Bachelorproject

Rasmus Løvstad

Block 1+2, 2021

Abstract

This project regards an analysis of recent advances in solving the Approximate Jaccard Similarity Search Problem, specifically in regards to how one can achieve sublinear query time using parallel bit counting as presented by Knudsen[6]. This contains a theoretical analysis of both runtime and correctness of the bit-counting algorithm as well as an empirical comparison to existing methods. The findings include both a theoretical and empirical run time advantage to using parallel bit counting compared to a simple, linear time algorithm. Furthermore, reflections are made on how the results might scale on more specialized hardware.

Contents

1 Introduction

The *Approximate Similarity Search Problem* regards efficiently finding a set A from a corpus $\mathcal{F}$ that is approximately similar to a query set Q in regards to the *Jaccard Similarity* metric $J(A, Q) = \frac{|A \cap Q|}{|A \cup Q|}$[4][6]. Practical applications includes searching through large corpi of high-dimensional text documents like plagiarism-detection or website duplication checking among others[5]. The main bottleneck in this problem is the *curse of dimensionality*. Any trivial algorithm can solve this problem in $O(nd|Q|)$ time, but algorithms that query in linear time to the dimensionality of the corpus scale poorly when working with high-dimensional datasets. Text documents are especially bad in this regard since they often are encoded using *w-shingles* (w contigous words) which Li, Shrivastava, Moore, *et al.* [3] shows easily can reach a dimensionality upwards of $d = 2^{83}$ using just 5-shingles.
The classic solution to this problem is the MinHash algorithm presented by Broder [1] to perform website duplication checking for the AltaVista search engine. It preprocesses the data once using hashing to perform effective querying in $O(n + |Q|)$ time, a significant improvement independent of the dimensionality of the corpus. Many improvements have since been presented to both improve processing time, query time and space efficiency. Notable mentions includes (but are not limited to) *b-bit minwise hashing*[2], *fast similarity sketching*[4] and *parallel bit-counting*[6] (the latter of which is the main focus of this project). These contributions have brought the query time down to sublinear time while keeping a constant error probability.
The addition of parallel bit-counting for querying

2 Theory

2.1 Problem Definition

2.1.1 Jaccard Similarity

2.1.2 Similarity Search Problem

2.1.3 Approximate Similarity Search Problem

2.2 Trivial Solution

2.3 MinHash

2.4 Fast Similarity Sketching

2.5 Comparison

3 Methods

3.1 Hypothesis

3.2 Benchmarking

3.3 Expected Results

4 Implementation

4.1 Technology

4.2 Design

4.3 Assumptions

4.4 Challenges

4.5 Correctness

4.6 Benchmarking

5 Results

6 Discussion

7 Conclusion

References

[1] A. Broder, *On the resemblance and containment of documents*, 1997. DOI: 10.1109/SEQUEN.1997.666900.

[2] P. Li and A. C. König, "Theory and applications of b-bit minwise hashing," *Commun. ACM*, vol. 54, no. 8, pp. 101–109, Aug. 2011, ISSN: 0001-0782. DOI: 10.1145/1978542.1978566. [Online]. Available: https://doi.org/10.1145/1978542.1978566.

[3] P. Li, A. Shrivastava, J. Moore, and A. C. Konig, *Hashing algorithms for large-scale learning*, 2011. arXiv: 1106.0967 [stat.ML].

[4] S. Dahlgaard, M. B. T. Knudsen, and M. Thorup, *Fast similarity sketching*, 2017. arXiv: 1704.04370 [cs.DS].

[5] S. Vassilvitskii, *Coms 6998-12: Dealing with massive data (lecture notes, columbia university)*, 2018.

[6] J. B. T. Knudsen, *Fast similarity search with low error probability*, 2021.