How to count without counting

Keith Briggs

Keith.Briggs@bt.com

research.btexact.com/teralab/keithbriggs.html

Tempura seminar 2003 September 04 15:00
The eprint

- Loglog counting of large cardinalities
- Marianne Durand and Philippe Flajolet
- Engineering and applications track of the 11th Annual European symposium on algorithms (ESA 2003, Budapest Sept 15-20)
- to be published by Springer, Lecture Notes in Computer Science
- algo.inria.fr/flajolet/Publications/DuFl03.ps.gz
Algorithms

★ A precisely defined, provably correct (for valid inputs) computational procedure for a specific problem ★

★ Abu ʿAbd Allâh Muḥammad ibn Mūsâ al-Khwârizmî

▷ born: about 780 in Baghdad
▷ died: about 850

★ e.g. Euclid’s algorithm for the greatest common divisor of two positive integers:

```python
def gcd(x, y):
    while y:
        y, x = x % y, y
    return x
```
Types of computational procedure

★ deterministic algorithm

- always returns the same output for the same input
- output proved always correct
- always terminates in finite time
- involves no random (stochastic) steps

★ heuristic

- not proved to always return the correct result
- usually involves some ‘rules of thumb’ - arbitrary but reasonable-looking steps
- not proved to terminate in finite time
- an ‘engineering’ solution

★ stochastic algorithm

- output proved usually correct, within certain probabilistic bounds
- may involve random (stochastic) steps
- may be much faster than a deterministic algorithm for the same problem
A deterministic counting algorithm

Problem: given a multiset $M$ (a collection of objects, possibly with repeats), determine how many different objects there are in $M$.

obvious algorithm:
- set $D = \{\}$ (the empty set)
- for each $x$ object in $M$...
  - see if $x$ is in $D$, and if not, add it to $D$
- count the numbers of elements in $D$, and return it

$D$ is a list which grows, so a lot of time is wasted in memory allocation.

as $D$ becomes large, it becomes slower and slower to find whether a given $x$ is in $D$.

Can we do better with a stochastic algorithm?
The Durand and Flajolet algorithm

- define $\rho(b_1 b_2 b_3 \ldots) \equiv \text{argmin}_k \{k \text{ such that } b_k = 1\}
- choose parameter $k$ (typically 10 to 16)
- $m = 2^k$, buckets $M_1, M_2, M_3, \ldots, M_m$, initialized to 0
- $h$ = a hash function (e.g. 32 bits)

for each word $x$ in the file:
- $y = h(x)$
- $j$=value of first $k$ bits of $y$
- $l$=value of last (hash size$-k$) bits of $y$
- set $M_j$ to the maximum of $M_j$ and $\rho(l)$

size estimate is $E = m \left[\Gamma(-1/m) \frac{2^{-1/m} - 1}{\log 2}\right]^{-m} 2^{(\sum_j M_j)/m - 1}$
The Durand and Flajolet algorithm

- Buckets need to be only about $\log \log(n_{\text{max}})$ bits

- $E$ is unbiased:
  - as $n \to \infty$, $\langle E \rangle/n = 1 + \theta_1 + o(1)$
  - $|\theta_1| < 10^{-6}$

- The standard error $S$ (the standard deviation divided by $n$) of $E$ satisfies
  - as $n \to \infty$, $S = \beta_m/\sqrt{m} + \theta_2 + o(1)$
  - $|\theta_2| < 10^{-6}$
  - $\beta_m \approx 1.3$

- Practical formula: $S \approx 1.3/\sqrt{m} = 1.3 \times 2^{-k/2}$

- An improved version has $S \approx 1.05 \times 2^{-k/2}$
The function $\rho$ is easily implemented in C:

/* index of first 1 bit in x, counting from leftmost=0 */
unsigned int rho(int x) {
    for (int i=0; i<32; i++) {
        if (x<0) return i;
        x<<=1;
    }
    return 32;
}
Hash

In this context, a *hash function* is a mapping from \( \{0, 1\}^n \) to itself with the properties:

- it is bijective: injective (one-to-one) and surjective (onto)
- it has high entropy (on average, close inputs map to distant outputs)

```c
unsigned int hash(unsigned int x) {
    x += ~ (x << 15);
    x ^= (x >> 10);
    x += (x << 3);
    x ^= (x >> 6);
    x += ~ (x << 11);
    x ^= (x >> 16);
    return x;
}
```
Results 1: English dictionary

‘aardvark aback abaft abandon abandoned abandoning abandonment abandons . . . ’
'When on board H.M.S. Beagle as naturalist, I was much struck with certain facts in the distribution of the organic beings inhabiting South America, . . . '
‘To be, or not to be: that is the question:
Devoutly to be wish’d. To die, to sleep; . . . ’
'Kent: I thought the king had more affected the Duke of Albany than Cornwall. . . .'
Results 5: Julius Caesar

‘Scene I. Rome. A street. Enter Flavius, Marullus, and certain Commoners. . . . ’
Results 6: Ṛgveda

\[ \text{estimate} \]

\[ k \]

\[ 1.4 \times 10^5 \]

\[ 1.2 \times 10^5 \]

\[ 1.0 \times 10^5 \]

\[ 8 \times 10^4 \]

\[ 6 \times 10^4 \]

\[ 4 \times 10^4 \]

\[ 4 \]

\[ 6 \]

\[ 8 \]

\[ 10 \]

\[ 12 \]

\[ 14 \]

\[ tvám agne dyúbhis tvám āśuśukṣāṇis tvám adbhyās tvám áśmanas pári; tváṃ vānebhyas tváṃ óśadhībhyas tváṃ nṛṇāṃ nṛpate jāyase śucih; távāgne hotrāṃ táva potrāṃ ṛtvīyaṃ táva neṣḍrāṃ tváṃ agnīd ṛtāyatāḥ . . . \]
'17th. Up, and with my wife, setting her down by her father’s in Long Acre, in so ill looked a place, among all the whore houses...’
Results 8: Y-chromosome (word=block of 16 codons)

\[ \text{'GAATTCTAGGCTTTCTTTGAAGAGGTAGTAATCTGTAGCCCTCACCCTAGG...'} \]
Conclusion

If approximate counts are sufficient, they may be obtained very rapidly, and with small, constant memory usage and with known standard error