Random Sampling from Databases – A Survey

Frank Olken and Doron Rotem
Presenter: Alin Dobra

January 28, 2004

Why Sample?

- External use: provide a sample of the result or approximate answer for evaluation purposes
  - estimate result of aggregate queries
  - retrieve a sample of records from a database query for subsequent processing
- Internal use: support query optimization
- Provide privacy protection for records on individuals contained in statistical databases: privacy preserving data-mining

Advantages of Sampling

- Reduces time to answer queries
- Reduces the time to *post-process* the query results
  - computation, financial auditing, physical inspection
- Helps in making query optimization decisions

Sampling Operator: useful within DBMS

Sampling Queries

Sample set/bag that is result of SQL query:

```
SAMPLE 100 OF
    SELECT EMP_ID, ADDRESS
    FROM EMPLOYEE WHERE SALARY>$50K
```

Sampling for approximation purposes:

```
SELECT AVG(SALARY), MAX(SALARY)
FROM EMPLOYEE
WHERE EMP_ID IN ( SAMPLE 100 OF
    SELECT EMP_ID
    FROM DOCTORS
    WHERE SPECIALITY = 'SURGEONS' )
```

Sampling Operator: useful for processing queries of the above type

- Generally useful only if it results in significant savings

Sampling Terminology

- **Population:** set of record (tuples) out of which the sample is selected
- **Fixed size random sample:** sample size is a specified constant
- **Binomial random sample:** size of the sample is a binomial random variable
  - scanning the set of records and including each record with fixed probability
- **Simple random sample without replacement (SRSWOR):**
  - subset of a population
  - each element of the population is equally probable to be included
  - no duplicates are allowed
  - **Generation:**
    - scan file sequentially and pick elements randomly
    - pick random permutation of data and take the first elements
  - **Properties:**
    - elements of the sample (as random variables) are **not independent**

Sampling Terminology (cont.)

Simple random sample with replacement (SRSWR):

- each element of the population is equally probable to be in any position(element) of the sample
- duplicates allowed
- **Generation:** pick an element randomly from the population; concatenate it to the sample
- **Properties:**
  - elements of the sample (as random variables) are **independent**

Stratified random sample: partition the population (e.g. by sex) than take SRS of specified size from each strata

- sample sizes usually allocated proportional to size of strata

Weighted random sample: inclusion probabilities are not uniform

- probability proportional with function of tuple
- probability proportional with size, dollar unit sample

Sampling Terminology (cont.)

Clustered Samples:

- **Generation:**
  - first sample a *cluster unit*
  - sample several elements within unit
- Cheaper to obtain generally
- More complex to analyze since they are not independent

Systematic Samples:

- Take every k-th element of a file with random starting point
- If tuples are not random in the file strange correlations can surface
- Analysis difficult

Types of Sampling Procedures

- **Single stage:** choose a sample of specific size
- **Two stage:**
  - choose a small sample
  - use information in small sample to decide how large second sample has to be
    - e.g. to obtain a certain level of accuracy
- **Sequential sampling:**
  - sample from the file iteratively
  - after each sample element decide if to continue sampling
  - alternative name: *adaptive sampling*
- **Group sequential sampling:** decide if to continue after groups of sample elements are obtained

Classification of Sampling Algorithms

- **Iterative:**
  - algorithms that loop over the data
  - one sample generated per loop
  - usually samples with replacement are generated
- **Batch:**
  - generate a group of sample elements at a time
  - avoid redundant reading of the disk pages
- **Sequential:** only sequential scan of the file is allowed
  - typically generate without replacement samples
  - file size might have to be known in advance
  - skipping is allowed – increases efficiency sometimes
- **Reservoir sampling:** sequential algorithm when size of the file is unknown
  - useful for sampling out of results of queries on-the-fly when size of result is not known

Design/Analysis of Sampling Algorithms

1. Define type of sampling to be performed
    - e.g. simple sample with replacement
2. Define constraints of the algorithm
    - e.g. sequential sampling
3. Come up with a candidate algorithm
4. Prove (formally) that the algorithm produces the desired type of sampling
    - show formally that the samples produced by the algorithm have the statistical properties desired
5. Argue/Prove that the algorithm satisfies the desired constraints
    - sometimes this is trivial, e.g. algorithm is sequential
    - sometimes requires a rigorous proof, e.g. algorithm terminates
6. Prove that the algorithm is optimal (very rare)

Reservoir Sampling Method (Fan,Muller,Rezucha 1962)

- **Type of sampling:** simple random sampling without replacement
- **Constraints:** sequential sampling, population size unknown
- **Algorithm:**

  1. Put first s elements in sequence in the *reservoir* of size s
  2. As each element of the sequence is encountered

     (a) accept this k-th element with probability $\frac{s}{k}$

     (b) if accepted, the k-th element displaces a random element in reservoir

- **Analysis:**

  – the main idea is to show that at any step the *reservoir* contains SRSWOR of sequence scanned

  – suppose this holds before looking at k-th element then show it holds after

- **Constraint satisfaction:** direct by design

Reservoir Sampling Method – Analysis

Invariant: Any tuple $i \in \{1 \ldots k\}$ is in reservoir R_k with probability $P[i \in R_k] = \frac{s}{k}$

Base Case: This clearly holds if $k = s$ since then $P[i \in R_s] = 1$

Induction: Suppose it holds for R_k and we encounter $k' = k+1$

$$
\begin{aligned}
P[i \in R_{k'}] &= P[(k' \notin R_{k'}) \wedge (i \in R_k) \quad \vee \quad (k' \in R_{k'}) \wedge (i \in R_k) \wedge (i \notin R_k - R_{k'})] \\
&= P[(k' \notin R_{k'}) \wedge (i \in R_k)] + P[(k' \in R_{k'}) \wedge (i \in R_k) \wedge (i \notin R_k - R_{k'})] \text{ (disjunct)} \\
&= P[k' \notin R_{k'}] P[i \in R_k] + P[k' \in R_{k'}] P[i \in R_k] P[i \notin R_k - R_{k'} | k' \in R_{k'} \wedge i \in R_k] \text{ (indep.)} \\
&= \left(1 - \frac{s}{k+1}\right) \frac{s}{k} + \frac{s}{k+1} \frac{s}{k} \left(1 - \frac{1}{s}\right) \\
&= \frac{s}{k} - \frac{s^2}{k(k+1)} + \frac{s^2}{k(k+1)} - \frac{s}{k(k+1)} \\
&= \frac{s}{k} \frac{k}{k+1} \\
&= \frac{s}{k+1} \text{ (the invariant is maintained)}
\end{aligned}
$$