Web Scale Photo Hash Clustering on A Single Machine

Yunchao Gong, Marcin Pawlowski, Fei Yang, Louis Brandy, Lubomir Boundev, Rob Fergus
Facebook

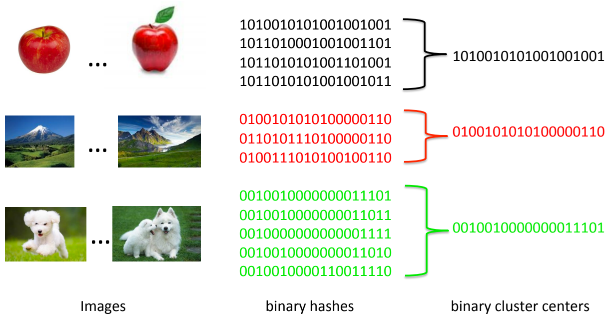


Figure 1: The problem setting of this paper. We are interested in clustering a large amount of image hash codes into compact binary centers.

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.

In this paper, we develop 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 [1, 3, 5], enabling us to store large number of photos in memory. Then we propose a variant of the classic *k*means 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 [4] to index the centers so that the nearest center lookup becomes extremely efficient. This can reduce the time complexity of the traditional *k*means from $O(nk)$ to $O(n)$, assuming we have n data points and k centers. This also significantly reduces the storage space of the centers, which we are representing as compact binary codes.

The binary hashing based k-means clustering mainly try to speedup the nearest center lookup. To enable efficient lookup of the nearest center, we use a constrained k-means formulation that constrains the mean to be binary. Given the means are binary, we can directly build a multi-index hash table [4] on the centers, and can efficiently find the *exact* nearest mean for any binary data point in constant time. It also significantly saves storage of the centers. We can have the following objective function:

$$\begin{aligned} &\min_{c_j} \textstyle\sum_i^n \sum_j^k \|x_i - c_j\|_2^2 \\ &\text{s.t.} \quad c_j \in \{-1, +1\}. \end{aligned} \tag{1}$$

Assuming c_j has already been computed, the problem is reduced to the assignment step of k-means, which can be easily accomplished by assigning each point x_i to its nearest center c_j. This can be done by building a multi-index hash table [4] on c_j and perform fast lookups for each x_i. When all the points have been assigned to its nearest center, the problem is how to optimize c_j with respect to the binary constrainss. By expanding Eq. (1), and only consider one cluster c_j and p points belonging to it, we have

$$\begin{aligned} &\min_{c_j} \textstyle\sum_i^p \|x_i - c_j\|_2^2 \\ = \quad &\textstyle\sum_i^p \|x_i\|_2^2 + \sum_i^p \|c_j\|_2^2 - \sum_i^p x_i c_j^T. \end{aligned} \tag{2}$$

We notice that $\sum_i^p \|x_i\|_2^2$ and $\sum_i^p \|c_j\|_2^2$ are both constants, because they are both binary variables. Thus the optimization problem can be reduced to:

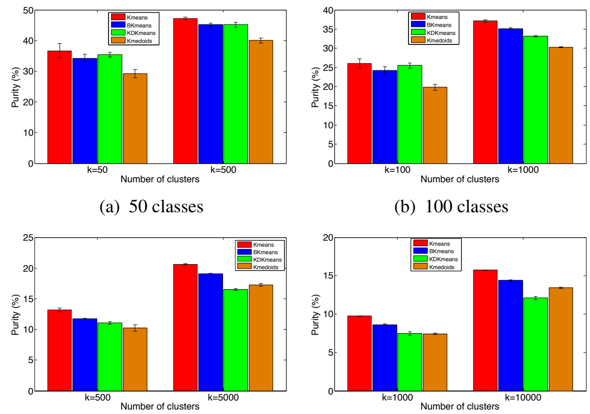
$$\begin{aligned} &\max_{c_j} \textstyle\sum_i^p x_i c_j^T = \max_{c_j} (\sum_i^p x_i) c_j^T \\ &\text{s.t.} \quad c_j \in \{-1, +1\}. \end{aligned} \tag{3}$$

The above problem can be solved by first computing the sum of all x_i as $m_j = \sum_i^p x_i$, and c_j can be obtained by:

$$c_{jk} = \text{sign}(m_{jk}) = \begin{cases} +1 & \text{if } m_{jk} \geq 0 \\ -1 & \text{if } m_{jk} < 0. \end{cases} \tag{4}$$

This gives an alternative optimization algorithm for solving Eq. (1), which iteratively solves two subproblems.

We have compared the proposed method with kmeans, kdtree based kmeans, and k-medoids, and some example results are shown in Figure 2. We also applied the method to online event detection, and got event photo clusters, as shown in Figure 3.



(a) 50 classes (b) 100 classes (c) 500 classes (d) 1000 classes

Figure 2: Comparison of clustering purity on subsets of ILSVRC2012 dataset. Each subset contains different number of classes.

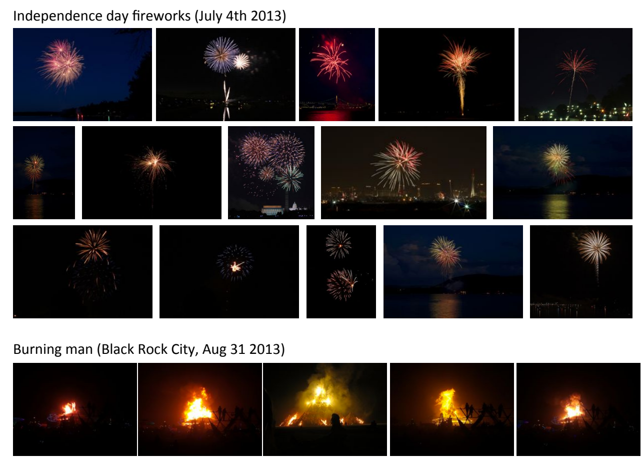


Figure 3: Example trending clusters our method found.

[1] Yunchao Gong, Svetlana Lazebnik, Albert Gordo, and Florent Perronnin. Iterative quantization: A Procrustean approach to learning binary codes for large-scale image retrieval. *PAMI*, 2012.

[2] Anil K. Jain and Richard C. Dubes. *Algorithms for Clustering Data*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1988. ISBN 0-13-022278-X.

[3] M. Norouzi and D. J. Fleet. Minimal loss hashing for compact binary codes. *ICML*, 2011.

[4] Mohammad Norouzi, Ali Punjani, and David J. Fleet. Fast search in hamming space with multi-index hashing. In *CVPR*. 2012.

[5] Yair Weiss, Antonio Torralba, and Rob Fergus. Spectral hashing. *NIPS*, 2008.

This is an extended abstract. The full paper is available at the Computer Vision Foundation webpage.