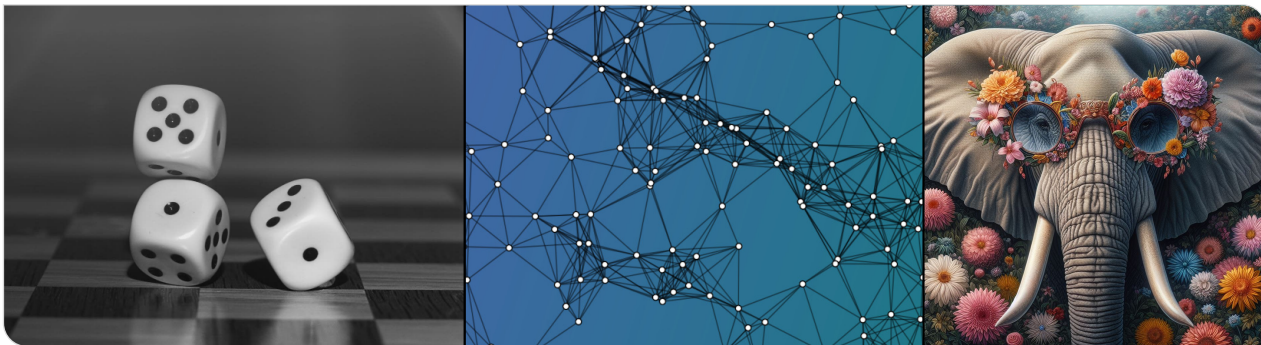


Probability and Computing – Bounded Differences and Bloom Filters

Stefan Walzer | WS 2024/2025



1. Method of Bounded Differences

2. What is a Filter or AMQ?

- Applications of Filters

3. The Bloom Filter Data Structure

4. Analysis of Bloom Filters

- Expected fraction of zeroes in Bloom filters
- Optimal tuning for Bloom filters
- Main Theorem on Bloom filters

5. Conclusion

More Concentration Bounds

Hoeffding: Let $X = X_1 + \dots + X_n$ where $a_i \leq X_i \leq b_i$, and X_1, \dots, X_n independent

$$\Pr[X - \mathbb{E}[X] \geq t] \leq \exp\left(-\frac{2t^2}{\sum_{i=1}^n (b_i - a_i)^2}\right). // \text{proof combines Chernoff with a different lemma}$$

Special Case: X_1, \dots, X_n are Bernoulli random variables

$$\Pr[X - \mathbb{E}[X] \geq t] \leq \exp(-2t^2/n). // \text{incomparable to the Chernoff bound we saw: } \Pr[X \geq (1 + \delta)\mathbb{E}[X]] \leq \left(\frac{e^\delta}{(1+\delta)^{1+\delta}}\right)^\mu$$

Generalisation: McDiarmid / Method of Bounded Differences

- Assume X_1, \dots, X_n are independent and $X = f(X_1, \dots, X_n)$.
- Assume for each i , a change in X_i changes $f(X_1, \dots, X_n)$ by at most c_i .

$$\Pr[X - \mathbb{E}[X] \geq t] \leq \exp\left(-\frac{2t^2}{\sum_{i=1}^n c_i^2}\right) // \text{same bound holds for } \Pr[\mathbb{E}[X] - X \geq t].$$

Proof uses Martingales... not here.

A Simple Use Case for the Method of Bounded Differences

Setting

- fixed graph $G = (V, E)$ and $s, t \in V$
- independent edge lengths $X_1, \dots, X_m \sim \mathcal{U}([0, 1])$
- let P be the shortest $s - t$ path. // unique with probability 1

McDiarmid / Bounded Differences

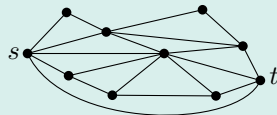
- X_1, \dots, X_n independent and $X = f(X_1, \dots, X_n)$.
 - changing X_i changes $f(X_1, \dots, X_n)$ by at most c_i .
- $\Rightarrow \Pr[X - \mathbb{E}[X] \geq t] \leq \exp\left(-\frac{2t^2}{\sum_{i=1}^n c_i^2}\right)$.

Example where McDiarmid might be useful

Let $L :=$ length of P .

- $L = f(X_1, \dots, X_m)$ for independent X_1, \dots, X_m . ✓
- Changing X_i changes L by at most 1. $\rightsquigarrow c_i = 1$. ✓

$\Pr[L - \mathbb{E}[L] \geq t] \leq \exp\left(-\frac{2t^2}{m}\right)$ // might be useful



A Simple Use Case for the Method of Bounded Differences

Setting

- fixed graph $G = (V, E)$ and $s, t \in V$
- independent edge lengths $X_1, \dots, X_m \sim \mathcal{U}([0, 1])$
- let P be the shortest $s - t$ path. // unique with probability 1

McDiarmid / Bounded Differences

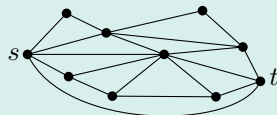
- X_1, \dots, X_n independent and $X = f(X_1, \dots, X_n)$.
 - changing X_i changes $f(X_1, \dots, X_n)$ by at most c_i .
- $\Rightarrow \Pr[X - \mathbb{E}[X] \geq t] \leq \exp\left(-\frac{2t^2}{\sum_{i=1}^n c_i^2}\right)$.

Example where McDiarmid seems useless

Let $H :=$ number of edges in P . // “hops”

- $H = f(X_1, \dots, X_m)$ for independent X_1, \dots, X_m . ✓
- Changing X_i might change H by $n - 2$. $\rightsquigarrow c_i = O(n)$.

$$\Pr[H - \mathbb{E}[H] \geq t] \leq \exp\left(-\frac{2t^2}{O(mn^2)}\right) // \text{too weak}$$



1. Method of Bounded Differences

2. What is a Filter or AMQ?

- Applications of Filters

3. The Bloom Filter Data Structure

4. Analysis of Bloom Filters

- Expected fraction of zeroes in Bloom filters
- Optimal tuning for Bloom filters
- Main Theorem on Bloom filters

5. Conclusion

Filter = Approximate Membership Query Data Structure

Setting

- universe D of possible keys
- a set $S \subseteq D$ of $n = |S|$
- a false positive probability ϵ

Want: Data structure representing S .

Space Requirement

- want $\mathcal{O}(n \log(1/\epsilon))$ bits
- *much* smaller than $\mathcal{O}(n \log |D|)$ bits needed for hash table

a set S :

Anja Blancani
Peter Sanders
Florian Kurpicz
Hape Lehmann
Thomas Worsch

Operations

- **insert** elements to S and **delete** elements from S (optional)
- **query**: given $x \in D$ answer “is $x \in S$?” *approximately*:

query(x) = **YES** for $x \in S$

$\Pr[\mathbf{query}(x) = \mathbf{NO}] \geq 1 - \epsilon$ for $x \notin S$

a filter for S :

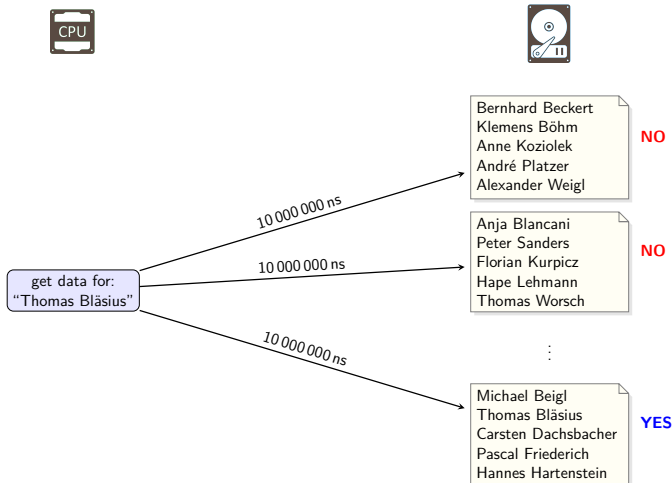
Anja
Peter
Florian
Hape
Thomas

query(Peter Sanders) = **YES**
query(Petra Mutzel) = **YES**
query(Donald Knuth) = **NO**

The **YES** answers are **unreliable**.

The **NO** answers are **reliable**.

Simplified Use Case for Filters



General Idea

If the reliable **NO** answers are frequent, a filter access can replace a (costly) access to a reliable data structure.

Further Example

Filter stores set of malicious URLs.

- Most accessed URLs will be non-malicious.
- Only false positives and true positives have to access reliable data structure (e.g. web-service).

1. Method of Bounded Differences

2. What is a Filter or AMQ?

- Applications of Filters

3. The Bloom Filter Data Structure

4. Analysis of Bloom Filters

- Expected fraction of zeroes in Bloom filters
- Optimal tuning for Bloom filters
- Main Theorem on Bloom filters

5. Conclusion

Method of Bounded Differences
○○○○

What is a Filter or AMQ?
○○

The Bloom Filter Data Structure
●○○

Analysis of Bloom Filters
○○○○○○○○

Conclusion
○○

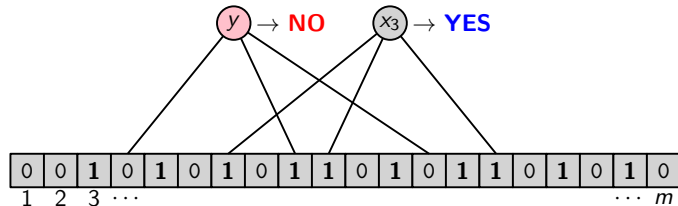
Simple Uniform Hashing Assumption (SUHA)

- We have access to $h \sim \mathcal{U}(R^D)$ for any R and D .
- h takes $\mathcal{O}(1)$ time to evaluate.
- h takes no space to store.

The Bloom Filter Data Structure

Parameters

- m length of a bit array $A[1..m]$ that we use
- $k \in \mathcal{O}(1)$ number of hash functions $h_1, \dots, h_k \sim \mathcal{U}([m]^D)$
- n number of keys in $S \subseteq D$ (dynamic)
- $\alpha \in \mathcal{O}(1)$ load n/m (dynamic)



insert(x):

```
for  $i \in [k]$  do  
   $A[h_i(x)] = 1$ 
```

query(x):

```
for  $i \in [k]$  do  
  if  $A[h_i(x)] = 0$  then  
    return NO  
return YES
```

delete(x):

\langle not supported \rangle

1. Method of Bounded Differences

2. What is a Filter or AMQ?

- Applications of Filters

3. The Bloom Filter Data Structure

4. Analysis of Bloom Filters

- Expected fraction of zeroes in Bloom filters
- Optimal tuning for Bloom filters
- Main Theorem on Bloom filters

5. Conclusion

Exercise: Some approximations of e

$$\forall n \in \mathbb{N} : \left(1 + \frac{1}{n}\right)^n \leq e \leq \left(1 + \frac{1}{n}\right)^{n+1}$$
$$\text{and } \left(1 - \frac{1}{n}\right)^n \leq e^{-1} \leq \left(1 - \frac{1}{n}\right)^{n-1}.$$

Corollaries

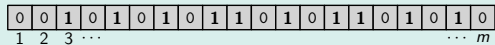
$$\forall n \in \mathbb{N} : \left(1 + \frac{1}{n}\right)^n = e - \mathcal{O}(1/n)$$
$$\text{and } \left(1 - \frac{1}{n}\right)^n = e^{-1} - \mathcal{O}(1/n).$$

Bloom Filter Analysis (i)

Lemma

Assume $S = \{x_1, \dots, x_n\}$ is inserted into the Bloom filter. Let $(A_1, \dots, A_m) \in \{0, 1\}^m$ be the random filter state and $Z := \sum_{i=1}^m (1 - A_i)$ the number of zeroes. Then

i $\mathbb{E}[\frac{Z}{m}] = (1 - \frac{1}{m})^{m\alpha k} = e^{-\alpha k} - o(1)$



ii For $y \notin S$: $\Pr[\text{query}(y) = \text{YES} \mid Z = z] = (1 - \frac{z}{m})^k$

Proof of (i).

$$\begin{aligned} \mathbb{E}[\frac{Z}{m}] &= \frac{1}{m} \mathbb{E}[\sum_{i=1}^m (1 - A_i)] = \frac{1}{m} \sum_{i=1}^m \Pr[A_i = 0] = \frac{1}{m} \sum_{i=1}^m \Pr[A_1 = 0] = \Pr[A_1 = 0] \\ &= \Pr[\forall x \in S : \forall i \in [k] : h_i(x) \neq 1] \stackrel{\text{SUHA}}{=} \prod_{x \in S} \prod_{i \in [k]} \Pr[h_i(x) \neq 1] \stackrel{\text{SUHA}}{=} \prod_{x \in S} \prod_{i \in [k]} (1 - \frac{1}{m}) \\ &= (1 - \frac{1}{m})^{nk} = (1 - \frac{1}{m})^{m\alpha k} = (e^{-1} - o(1))^{\alpha k} = e^{-\alpha k} - o(1). \end{aligned}$$

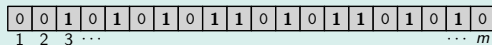
Bloom Filter Analysis (i)

Lemma

Assume $S = \{x_1, \dots, x_n\}$ is inserted into the Bloom filter. Let $(A_1, \dots, A_m) \in \{0, 1\}^m$ be the random filter state and $Z := \sum_{i=1}^m (1 - A_i)$ the number of zeroes. Then

i $\mathbb{E}\left[\frac{Z}{m}\right] = \left(1 - \frac{1}{m}\right)^{m\alpha k} = e^{-\alpha k} - o(1)$

ii For $y \notin S$: $\Pr[\text{query}(y) = \text{YES} \mid Z = z] = \left(1 - \frac{z}{m}\right)^k$



Proof of (ii).

$$\Pr[\text{query}(y) = \text{YES} \mid Z = z] = \Pr[\forall i \in [k] : A_{h_i(y)} = 1 \mid Z = z] \stackrel{\text{SUHA}}{=} \prod_{i \in [k]} \left(\frac{m - z}{m}\right) = \left(1 - \frac{z}{m}\right)^k.$$

How should a Bloom filter be configured?

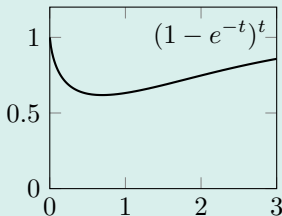
Approximate false positive rate

From the previous Lemma we get for $y \notin S$:

$$\begin{aligned}\varepsilon &= \Pr[\text{query}(y) = \text{YES}] \approx \Pr[\text{query}(y) = \text{YES} \mid Z = \mathbb{E}[Z]] \\ &\stackrel{\text{ii}}{=} \left(1 - \frac{\mathbb{E}[Z]}{m}\right)^k \stackrel{\text{i}}{=} (1 - e^{-\alpha k} + o(1))^k \approx (1 - e^{-\alpha k})^k.\end{aligned}$$

Which k minimises ε ? (when α is fixed)

$$\begin{aligned}&\arg \min_{k \in \mathbb{N}} (1 - e^{-\alpha k})^k \\ &= \arg \min_{k \in \mathbb{N}} (1 - e^{-\alpha k})^{\alpha k} \\ &\approx \frac{1}{\alpha} \arg \min_{t \in \mathbb{R}_+} (1 - e^{-t})^t \\ &= \frac{1}{\alpha} \arg \min_{t \in \mathbb{R}_+} t \ln(1 - e^{-t})\end{aligned}$$



- plot $(1 - e^{-t})^t \rightsquigarrow$ one global minimum.
- deriving $t \ln(1 - e^{-t})$ gives $\ln(1 - e^{-t}) + \frac{te^{-t}}{1 - e^{-t}}$
- $t = \ln(2)$ is root of the derivative.

$\hookrightarrow k = \ln(2)/\alpha$ is optimal for fixed α .
 \hookrightarrow choose α and k such that $\alpha k = \ln(2)$

Intuition for optimality of $\alpha k = \ln(2)$

- gives $\mathbb{E}[\frac{Z}{m}] \approx e^{-\alpha k} = \frac{1}{2}$
- maximises *entropy* of the filter bits

Theorem

A Bloom filter with $k \in \mathbb{N}$ hash functions and load factor $\alpha = \ln(2)/k$ has

space requirement $m = n/\alpha = \frac{kn}{\ln 2} \approx 1.44kn$ bits and

false positive probability $\varepsilon = 2^{-k} + o(1)$.

- space requirement ✓
- false positive probability: need a concentration bound first.

Concentration bound for Z

Lemma

- i $\Pr[Z \leq \mathbb{E}[Z] - t] \leq \exp(-\Theta(t^2/m))$ for any $t > 0$,
- ii $\Pr[Z \leq \mathbb{E}[Z] - m^{2/3}] \leq \exp(-\Theta(m^{1/3}))$ by setting $t = m^{2/3}$.

McDiarmid / Bounded Differences

- X_1, \dots, X_n independent and $X = f(X_1, \dots, X_n)$.
 - changing X_i changes $f(X_1, \dots, X_n)$ by at most c_i .
- $\Rightarrow \Pr[\mathbb{E}[X] - X \geq t] \leq \exp(-\frac{2t^2}{\sum_{i=1}^n c_i^2})$.

Proof of (i) using the method of bounded differences.

- Z is a function of kn independent hash values // SUHA
- each hash value can change Z by at most 1
- use method of bounded differences!

$$\Rightarrow \Pr[Z \leq \mathbb{E}[Z] - t] \leq \Pr[\mathbb{E}[Z] - Z \geq t] = \exp\left(-\frac{2t^2}{nk}\right) = \exp\left(-\frac{2t^2}{m\alpha k}\right) = \exp\left(-\frac{2t^2}{m \ln(2)}\right). \quad \square$$

Proof of the Main Theorem on Bloom filters (false positive probability).

By choice of k and α we have $\mathbb{E}\left[\frac{Z}{m}\right] = e^{-\alpha k} - o(1) = \frac{1}{2} - o(1)$.

Let $y \notin S$ and $B = \lfloor \mathbb{E}[Z] - m^{2/3} \rfloor$.

$$\begin{aligned} \varepsilon &= \Pr[\text{query}(y) = \text{YES}] \stackrel{\text{LTP}}{=} \sum_{z=1}^m \Pr[Z = z] \cdot \Pr[\text{query}(y) = \text{YES} \mid Z = z] = \sum_{z=1}^m \Pr[Z = z] \cdot \left(1 - \frac{z}{m}\right)^k \\ &\leq \sum_{z=1}^B \Pr[Z = z] + \sum_{z=B+1}^m \Pr[Z = z] \left(1 - \frac{B+1}{m}\right)^k \leq \Pr[Z \leq B] + \left(1 - \frac{B+1}{m}\right)^k \\ &\leq \Pr[Z \leq \mathbb{E}[Z] - m^{2/3}] + \left(1 - \frac{\mathbb{E}[Z] - m^{2/3}}{m}\right)^k \stackrel{\text{ii}}{\leq} \exp(-\Theta(m^{1/3})) + \left(1 - \frac{1}{2} + o(1)\right)^k = 2^{-k} + o(1). \quad \square \end{aligned}$$

How to Configure Your Bloom Filter

Theorem

A Bloom filter with $k \in \mathbb{N}$ hash functions and load factor $\alpha = \ln(2)/k$ has

space requirement $m = n/\alpha = \frac{kn}{\ln 2} \approx 1.44kn$ bits and

false positive probability $\varepsilon = 2^{-k} + o(1)$.

How to determine m and k (the parameters you actually need)

- 1 n : determined by input
- 2 ε : choose a trade-off between space usage and false positive probability
 - If utility comes from negative answers “ $x \notin S$, definitely” and running time is negligible, then:
 - want to maximise utility – disutility, where: (\propto means “proportional to”)
 - utility \propto negative answers = queries $\cdot \Pr[x \notin S] \cdot (1 - \varepsilon)$
 - disutility \propto space consumption = $1.44 \log(1/\varepsilon)n$ bits of RAM or cache
- 3 compute $k = \lceil \log(1/\varepsilon) \rceil$ // effectively restricts ε to powers of 2
- 4 compute $\alpha = \ln(2)/k$ and $m = \lceil n/\alpha \rceil$

Much, much more is known

- more functionality
 - ↔ counting Bloom filters support deletions
- better query times
 - ↔ blocked Bloom filters improve cache efficiency
- better space efficiency
 - ↔ cuckoo filters use $n \log(1/\varepsilon) + \mathcal{O}(n)$ bits rather than $\approx 1.44n \log(1/\varepsilon)$ bits
 - ↔ static filters (no insertions or deletions) use $n \log(1/\varepsilon) + o(n)$ bits.
- ...

- **Approximate Membership Queries.**
 - Decide “is $x \in S$?” with *false positive probability* ε .
 - The Bloom filter is the most widespread AMQ.
- **Space Efficient.** $\approx 1.44 \log(1/\varepsilon)$ bits per element
 - often fit into cache or RAM when proper set data structure does not
- **Used to prevent costly accesses.**
 - Reliable on **NO** answers.
 - Useful if **NO** answers are frequent.

- Approximate-Membership-Query Datenstrukturen im Allgemeinen
 - Welche Aufgabe hat eine AMQ Datenstruktur?
 - Was ist der Vorteil gegenüber einer exakten Datenstruktur?
 - Was wäre ein Anwendungsfall, in dem eine AMQ Datenstruktur nützlich ist?
- Bloomfilter
 - Wie ist ein Bloomfilter aufgebaut und welche Operationen unterstützt er?
 - Welche Parameter gibt es, und wie hängen diese zusammen?
 - Was hat unsere Analyse zur geschickten Wahl der Parameter zu sagen? Wie werden die übrigen Parameter gewählt? Welcher Speicherverbrauch ergibt sich?
 - Fragen zur Analyse
 - Welche Anzahl von Nullen bzw. Einsen erwarten wir?
 - Wie hängt die falsch-positiv Wahrscheinlichkeit mit der Anzahl Nullen bzw. Einsen zusammen?
 - Wir kann man argumentieren, dass die Anzahl Nullen bzw. Einsen im Bloomfilter nahe am Erwartungswert liegt?