Web Scale Photo Hash Clustering on A Single Machine

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

Facebook

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 the storage of the centers. We can have the following objective function:

$$\min_{c_j} \sum_{i} \sum_{j} p \left\| x_i - c_j \right\|_2^2$$

s.t. $c_j \in \{-1, +1\}.$

(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 constrainst. By expanding Eq. (1), and only consider one cluster $c_j$ and $p$ points belonging to it, we have

$$\min_{c_j} \sum_{i} p \left\| x_i - c_j \right\|_2^2 = \sum_{i} \left\| x_i \right\|_2^2 + \sum_{j} \left\| c_j \right\|_2^2 - \sum_{i} x_i c_j^T .$$

We notice that $\sum_{i} \left\| x_i \right\|_2^2$ and $\sum_{j} \left\| c_j \right\|_2^2$ are both constants, because they are.

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

References: