Knowledge-Based Systems 97 (2016) 40–47

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

BitHash: An efficient bitwise Locality Sensitive Hashing method with applications

Wenhao Zhang, Jianqiu Ji, Jun Zhu, Jianmin Li, Hua Xu*, Bo Zhang

State Key Lab of Intelligent Technology and Systems; Tsinghua National TNLIST Lab Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China

ARTICLE INFO

Article history:
Received 26 September 2015
Revised 11 January 2016
Accepted 18 January 2016
Available online 23 January 2016

Keywords:
Locality Sensitive Hashing
BitHash
Near-duplicate detection
Machine learning
Sentiment analysis
Storage efficiency

ABSTRACT

Locality Sensitive Hashing has been applied to detecting near-duplicate images, videos and web documents. In this paper we present a Bitwise Locality Sensitive method by using only one bit per hash value (BitHash), the storage space for storing hash values is significantly reduced, and the estimator can be computed much faster. The method provides an unbiased estimate of pairwise Jaccard similarity, and the estimator is a linear function of Hamming distance, which is very simple. We rigorously analyze the variance of One-Bit Min-Hash (BitHash), showing that for high Jaccard similarity. BitHash may provide accurate estimation, and as the pairwise Jaccard similarity increases, the variance ratio of BitHash over the original min-hash decreases. Furthermore, BitHash compresses each data sample into a compact binary hash code while preserving the pairwise similarity of the original data. The binary code can be used as a compressed and informative representation in replacement of the original data for subsequent processing. For example, it can be naturally integrated with a classifier like SVM. We apply BitHash to two typical applications, near-duplicate image detection and sentiment analysis. Experiments on real user's photo collection and a popular sentiment analysis data set show that, the classification accuracy of our proposed method for two applications could approach the state-of-the-art method, while BitHash only requires a significantly smaller storage space.

1. Introduction

As the data volume increasingly grows, efficient similarity measurement becomes a very important building block in information retrieval and machine learning. However, for large-scale high dimensional data, computing the pairwise similarity between each pair of data samples can be expensive and the storage cost is high. The applications of natural language processing and computer vision will also suffer from the curse of dimensionality, because the words and visual descriptors might have millions of dimensions. For this problem, Locality Sensitive Hashing is a powerful method, which enables efficient similarity computation and search.

Min-hash is one of such methods, which is simple and has been widely used in search engines and clustering tasks. Min-hash is first designed for near-duplicate web document detection [1], where each web document is represented as a set of shingles and Jaccard similarity is used as the similarity measurement. Since min-hash serves as a scalable way of estimating the Jaccard similarity between two sets, it can be used for any data that can be represented as sets. Later Chum et al. [2] apply min-hash to detect shots in videos, and then they use min-hash to detect near-duplicate images [3]. This is analogous to near-duplicate web document detection since images and videos can also be represented as sets, e.g. sets of visual words. There are also various near-duplicate image detection methods based on different techniques, e.g. [4,5]. However, these methods have not been proven to be scalable to handle large data sets of images. This work is inspired by [6], which demonstrates a hashing learning method. It proves that b-Bit Min-Hash method's estimators could be naturally integrated with learning algorithms such as SVM.

As a journal extension of our previous work [7], we exploit BitHash to generate binary hash values, and apply it to near-duplicate image detection. Compared with Min-Hash, by using only one bit per hash value, BitHash significantly reduced storage space for storing hash values, and its estimator can be computed much faster. The method provides an unbiased estimate of pairwise Jaccard similarity, and the estimator is a linear function of Hamming distance, which is very simple. We rigorously analyze the variance of BitHash, showing that as pairwise Jaccard similarity increases, the variance ratio of BitHash over the original min-hash

* Corresponding author. Tel.: +86 10 62796450; fax: +86 10 62782266.
E-mail address: xuhua@mail.tsinghua.edu.cn (H. Xu).

http://dx.doi.org/10.1016/j.knosys.2016.01.022
0950-7051/

decreases. And we examine BitHash in two typical applications, near-duplicate image detection and sentiment analysis.

Near-duplicate image detection is one of the most important problems in computer vision and multimedia. Large amount of near-duplicate images frequently occur on the web, e.g. photos of the same object or scene that are taken with subtle differences in personal photo albums. Efficient detection of near-duplicate images emerges as a more and more important problem in the context of image indexing and retrieval. One of the most popular representations of images is bag-of-visual-words, e.g. bag-of-SIFT-features. In the context of near-duplicate image detection, a reasonable similarity measurement based on this representation is the Jaccard similarity, since it is straightforward to expect that two near-duplicate images will have many visual words in common. To accelerate the similarity computation between queries and data in the database, BitHash can compress the data samples into compact binary codes, and the approximate nearest neighbor search can be conducted very efficiently by searching in Hamming space.

Sentiment classification (i.e., whether the sentiment orientation of the text is positive/negative) is one of the most important tasks in sentiment analysis [8]. Many machine learning approaches have been applied to this task, where the representation of a document plays a key role in classification. Bag of words, N-grams are simple and effective ways to build language models. However, these representations need large amounts of memory for large-scale classification. As the scale of text data on the Internet becomes larger, there is an emerging need to scale up sentiment analysis methods. In the field of sentiment analysis, researchers are usually aiming at improving the classification accuracy [9,10], but there is little literature about reducing the storage for large-scale corpus. In this paper, BitHash compresses each data sample into a compact binary hash code while preserving the pairwise similarity of the original data. The binary code can be used as a compressed and informative representation in replacement of the original data for subsequent processing.

We have three key contributions in this paper. Firstly, we propose One Bit Min-Hash (BitHash) method, which provides an unbiased estimate of pairwise Jaccard similarity, and the estimator is a linear function of Hamming distance; Secondly, we are the first to combine Locality Sensitive Hashing technique with machine learning method to scale up sentiment analysis; Finally, we apply BitHash into near-duplicate image detection and sentiment analysis, which can significantly reduce the feature dimensions as a more compressed and informative representation method, and help reduce the storage for large-scale images and texts substantially.

In Section 2, we introduce some related work on fundamental Locality Sensitive Hashing (LSH) technique Min-Hash method. After briefly reviewing Min-Hash, we introduce BitHash in Section 3, and rigorously analyze its variance. We conduct Jaccard similarity estimation experiment in Section 3.2, to verify the variance analysis in Section 4. In Section 5, we show experimental results of Near-duplicate image detection with BitHash. In Section 6, we describe how to integrate Locality Sensitive BitHash representation with SVM in order to deal with large-scale text sentiment classification problem. In Section 7 , we show our experimental results of sentiment analysis with BitHash on IMDB movie reviews data set. Finally, Section 8 concludes this paper.

2. Related work

In recent years, there exist some works about hashing methods for vectors [11–14], sets [7,15–17], and other more complicated structures like subspaces [18]. Ji et al. [11] proposed a batch-orthogonal hashing method for angular similarity and Liu et al. [12] introduced decorate graph hashing method. Ji et al. [18] proposed an effective method to generate angular-similarity binary hash codes for linear subspaces. Some other works introduced set similarity hashing methods. Zhao et al. [15] extended Min-Hash, and introduced a real-valued vectors set hashing method, and Ji et al. [16] introduced set similarity method Min-Max Hash, which is both effective and efficient. Our BitHash method is one of integer set similarity hashing methods. It also extends from Min-Hash and could be easily applied in near-duplicate image detection and sentiment analysis works. Apart from Min-Hash, the most relevant work is Min-Max Hash, which extended Min-Hash for integer set similarity problem, but could reduce the hashing time while achieving more accurate results. First, we take a brief review of Min-Hash method.

2.1. Min-Hash review

Before introducing BitHash method, we take a brief review of Min-Hash, which is a building block of our proposed BitHash.

Min-hash [1] is a popular hashing method for Jaccard similarity, which is the most widely-used pairwise similarity between two sets A and B, defined as

$$J(A,B)=\frac{|A\bigcap B|}{|A\bigcup B|}. \tag{1}$$

For data of bag-of-words representation, e.g. texts can be represented as sets of words or Ngrams, denote the dictionary (the whole set of elements) by W of which the cardinality is $|W|=d$. We assign each word with a unique ID from the non-negative integer set $I=\{0,1,2,\ldots,d-1\}$, and thus any set of words S is represented by a subset of non-negative integers in I.

Min-Hash outputs a hash value by first generating a random permutation $\pi: I\rightarrow I$, and then taking the smallest permuted ID, i.e. $min(\pi(S)):=min_{i\in S}\pi(i)$. It is proven [1] that for any two non-empty sets A and B,

$$Pr[min(\pi(A))=min(\pi(B))]=\frac{|A\bigcap B|}{|A\bigcup B|}=J(A,B). \tag{2}$$

Define random variable

$$X_\pi=\begin{cases}1, & min(\pi(A))=min(\pi(B))\\ 0, & \text{otherwise}\end{cases}.$$

Then

$$\mathbb{E}[X_\pi]=Pr[X_\pi=1]=J(A,B). \tag{3}$$

Generate K random permutations π_1, π_2, ..., π_K independently, and define correspondingly $X_{\pi_1},X_{\pi_2},\ldots,X_{\pi_K}$. The estimator of Min-Hash is

$$X=\frac{1}{K}\sum_{i=1}^{K}X_{\pi_i}=\frac{1}{K}\sum_{i=1}^{K}1_{min(\pi_i(A))=min(\pi_i(B))}, \tag{4}$$

which is an unbiased estimator of $J(A, B)$, and its variance is

$$Var[X]=\frac{1}{K}J(A,B)(1-J(A,B)). \tag{5}$$

However, the feature representation by using Min-Hash is consisting with hash integers, which couldn't be adapted directly with linear learning method such as SVM. So we propose to use a new approach called BitHash to deal with this problem.

3. BitHash

3.1. BitHash

BitHash is short for One-Bit Min-Hash, which is based on Min-Hash, while producing more compact hash code. One shortage of Min-Hash is that each of its hash value is an integer represented