Random Sampling

Florian Schoppmann

August 24, 2010

Sampling Algorithms

Input:

- List$[1 \ldots N]$
- Length of database N (if known)
- Length of sample n

Output:

- Sample$[1 \ldots n]$

Considerations

- Online vs. random-access
- Sequential vs. non-sequential
- Samples for independent categories

Desiderata:

- Parallelizable
- If random access, running time close to $O(n)$
- Constant memory

Random Indices

1: $m \leftarrow 0$
2: **while** $m < n$ **do**
3: $\quad R \leftarrow \text{random}(\{1 \ldots N\})$
4: $\quad$ **if** $\text{List}[R] \notin \text{Sample}$ **then**
5: $\quad\quad m \leftarrow m + 1$
6: $\quad\quad \text{Sample}[m] \leftarrow \text{List}[R]$

- ca. $N \ln \frac{N}{N-n+1}$ iterations in expectation
- Space/time trade off in line 4

Random Remaining Indices

for $m \leftarrow 1, \ldots, n$ **do**
    $R \leftarrow \text{random}(\{1 \ldots N - m + 1\})$
    $j \leftarrow$ index of R'th non-null element in List
    $\text{Sample}[m] \leftarrow \text{List}[j]$
    $\text{List}[j] \leftarrow \text{null}$

- Prohibitive running time $\Theta(nN)$
- Modifies List

The Fisher-Yates Shuffle

for $m \leftarrow 1, \ldots, n$ **do**
    $R \leftarrow \text{random}(\{m \ldots N\})$
    Swap $\text{List}[m]$ and $\text{List}[R]$
$\text{Sample}[1 \ldots n] \leftarrow \text{List}[1 \ldots n]$

- Running time $\Theta(n)$
- Modifies List

Probabilistic Sampling

for $t \leftarrow 1, \ldots, N$ **do**
    with probability $\frac{n}{N}$ **do**
        Append List$[t]$ to Sample

- Running time $\Theta(N)$
- Only expected sample size n (mean of $B(N, \frac{n}{N})$)
- Standard deviation $\sqrt{n(1 - n/N)}$

Selection Sampling

$m \leftarrow 0$
for $t \leftarrow 1, \ldots, N$ **do**
    with probability $\frac{n-m}{N-t}$ **do**
        $m \leftarrow m + 1$
        $\text{Sample}[m] \leftarrow \text{List}[t]$

- Running time $\Theta(N)$
- Completely unbiased!

Random Number Generation

(Digression)

Running time $O(n)$ possible by skipping rows?

Idea 1:

- Let $S \in \{0 \ldots N - n\}$ RV for # rows to skip

$$\Pr[S \leq s] = 1 - \frac{(N-n)^{\underline{s+1}}}{N^{\underline{s+1}}}$$

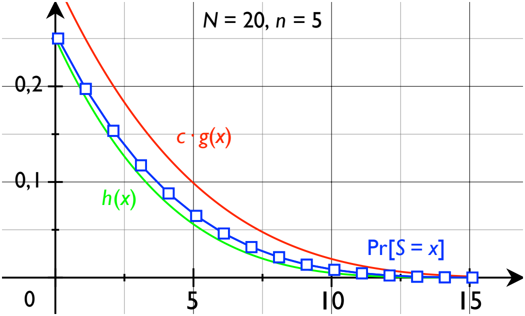
Idea 2 (Vitter, 1984):

- von Neumann's rejection & "squeeze" method

Vitter (1984)

$$c = \frac{N}{N-n+1}, \quad g(x) = \frac{n}{N}\left(1 - \frac{x}{N}\right)^{n-1},$$

$$h(x) = \frac{n}{N}\left(1 - \frac{x}{N-n+1}\right)^{n-1}$$



Reservoir Sampling

```
Sample[1 ... n] ← List[1 ... n]
for t ← n + 1, ..., N do
    with probability n/t do
        R ← random({1 ... n})
        Sample[R] ← List[t]
```

- Completely unbiased!
- $O(n(1 + \log \frac{N}{n}))$ by optimizing (Vitter, 1985)

Reservoir, with Replacement

for $t \leftarrow 1, \ldots, N$ **do**
    for $i \leftarrow 1, \ldots, n$ **do**
        with probability $\frac{1}{t}$ **do**
            $\text{Sample}[i] \leftarrow \text{List}[t]$

- Completely unbiased!

Bibliography

- Knuth (1997):
  The Art of Computer Programming, Vol. 2
- Vitter (1984):
  Faster methods for random sampling
- Vitter (1985):
  Random sampling with a reservoir
- Park, Ostrouchov, Samatova, Geist (2004):
  Reservoir-Based Random Sampling with Replacement from Data Stream