Fachhochschule Nordwestschweiz
Hochschule für Technik

Benjamin Leber, Janis Peyer, Samuel Merki

Zusammenfassung Bloomfilter

Diskrete Stochastik

Idee

Die Idee hinter dem Bloomfilter ist, das in einem Datenstrom möglichst schnell festgestellt werden kann ob Daten aus dem Strom schon vorgekommen sind. Dazu wird ein BitSet und mehrere Hashfunktionen verwendet. Nun wird für die Daten mit jeder Hashfunktion der Index im Bitset berechnet. Beim Hinzufügen von Wörtern werden diese Indexe auf 1 gesetzt, falls man prüfen will ob die Daten schon enthalten sind, muss in jedem Index der berechnet wurde ein 1 stehen, ansonsten sind die Daten nicht enthalten.

Vor- und Nachteile

Vorteile

+ Effizient
+ Kompaktes Speichern der Daten

Nachteile

– Keine 100% Wahrscheinlichkeit, dass ein gefundenes Wort wirklich enthalten ist.

Beispiel eines Bloomfilters

Der Proxyserver Squid verwendet den Bloomfilter zum Abgleich von Cache Einträgen. Dies dient dazu um schnell zu prüfen ob die angefragte URL im Cache liegt.
Siehe auch: http://wiki.squid-cache.org/SquidFaq/CacheDigests

Hashwert zu Index

Der Hashwert wird bei uns auf eine einfache Art in einen Index umgewandelt. Mit einer Modulo Rechnung durch die Grösse des BitSets bringen wird den Hashwert auf die richtige grösse. Davon nehmen wir den absolut Wert als Index.

```
Math.abs(hashCode % sizeBitSet)
```

Test Fehlerwahrscheinlichkeit

Um die Fehlerwahrscheinlichkeit zu testen, haben wir eine zweite Liste mit Wörtern erstellt die nicht in der ersten vorkommen, dazu haben wir jeden Buchstaben der Wörter aus der normalen Liste verdoppelt.
Nun zählen wir alle Wörter dieser Liste welche vom Filter gefunden werden, daraus lässt sich dann die Fehlerwahrscheinlichkeit in Prozent berechnen.

```
double percent =  100/(double)words.size() * correctWordsFound;
```

Tests

words.size() = 58110

Fehlerwahrscheinlichkeit p	Anzahl gefundener Worte	Prozent
10%	5962	10.26%
25%	14619	25.16%
50%	29123	50.12%