Supervised Discrete Hashing

Fumin Shen[1], Chunhua Shen[2], Wei Liu[3], Heng Tao Shen[4]
[1]University of Electronic Science and Technology of China. [2] University of Adelaide; and Australian Centre for Robotic Vision. [3]IBM Research. [4]The University of Queensland.

Recently, learning based hashing techniques have attracted broad research interests due to the resulting efficient storage and retrieval of images, videos, documents, etc. However, a major difficulty of learning to hash lies in handling the discrete constraints imposed on the needed hash codes. In general, the discrete constraints imposed on the binary codes that the target hash functions generate lead to mixed-integer optimization problems—which is generally NP hard. To simplify the optimization involved in a binary code learning procedure, most of the aforementioned methods choose to first solve a relaxed problem through directly discarding the discrete constraints, and then threshold the continuous outputs to be binary. This greatly simplifies the optimization but, unfortunately, the approximated solution is typically of low quality and often makes the final hash functions less effective, possibly due to the accumulated quantization errors. This is especially the case when long-length codes are needed.

Directly learning the binary codes without relaxations would be preferred if (and only if) a tractable and scalable solver is available. The importance of discrete optimization in hashing has been rarely taken into account by most existing hashing methods. Iterative Quantization (ITQ) [1] is an effective approach to decrease the quantization distortion by applying an orthogonal rotation on the transformed data. A limitation of ITQ is that it learns the orthogonal rotations on the pre-computed mappings (e.g., using PCA or CCA). The separate learning procedure means that ITQ usually achieves a suboptimal solution.

In this work, we propose a new supervised hashing framework, which aims to directly generate the binary hash codes, *effectively* and *efficiently*. To leverage the supervised information, we formulate the hashing problem in the framework of linear classification, where the learned binary codes (nonlinear feature vectors) are expected to be optimal for linear classification. More specifically, the learned binary codes can be viewed as nonlinear feature vectors of the original data. The label information is exploited such that these binary feature vectors are easy to be classified. Similar to the discrete boosting learning at the high level, we transform the original data into a nonlinear binary space, and then classify the original data in this binary space.

To implement this idea, we propose a joint optimization problem which jointly learns the binary embeddings and the linear classifier. Suppose that we have n samples $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^n$. We aim to learn a set of binary codes $\mathbf{B} = \{\mathbf{b}_i\}_{i=1}^n \in \{-1,1\}^{L\times n}$ to well preserve their semantic similarities, where the i^{th} column $\mathbf{b}_i$ is the L-bits binary codes for $\mathbf{x}_i$. We adopt following multi-class classification formulation $\mathbf{y} = G(\mathbf{b}) = \mathbf{W}^\top\mathbf{b} = [\mathbf{w}_1^\top\mathbf{b}, \cdots, \mathbf{w}_C^\top\mathbf{b}]^\top$ where $\mathbf{w}_k \in \mathbf{R}^{L\times 1}$, $k = 1, \cdots, C$ is the classification vector for class k and $\mathbf{y}$ is the label vector, of which the maximum item indicates the assigned class of $\mathbf{x}$. We then formulate the hashing problem as follows,

$$\min_{\mathbf{B},\mathbf{W},F} \sum_{i=1}^n L(\mathbf{y}_i, \mathbf{W}^\top\mathbf{b}_i) + \lambda||\mathbf{W}||^2 + \nu\sum_{i=1}^n ||\mathbf{b}_i - F(\mathbf{x}_i)||^2 \quad \text{s.t. } \mathbf{b}_i \in \{-1,1\}^L. \tag{1}$$

Here $L(\cdot)$ is the loss function. In this formulation, the hash functions $F(\mathbf{x}_i)$ are simultaneously learned to fit the generated binary bits. To better capture the nonlinear structure of the data, the hash functions are learned in a kernel space. The overall optimization problem is then solved in an iterative fashion with three corresponding sub-problems: each sub-problem solves for one variable while keeping other two variables fixed. With the ℓ_2 loss, it is trivial to solve for $\mathbf{W}$ and F. The most important $\mathbf{B}$ sub-problem writes

$$\min_{\mathbf{B}} \quad ||\mathbf{W}^\top\mathbf{B}||^2 - 2\mathrm{Tr}(\mathbf{B}^\top\mathbf{Q}) \quad \text{s.t.} \quad \mathbf{B} \in \{-1,1\}^{L\times n}. \tag{2}$$

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

Table 1: Comparative results of our method with discrete constraints or relaxed ones.

	Constraint	32 bits	64 bits	96 bits
Precision	Discrete	0.5090	0.4229	0.3515
	Relaxed	0.4718	0.3354	0.1685
MAP	Discrete	0.4307	0.4555	0.4582
	Relaxed	0.3777	0.4150	0.4244

Table 2: Comparison with state-of-the-arts on CIFAR with 64 bits.

Method	Precision	MAP	Training time	Test time
BRE	0.1299	0.1156	12042.0	6.4e-5
MLH	0.2251	0.1730	2297.5	3.2e-5
KSH	0.1656	0.3822	2625.0	3.1e-6
SSH	0.2860	0.2091	96.9	3.6e-6
CCA-ITQ	0.3524	0.3379	4.3	1.7e-6
FastHash	0.1880	0.4187	1340.7	7.1e-4
SDH	**0.4229**	**0.4555**	62.6	2.6e-6

where $\mathbf{Q} = \mathbf{W}\mathbf{Y} + \nu F(\mathbf{X})$ and $\mathrm{Tr}(\cdot)$ is the trace norm.

We choose to learn the binary codes $\mathbf{B}$ by the *discrete cyclic coordinate descent (DCC)* method. In other words, We learn $\mathbf{B}$ bit by bit. Let $\mathbf{z}^\top$ be the l^{th} row of $\mathbf{B}$, $l = 1, \cdots, L$ and $\mathbf{B}'$ the matrix of $\mathbf{B}$ excluding $\mathbf{z}$. Then $\mathbf{z}$ is one bit for all n samples. Similarly, let $\mathbf{q}^\top$ be the l^{th} row of $\mathbf{Q}$, $\mathbf{Q}'$ the matrix of $\mathbf{Q}$ excluding $\mathbf{q}$, $\mathbf{v}^\top$ the l^{th} row of $\mathbf{W}$ and $\mathbf{W}'$ the matrix of $\mathbf{W}$ excluding $\mathbf{v}$. Then we have w.r.t. $\mathbf{z}$:

$$\min_{\mathbf{z}} \quad (\mathbf{v}^\top\mathbf{W}'^\top\mathbf{B}' - \mathbf{q}^\top)\mathbf{z} \quad \text{s.t.} \quad \mathbf{z} \in \{-1,1\}^n. \tag{3}$$

This problem has the optimal solution

$$\mathbf{z} = \mathrm{sgn}(\mathbf{q} - \mathbf{B}'^\top\mathbf{W}'\mathbf{v}). \tag{4}$$

By carefully choosing loss functions of the classifier, the DCC method produces the *optimal* hash bits in a closed form, which consequently makes the entire optimization procedure very efficient. Therefore, the proposed binary code learning method can easily deal with large-scale datasets. We name the proposed supervised hashing method employing discrete cyclic coordinate as Supervised Discrete Hashing (SDH).

Discrete or Not? To see how much the discrete optimization will benefit the hash code learning, we perform a comparison of our hash formulation (1) with or without the discrete constraints. The comparative results on CIFAR are shown in Table 1. As can be seen, our discrete hashing framework SDH consistently yields better hash codes than the relaxed one by removing the sign function. In particular for precision, the performance gaps between these two methods are increased with longer hash bits.

Conclusion By the cyclic coordinate descent method, the proposed SDH attained a high-quality discrete solution with a very low computational cost. The efficacy of SDH is validated by the superior results over several state-of-the-art hashing methods (see Table 2). Refer to [2] for full details of this work. The code for SDH is available at `https://github.com/bd622/DiscretHashing`.

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

[2] F. Shen, C. Shen, W. Liu, and H. T. Shen. Supervised Discrete Hashing. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2015.