Probabilistic Method and Random Graphs

Lecture 6. Hashing and Random Graphs [1]

Xingwu Liu

Institute of Computing Technology
Chinese Academy of Sciences, Beijing, China

[1] The slides are mainly based on Chapter 5 of the textbook *Probability and Computing* and Lectures 12&13 of Ryan O'Donnell's lecture notes of *Probability and Computing*.

Preface

Questions, comments, or suggestions?

A recap of Lecture 5

Joint distribution of bin loads

$\Pr(X_1 = k_1, \dots X_n = k_n) = \frac{m!}{k_1! k_2! \cdots k_n! n^m}$

Poisson approximation theorem

- $(X_1^{(m)}, X_2^{(m)}, \dots X_n^{(m)}) \sim (Y_1^{(\mu)}, Y_2^{(\mu)}, \dots Y_n^{(\mu)} | \sum Y_i^{(\mu)} = m)$
- $\mathbb{E}[f(X_1^{(m)}, \dots X_n^{(m)})] \le e\sqrt{m}\mathbb{E}[f(Y_1^{(m)}, \dots Y_n^{(m)})]$
  - $\Pr(\mathcal{E}(X_1^{(m)}, \dots X_n^{(m)})) \le e\sqrt{m}\Pr(\mathcal{E}(Y_1^{(m)}, \dots Y_n^{(m)}))$
  - $e\sqrt{m}$ can be improved to 2, if f is monotonic in m

Applications

- For the coupon collector's problem, $\lim_{n\to\infty} \Pr(X > n \ln n + cn) = 1 - e^{-e^{-c}}$
- Max load: $L(n, n) > \frac{\ln n}{\ln \ln n}$ with high probability

Application: Hashing

Used to look up records, protect data, find duplications ...

Membership problem: password checker

Binary search vs Hashing

Hash table (1953, H. P. Luhn @IBM)

Hash functions: efficient, deterministic, uniform, non-invertible

- Random: coin tossing, SUHA
- SHA-1 (broken by Wang et al., 2005)

Bins&Balls model

Efficiency

Search time for m words in n bins: expected vs worst.

Space: $\geq 256m$ bits if each word has 256 bits.

- Potential wasted space: $\frac{1}{e}$ in the case of $m = n$.

Trade space for time. Can we improve space-efficiency?

Information Fingerprint

Fingerprint

Succinct identification of lengthy information

Fingerprint hashing

Fingerprinting $\rightsquigarrow$ sorting fingerprints (rather than original data)
$\rightsquigarrow$ binary search.

Trade time for space

Performance

False positive: due to loss of information

No other errors

Partial correction using white lists

False positive

Probability of a false positive: m words, b bits

Fingerprint of an acceptable differs from that of a bad: $1 - \frac{1}{2^b}$.
Probability of a false positive: $1 - \left(1 - \frac{1}{2^b}\right)^m \geq 1 - e^{-\frac{m}{2^b}}$.

Determine b

For a constant c, false positive $< c \Rightarrow e^{-\frac{m}{2^b}} \geq 1 - c$.
So, $b \geq \log_2 \frac{-m}{\ln(1-c)} = \Omega(\ln m)$.
If $b \geq 2 \log_2 m$, false positive $< \frac{1}{m}$.
2^{16} words, 32-bit fingerprints, false positive $< 2^{-16}$.
Save a factor of 8 if each word has 256 bits.

Can more space be saved while getting more time-efficient?

Bloom Filter

1970, CACM, by Burton H. Bloom.

Used in Bigtable and HBase.

Basic idea

Hash table + fingerprinting
Illustration

False positive is the only source of errors.

False positive: m words, n-bit array, k mappings

A specific bit is 0 with probability $\left(1 - \frac{1}{n}\right)^{km} \approx e^{-\frac{km}{n}} \triangleq p$.
Resonable to assume that a fraction p of bits are 0.
By Poisson approximation and Chernoff bounds.
False positive probability: $f \triangleq \left(1 - \left(1 - \frac{1}{n}\right)^{km}\right)^k \approx \left(1 - e^{-\frac{km}{n}}\right)^k$

Determine k for fixed m, n

Objective

Minimize f.
Dilemma of k: chances to find a 0-bit vs the fraction of 0-bits.

Optimal k

$\frac{d\ln f}{dk} = \ln\left(1 - e^{-\frac{km}{n}}\right) + \frac{km}{n}\frac{e^{-\frac{km}{n}}}{1-e^{-\frac{km}{n}}}$.
$\frac{d\ln f}{dk}\big|_{k=\frac{n}{m}\ln 2} = 0$.
$f\big|_{k=\frac{n}{m}\ln 2} = 2^{-k} \approx 0.6185^{n/m}$.
$f < 0.02$ if $n = 8m$, and $f < 2^{-16}$ if $n = 23m$, saving 1/4 space

Remark

Fix n/m, the #bits per item, and get a constant error probability.
In fingerprint hashing, $\Omega(\ln m)$ bits per item guarantee a constant error probability

Reference

Lectures 12 of the CMU lecture notes by Ryan O'Donnell.