Searching Web Data using MinHash LSH

BiChen Rao
University of Toronto
erao@cs.toronto.edu

Erkang Zhu
University of Toronto
ekzhu@cs.toronto.edu

ABSTRACT

In this extended abstract, we explore the use of MinHash Locality Sensitive Hashing (MinHash LSH) to address the problem of indexing and searching Web data. We discuss a statistical tuning strategy of MinHash LSH, and experimentally evaluate the accuracy and performance, compared with inverted index. In addition, we describe an on-line demo for the index with real Web data.

1. INTRODUCTION

Numerous datasets are published on the Web and can be used freely. The Web Data Common Web Table project extracted over 200 millions tabular datasets from HTML pages on the Web [6]. With the large amount of data available, it is difficult for researchers to discover datasets covering similar topics that are relevant to their interests, due to heterogeneous sources and schemas and the lack of search infrastructure. Datasets can be huge in size and the number of datasets is usually massive, causing additional challenges for Web data search.

When searching for datasets, we can use the Jaccard distance between *attributes* of the datasets as a relevance measure. An attribute is a set of values. It may be a column of a tabular dataset, or a full path in a semi-structured dataset. Due to the massive size and number of attributes, computing Jaccard distance directly using data values can be very expensive. We can represent an attribute with a fixed-size summary called *MinHash*, which has been used to reduce the memory usage of attributes and increase the query and insertion speed [2]. However, over millions of attributes, linear scan over MinHashes is still very slow. MinHash LSH can be used to index the MinHashes of attributes [5]. Similar attributes can be searched by querying a number of hash tables whose hash keys are concatenations of hash values in MinHash, in sub-linear time with respect to the number of attributes. SimHash is an alternative LSH index that also supports sub-linear search time [3], however, a recent paper showed that MinHash LSH is more computationally efficient than SimHash [7].

SIGMOD/PODS'16 June 26 - July 01, 2016, San Francisco, CA, USA
© 2016 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-3531-7/16/06.
DOI: http://dx.doi.org/10.1145/2882903.2914838

In this extended abstract, we present a new evaluation of MinHash LSH with a statistical tuning strategy proposed by Dong et al. [4] using real Web data.

2. TUNING MINHASH LSH

Minhash LSH has two parameters: L the number of hash tables and M the size of the hash key, which is a concatenation of M hash values in MinHash. The parameters typically need to be hand-tuned in order to balance accuracy and performance.

We used a statistical tuning strategy based on the one proposed by Dong et al. for Euclidean distance [4]. In order to use this strategy, the following distributions need to be obtained from the data: (1) the distribution of all-pair squared Jaccard distance, and (2) given K, the distribution of squared Jaccard distance for the 1^{st} to K^{th} ranked attributes. The tuning is formulated as an optimization problem, where the objective is to minimize *selectivity*, which is defined as the number of matches returned by LSH divided by the total number of attributes in the index, while maintaining a minimum level of *recall*. In other words, we want to minimize the processing cost for the returned matches without loosing too much accuracy. The selectivity can be expressed as:

$$selectivity = \int_0^{1.0} \rho(\sqrt{x}) f(x) dx \tag{1}$$

And the recall can be expressed as:

$$recall = \frac{1}{K} \sum_{k=1}^{K} \int_0^{1.0} \rho(\sqrt{x}) f_k(x) dx \tag{2}$$

f is the squared Jaccard distance distribution of all-pairs and f_k is the squared Jaccard distance distribution of the Kth result. ρ is the probability of LSH returning the result given the distance to the the query data. ρ can be obtained using a closed form equation given M and L.[8].

3. EXPERIMENTAL EVALUATION

To evaluate the accuracy and performance of MinHash LSH index with the statistical tuning strategy, we downloaded the English relational subset of the 2015 Web Data Common Web Table [6]. It contains 51 million tables and 262,893,349 attributes in total.

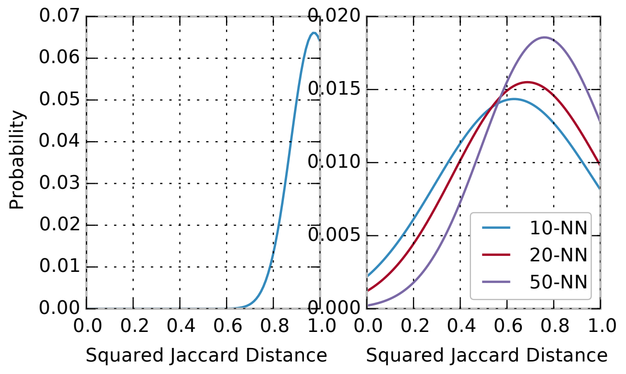

Figure 1: Squared Jaccard Distance Distribution for Attributes in Web Table

K	Precision	Recall	F Score
10	0.716	0.584	0.643
30	0.738	0.564	0.639
50	0.775	0.583	0.665

Table 1: Precision, Recall and F-score for different Ks

Accuracy. We take a random sample of 1,000 tables to estimate the distributions required for index tuning. The all-pair square Jaccard distance distribution is plotted on left side of Figure 1. The right side of Figure 1 shows the squared Jaccard distance distributions for the 10^{th}, 20^{th}, and 50^{th} rank in the top-K matches. We fit the distributions with gamma functions, and use the fitted gamma functions to approximate the probability density functions of the original distributions. For index tuning, we choose the minimum recall to be 0.6. The optimal parameters are $M = 2$ and $L = 256$. We build the index using 1,000 random selected tables. Table 1 presents the average precision and recall of 1,000 randomly selected query attributes. We can see the recall is very close to the minimum recall set in index tuning.
Performance. An alternative search index is inverted index. In our experiment, the inverted index is constructed by building a mapping from data value to attributes. For query, MinHash LSH has a time complexity of $O(L \cdot M)$ since it only needs to query L hash tables to retrieve the appropriate attributes, and each hash key is a concatenation of M hash values. The inverted index has a time complexity of $O(log(N) \cdot q)$ where N is the number of indexed attributes and q is the number of distinct data values in the query attribute. For each data value, the index finds all attributes that contain it. Figure 2 shows the query time of inverted index vs. MinHash LSH with respect to increasing number of indexed attributes. The query time of the MinHash LSH index stays constant, while the query time of the inverted index increases. Thus, the result confirms the previous analysis.

4. DEMO

A demo of the MinHash LSH index with Web Table data can be accessed at `vldb.cs.toronto.edu:4001`. User can upload a CSV file through the web interface and find tables that contain similar columns as the input CSV.

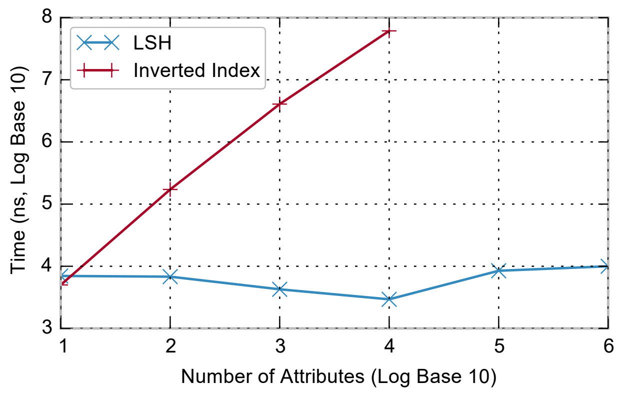

Figure 2: MinHash LSH with Tuning vs Inverted Index Performance

5. CONCLUSION

We have shown that MinHash LSH is very efficient for searching datasets, and a statistical tuning strategy can be used to balance the performance and recall accuracy. Alternative LSH indexes such as LSH Forest [1] can be used to improve recall. Furthermore, set containment is sometimes preferred over Jaccard distance as a relevance measure, and LSH Ensemble [8] can be used for set containment search. Applying the statistical tuning strategy to LSH Ensemble is another potential direction of research.

6. REFERENCES

[1] Mayank Bawa, Tyson Condie, and Prasanna Ganesan. LSH Forest: Self-tuning Indexes for Similarity Search. In *WWW*, pages 651–660. ACM, 2005.

[2] A. Broder. On the Resemblance and Containment of Documents. In *SEQUENCES*, pages 21–. IEEE Computer Society, 1997.

[3] Moses Charikar. Similarity Estimation Techniques from Rounding Algorithms. In *STOC*, pages 380–388. ACM, 2002.

[4] Wei Dong, Zhe Wang, William Josephson, Moses Charikar, and Kai Li. Modeling LSH for Performance Tuning. In *CIKM*, pages 669–678. ACM, 2008.

[5] Piotr Indyk and Rajeev Motwani. Approximate Nearest Neighbors: Towards Removing the Curse of Dimensionality. In *STOC*, pages 604–613. ACM, 1998.

[6] Oliver Lehmberg, Dominique Ritze, Robert Meusel, and Christian Bizer. A Large Public Corpus of Web Tables Containing Time and Context Metadata. In *WWW (Companion Volume)*, pages 75–76. ACM, 2016.

[7] Anshumali Shrivastava and Ping Li. In Defense of Minhash over Simhash. In *AISTATS*, volume 33 of *JMLR Proceedings*, pages 886–894. JMLR.org, 2014.

[8] Erkang Zhu, Fatemeh Nargesian, Ken Q. Pu, and Renée J. Miller. LSH Ensemble: Internet Scale Domain Search. *CoRR*, abs/1603.07410, 2016.