small counts are less likely to affect the maximum point of the posterior.

Experiments

Dataset : DBLP XML data. 5M leaf nodes (path-string pairs), 2M are distinct.

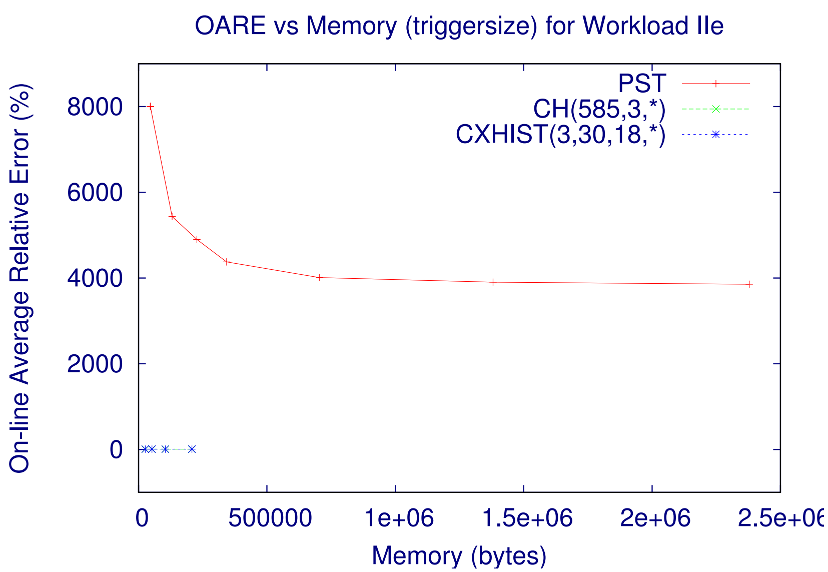
Workload generation : sample from 2M distinct pairs using Gaussian distribution.

Query workload type : exact match, substring match and mixed.

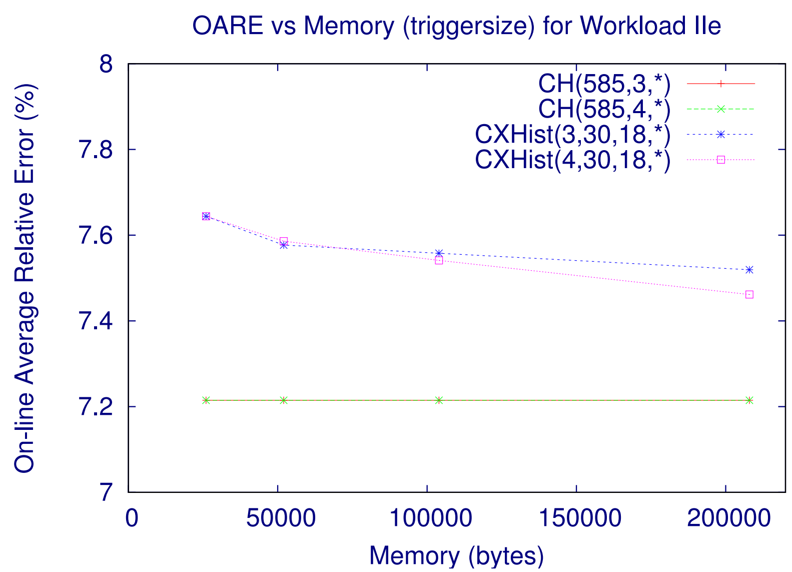
Comparisons : one pruned suffix tree (PST) per pathID, compressed histogram (CH).

Metric : on-line average relative error.

Accuracy vs Memory: Exact Match



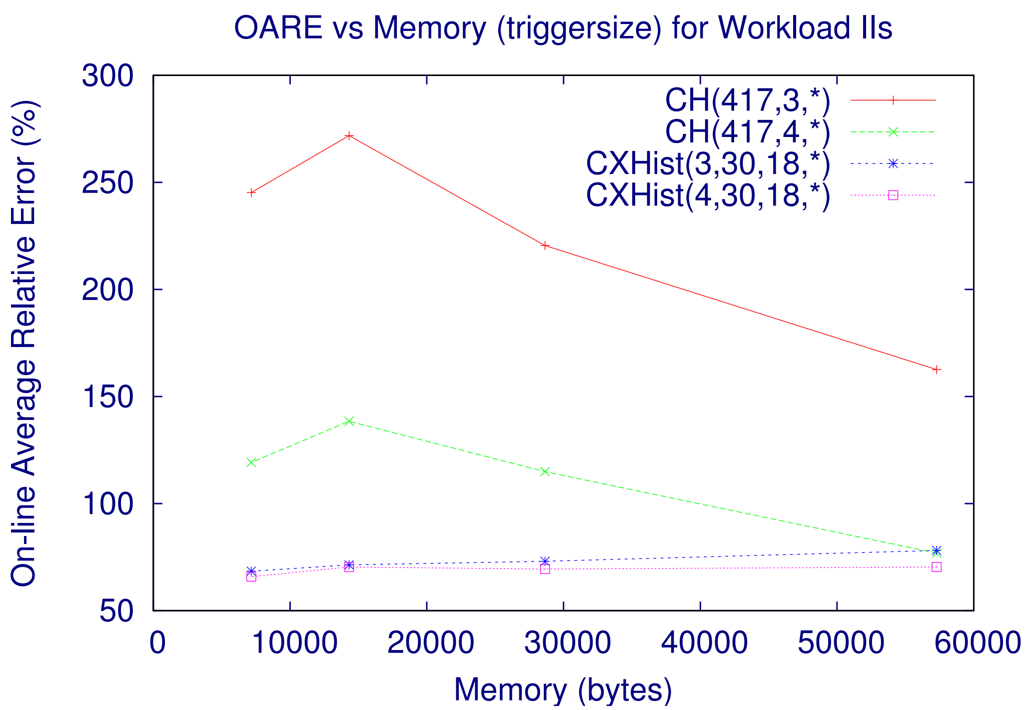
Overview



Detailed

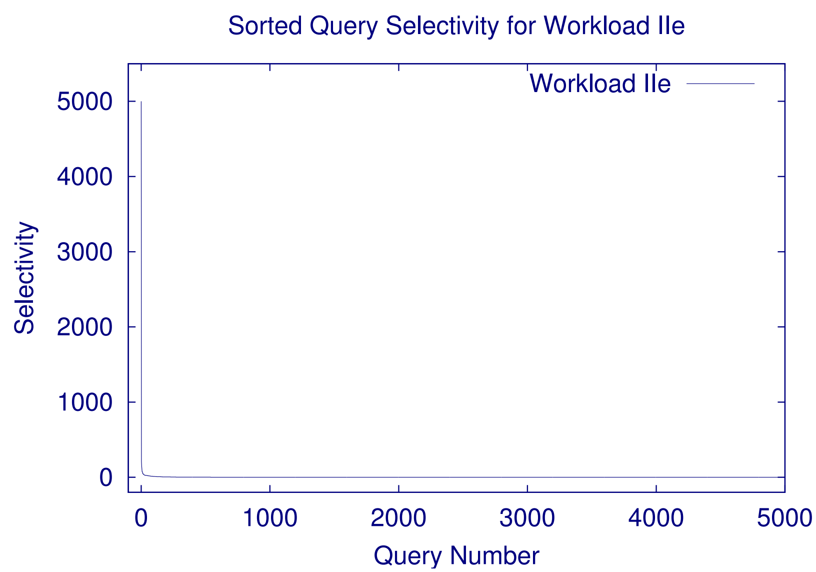
CH is slightly better than CXHist for exact match workload.

Accuracy vs Memory: Substring Match

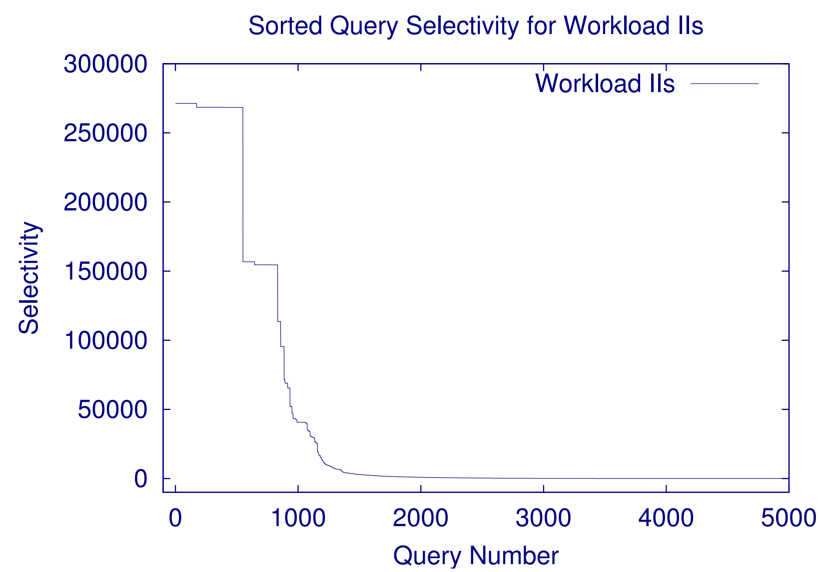


CXHist is more accurate for substring match workload.

Selectivity Distribution in Workload



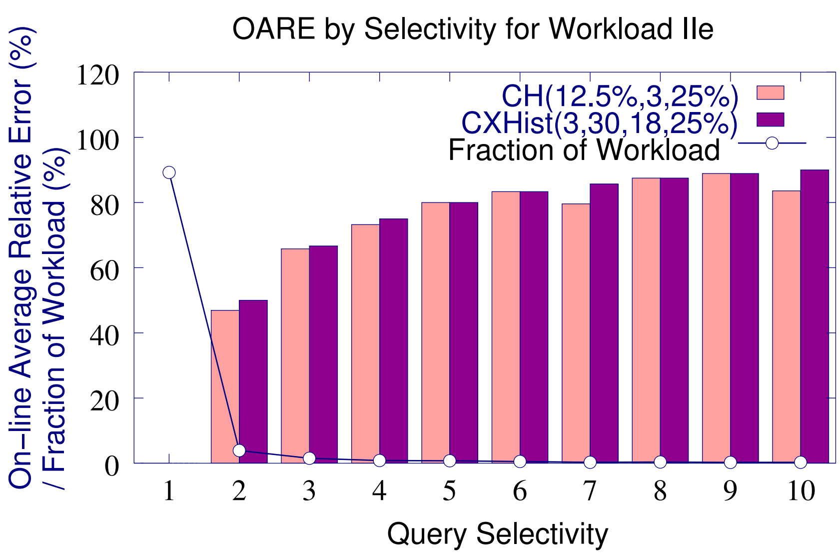
Exact Match Workload



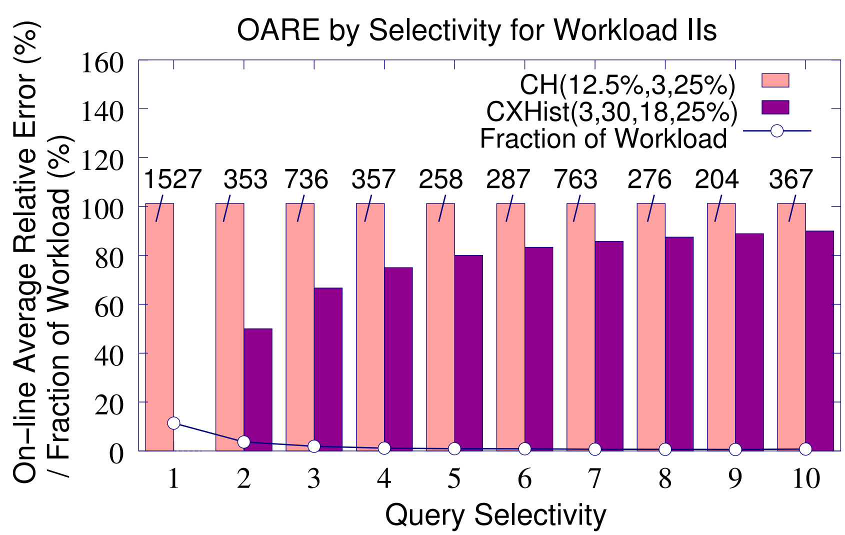
Substring Match Workload

Exact match workload is more skewed than substring match workload.

Accuracy Partitioned by Selectivity



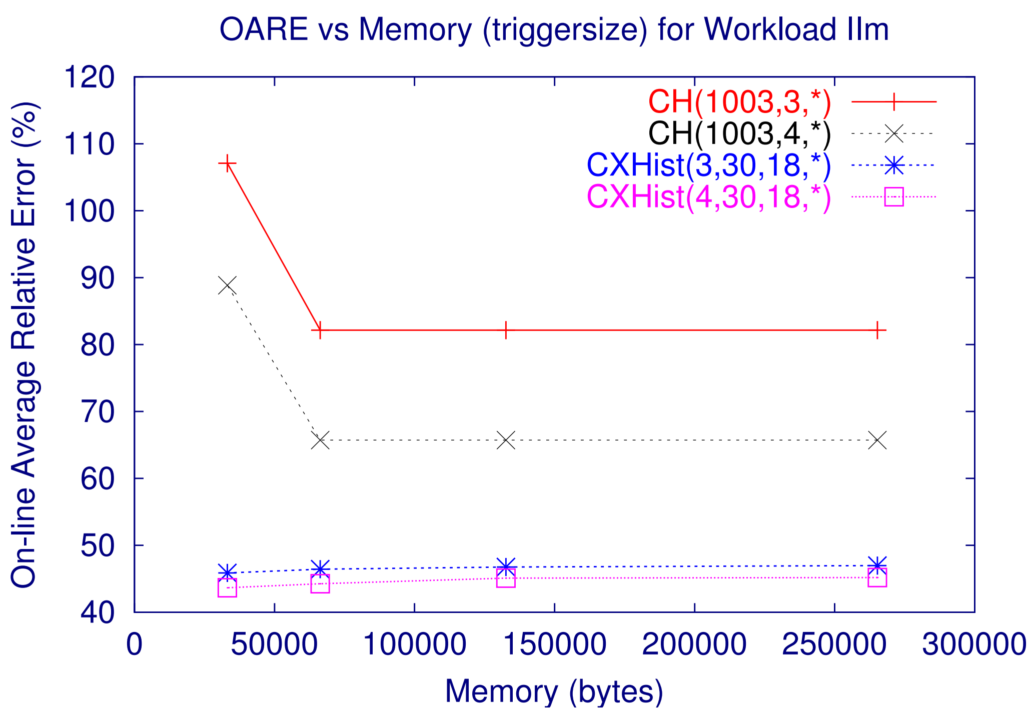
Exact Match Workload



Substring Match Workload

Performance of CXHist is consistent, but CH performs poorly on substring workload.

Accuracy of CXHist on Mixed Workload



Workload: a mixture of 5000 substring queries and 5000 exact match queries.

Conclusions

- CXHist is a new type of histogram that uses feature distributions and Bayesian classification techniques to capture the mapping between queries and their selectivity.
- CXHist is on-line: it gathers statistics from query feedback rather than from costly data scans, and hence adapts to changes in workload characteristics and in the underlying data.
- CXHist is general and not limited to XML data: it can be used for multidimensional string data in relational databases as well.
- CXHist can be easily implemented and deployed in practice.