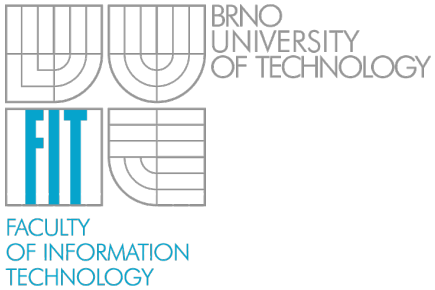
Bloom Filters Basics
Accelerated Network Technologies
Research Group

Martin Žádník

Brno University of Technology, Faculty of Information Technology
Bozetechova 2, 612 00 Brno, CZ
http://merlin.fit.vutbr.cz/ant/



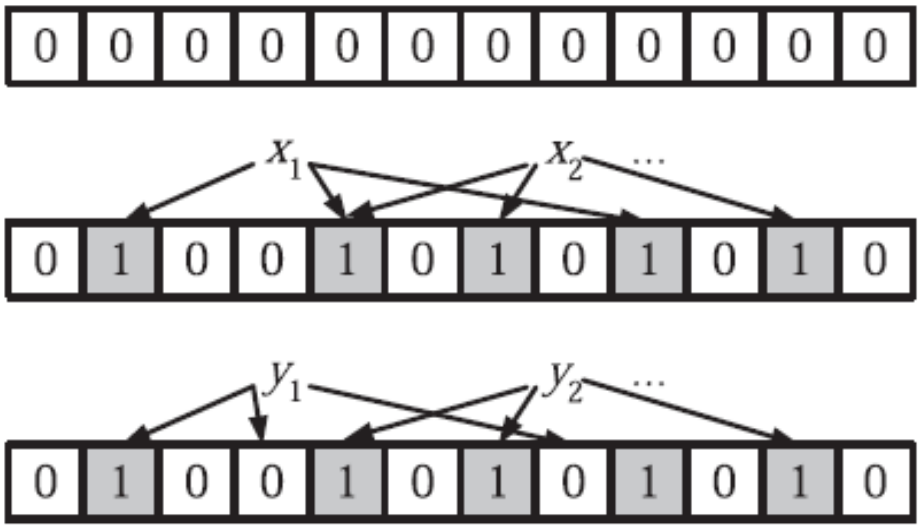
INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Motivace

- Struktura pro uložení informace o náležitosti prvku do množiny
  - Šetření místem
  - Rychlý a jednoduchý přístup
  - Velikost paměti x přesnost
- Není možné provést výčet prvků
- Obsahuje falešně positivní – tedy prvky, které nejsou součástí množiny

Princip

- Vložení znamená nastavení několika bitů v poli
  - Hash udává, které bity
  - Pokud je bit nastaven na 1, pak 1 zůstává
  - Velikost paměti x přesnost



Vlastnosti

- Notace
  - m … počet bitů filtru
  - n … počet prvků množiny
  - k … počet hash funkcí
- Pravděpodobnost, že nějaký bit po vložení *n* prvků na *k* míst je stále nula

Pravděpodobnost nastavení nějakého bitu na 1

$$p' = \left(1 - \frac{1}{m}\right)^{kn} \approx \mathrm{e}^{-kn/m}$$

Pravděpodobnost, že bit na 1 nebude nastaven

Vlastnosti

- Pravděpodobnost falešně positivních
  - s rostoucím počtem nastavených 1 roste falešně positivní
  - růst je exponenciální

$$f = \left(1 - \mathrm{e}^{-kn/m}\right)^k = (1 - p)^k$$

Pravděpodobnost, že bit je stále nula

- m … počet bitů filtru, n … počet prvků množiny, k … počet hash funkcí

Vlastnosti

- Minimalizace falešně positivních
  - více hash funkcí
  - menší počet nastavených jedniček

Sprostá úprava, aby to šlo derivovat

$$f = \left(1 - \mathrm{e}^{-kn/m}\right)^k = (1-p)^k$$

Vnitřek derivujeme

$$f = \exp(k \ln(1 - \mathrm{e}^{-kn/m}))$$

$$\frac{\partial g}{\partial k} = \ln\left(1 - \mathrm{e}^{-\frac{kn}{m}}\right) + \frac{kn}{m} \frac{\mathrm{e}^{-\frac{kn}{m}}}{1 - \mathrm{e}^{-\frac{kn}{m}}}$$

- Derivace je nula, když k = (m / n) * ln 2 = 0.69 * (m / n)
  - m … počet bitů filtru, n … počet prvků množiny, k … počet hash funkcí

Vlastnosti

- Efektivita, aneb počet bitů *m* pro dosažení false rate *eps*

Výčet všech filtrů délky m

Možné kombinace množin nad universem

$$2^m \binom{n + \epsilon(u-n)}{n} \geq \binom{u}{n}$$

Kombinace i s chybou

Redundance oproti prosté kombinaci

$$m \geq \log_2 \frac{\binom{u}{n}}{\binom{n+\epsilon(u-n)}{n}} \approx$$

$$n \log_2 \mathrm{e} \cdot \log_2(1/\epsilon)$$

- Prostorová složitost ~ 1.44*n
  - m … počet bitů filtru, n … počet prvků množiny, u … počet prvků univer

Bloom vs. Hash

- Nechť m = n * j, pak falešně positivní je $(0.62)^j$
- Hash funkce a různá rozšíření Bloom filtru umožňují dosáhnout lepší hranice $(0.5)^j$
- Jednoduchá hash je vlastně Bloom filtr s jednou hash

- Operace nad Bloom filtry
  - spojení dvou filtrů – OR
  - průnik dvou filtrů – AND

Rozšíření

- Čítající Bloom filtr
  - Místo jednoho bitu, celý čítač
  - Je možné i odebírat
- Velikost čítače by měla postačovat 4 bity pro většinu případů, tj. pokud dimenzujeme správně velikost a hash

$$\mathbb{P}(c(i) = j) = \binom{nk}{j} \left(\frac{1}{m}\right)^j \left(1 - \frac{1}{m}\right)^{nk-j}$$

$$\mathbb{P}(\max_i c(i) \geq 16) \leq 1.37 \times 10^{-15} \times m$$

- Jaká je pravděpodobnost úspešného odebrání?
  - m … počet bitů filtru, n … počet prvků množiny, k … počet hash funkcí

Aplikace

- Slovníky
- Databáze
- Distribuované cache
- Distribuované uložistě, např. P2P
- Routing, detekce smyček, multicast
- Měření, např. heavy toky