Web Scale Photo Hash Clustering on A Single Machine

Xuewen Yang

June 2 2018

Abstract

This paper addresses the problem of clustering a very large number of photos in a stream into millions of clusters. This is particularly important as the popularity of photo sharing websites, such as Facebook, Google, and Instagram. Given large number of photos available online, how to efficiently organize them is an open problem.To address this problem, we propose to cluster the binary hash codes of a large number of photos into binary cluster centers. We present a fast binary k-means algorithm that works directly on the similarity-preserving hashes of images and clusters them into binary centers on which we can build hash indexes to speedup computation. The proposed method is capable of clustering millions of photos on a single machine in a few minutes.We show that this approach is usually several magnitude faster than standard k-means and produces comparable clustering accuracy. In addition, we propose an online clustering method based on binary kmeans that is capable of clustering large photo stream on a single machine, and show applications to spam detection and trending photo discovery.

1. Introduction

Photo sharing websites are becoming extremely popular,hundreds of millions of photos are uploaded every day.For example, Facebook announced it has about 300 million photo uploads every day. However, how to efficiently organize such huge online photo collections is becoming a challenge. In this paper, we propose to study the problem of clustering large photo collections at the scale of hundreds millions a day, This process has many practical applications.For example, clustering large photo collections into near-duplicate image clusters can help find spam photos.Online clustering photos into semantic clusters can be used to find time-sensitive photo clusters and trending events. For these scenarios, we need online clustering methods which can handle hundreds of millions photos a day and can store a very large number of centers in memory.

Image clustering is a well-studied problem in the literature [3]. However, how to efficiently cluster such huge collections of photos on a single machine has received little attention. This problem is challenging because 1) it is hard to compactly represent such huge photo collections; 2) it is computationally very inefficient to perform clustering on large datasets; 3) it is very inefficient to store and index increasing large number of cluster centers. The first problem has been addressed by recent works on similarity preserving hashing [2], that try to represent images as compact hash codes. For the second challenge, there is work using kd-tree to speed up the clustering process, but it does not address the third challenge, as kd-tree needs to store all the real valued centers in memory. Photo clustering will become infeasible when the number of clusters accumulates to tens of millions or even more.

In this paper, we try to address three challenges by developing a method that clusters image similarity binary codes into a set of compact binary centers, which can be easily indexed.The basic idea is illustrated in Figure 1. We first represent the photos using similarity preserving binary codes [4], enabling us to store large number of photos in memory. Then we propose a variant of the classic kmeans algorithm denoted as Binary k-means (Bk-means) that constrains the centers to be binary. The centers also live on the Hamming cube. This enables us to easily use a multi-index hash table to index the centers so that the nearest center lookup becomes extremely efficient.

2. Related Work

Image clustering is one of the fundamental research areas in computer vision research. It has been widely used in many different vision applications such as automatic discovery of object categories, finding trending or iconic photos, reconstructing story lines from photo streams, and 3D reconstruction from photo collections. One important application of image clustering in recent years is to automatically organize Internet photo collections.Many very large-scale datasets have been proposed to better study the problem of Internet computer vision.

Deep learning methods have been extremely successful for visual recognition and retrieval in recent years.In par-

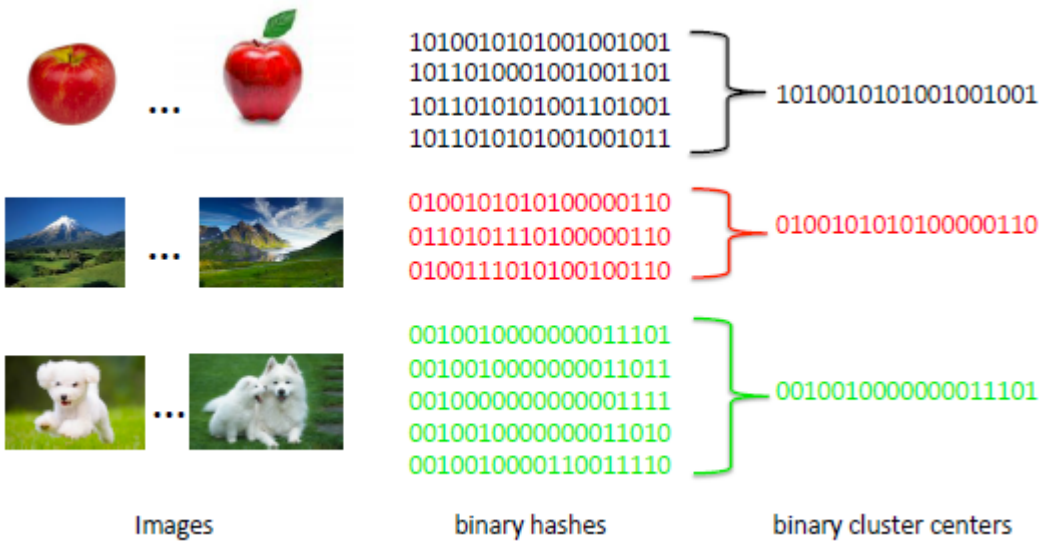

Figure 1. The problem setting of this paper. We are interested in clustering a large amount of image hash codes into compact binary centers.

ticular, training deep convolutional neural networks on large Internet collections has achieved great success.

Online clustering is another important problem that has been extensively studied in machine learning. Usually,it assumes one has very large data stream, and is unable to perform batch processing to such streams.

3. Fast Clustering on Binary Hash Codes

We introduce the methods for batch clustering on hashes in this section.First, we show that traditional k-means and kmedoids clustering methods can be adapted to this case to cluster hashes efficiently. Then we introduce the Binary k-means (Bk-means) algorithm.

We first briefly discuss how to perform standard k-means clustering on hashes. The hash codes use a binary representation and lie on the vertices of the hypercube. Every hash corresponds to one binary vector1 with values +1.Thus it is natural to perform k-means clustering on these vectors by treating them as floating binary vectors. In practice,we are not able to store large amount of such floating vectors in memory but can only store the compact hashes.Thus, we need to perform a decoding during the k-means algorithm.We first construct a lookup table from compact hash keys to floating vectors, and during the optimization of k-means, we directly lookup the floating vectors for the hash keys and perform updates. This method produces exactly the same result as standard k-means and has the same running time.

Another approach to explore the special properties of hashes to speedup clustering is applying k-medoids clustering,as described in [1]. Specifically, k-medoids directly picks original data points as centers, which guarantees that the centers are binary. This enables us to use fast distance computation to find the nearest center for every data point during optimization. In [1],they proposed to directly apply Hamming distance to compute the distance between every point to the centers.

References

[1] J. M. Frahm, P. Fitegeorgel, D. Gallup, T. Johnson, R. Raguram, C. Wu, Y. H. Jen, E. Dunn, B. Clipp, and S. Lazebnik. Building rome on a cloudless day. In *ECCV*, pages 368–381, 2010. 2

[2] Y. Gong, S. Lazebnik, A. Gordo, and F. Perronnin. Iterative quantization: a procrustean approach to learning binary codes for large-scale image retrieval. *PAMI*, 35(12):2916–2929, 2013. 1

[3] G. Kim, C. Faloutsos, and M. Hebert. Unsupervised modeling of object categories using link analysis techniques. In *CVPR*, pages 1–8, 2008. 1

[4] M. Norouzi and D. J. Fleet. Minimal loss hashing for compact binary codes. In *ICML*, pages 353–360, 2011. 1