Abstract

The goal of binary hashing is to learn a hash function that maps high-dimensional points to bit codes, which can be used to speed up search on large databases. Most papers use optimization approaches based on a suitable objective function with a difficult and inexact optimization. Recently, it has been shown that the hash function for a code bit may be learned independently from that of the other code bits. One simply optimizes a single-bit objective function defined on a random data sample, and then fits a binary classifier to the resulting codes. We show that it is even possible to dispense with the single-bit optimization, by assigning binary codes to the points based on their similarity to a randomly chosen seed point. This procedure is very simple, scalable, and is competitive with the state-of-the-art methods in retrieval metrics.

Advantages of the ensemble diversity approach

The ensemble-based approach gives several advantages:

- It is better or comparable to the optimization-based methods in terms of retrieval performance.
- Much simpler optimization: ILH deals with b independent problems each over N binary codes rather than 1 problem with Nb binary codes.
- Hence, faster training and better accuracy, because ILH deals with optimization problems of a smaller size.
- Training the b functions can be parallelized.

In this paper we show that by guessing the binary codes it is even possible to dispense with the single-bit optimization.

Previous works: learning the hash function

Most hashing papers try to minimize an affinity-based objective, which directly tries to preserve the original similarities in the binary space:

$$\min \mathcal{L}(h) = \sum_{x \in D} L(h(x)), \quad \text{h}(x), \quad \text{y}\in\mathbb{R}^D$$

where $x \in \mathbb{R}^D$ is the ith input data, h is the parameters of the hash function, $L(h)$ is a loss function that compares the codes for two images with the ground-truth value $y_{mn}$ that measures the affinity in the original space between the two images $x_{m}$ and $x_{n}$. Many such loss functions $L(z_{m}, z_{n}; y_{mn})$ exist, e.g.:

- KSH: $L(z_{m}, z_{n}; y_{mn}) = \|z_{m} - z_{n}\|^2$
- Laplacian: $L(z_{m}, z_{n}; y_{mn}) = y_{mn} \|z_{m} - z_{n}\|^2$

Optimizing $\mathcal{L}(h)$ is difficult because h is discrete.

Optimization-based approach: Many optimization-based methods have been proposed to optimize the objective approximately over the b-bit hash function. These methods have several limitations:

- The hash function outputs binary values, hence the problem is nonconvex and nonsmooth. An NP-complete problem over Nb variables has to be solved.
- They do not scale beyond a few thousand training points.
- In the end, there is little practical difference between the different objective functions and optimization algorithms proposed.

Ensemble-based approach: Rather than coupling the b hash functions, a recent method, Independent Laplacian Hashing (ILH) (Carreira-Perpiñán and Raziperchikolaei, NIPS 2016), proposed to train each hash function independently from each other.

To get good retrieval results, the single-bit hash functions have to be different from each other. ILH uses the ensemble learning techniques, like using different training subsets or different initializations for each single-bit hash function, to make the hash functions different from each other.

ILH minimizes the objective $\mathcal{L}(h)$ over a single-bit hash function $h$:

$$\min_h \mathcal{L}(h) = \sum_{n,m} y_{mn} h(x_n) h(x_m)$$

where $h(x_n) = (h(x_n_1), \ldots, h(x_n_N)) \in \{-1,1\}^N$ is a row vector of N bits, $h(x_n) = \mathcal{I}(h(x_n)), \quad \mathcal{I}(t) = 1$ if $t \geq 0$ and $-1$ if $t < 0$.

Experiments

Infinite MNIST contains 1 000 000 / 2 000 images for training/test, in 10 classes. The groundtruth is defined based on the labels of the images.

Sun dataset contains 23 537 / 2 000 images as training/test. The groundtruth is defined based on the commonality of objects among the two images.

We report precision as a function of number of bits. ISH-clever selects seeds by cycling over the C classes in the labeled datasets and achieves better results than randomly selecting the seeds (done by ISH). SVM-class trains C one-vs-all classifiers and reports the classification accuracy. When the ground-truth is given by the class label, SVM-class gives better precision than the hashing methods, but is inapplicable to the SUN dataset.