Web Scale Photo Hash Clustering on A Single Machine

Xuewen Yang

June 4 2018

1. Experiments

1.1. Datasets, Features, and Protocols

We perform quantitative experiments on the ILSVRC2012 dataset, which is a subset of ImageNet [1]. It contains 1000 object categories and more than 1.2 million images. Each category roughly has more than 1000 images. We randomly pick 50, 100, 500, and 1000 classes from all 1000 classes, and construct four different datasets.

We use the activations from a deep neural network as the image features. The network is trained using a similar architecture as the Krizhevsky *et al.* [3]. We extract the 4096 dimensional activations from the last hidden layer as our image feature and normalize them to have unit norm. Then we apply the Iterative Quantization (ITQ) method [2] to quantize the features into 256-bit binary codes. These binary codes are the input of all the algorithms studied below.

1.2. Clustering Accuracy

We first evaluate the clustering performance using purity on various datasets, and the results are shown in Figure 1. We can find k-means always performs the best, as most other methods are approximating it or introduce certain constrainss. The Bk-means method is usually slightly (1-2%) worse than k-means. KDk-means works well for small number of centers, while its performance becomes worse when the dataset size and number of centers become large. Overall, Bk-means is still better than KDk-means,and we will show later that it is also faster. k-medoids works reasonably well, but is always worse than Bk-means.This is probably because Bk-means is not restricted to using data points as centers and allows more degrees of freedom.Some sample image clusters on ILSVRC are shown in Figure 2.

1.3. Clustering Speed

report the running time for different clustering methods on different dataset sizes in Table 2. All methods have achieved a faster speed than k-means. Overall, Bkmeans and k-medoids are fastest and are slightly faster than KDk-means. In particular, Bk-means method has achieved a significant speedup over k-means. For relatively small datasets, such as those with 50 classes and 50 clusters Bk-means achieved comparable speed to k-means. This is mainly because the number of centers is small, which limits the speedup factor of hashing. When we use more centers, the speedup becomes more significant. For example, with 1000 classes and 10,000 centers, we are able to achieve about 60-70 times speedup. We have also tested kmeans and bk-measuring 20M images and 1M centers, and bk-means can achieve over 400 times speedup than k-means.

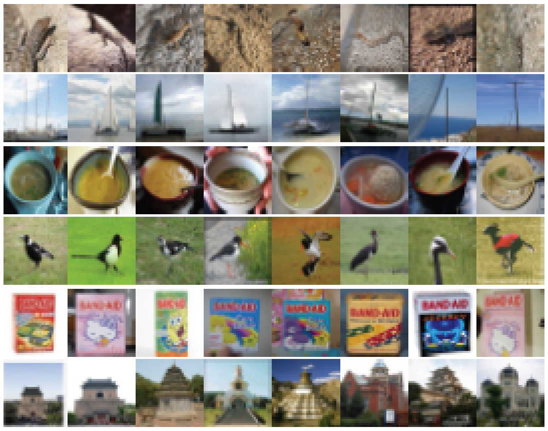
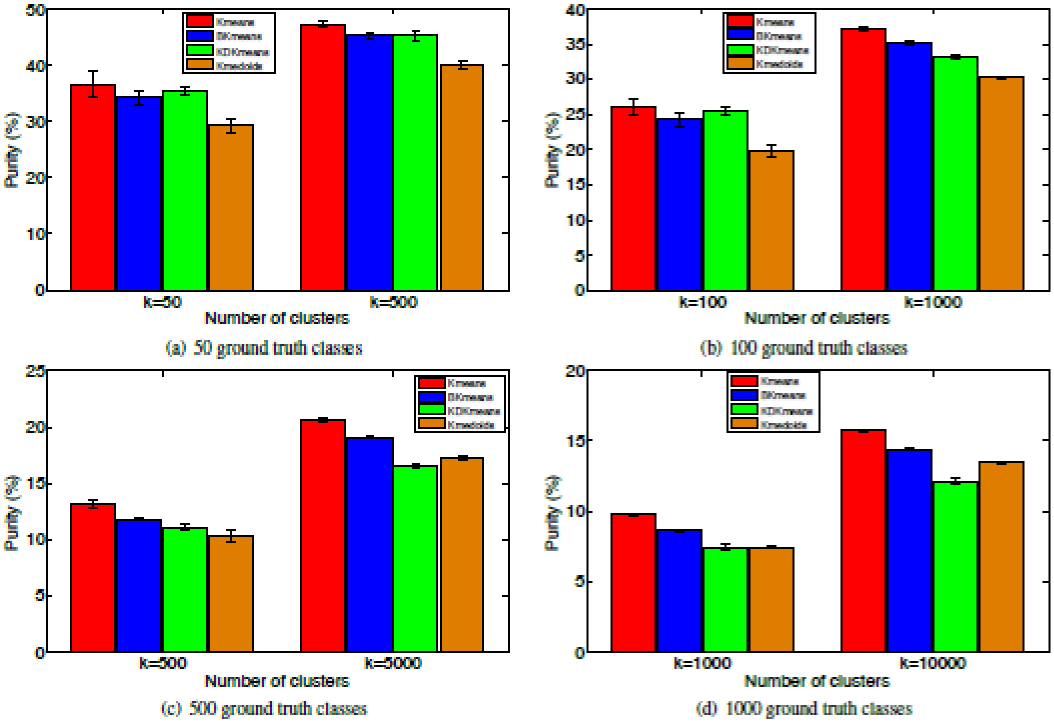


Figure 2. Sample clusters from the ILSVRC2012 dataset. Each line shows images in one cluster.

2. Online Clustering of Large Photo Streams

In this section, we introduce an online clustering algorithm that performs clustering on large photo stream. This has real world applications since photos are always uploaded as a stream. Our work is similar to previous works on online k-means clustering, but the main difference is we do not know the number of clusters and the number of clusters will continue to increase.

We describe the online clustering algorithm in Algorithm 2. The algorithm works as follows. We have an online photo stream coming in every second, and we use a fixed size Least Recently Used (LRU) cache to store the data. When the size of the cache grows to n, we run our Bkmeans clustering method on this batch of data and clus-



(a) 50 ground truth classes

(b) 100 ground truth classes

(c) 500 ground truth classes

(d) 1000 ground truth classes

Figure 1. Comparison of clustering purity on subsets of ILSVRC2012 dataset. Each subset contains different number of classes.

ter them into n/10 centers. Then we filter out images that are not visually consistent in each cluster. For every point in each cluster, we filter out images that having distance r away from its center. After filtering, all the clusters smaller than size t will be removed. In the meantime of clustering, the LRU cache collects another set of data, and when the clustering is done, we perform clustering to the new batch. For this new batch, we first assign its points to all the existing clusters when they are within r Hamming distance to the cluster centers. Then we run Bk-means to the rest and repeat the above steps. In the following experiments, we will set n = 1000000, t = 10, and r depends on particular application. This online clustering setting makes the need of compact cluster centers a even more desirable property, as the number of centers will grow significantly after running it for years. It enables us to process large online photo streams, and we will show two applications for it. Clustering 1 million images into 100,000 clusters usually only takes 4 minutes. This system enables us to cluster about 360 million photos in a day on a single machine

References

[1] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and F. F. Li. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009. 1

[2] Y. Gong, S. Lazebnik, A. Gordo, and F. Perronnin. Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval. *IEEE TPAMI*, 2013. 1

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *NIPS*, 2012. 1