Similarity Search
A Survey

Marcus Gregersen
mabg@itu.dk

Martin Faartoft
mlfa@itu.dk

Rick Marker
rdam@itu.dk

May 21st 2014

Abstract

The abstract goes here

1 Introduction

We will analyse and compare two different approaches to solving the problem of querying a set for an approximate match on a query element.

2 Related work

2.1 Distance-Sensitive Bloom Filters

TODO 2 paragraffer om klassiske bloom filtre
TODO 2 paragraffer om LSH i bit-vector context

Kirsch and Mitzenmacher propose in the paper 'Distance-Sensitive Bloom Filters'[1] a novel way of using Bloom filters. The aim is to expand upon classical Bloom filters, enabling them to answer the question: "Does a set contain an element similar to a given element?". They note that such a data structure has a number of practical uses, if it can be made sufficiently effective in both time and space.

The way they accomplish this is by using locality sensitive hashing algorithms (TODO REF), such that two elements that are sufficiently similar will, with a high probability, hash to the same value. They also use a partitioned Bloom filter, which is a like a standard Bloom filter except each locality sensitive hashing algorithm maps into its own bit array, instead of all the hashing algorithms share the same bit array.

When querying the Bloom filter for an approximate match, the element is hashed with all the hashing algorithms, and the corresponding indiceses are checked. If there are more bits, than a specified threshold (T), that is set to 1, it returns that the data contains an element that is close to the query element. The use of a threshold is the reason why it is possible to have false positives using this data structure.

A disadvantage to this approach, compared to a classical Bloom filter, is that it introduces the possibility of false negatives.

2.2 Kinesere

3 Theory

4 Experiments

5 Conclusion

References

[1] Adam Kirsch, Michael Mitzenmacher Distance-Sensitive Bloom Filters

Appendix