Direct Hashing without Pseudo-Labels

Feng Zheng, Heng Huang*
Electrical and Computer Engineering, University of Pittsburgh
3700 O'Hara Street, Pittsburgh, PA, USA 15261
{feng.zheng, heng.huang}@pitt.edu

Abstract

Recently, binary hashing has been widely applied to data compression, ranking and nearest-neighbor search. Although some promising results have been achieved, effectively optimizing $\mathbf{sign}$ function related objectives is still highly challenging and thus pseudo-labels are inevitably used. In this paper, we propose a novel general framework to simultaneously minimize the measurement distortion and the quantization loss, which enable to learn hash functions directly without requiring the pseudo-labels. More significantly, a novel W-Shape Loss (WSL) is specifically developed for hashing so that both the two separate steps of relaxation and the NP-hard discrete optimization are successfully discarded. The experimental results demonstrate that the retrieval performance both in uni-modal and cross-modal settings can be improved.

Introduction

Actually, almost all existing methods of binary embedding (hashing) choose $H(x) = \mathbf{sign}(F(x)) \in \{1, -1\}^K$ as the hash function to binarize samples $x \in X$, where F captures some specific properties, X is a sample set and K is the length of codes. Generally, a minimization problem to keep the measurement consistency of relationships between the original and learned Hamming spaces can be written as:

$$H^* = \arg\min_H \mathcal{R}(H, X), \tag{1}$$

where $\mathcal{R}(H, X) = \sum_{i,j} \mathcal{M}(S(H(x_i), H(x_j)), S(x_i, x_j))$. S describes the relationships between any pair of samples in the corresponding space and $\mathcal{M}$ is used to measure the difference between the two measurements of relationship. However, the immediate challenging is how to optimize $\mathbf{sign}$ related ideal objective functions, because the step function $\mathbf{sign}$ is non-differentiable at 0 in the ordinary sense.

To overcome the problems induced by $\mathbf{sign}$ function, recently, three strategies are normally used. Firstly, the discrete codes $H(x)$ can be obtained from the objective by discrete optimization and, then, the best hash functions would be the ones approach to these codes mostly (Lin et al. 2013). Nevertheless, discrete optimization is an NP hard problem when the length of codes is large and, in practice, some relaxation tricks are inevitably needed. Moreover, the originally preserved properties are likely to be ruined after the two separated steps of relaxation and approximation, since the codes were learned without knowledge of functions. Secondly, to improve the two-step hashing, several works (Gong et al. 2013b) consider to optimize the codes and the functions, alternatively. In fact, beside the NP hard problem, iterative optimization between codes and functions is computationally expensive. More importantly, once the model generates a bad results in one step, then the effect of decreasing would be amplified in another step, due to the second optimum is the one which fits the first results mostly. Thirdly, some constraints, such as hinge loss (Rastegari et al. 2013), are introduced to learn functions to avoid that $F(x)$ is close to zero, when $F(x)$ is directly used. However, the hinge loss needs that the codes should be given in advance and thus, an alternative optimization is also required.

In addition to the above essential problems, a common inferior trait is that, before learning the hash function, the certain binary codes $H(x)$ need to be given in advance. Then, learning hash functions could be considered as training a set of binary classifiers, which regard $H(x)$ as the ground-truth label. However, the tasks of hashing and classification are fundamental different and the codes between them have varied practical significance as well. Roughly speaking, the loss in classification is used to measure the difference between $F(x)$ and a fixed label (one of them 1 or -1) but the loss in hashing only needs to measure the difference between $F(x)$ and either of them (1 or -1). For example, to improve the measurement consistency, $F(x)$ in hashing could freely jump from -1 to 1 and vice versa, whilst for classification, it will make a big classification loss.

Can we learn a set of hash functions as training classifiers for classification but without introducing the surrogate ground-truth labels? The answer is definitely yes. In this paper, we propose a novel general framework to directly learn a set of hash functions without the ground-truth labels, which can minimize the quantization loss and the measurement distortion, simultaneously. More significantly, a new W-Shape Loss (WSL) function is developed so that the quantization loss between $F(x)$ and the binary codes could be directly minimized and the most important feature of W-shape loss having the two equal minimums at 1

*To whom all correspondence should be addressed.

and -1 makes it rightly suitable to the hash function learning. Indeed, the essential role of WSL is to minimize the minimum distance ($i.e.\ \min(|f(x)-1|, |f(x)-(-1)|)$) between the mapped value $f(x)$ and the set $\{-1, 1\}$ with less measurement distortion, where f is a function for one code. Moreover, any kind of measurement consistency considered as the regularization item could be seamlessly incorporated into the framework. Finally, by minimizing the overall objective (loss + regularization), we can directly obtain the parameters of functions without both the NP-hard discrete optimization and the two separated steps of relaxation between $H(x)$ and $F(x)$.

In summary, our contributions are as follows: 1) Our method does not need pseudo-labels which are usually obtained by a separate optimization in previous work. 2) WSL avoids the NP-hard discrete optimization required in previous work, and does not need the alternative optimization between codes and mappings in previous work. 3) We introduce a new quantization loss to minimize the minimum distance between the mapped value and the code set. 4) We propose a general framework which enables to seamlessly incorporate any kind of measurement consistency and quantization loss.

Related Work

Self-taught hashing (Zhang, Wang, and Lu 2010) firstly decomposes the learning procedure into two steps and, then the idea is extended in (Lin et al. 2013). The first step of codes learning can typically be formulated as binary quadratic problems, and the second step of function learning can be accomplished by training standard binary classifiers. (Shen et al. 2015) formulate the hashing framework as a multi-class classification, where the learned binary codes (surrogate labels) are expected to be optimal for classification. To improve the two-step hashing, several works (Gong et al. 2013b) consider to optimize the codes and the functions, alternatively. Another group methods follow the similar scheme of two-step hashing but, the first step considers the affinity-based loss as a smooth problem and the second step minimizes the quantization loss by finding the thresholds for the learned continuous variables. Iterative Quantization (ITQ) (Gong et al. 2013b) minimize the quantization error of mapping the PCA-projected data to vertices of the binary hypercube. Similarly, Isotropic hashing (Kong and Li 2012) can produce embedded dimensions for the projected data with isotropic variances thus reduce the quantization error. Followed the alternative quantization, various techniques including locally linear reconstruction weight (Irie et al. 2014), graph Laplacian matrix (Zhao, Lu, and Mei 2014) and bilinear projection (Gong et al. 2013a) are applied into the first step.

Moreover, the hinge loss (Mu, Shen, and Yan 2010; Rastegari et al. 2013; Zheng, Tang, and Shao 2016; Zheng and Shao 2016) could be used to minimize the risk of the auxiliary variable close to 0. In (Mu, Shen, and Yan 2010), a hinge loss is used to learn hash function one by one with considering the similarity-similarity difference. Recently, the hinge loss is used to learn a set of hash functions in cross-modal setting (Rastegari et al. 2013; Zheng, Tang, and Shao 2016). (Zheng, Tang, and Shao 2016) proves that, by incorporating the hinge loss, the discrete optimization problem could be solved certainly by minimizing an differentiable upper bound. In (Gong et al. 2012), a orthogonal transformation is searched so that the sum of cosine similarities (Angular quantization loss) between each data point and its corresponding binary landmark is maximized.

Hashing can be applied for fast realizing various tasks. In (Zheng and Shao 2016), binary codes are learned to fast person re-identification. (Liu et al. 2014) proposes a collaborative hashing scheme for image search and recommendation. A structure sensitive hashing based on cluster prototypes is designed in (Liu et al. 2016) to discover prototypes. Later, (Liu et al. 2017) proposes an adaptive binary quantization method that learns a discriminative hash function with prototypes associated with small unique binary codes. In (Guo, Ding, and Han 2017), an $l_{p,q}$-norm loss function is proposed to conduct the l_p-norm similarity search. The l_∞-norm distortion is also introduced for robust hashing. Sparse hashing is designed to optimize the integration of anchors so that the features can be better binarized (Guo et al. 2017). Furthermore, we think that hashing can be applied for other tasks such as tracking (Chen et al. 2017), texture synthesis (You et al. 2016) and segmentation (You et al. 2011) etc.

It is worth to point out that the square loss, the hinge loss and the angular quantization loss are from the area of classification. More importantly, surrogate labels must be given in advance and then the problem of hashing could be solved as training a set of classifiers. In this paper, we propose a novel framework which can minimize the quantization loss and the affinity distortion, simultaneously.

The proposed method

Given a dataset $X = (x_1, \cdots, x_N) \subset R^d$, the hashing is to learn a set of hash functions $H = \{h_k : k = 1, \cdots, K\}$ to embed the x_i into binary codes $Y = (y_1, \cdots, y_N) \subset \{-1, 1\}^K$ with considering certain consistencies between the two spaces. Among them, N is the number of samples and d is the dimension of the original feature space. For simplicity, we depict the binary codes as $y_i = H(x_i) = (h_1(x_i), \cdots, h_K(x_i))^T$ and the kth code as $y_i^k = h_k(x_i)$.

Measurement consistency

Generally, the core of hashing is that the relationship measurements between samples represented in the original feature space and the learned binary code space should be consistent (Wang et al. 2016). Considering varied properties required by certain tasks, different models exploited diverse frameworks but, in fact, without complicated mathematical operations, all of them could be derived from the general problem in Eq. 1: $\mathcal{R}(H, X) = \sum_{i,j} \mathcal{M}(S(y_i, y_j), S(x_i, x_j))$. $\mathcal{M}$ which depends on the choices of measurements could be the operation of product or the square of deviation.

On the one hand, the most popular $S(y_i, y_j)$ is the Hamming distance $D_h(y_i, y_j)$. Moreover, if we have $y_i, y_j \in \{-1, 1\}^K$, thus the following two equations hold: $2D_h(y_i, y_j) = K - y_i^T y_j$ and $||y_i - y_j||_2^2 = 2K - 2y_i^T y_j$.

Thus, the inner product and the Hamming distance could be directly connected. On the other hand, $S(x_i, x_j)$ is a kind of pair-wise relationships in the original space. Local structures, global statistical constraints and semantic attributes can be used to define $S(x_i, x_j)$ and then guide the leaning of hash functions. Furthermore, beside the pair-wise relationships, some other types of list-wise measurements, such as triplet (x_i, x_j, x_l) and multi-group (similar and dissimilar groups), could be also derived as the combination of several pair-wise measurement items.

If the Hamming distance is selected for $S(y_i, y_j)$, $S(x_i, x_j)$ is a kind of similarity in the original space and the operation of product is used in $\mathcal{M}$, then we have the minimizing problem as $\mathcal{R}(H, X) = \sum_{i,j} D_h(y_i, y_j)S(x_i, x_j) = \sum_{i,j} ||y_i - y_j||_2^2 S(x_i, x_j)/2$. Furthermore, we have

$$\mathcal{R}(H, X) = \sum_k \mathcal{R}(h_k, X), \tag{2}$$

where $\mathcal{R}(h_k, X) = \sum_{i,j} ||y_i(k) - y_j(k)||_2^2 S(x_i, x_j)/2$. Simply, the learned distance is expected to be smaller if the similarity in the original space is larger and vice versa.

Quantization loss

Without exception, all hashing methods choose $\mathbf{sign}$ function to fill the gap between H and F as $y_i = H(x_i) = \mathbf{sign}(F(x_i))$, where $F = \{f_k : R^d \to R, k = 1, \cdots, K\}$ is the corresponding continuous functions and $y_i^k = h_k(x_i) = \mathbf{sign}(f_k(x_i))$. Without ambiguity, we denote binary representation matrix of X by all hash functions as $Y = \mathbf{sign}(F(X))$ and a row vector of X by function f_k only as $y^k = \mathbf{sign}(f_k(X))$. The parameters of functions including deep neural networks, eigenfunctions, kernel-based and linear functions, will be optimized in the stage of training.

However, $\mathbf{sign}$ function is non-differentiable at 0. Thus, most optimization strategies which used the derivative of objective function are unsuitable to this problem. In this paper, we resort to an alternative novel scheme to learn the function F directly, by simultaneously minimizing a novel quantization loss and measurement distortion defined in Eq. 1. For every sample, a set of multiple labels $s = \{1, -1\}$ is given. Hashing is to learn a function which embeds x into its set s. The quantization loss is generated only when $f(x)$ is far from the corresponding set s. Thus, with minimizing the measurement distortion, the nature of hashing is to minimize the following minimum:

$$\mathcal{L}(f(x), s) = \min(|f(x) - 1|, |f(x) - (-1)|) \tag{3}$$

$\mathcal{L}(f(x), s)$ is the quantization loss. The following theorem guarantees that zero loss can be achieved.

Theorem 0.1. *Given a dataset X, there must exist a differentiable function f so that the equation holds: $\mathcal{L}(f_k(X), s) = 0$, where $\mathcal{L}(f_k(X), s) = \sum_i \mathcal{L}(f_k(x_i), s)$.*

Corollary 0.1. *Given a dataset X, if all members in F can achieve zero quantization loss, then we have $F(X) \circ F(X) = J_{KN}$, where J_{KN} is a $K \times N$-size matrix whose elements are 1 and the symbol $\circ$ denotes the Hadamard product between the two matrices.*

All proofs of theory, lemma and corollary will be given in our supplementary materials. Hadamard product is an operation that takes two matrices of the same dimensions, and produces a matrix whose element is the product of corresponding elements in the original two matrices. Therefore, by considering both Eq. 1 and quantization loss, our objective can be defined as:

$$F^* = \arg\min_F \mathcal{R}(F, X), \ s.t. \ F(X) \circ F(X) = J_{KN}. \tag{4}$$

Obviously, we have $\forall k, \ f_k(X) \circ f_k(X) = J_{1N}$ and $\forall i, \ F(x_i) \circ F(x_i) = J_{K1}$ where J_{1N} is a row vector with same size of $f_k(X)$ and J_{K1} is a column vector with same size of $F(x_i)$.

The advantages of the objective function in Eq. 4 are obvious and distinctive. Firstly, the $\mathbf{sign}$ function has been removed from the objective function thus the classical derivative based methods (gradient descend) could be used while, more importantly, the gap between the real value and discrete number has been considered as well. Secondly, the complicated procedure of two-step optimization to search the suboptimal Y and F iteratively is successfully avoided. Finally, the properties can be directly preserved without any further loss in the step of quantization.

W-Shape Loss

Rather than satisfy equations in Eq. 4 strictly, allowing certain loss for some samples is acceptable: $F(X) \circ F(X) - J_{KN} = B(F, X)$, where $B(F, X)$ is a $K \times N$ matrix whose all elements are required to be close to zero. However, instead of approximating to zero from two directions, we use a substitute quantity defined as:

$$\mathcal{L}(F, X) = J_{1K}(\mathbf{ln}(B(F, X) \circ B(F, X)))J_{N1} \tag{5}$$

where $\mathbf{ln}(\cdot)$ is the natural logarithm function which executes on every element of the matrix. It is obvious that $\mathbf{ln}$ is a monotonous and differential function. Moreover, it is easy to prove that minimizing $\mathcal{L}(F, X)$ can guarantee that all the elements of $B(F, X)$ will be close to zero. This step is similar to the barrier method in Lagrangian multiplier, which is used to transfer inequalities into equalities. In our framework, we use $B(F, X)$ to relax the equalities.

Next, we will see the essential roles of this relaxation when we investigate the function individually. Thus, for $f_k(x_i)$, the constraint requires that the following item should be minimized:

$$\mathcal{L}(f_k, x_i) = \mathbf{ln}((f_k(x_i)^2 - 1)^2) \tag{6}$$

Obviously, $\mathcal{L}(f_k, x_i)$ is differential but nonconvex. The plot of the function has been illustrated in Fig. 1. The most significant property is the W-shape and, more importantly, it depict the distance between $f_k(x_i)$ and 1 or -1. Thus, we name it as "W-shape" loss, which can be used to learn hash function. The W-shape makes the loss feasible to be the minimum when $f_k(x_i)$ is 1 or -1.

We derive our framework from one case of zero quantization loss to a more reasonable problem allowing some acceptable quantization losses and then drive our W-shape loss. That is to say, beside the objective function in Eq. 4,

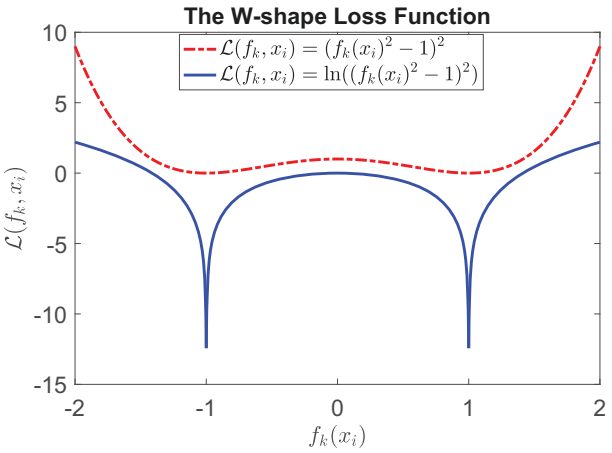

Figure 1: W-shape loss functions. A function $\mathcal{L}(f_k, x_i) = (f_k(x_i)^2 - 1)^2$ of same characteristics has been shown as well. The domain of the loss in the plots is $f(x) \in [-2, 2]$ and, actually, the loss functions are monotonous outside this domain. The W-shape functions have two equal minimums at $f(x) = 1$ and -1.

the quantization loss should be minimized simultaneously. Therefore, we can obtain our new general framework for hash function learning:

$$F^* = \arg\min_F \mathcal{L}(F, X) + \mu\mathcal{R}(F, X), \tag{7}$$

where μ is used to balance the two parts. In this objective, the first item is the quantization loss while the second one is the regularization item for measurement distortion. Apparently, the overall objective function is non-convex but, importantly, it is differential. In this general hashing framework, different W-shape losses can be explored while varied consistency preserving strategies can be incorporated into the objective as well.

Algorithm 1 WSL Hashing

Input: Training dataset X and parameters: μ, ν and K.
Output: $F(x) = (w_1, \cdots, w_K)^T x$.
Initialisation:
Randomly initiate $w_1^0, \cdots, w_K^0$.
Construct S for X and calculate matrix L.
For $t = 1, \cdots,$; $k = 1, \cdots, K$
Compute values $f_k^{t-1}(X)$ on X using w_k^{t-1}.
Calculate C_k using $w_1^t, \cdots, w_{k-1}^t, w_{k+1}^{t-1}, \cdots, w_K^{t-1}$.
Compute barriers $B(f_k^{t-1})$.
Calculate derivatives $\frac{\partial \mathcal{R}(f_k)}{\partial f_k}$, $\frac{\partial \mathcal{O}(f_k, C_k)}{\partial f_k}$ and $\frac{\partial \mathcal{L}(f_k)}{\partial f_k}$.
Run optimizer $\mathbf{MMA}(\mathbf{L}(f_k, \mu, \nu), \frac{\mathbf{L}(f_k, \mu, \nu)}{\partial f_k}, w_k^{t-1})$.
Output w_k^t and compute $\bigtriangledown \mathbf{L}_k^t$ and $\bigtriangledown \mathcal{O}_k^t$.
If the two conditions in Eq. 12 are false
$w_k^t = w_k^{t-1}$.
End
End If satisfy conditions: **Exit**.

Optimization

We will solve the differentiable problem in Eq. 7 with considering the orthogonality of the learned codes.

Orthogonal constraints

To avoid the trivial solutions, two orthogonal constraints are necessarily considered so the samples could be evenly mapped to each hash code. The first one is the bit balance means that each bit has a around fifty percent chance of being 1 or -1. And the second one is the bit uncorrelation means that different bits need to be independent. The mathematical formulations of the two constraints are given as: (1) $\forall k,\ f_k(X) J_{1N}^T = 0$. (2) $\forall k \neq l,\ f_k(X) f_l(X)^T = 0$. To unify them, we consider J_{1N}^T as a function where $J_{1N}^T(X) = J_{1N}^T$. We firstly select one projection f_k, then the remaining including $f_l(X), l \neq k$ and $J_{1N}^T(X)$ are required to be orthogonal to $f_k(X)$. For simplicity, X will be omitted from $f_k(X)$ and $F(X)$ in the following.

Theorem 0.2. *Given a dataset X and a hypothesis set F, if one function $f_k \in F$ satisfies:*

$$\mathcal{O}(f_k, C_k) = f_k C_k f_k^T = 0, \tag{8}$$

where $C_k = \sum_{l \neq k} f_l^T f_l + J_{1N}^T J_{1N}$, then the two constraints bit balance (1) and bit uncorrelation (2) hold with respect to the given f_k.

Using Theory 0.2, we transfer the two constraints into an unified condition. Totally, considering both the general framework introduced in 7 and the integrated constraint given in 8, we can obtain our final optimization objective:

$$F^* = \arg\min_F \mathcal{L}(F) + \mu\mathcal{R}(F),\ s.t.\ \forall k, \mathcal{O}(f_k, C_k) = 0. \tag{9}$$

The algorithm is summarized in Algorithm 1.

Divide and conquer

Obviously, the objective in Eq. 9 is a complicated nonlinear and non-convex optimization problem with a nonlinear equality constraint. To solve this objective, we divide the problem into several sub-problems which are relatively independent with each other. Then, a greedy algorithm is adopted to optimize the sub-problem individually. To this end, we first choose one function f_k to be optimized and fix all remaining functions. Therefore, we obtain the objective function of sub-problem:

$$f_k^* = \arg\min_{f_k} \mathcal{L}(f_k) + \mu\mathcal{R}(f_k), s.t.\ \mathcal{O}(f_k, C_k) = 0. \tag{10}$$

In this sub-objective for f_k, the consistency item is $\mathcal{R}(f_k) = \sum_{i,j} ||f_k(x_i) - f_k(x_j)||_2^2 S(x_i, x_j)/2 = f_k L(f_k)^T/2$, where $L = D - S$ is a Laplacian matrix and D is a diagonal matrix, whose elements are $D(x_i, x_i) = \sum_j S(x_i, x_j)$. And the loss function is $\mathcal{L}(f_k) = (\mathbf{ln}(B(f_k) \circ B(f_k)))J_{N1}$, where $B(f_k)$ is the kth row of $B(F)$. The following theory can be used to demonstrate the relationship between the original problem and the sub-problems.

Theorem 0.3. *Using a Lagrange multiplier ν, the Lagrangian for the original problem is:* $\mathbf{L}(F, \mu, \nu) = \mathcal{L}(F) +$

$\mu\mathcal{R}(F) + \nu\sum_k \mathcal{O}(f_k, C_k)$ *and the Lagrangian for the sub-problem for function* f_k *is:* $\mathbf{L}(f_k, \mu, \nu) = \mathcal{L}(f_k) + \mu\mathcal{R}(f_k) + \nu\mathcal{O}(f_k, C_k)$. *Then the following equation holds:*

$$\mathbf{L}(F, \mu, \nu) = \sum_k \mathbf{L}(f_k, \mu, \nu). \tag{11}$$

From the theory 0.3, we can see that the original problem consists of these sub-problems and is completely without any relaxation. Every sub-problem, which only focuses on optimizing one function, is connected to others using the orthogonal constraint only. In fact, if the sub-problem is convex, then the divide-conquer strategy guarantees that the alternative optimization can converge to a local optimum because, in every step, the reduction of overall objective must be non-negative[1]. However, in our general framework, we consider to use more relaxed nonlinear and non-convex items both for loss and consistency to improve the flexibility of the framework. To achieve this, an acceptance-rejection optimizer is proposed to selectively accept the local optimum f_k^t of the sub-problem in the current round t. The following theory guarantees that the Lagrange of original problem $\mathbf{L}(F, \mu, \nu)$ must converge into a local minimum at least.

Theorem 0.4. *If* f_k^t *is accepted in the* t*th round of optimization only when the equalities holds:*

$$\bigtriangledown\mathbf{L}_k^t > 0, \bigtriangledown\mathbf{L}_k^t > \bigtriangledown\mathcal{O}_k^t, \tag{12}$$

where $\bigtriangledown\mathbf{L}_k^t = \mathbf{L}(f_k^{t-1}, \mu, \nu) - \mathbf{L}(f_k^t, \mu, \nu)$ *and* $\bigtriangledown\mathcal{O}_k^t = \mathcal{O}(f_k^t, C_k^t) - \mathcal{O}(f_k^{t-1}, C_k^t)$*, then, the Lagrange of original problem* $\mathbf{L}(F, \mu, \nu)$ *can consistently decrease.*

In this theory, f_k^{t-1} is the accepted local optimum in the last round $t-1$. Obviously, $\bigtriangledown\mathbf{L}_k^t$ in Eq. 12 is likely to be non-negative, because f_k^{t-1} is the initial and f_k^t is the optimal in current round of optimization. This step is similar to the module of rejection in Sequential Monte Carlo methodologies, where the undesired samples will be abandoned in the current round of sampling. The division of optimization makes it feasible to derive the stochastic gradient descend for hashing or ensemble learning based hashing (Carreira-Perpinan and Raziperchikolaei 2016), in which the mini batches of X could be used sequentially or treated separately for different bits.

Derivative of sub-problem

Here, we offer the derivatives of $\mathbf{L}(f_k^t, \mu, \nu)$ with respect to f_k and then, the derivative of Lagrange with respect to the parameters θ of function f_k could be easily calculated using chain rule $\frac{\partial\mathbf{L}}{\partial\theta} = \frac{\partial f_k}{\partial\theta}\frac{\partial\mathbf{L}}{\partial f_k}$. Thus, we have $\frac{\partial\mathbf{L}(f_k,\mu,\nu)}{\partial f_k} = \frac{\partial\mathcal{L}(f_k)}{\partial f_k} + \mu\frac{\partial\mathcal{R}(f_k)}{\partial f_k} + \nu\frac{\partial\mathcal{O}(f_k,C_k)}{\partial f_k}$. It is easy to obtain the last two derivatives: $\frac{\partial\mathcal{R}(f_k)}{\partial f_k} = Lf_k^T$ and $\frac{\partial\mathcal{O}(f_k,C_k)}{\partial f_k} = 2C_kf_k^T$. However, due to the Hadamard product in the loss, the derivative of the first item could not be computed directly. In (Bentler and Lee 1978), the derivative of a function defined by Hadamard product is fully discussed. Hence, we can use the following corollary to calculate the derivative of the W-shape loss.

Corollary 0.2. $\frac{\partial\mathcal{L}(f_k)}{\partial f_k} = \mathbf{diag}(f_k)\frac{4}{B(f_k)^T}$*, where the division in* $\frac{4}{B(f_k)^T}$ *is executed on each element separately and results in a vector of same size to* $B(f_k)^T$.

The Lagrangian $\mathbf{L}(f_k^t, \mu, \nu)$ is a nonlinear and non-convex optimization problem without constraint. Our desired hash function can be searched at the minimum of the Lagrangian. Given a specific function f_k and consistency, we can calculate the value of f_k and its derivative w.r.t. its parameters θ. Then, the value and the derivative of $\mathbf{L}(f_k^t, \mu, \nu)$ can be computed as well. Using these quantities, several existing nonlinear and non-convex optimizers (Bottou, Curtisy, and Nocedal 2016; Johnson 2008) could be directly used. In this paper, a method of globally convergent method-of-moving-asymptotes (MMA) algorithm for gradient based local optimization (Svanberg 2002) is selected and others in this library (Johnson 2008) can achieve similar results.

To make our optimization feasible when $|f(x)|$ approximates to 1, we only need to use a surrogate gradient which is computed from $(f(x)^2-1)^2$ to update the parameters. While the loss can be also computed based on original loss function $log(f(x)^2-1)^2$. Surrogate gradient algorithm (we mentioned above) is a general scheme to optimize the parameters when the original gradient is singular or non-existence. Intuitively, when $|f(x)^2 - 1|$ is less than a very small value, it is unnecessary to further update the parameters because the goal of minimizing quantization loss is achieved.

Specific Function and Consistency

Actually, any kind of functions, including linear, kernel-based and neural networks based models, could be used in our framework to embed samples into the codes in Hamming space, as long as they have derivatives or some surrogates of derivative. In this paper, to mainly illustrate the performance of W-shape loss function, we select linear functions to learn the binary codes. Therefore, assuming the $f(X) = w^TX$, we have $\frac{\partial f(X)}{\partial w} = X$. Actually, all the related models introduced in (Yan et al. 2007), which are originally proposed for dimensionality reduction, can be used as the definition of measurement consistency $f(X)Lf(X)^T$.

Moreover, the proposed general framework can be used to learn hash codes in a cross-modal setting. We consider a recent proposed method of Hetero-manifold regularization (Zheng, Tang, and Shao 2016) which incorporates both within-modality local structure and between-modality supervised information, as the consistency preserving. We use linear functions $f(X) = w^TX$ for M number of modalities, where $w^T = ((w^1)^T, (w^2)^T, \cdots, (w^M)^T)$ consists of projections from all the M modalities. In this case, we have the multi-modal data matrix $X = \mathbf{diag}(X^1, \cdots, X^M)$, where X^m is the samples from the m modality. Thus, the derivative of $f(x)$ is given by: $\frac{\partial f(X)}{\partial w} = \mathbf{diag}(X^1, X^2, \cdots, X^M)$, and the Hetero-manifold regularization is defined as: $f(X)Lf(X)^T$ where L is a Laplacian exploited on the Hetero-manifold.

[1]The 0 reduction occurs when f_k in last round is still the optimum of sub-problem in current round.

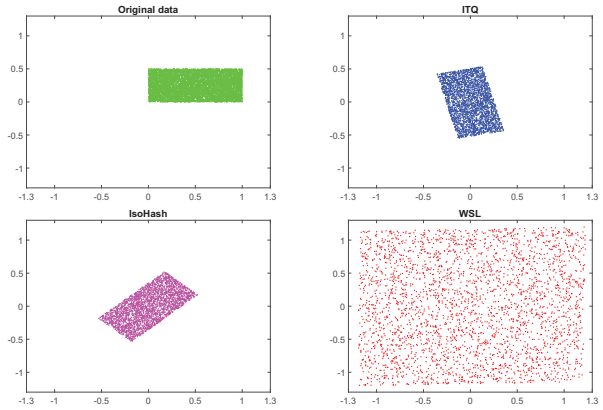

Figure 2: 2D toy examples.

Table 1: Various loss comparisons on a toy set between ITQ, IsoHash and W-Shape. Four types of quantization losses including $|f(x) - \mathbf{sign}(f(x))|$ (Abs Loss), $(f(x) - \mathbf{sign}(f(x)))^2$, $max\{1 - \mathbf{sign}(f(x))f(x), 0\}$ and $(f(x)^2 - 1)^2$ are used to describe the loss of binarization.

Methods	ITQ	IsoHash	WSL
Abs Loss	1.623	1.613	0.858
Squire Loss	1.350	1.332	0.553
Hinge Loss	1.623	1.613	0.816
W-Shape Loss	1.802	1.801	0.903
Average	1.599	1.59	0.783

Experiment

To validate the proposed framework, we compare it with the state-of-the-art methods in five datasets: toy set, SIFT1M (Jegou, Douze, and Schmid 2011) CIFAR-10 (Krizhevsky and Hinton 2009), MNIST[2] and VIPeR (Gray and Tao 2008). The parameters of the proposed model are set as $\mu = 0.05$ and $\nu = 0.6$.

Toy set

Our first experiment works on a toy dataset, in which the points are randomly generated in the area $[0, 1] \times [0, 0.5]$. It is worth to point out that the variance in two directions are different. The purpose of hash function learning is to project the points close to -1 or 1. Two classical methods concentrating on minimizing quantization loss: ITQ (Gong et al. 2013b) and IsoHash (Kong and Li 2012) are compared.

Fig. 2 illustrates the learned real-valued features before binarization. From Table 1 and Fig. 2, we can see that, no matter what types of criterion are used, the quantization loss of the proposed method is almost only half of other two methods. Because the structure of the original data is simple, all the three methods can preserve the basic shape of dataset. However, only the proposed method can project the data closer to 1 or -1 and the other two methods focus more on balancing the variances between two directions.

[2]http://yann.lecun.com/exdb/mnist/.

We also observed that ITQ and IsoHash are significantly influenced by the range of data but WLS is not. The inherent reason is that the linear models $y = f(x) = w^T x$ in two methods are not well regularized. Both methods focus on the balance of variances (holistic view) and thus achieve very similar quantization loss. Then, the magnitude of the projected values for certain coordinates maybe widen to satisfy the variance balance. However, in the quantization step, the intrinsic structure (local) is very likely to be ruined, due to the very large quantization loss, which will result in bad binary representation.

Uni-modal Hashing

The second group of datasets used to validate the performance of the proposed method includes SIFT1M (Jegou, Douze, and Schmid 2011) CIFAR-10 (Krizhevsky and Hinton 2009) and MNIST. SIFT1M consists of 1M 128-dimensional SIFT vectors as the reference set, 100K vectors as the learning set, and 10K vectors as the query set. The recall@R defined as the fraction of the relevant samples in the retrieved R items to the ground truth neighbours are used to measure the retrieval performances of different methods. The experimental setting for CIFAR-10 and MNIST, in which label information is provided, is the same as that in (Liong et al. 2015). The mean average precision of the semantic retrieval on the two datasets is reported.

To make clear that the improvement is exactly raised by WSL, we use the purely simplest setting (linear and unsupervised) for WSL. We compare WSL method with MLH (Norouzi and Fleet 2011), LSH (Indyk and Motwani 1998), ITQ (Gong et al. 2013b), SH (Weiss, Torralba, and Fergus 2009), SpH (Heo et al. 2015), PCAH (Wang, Kumar, and Chang 2012) and AGH (Liu et al. 2011).

From the Fig. 3 (a), we can see that the proposed WSL can achieve better results than MLH, LSH, ITQ and SH. ITQ is a pioneer method which reduces the square based quantization loss and an alternative optimization strategy is used. Same to SH, WSL uses the basic Laplacian matrix in the measurement item but WSL outperforms SH because the quantization loss is minimized as well. Moreover, WSL achieves better results than AGH. Actually, linear functions are used in WSL but in AGH, nonlinear eigenfunctions are used. Thus the computation burden of the retrieval stage in WSL is much lower than that in AGH. From the Tables 2 and 3, WSL achieves advanced performance for the semantic retrieval on datasets CIFAR-10 and MNIST.

In Fig. 4, the embedded values *w.r.t.* the parameter μ are investigated. On the one hand, when μ is too small, the norm of projection would be small as well and then, all samples will be projected close to 0. In fact, W-shape loss plays the same role as the norm regularization item $||F||_2$. On the other hand, surprisingly, large value of μ also results in poor performance. We observed in our experiment that the initial of projection is very important and would converge into a poor local optimum. In general, the initial could be selected from the ones which could minimize the measurement item firstly. The optimizing procedure of loss is illustrated in Fig. 3 (b). We can see that both the quantization loss and the measurement distortion decrease consistently and the mea-

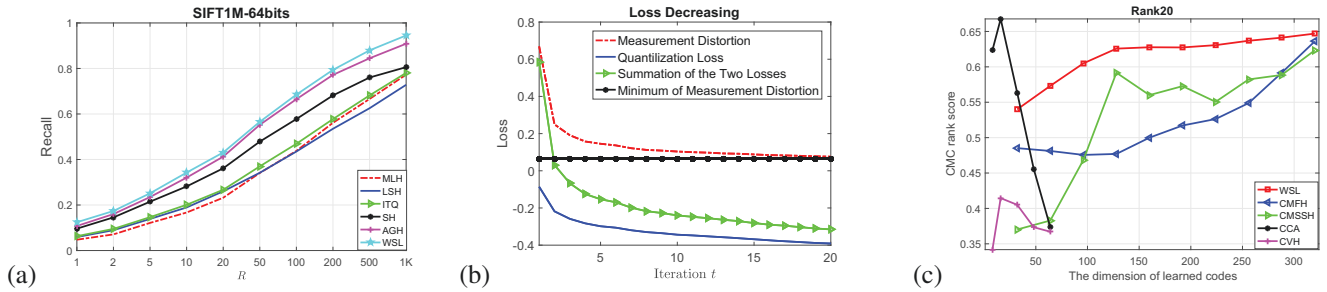

Figure 3: (a) The performance comparison of uni-modal retrieval on dataset SIFT1M. (b) The investigation of loss decreasing with respect to the iteration t. (c) The performance comparison of cross-modal retrieval on dataset VIPeR.

Table 2: Ranking performance on CIFAR-10. Mean average precision (MAP) at different number of bits is calculated.

#bits	LSH	ITQ	PCAH	SH	SpH	WSL
16	0.140	0.167	0.135	0.136	0.148	0.183
32	0.141	0.174	0.130	0.130	0.154	0.210
64	0.127	0.179	0.124	0.132	0.159	0.227

surement distortion may approximate to the possible lowest value (Black line) which is computed in the corresponding continuous model.

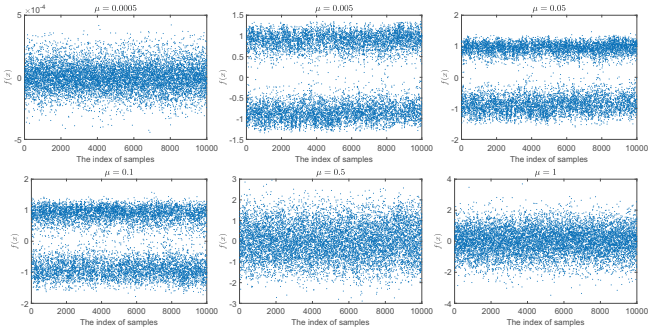

Figure 4: The embedded points $f(x)$ of 10000 query samples in the dataset SIFT1M with the varied parameter $\mu = 0.0005, 0.005, 0.05, 0.1, 0.5, 1$.

Cross-modal Hashing

We also test the proposed method in a cross-modal setting for fast cross-camera person re-identification. Two cameras setting in different places of a campus environment are used to collect the samples. The task of cross-camera person re-identification is to, given a image of a person in the view of one camera, search the images of same person in the view of other cameras. The VIPeR (Gray and Tao 2008) contains 632 pedestrian image pairs in an outdoor environment. Half of the dataset including 316 images for each view is used for training the algorithms and the reminding (316 pedestrian) is used for testing. For a semantic identification systems, the Cumulated Matching Characteristics (CMC) are used for performance evaluation and measuring how well an identification system ranks the identities in the gallery with respect to a probe sample (Rank 20).

Table 3: Ranking performance (MAP) on MNIST.

#bits	LSH	ITQ	PCAH	SH	SpH	WSL
16	0.224	0.410	0.276	0.272	0.268	0.434
32	0.246	0.434	0.245	0.259	0.323	0.479
64	0.320	0.456	0.212	0.251	0.356	0.522

Three cross-modal hashing methods including CMFH (Ding, Guo, and Zhou 2014), CMSSH (Bronstein and Bronstein 2010), CVH (Kumar and Udupa 2011) and one non-hashing methods CCA (Hotelling 1936) are compared. From Fig. 3 (c), we can see that the proposed WSL method can consistently outperform the three hashing methods. It is worth to point out that, due to the dimension limitation of covariance, CCA and CVH can learn a few bits. The best performances of WSL and CCA are close but CCA is a continuous method without any quantization loss.

Conclusion

In this paper, a novel general framework is raised for hashing which could be used in both uni-modal and cross-modal settings. We test the proposed method on five datasets and the results demonstrate that the W-Shape Loss (WSL) actually benefits to simultaneously reduce the measurement distortion and the quantization loss. In the future, it is valuable to propose more types of W-shape loss. Moreover, the theoretical aspect of the WSL needs to be investigated as well. More importantly, the proposed W-shape loss would be used to guide the binary weights learning for deep architecture machine, which further speeds up the step of on-line testing.

Acknowledgement

This work was partially supported by the following grants: NSF-IIS 1302675, NSF-IIS 1344152, NSF-DBI 1356628, NSF-IIS 1619308, NSF-IIS 1633753, NIH R01 AG049371.

References

Bentler, P., and Lee, S.-Y. 1978. Matrix derivatives with chain rule and rules for simple, hadamard, and kronecker products. *Journal of Mathematical Psychology* 17(3).

Bottou, L.; Curtisy, F. E.; and Nocedal, J. 2016. Optimization methods for large-scale machine learning. *arXiv:1606.04838v1*.

Bronstein, M. M., and Bronstein, A. M. 2010. Data fusion through cross-modality metric learning using similarity-sensitive hashing. In *CVPR*.

Carreira-Perpinan, M. A., and Raziperchikolaei, R. 2016. An ensemble diversity approach to supervised binary hashing. In *NIPS*.

Chen, Z.; You, X.; Zhong, B.; Li, J.; and Tao, D. 2017. Dynamically modulated mask sparse tracking. *IEEE Transactions on Cybernetics* 47(11).

Ding, G.; Guo, Y.; and Zhou, J. 2014. Collective matrix factorization hashing for multimodal data. In *CVPR*.

Gong, Y.; Kumar, S.; Verma, V.; and Lazebnik, S. 2012. Angular quantization-based binary codes for fast similarity search. In *NIPS*.

Gong, Y.; Kumar, S.; Rowley, H. A.; and Lazebnik, S. 2013a. Learning binary codes for high-dimensional data using bilinear projections. In *CVPR*.

Gong, Y.; Lazebnik, S.; Gordo, A.; and Perronnin, F. 2013b. Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval. *IEEE TPAMI* 35(12).

Gray, D., and Tao, H. 2008. Viewpoint invariant pedestrian recognition with an ensemble of localized features. In *Proc. ECCV*.

Guo, Y.; Ding, G.; Liu, L.; Han, J.; and Shao, L. 2017. Learning to hash with optimized anchor embedding for scalable retrieval. *IEEE TIP* 26(3).

Guo, Y.; Ding, G.; and Han, J. 2017. Robust quantization for general similarity search. *IEEE TIP*.

Heo, J.-P.; ; Lee, Y.; He, J.; Chang, S.-F.; and Yoon, S.-E. 2015. Spherical hashing: Binary code embedding with hyperspheres. *IEEE TPAMI* 37(11).

Hotelling, H. 1936. Relations between two sets of variates. *Biometrika* 28:321–377.

Indyk, P., and Motwani, R. 1998. Approximate nearest neighbors: towards removing the curse of dimensionality. In *STOC*.

Irie, G.; Li, Z.; Wu, X.-M.; and Chang, S.-F. 2014. Locally linear hashing for extracting non-linear manifolds. In *CVPR*.

Jegou, H.; Douze, M.; and Schmid, C. 2011. Product quantization for nearest neighbor search. *IEEE TPAMI* 33(1).

Johnson, S. G. 2008. The nlopt nonlinear-optimization package. In *http://ab-initio.mit.edu/nlopt*. MIT.

Kong, W., and Li, W.-J. 2012. Isotropic hashing. In *NIPS*.

Krizhevsky, A., and Hinton, G. 2009. Learning multiple layers of features from tiny images. Technical report, Technical report, University of Toronto.

Kumar, S., and Udupa, R. 2011. Learning hash functions for cross-view similarity search. In *IJCAI*.

Lin, G.; Shen, C.; Suter, D.; and van den Hengel, A. 2013. A general two-step approach to learning-based hashing. In *ICCV*.

Liong, V. E.; Lu, J.; Wang, G.; Moulin, P.; and Zhou, J. 2015. Deep hashing for compact binary codes learning. In *CVPR*.

Liu, W.; Wang, J.; Kumar, S.; and Chang, S.-F. 2011. Hashing with graphs. In *ICML*.

Liu, X.; He, J.; Deng, C.; and Lang, B. 2014. Collaborative hashing. In *CVPR*.

Liu, X.; Du, B.; Deng, C.; Liu, M.; and Lang, B. 2016. Structure sensitive hashing with adaptive product quantization. *IEEE Transactions on Cybernetics* 46(10).

Liu, X.; Li, Z.; Deng, C.; and Tao, D. 2017. Distributed adaptive binary quantization for fast nearest neighbor search. *IEEE TIP* 26(11).

Mu, Y.; Shen, J.; and Yan, S. 2010. Weakly-supervised hashing in kernel space. In *CVPR*.

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

Rastegari, M.; Choi, J.; Fakhraei, S.; III, H. D.; and Davis, L. S. 2013. Predictable dual-view hashing. In *ICML*.

Shen, F.; Shen, C.; Liu, W.; and Shen, H. T. 2015. Supervised discrete hashing. In *CVPR*.

Svanberg, K. 2002. A class of globally convergent optimization methods based on conservative convex separable approximations. *SIAM Journal on Optimization*.

Wang, J.; Zhang, T.; Song, J.; Sebe, N.; and Shen, H. T. 2016. A survey on learning to hash. *arXiv:1606.00185*.

Wang, J.; Kumar, S.; and Chang, S.-F. 2012. Semi-supervised hashing for large-scale search. *IEEE TPAMI* 34(12).

Weiss, Y.; Torralba; and Fergus, R. 2009. Spectral hashing. In *NIPS*.

Yan, S.; Xu, D.; Zhang, B.; jiang Zhang, H.; Yang, Q.; and Lin, S. 2007. Graph embedding and extensions: A general framework for dimensionality reduction. *IEEE TPAMI* 29(1).

You, X.; Peng, Q.; Yuan, Y.; ming Cheung, Y.; and Lei, J. 2011. Segmentation of retinal blood vessels using the radial projection and semi-supervised approach. *Pattern Recognition*.

You, X.; Guo, W.; Yu, S.; Prncipe, K. L. J. C.; and Tao, D. 2016. Kernel learning for dynamic texture synthesis. *IEEE TIP* 25(10).

Zhang, D.; Wang, J.; and Lu, D. C. J. 2010. Self-taught hashing for fast similarity search. In *SIGIR*.

Zhao, K.; Lu, H.; and Mei, J. 2014. Locality preserving hashing. In *AAAI*.

Zheng, F., and Shao, L. 2016. Learning cross-view binary identities for fast person re-identification. In *IJCAI*.

Zheng, F.; Tang, Y.; and Shao, L. 2016. Hetero-manifold regularisation for cross-modal hashing. *IEEE TPAMI* 99.