DATABASE HISTOGRAMS

E0 261

Jayant Haritsa
Computer Science and Automation
Indian Institute of Science

MOTIVATION

- System R’s assumption of uniform distribution of values over data domain rarely holds true in practice

$\Rightarrow$ Garbage In, Garbage Out

Solutions

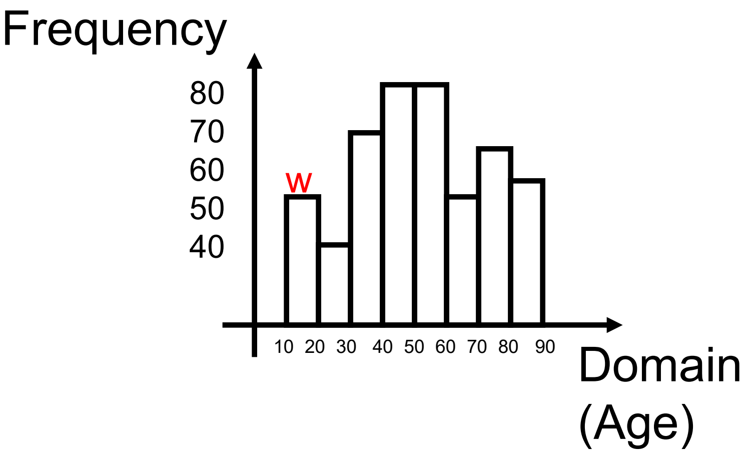
- Use a “classical” distribution (e.g. Gaussian, Exponential, Zipf, etc.) to model the data
  - Problem is that real-life data is often not like these also !
  - Further, not all distributions are easily computable !
- Approximate the distribution using histograms
  - Several flavors available
  - Maintenance could be a pain
- Use sampling for dynamic estimation
  - Expensive since done at run-time
  - Space-efficient

HISTOGRAMS

- Domain of attribute A is partitioned into B buckets, and a uniform distribution is assumed within each bucket. That is, frequency of a value in a bucket is approximated by the average of the frequencies of all domain values assigned to the bucket.

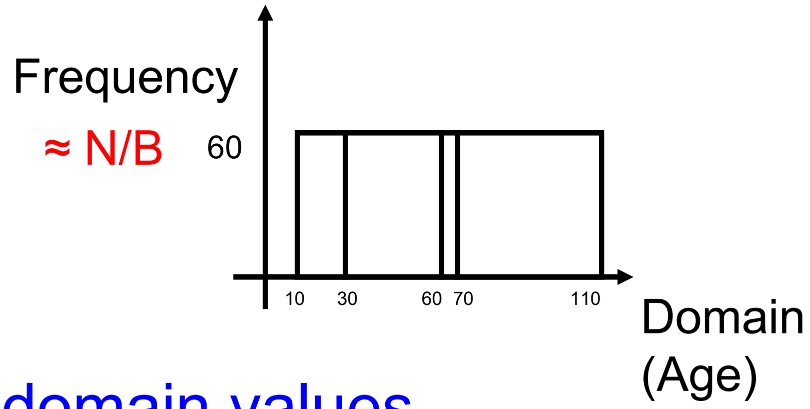
- Trivial histogram: Single bucket that assigns the same frequency (N ÷ V) to all attribute values.
  - equivalent to System R approach

EQUI-WIDTH HISTOGRAMS



- Frequency versus ordered domain values
- All buckets of same width
- MaxErr [Sel <X] = FreqFraction [X_{PB}]
- Easy to construct and maintain
- Example: CRICKET !

EQUI-DEPTH HISTOGRAMS



- Frequency versus ordered domain values
- All buckets of same height (or depth)
- MaxErr [Sel <X] = 1 / B
- Comparatively difficult to construct and maintain
  - Require to sort the relation first to find the B “quantiles”
  - Or use index if available
  - Cheaper approximate technique based on sampling typically used

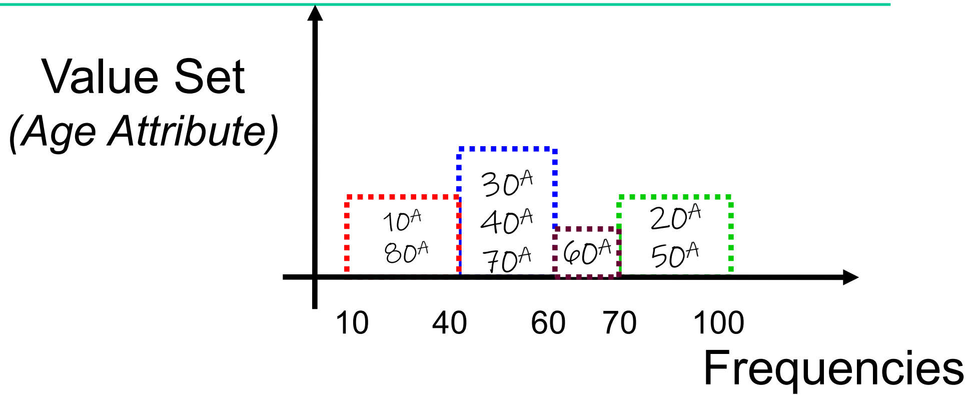
Sampling Technique

- Take a random sample of database, sort the sampled tuples, and use the boundaries established by them to form the approximate buckets
- How many samples to take ?
  - Based on Kolmogorov statistic

Kolmogorov Statistic

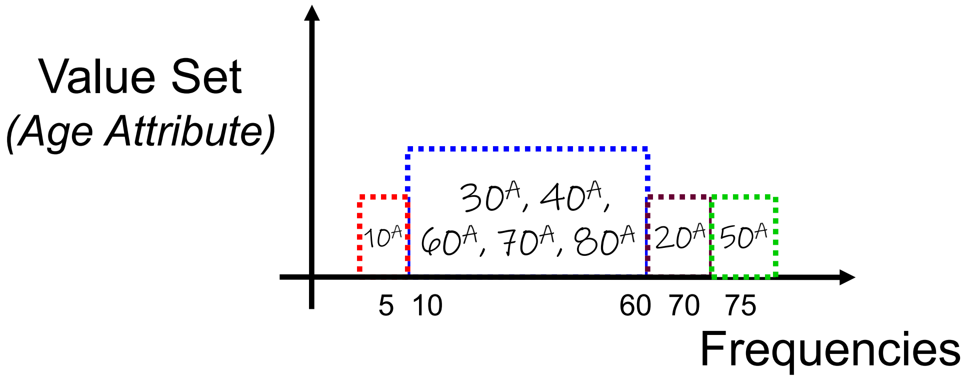
- Let α be the fraction of tuples that satisfy a property (in our case, property is query box). Let β be the fraction of tuples in sample that lie in same box. Then, the K-statistic is that $|\alpha-\beta| \leq d$ (precision) with probability $\geq p$ (confidence) if sample size is $\geq n$.
- Given p and d, n can be evaluated. For example, if d = 0.05 and p = 0.99, n = 1064
  - Note, does not depend on number of tuples in database!

SERIAL HISTOGRAMS



- Domain values versus ordered frequencies !
- Frequency assigned to domain values in a bucket is avg frequency of bucket or mid-frequency of bucket range (based on information stored).
- Optimal histogram is a serial histogram
  - paper covered in TIDS (E0 361) course
  - in particular, the serial histogram that minimizes $\sum n_i V_{i,,}$ where n_i is number of attribute values placed in bucket B_i , and V_i is variance of these frequencies
- Difficult to construct and maintain
  - Queries are usually on value domain, not frequency domain
  - Have to explicitly represent all domain values – ranges not possible
  - Identification of optimal is exponential in number of buckets

END-BIASED HISTOGRAMS



- Special case of serial histograms
- Highest and lowest sets of frequencies are explicitly maintained in separate individual buckets
- Remaining (middle) frequencies are approximated together in a single bucket.
- Optimal end-biased histogram minimizes $\sum n_i V_i$
- Performance "close" to that of optimal serial histogram.
- Easy to construct and maintain
  - Attribute Values corresponding to the "middle" bucket do not have to be stored explicitly: can be identified by negation.
  - Linear time complexity in number of buckets to identify optimal

Serial Histogram Construction

- Data Collection:

```
select A, Count(*)
from R
group by A
```

  – requires sorting

- Approximate by sampling when building high-end-biased serial histograms

In Practice

- Histograms used in most commercial systems.
- Approximate equi-depth histograms built on individual attributes with about 10-20 buckets using a one-pass sampling (usually 1000-2000 samples) approach.
- Results in large errors for multi-attribute queries, but technique proposed in SIGMOD 99 for efficiently constructing and maintaining multi-dimensional histograms (covered in TIDS course)

END DATABASE HISTOGRAMS

E0 261