Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16)

Towards Optimal Binary Code Learning via Ordinal Embedding

Hong Liu[†‡], Rongrong Ji[†‡], Yongjian Wu[♮], Wei Liu[♯]
[†]Fujian Key Laboratory of Sensing and Computing for Smart City, Xiamen University, 361005, China
[‡]School of Information Science and Engineering, Xiamen University, 361005, China
[♮]BestImage, Tencent Technology (Shanghai) Co.,Ltd, China
[♯]Department of Electrical Engineering, Columbia University, New York, NY 10027, USA
lynnliuxmu@outlook.com, rrji@xmu.edu.cn, littlekenwu@tencent.com, wliu@ee.columbia.edu

Abstract

Binary code learning, *a.k.a.*, hashing, has been recently popular due to its high efficiency in large-scale similarity search and recognition. It typically maps high-dimensional data points to binary codes, where data similarity can be efficiently computed via rapid Hamming distance. Most existing unsupervised hashing schemes pursue binary codes by reducing the quantization error from an original real-valued data space to a resulting Hamming space. On the other hand, most existing supervised hashing schemes constrain binary code learning to correlate with pairwise similarity labels. However, few methods consider *ordinal relations* in the binary code learning process, which serve as a very significant cue to learn the optimal binary codes for similarity search. In this paper, we propose a novel hashing scheme, dubbed Ordinal Embedding Hashing (OEH), which embeds given ordinal relations among data points to learn the ranking-preserving binary codes. The core idea is to construct a directed unweighted graph to capture the ordinal relations, and then train the hash functions using this ordinal graph to preserve the permutation relations in the Hamming space. To learn such hash functions effectively, we further relax the discrete constraints and design a stochastic gradient decent algorithm to obtain the optimal solution. Experimental results on two large-scale benchmark datasets demonstrate that the proposed OEH method can achieve superior performance over the state-of-the-arts approaches. At last, the evaluation on query by humming dataset demonstrates the OEH also has good performance for music retrieval by using user's humming or singing.

Introduction

Large-scale visual search has attracted extensive research focus recently in computer vision and artificial intelligence communities (Dean et al. 2013; Grauman and Fergus 2013; Li et al. 2011). One fundamental challenge lies in its heavy computational cost, *i.e.* linearly comparing massive real-valued features to find the nearest neighbours of the query. Hashing techniques have been recently popular to tackle such challenge, which encodes the real-valued data points into binary codes with significant efficiency in storage and computation. In principle, most existing hashing methods learn a set of hash functions $\{h^k : R^d \mapsto \{0,1\}\}_{k=1}^r$, which map the original data from a d-dimensional real-valued feature space to a r-bits Hamming space, such that the distance among original data can be approximated by Hamming distance efficiently.

One issue in binary code learning is how to preserve the similarity among data points in the high-dimensional real-valued space. To this end, the existing binary code learning schemes can be classified into either data-independent or data-dependent ones. Data-independent hashing like Locality Sensitive Hashing (LSH) (Gionis et al. 1999) typically adopts random projection to find a set of random sign functions to produce binary codes. For instance, Shift-Invariant Kernel Hashing (Raginsky and Lazebnik 2009) learns binary code by using random Fourier features with shift-invariant kernel transformation. For another instance, Kernelized Locality Sensitive Hashing (Kulis and Grauman 2012) extends the general LSH to Kernelized space to support arbitrary similarity. However, the above data-independent hashing methods always require long bits to achieve satisfactory search accuracy.

Data-dependent hashing can be categorized into either supervised or unsupervised ones, Unsupervised hashing, such as Spectral Hashing (Weiss, Torralba, and Fergus 2009), Anchor Graph Hashing (Liu et al. 2011), Iterative Quantization (Gong et al. 2013), Spherical Hashing (Heo et al. 2012), Discrete Graph Hashing (Liu et al. 2014), and Scalable Graph Hashing (Jiang and Li 2015), models the data structure or distribution as constraints to achieve high search accuracy with short binary codes. Differently, supervised hashing preserves the label relation (*e.g.*, pairwise similarity or dissimilarity) in Hamming space to learn semantic-aware binary codes, for instance Binary Reconstructive Embedding (Kulis and Darrell 2009), Minimal Loss Hashing (Norouzi and Blei 2011), Kernel-based Supervised Hashing (Liu et al. 2012), Hamming Distance Metric Learning (Norouzi, Blei, and Salakhutdinov 2012), and Supervised Discrete Hashing (Shen et al. 2015). Although supervised hashing typically provides superior performance, it is typically labour-intensive to obtain sufficient semantic labels in real application.

In this paper, we focus on unsupervised hashing. To learn accurate binary codes, the key design lies in preserving the similarity among data points in the high-dimensional real-valued feature space. To this end, existing unsupervised

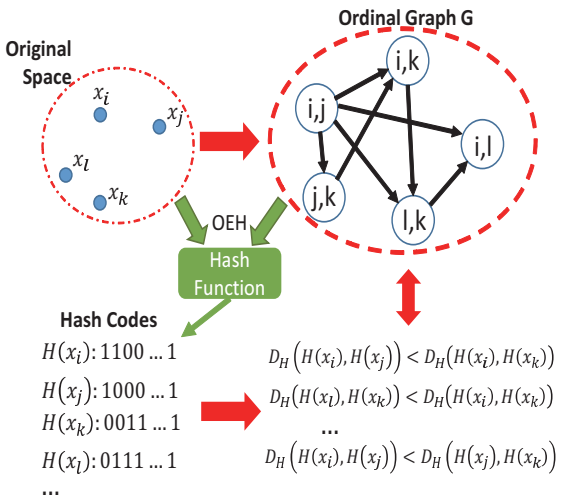


Figure 1: Framework of the proposed OEH method.

methods only consider the preservation of pairwise similarity between data points. Taking graph hashing (*e.g.* Spectral Hashing (Weiss, Torralba, and Fergus 2009), Anchor Graph Hashing (Liu et al. 2011), Discrete Graph Hashing (Liu et al. 2014) and Scalable Graph Hashing (Jiang and Li 2015)) for instance, pairwise similar/dissimilar constraints are embedded into neighbor graph construction and decomposition, which typically result in high complexity, and cannot capture the relative orders among data points. To the best of our knowledge, none existing hashing methods can consider the *ordinal relation* among data points, which is however cheaply available and can be obtained in an unsupervised manner.

In this paper, we propose a novel hashing method, termed Ordinal Embedding Hashing (OEH), which makes the first attempt towards preserving the ordinal relation from the high-dimensional real-valued feature space to the Hamming space. Our work is inspired by the work in (McFee and Lanckriet 2011; Terada and Luxburg 2014) which were proposed for embedding arbitrary ordered structures into the Euclidean space. Different from the existing pairwise similarity constraints, we argue that, it is the relative order between data pairs, rather than the absolute distance, which must be preserved in the Hamming space. To that effect, our approach constructs a directed, unweighted ordinal graph to capture the original relation among data points, upon which we learn hash functions via a novel stochastic gradient decreasing (SGD) algorithm. The whole framework of our proposed OEH approach is shown in Figure 1. We compare the proposed OEH method against various state-of-the-art unsupervised hashing methods on two widely-used image retrieval benchmarks, *i.e.*, **CIFAR10** and **LabelMe**. Quantitative results demonstrate that OEH outperforms the existing unsupervised hashing methods with a significant margin. At last, we propose a two-step framework for the system of query by humming, and the evaluation on **MIR-QBSH** demonstrate the efficiency of our scheme that can quickly retrieve similar music by users humming or singing. By working with the BestImage team (bestimage.qq.com/) in Tencent, the proposed scheme has been integrated into Tencent QQ Music product (see details in the experiments).

The rest of this paper is organized as follows: In Section 2, we introduce the preliminary formulation of ordinal embedding. The proposed OEH and the iterative SGD based optimization are introduced in Section 3. In Section 4, we show the experimental results and analysis. Finally, we conclude this paper in Section 5.

Preliminaries

In this section, ordinal embedding and isotonic functions are introduced, which serve as the basis for the proposed OEH method. Ordinal embedding, also known as ordinal scaling, non-metric multidimensional scaling, or isotonic embedding (Arias-Castro 2015), targets at finding an embedding of items by comparing their pairwise distances. Let us denote a set of n items each with dimension d_1, and also denote $\delta_{ij} \geq 0$ as the dissimilarity between the i-th and the j-th item. Suppose that $\delta_{ii} = 0$ and $\delta_{ij} = \delta_{ji}$, an order relation subset C can be written as follows

$$\delta_{ij} < \delta_{kl}, \forall (i,j,k,l) \in C. \tag{1}$$

Given C and a dimension d_2 for the new space, the goal of ordinal embedding is to embed items in C as new feature representation $x_1, ..., x_n \in \mathbb{R}^{d_2}$ such that the ordinal constraints are preserved as

$$\delta_{ij} < \delta_{kl} \Rightarrow \|x_i - x_j\| \leq \|x_k - x_l\|, \forall (i,j,k,l) \in C, \tag{2}$$

where $\|\cdot\|$ denotes the Euclidean norm. Eq. (2) is referred to the Euclidean embedding[1]. There are two common situations, *i.e.*, quadruple comparisons where $(i,j,k,l) \in C = [n]^4$, and triplet comparisons where $(i,j,i,l) \in C = [n]^3$. Both settings have widely been studied in the machine learning community. (Shepard 1962; Young 2013; von Luxburg and others 2014)

We further denote $\Omega \subseteq \mathbb{R}^d$ and a transformation function $f : \Omega \to \mathbb{R}^d$, where for all $\{x, y\} \in \Omega$ and some $\lambda > 0$. Thus we have $\|f(x) - f(y)\| = \lambda \|x - y\|$, where f is isotonic if for all $\{x, y, z, w\} \in \Omega$, the following constraints hold as:

$$\|x - y\| < \|z - w\| \Rightarrow \|f(x) - f(y)\| < \|f(z) - f(w)\|. \tag{3}$$

Equivalently, $f : \mathbb{R}^d \to \mathbb{R}^d$ is a linear transformation, given by $f(x) = Wx + b$ with orthogonal matrix W and an offset $b \in \mathbb{R}^d$. f is weakly isotonic if Eq. (3) holds only for $\{x, y, z, w\} \in \Omega$ with $x = z$. And f is locally isotonic for each points $x \in \Omega$ with its neighborhood set U.

Ordinal Embedding Hashing

We first introduce some notations before describing the proposed hashing algorithm. Let $X = \{x_1, x_2, ..., x_n\} \in \mathbb{R}^{d \times n}$ be the data matrix with n samples, where x_i is the i-th column of X with d dimensions. As mentioned, an order relation subset C is used as the ordinal constraints to present

[1] In Section 3, we further extend such Euclidean embedding into a binary case for binary code learning.

the relative dissimilarity degree between δ_{ij} and δ_{kl}. According to C, we construct a directed unweighted graph $G = (V, E) = [n^4]$, where each node $v_{ij} \subseteq V$ represents the dissimilarity degree between the i-th and the j-th sample, and each directed edge $e_{(i,j,k,l)\in C} = (v_{ij} \rightarrow v_{kl}) \subseteq E$ represents $\delta_{ij} < \delta_{kl}$.

OEH aims at learning a set of mapping functions $H(x) = \big\{h_1(x), h_2(x), ..., h_r(x)\big\}$, which map the real-valued sample points to the corresponding binary codes $B = \{b_1, b_2, ..., b_n\} \in \{0, 1\}^{r\times n}$, where r is the length of binary codes. Specifically, our goal is to ensure the learned hash function to hold the following constraint

$$\forall(v_{ij}, v_{kl}) \in G : \|H(x_i) - H(x_j)\|_1 < \|H(x_k) - H(x_l)\|_1 . \tag{4}$$

More specifically, given G and the code length r, the hash function should preserve the ordinal relations as much as possible. By using a linear transform function with orthogonal mapping matrix, we propose to formulate the following hashing function

$$h_k = sgn\big(f_k(\hat{x})\big), k = 1, 2, ..., r, \tag{5}$$

where $sgn(\cdot)$ is the sign function, which returns 1 if $f_k(\cdot) > 0$ and -1 otherwise, $f_k(\hat{x}) : \mathbb{R}^r \rightarrow \mathbb{R}^r$ is a linear transformation as described in Section 2, and $\hat{x} \in \mathbb{R}^r$ is the non-linear principle component of the original feature, calculated via kernel transformation (Liu et al. 2012; Shen et al. 2015) and PCA. For simplicity, we define the whole hashing function as $H(\hat{x}) = sgn(W^T\hat{x})$, with the orthogonal matrix $W = [w_1, w_2, ..., w_r] \in \mathbb{R}^{r\times r}$.

To embed the ordinal relations into $H(\hat{x})$, we learn hash codes with the ordinal graph G instead of the pairwise distance. For edge $e_{(i,j,k,l)\in C} \subseteq E$ in G, we expect the relation of dissimilar degrees can be preserved by hash function $H(\hat{x})$ under the constraint of Eq. (4). In other words, the Hamming distance between b_i and b_j should be smaller than that between b_k and b_l. We therefore write the objective function for our proposed OEH method as

$$\begin{aligned} &min \sum_{(v_{ij},v_{kl})\in G} I\big(D_H(b_i, b_j) \geq D_H(b_k, b_l)\big) \\ &s.t.\ b_i = sgn(W^T\hat{x_i}), \\ &\quad W^TW = I, \end{aligned} \tag{6}$$

where $I(\cdot)$ is an indicator function which returns 1 if the condition is satisfied and 0 otherwise. Function $D_H(b_i, b_j)$ represents the Hamming distance between binary code b_i and b_j. This objective function aims at counting up the number of incorrect order relations, which indicates the Hamming distance of $pair(b_i, b_j)$ is lager than that of $pair(b_k, b_l)$. Note that the objective function in Eq. (6) is discrete and hard to optimize. We tackle this issue by an alternative function, which is equivalent to Eq. (7) with a scale parameter $\beta > 0$, *i.e.*,

$$min \sum_{(v_{ij},v_{kl})\in G} o(v_{ij}, v_{kl}) max\big[0, D_H(b_i, b_j) + \beta - D_H(b_k, b_l)\big], \tag{7}$$

where function $o(v_{ij}, v_{kl}) = 1$ if there is an directed edge from vertex v_{ij} to v_{kl}, and $o(v_{ij}, v_{kl}) = 0$ otherwise.

Although the order relation can be generated easily, the total number of order constraints is too large to be used for training. To solve this problem, inspired by (Terada and Luxburg 2014), we relax the above *global* ordinal constraints to *local* ordinal constraint, which significantly reduces the number of order constraints in ordinal graph without decreasing the embedding quality. We adopt a landmark-based ordinal embedding (Arias-Castro 2015) which considers the comparison from any point to the landmarks. Under such a circumstance, the number of ordinal constraints reduces from n^4 to $n \cdot L^2$, where L is the number of landmarks. Generally, with triple constraints instead of quadruple, the landmark-based ordinal embedding can be written as the following constraints:

$$(i, j, k) \in C : \tilde{\delta}_{ij} < \tilde{\delta}_{ik} \Rightarrow \|b_i - bl_j\|_1 < \|b_i - bl_k\|_1 , \tag{8}$$

where bl_j is the corresponding binary code of landmark $l_j \in \mathbb{R}^r$. With this constraint, Eq. (7) can be rewritten as

$$min \sum_{i=1}^{n} \sum_{j,k=1}^{L} o(v_{ij}, v_{ik}) max\big[0, D_H(b_i, bl_j) + \beta - D_H(b_i, bl_k)\big]. \tag{9}$$

To describe function $o(\cdot, \cdot)$ more conveniently, we formulate the neighbor information in the form of k-nearest neighbor graph $A^1 \in \mathbb{R}^{n\times L}$. Neighbor graph A^1 measures the directed unweighted constraints, where $a^1_{ij} = 1$ means sample x_i is connected with its k-nearest neighbor l_j, and $a^1_{ij} = 0$ otherwise. For a given directed unweighted graph A^1, the function can be rewritten as $o(i, j, k) = a^1_{ij}(1 - a^1_{ik})$, which reflect the ordinal constraints $\delta_{i,j} < \delta_{i,k}$. To further integrate the ordinal constraint among landmarks, for a given sample x_i, we select the m-nearest neighbor landmarks as a subset M and construct a new neighbor graph A^2 as before. Based on A^2, we introduce a new objective function for landmark-based ordinal preserving as follows:

$$\begin{aligned} min \sum_{m=1}^{M} \sum_{j,k\neq m}^{L} \big\{ &o(v_{mj}, v_{mk}) \cdot \\ &max[0, D_H(l_m, l_j) + \beta - D_H(l_m, l_k)]\big\}. \end{aligned} \tag{10}$$

Correspondingly, we rewrite the overall model for the proposed OEH approach as follows:

$$\begin{aligned} F(W|G) = \sum_{i=1}^{n} \big\{ &\sum_{j,k=1}^{L} o(i, j, k) max[0, D_H(b_i, bl_j) \\ &+ \beta - D_H(b_i, bl_k)] + \alpha \sum_{m=1}^{M} \sum_{j,k\neq m}^{L} o(m, j, k) \cdot \\ &max[0, D_H(bl_m, bl_j) + \beta - D_H(bl_m, bl_k)]\big\} \\ &s.t.\ b_i = H(\hat{x}_i), bl_j = H(l_j), \\ &\quad W^TW = I. \end{aligned} \tag{11}$$

where α is the tradeoff among the ordinal information of sample points and landmarks. Note that different from Euclidean based ordinal embedding (McFee and Lanckriet 2011; Terada and Luxburg 2014), our work measures the ordinal relation as Hamming distance which is discontinuous. Meanwhile our target is to learn the hash functions that preserve the ordinal information in the compact Hamming space. As a result, a new optimization scheme should be designed, which is introduced subsequently in Section 3.

Optimization

Problem Relaxation

Directly minimizing the objective function in Eq. (11) is intractable, as the coding function is discrete while Hamming space is not continuous. To solve this problem, we relax the discrete constraints from the Hamming space to an approximated continuous space.

To that effect, we first relax the hashing function $H(\hat{x}_i) = sgn(W^T\hat{x}_i)$ as follows:

$$\bar{H}(\hat{x}_i) = tanh(W^T\hat{x}_i), \tag{12}$$

where $tanh(\cdot)$ is a good approximation for $sign(\cdot)$, which transforms the binary codes from $\{0, 1\}$ to $\{-1, 1\}$. Correspondingly the Hamming distance could be calculated as:

$$D_H(b_i, b_j) = \frac{1}{2}(r - \bar{H}^T(\hat{x}_i) \cdot \bar{H}(\hat{x}_j)). \tag{13}$$

Finally, we introduce another function $p(\cdot, \cdot, \cdot)$ to relax the maximum function that is not easy to optimize. Function $p(b_i, l_j, l_k)$ is defined as:

$$p(b_i, bl_j, bl_k) = \frac{1}{1 + exp(D_H(b_i, bl_k) - \beta - D_H(b_i, bl_j))}.$$

Based upon the above relaxations, the objective function in Eq. (10) can be rewritten as:

$$\begin{aligned}\bar{F}(W|G) = &\sum_{i=1}^{n}\{\sum_{j,k}^{L} o(i,j,k) \cdot p(b_i, bl_j, bl_k) \\ &+ \alpha \sum_{m=1}^{M}\sum_{j,k\neq m}^{L} o(m,j,k) \cdot p(bl_m, bl_j, bl_k)\} \\ &+ \gamma \left\| W^T W - I \right\|.\end{aligned} \tag{14}$$

Intuitively, the gradient descent approach can be used to carry out an iterative optimization for Eq. (14). However, due to the large number of ordinal constraints, the efficiency of such a direct gradient descend is low. In the following, we further introduce an alternative stochastic gradient descent algorithm to solve this problem efficiently.

Stochastic Gradient Descent

We first conduct PCA for all samples and landmarks before optimization. Then, a stochastic gradient descent is done to learn parameters in OEH: Given a data point $\hat{x}_i$ uniformly sampled from the training set, a set of landmarks and a relation graph A^2 representing the order information between landmarks are built, based on which we generate a unweighted KNN graph A^1 between the sample and the landmarks.

We then build the function $p(i, j, k)$, by which the Hamming distance between landmarks and samples can be easily calculated by the projection matrix W (W is generated during the last iteration). In this way, the gradient of Eq. (14) is given by:

$$\begin{aligned}&\frac{\partial \bar{F}(W|A^1, A^2, \hat{x}_i)}{\partial W} = \\ &\gamma W + \sum_{j,k}^{L} o(i,j,k) \cdot \big(p(b_i, bl_j, bl_k)(1 - p(b_i, bl_j, bl_k))\big) \cdot \\ &\Big[\frac{\partial D_H(b_i, bl_k)}{\partial W} - \frac{\partial D_H(b_i, bl_j)}{\partial W}\Big] + \\ &\alpha \sum_{m=1}^{M}\sum_{j,k\neq m}^{L} \{o(m,j,k) \cdot \big(p(bl_m, bl_j, bl_k)(1 - p(bl_m, bl_j, bl_k))\big) \cdot \\ &\Big[\frac{\partial D_H(bl_m, bl_k)}{\partial W} - \frac{\partial D_H(bl_m, bl_j)}{\partial W}\Big]\},\end{aligned} \tag{15}$$

where the gradient of Hamming distance is formulated as:

$$\begin{aligned}\frac{\partial D_H(b_i, bl_j)}{\partial W} = &-\frac{1}{2}\big\{\hat{x}_i \cdot \big[\big(1\text{-}\bar{H}^2(\hat{x}_i)\big) \odot \bar{H}(l_j)\big]^T \\ &+ l_j \cdot \big[\big(1\text{-}\bar{H}^2(l_j)\big) \odot \bar{H}(\hat{x}_i)\big]^T\big\}.\end{aligned} \tag{16}$$

In Eq. (16), $\odot$ is the Hadamard product which represents the element-wise product. Based on the gradient, we perform the update procedure for matrix W as:

$$W_{t+1} = W_t - \eta\frac{\partial \bar{F}(W|A^1, A^2, \hat{x}_i)}{\partial W}, \tag{17}$$

where η is the parameter of the learning rate and t is the number of iterations. The details of the proposed SGD for OEH in shown in Algorithm 1. The complexity of the proposed gradient updating is $O\big((m+1)r^3L^2\big)$

Algorithm 1 Ordinal Embedding Hashing (OEH)

Input: Data samples $X = \{\hat{x}_1, \hat{x}_2, ..., \hat{x}_n\}$, landmarks $\{l_1, l_2, ..., l_L\}$, and parameters α, β, γ and η.
Output: The projection matrix $W \in R^{r\times r}$.

```
repeat
    randomly pick up an sample point x̂_i;
    generate the KNN graph A^1 based on x̂_i and landmarks;
    generate the landmark ordinal graph A^2;
    calculate the gradient according to Eq. (15);
    make a gradient descent according to Eq. (17);
until convergence or reaching the maximum iteration number.
```

Experiments

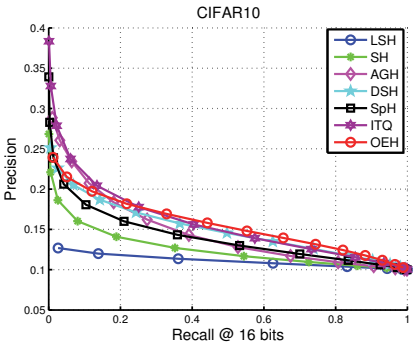
In this section, we evaluate the proposed OEH approach on two widely used benchmark datasets: **CIFAR10** (Krizhevsky 2009), and **LabelMe** (Torralba, Fergus, and Weiss 2008). We also evaluate the proposed OEH method on **MIR-QBSH** dataset with the application of query by humming, at the last sub-section.

Datasets

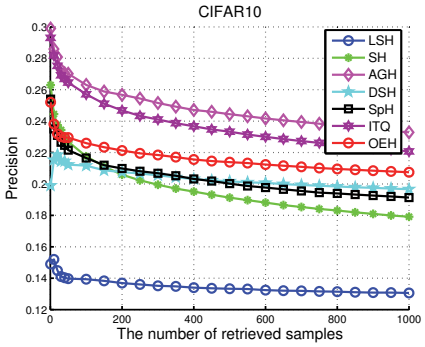
The **CIFAR10** dataset consists of 60,000 color images in 10 classes (6,000 images per class) each with 32×32 spatial resolution. Each image is represented by a 512-dimensional GIST feature (Oliva and Torralba 2001). 1,000 images are randomly selected as the test set, and the remaining are used

Table 1: *m*AP comparision using Hamming ranking on both datasets with different hash bits.

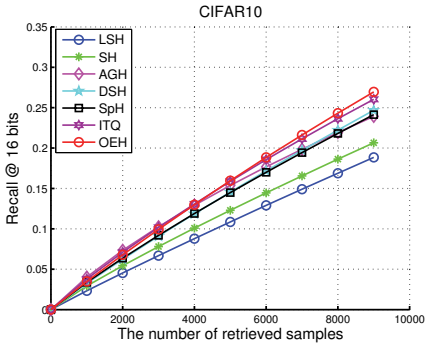
	CIFAR10				LabelMe			
Methods	8	16	32	64	8	16	32	64
LSH	0.1170	0.1122	0.1243	0.1271	0.1412	0.2473	0.2444	0.3112
SH	0.1243	0.1287	0.1276	0.1259	0.2044	0.2437	0.2779	0.3802
AGH	0.1507	0.1575	0.1483	0.1440	0.2381	0.2603	0.3601	0.4075
DSH	0.1418	0.1472	0.1625	0.1675	0.0629	0.2521	0.3354	0.4148
SpH	0.1465	0.1487	0.1537	0.1617	0.0570	0.2913	0.4149	0.4356
ITQ	0.1512	0.1650	0.1693	0.1761	**0.3516**	0.3011	0.4222	0.4467
OEH	**0.1682**	**0.1728**	**0.1759**	**0.1797**	0.3022	**0.3496**	**0.4240**	**0.4512**



(a) Precision-Recall curve on CIFAR10.

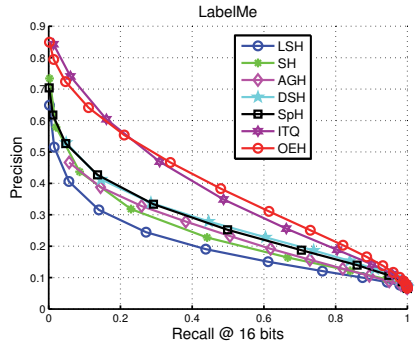


(b) Precision@K on CIFAR10.

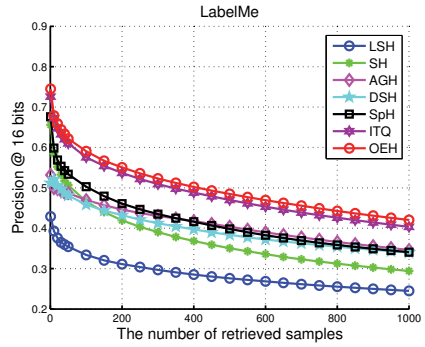


(c) Recall@K on CIFAR10.

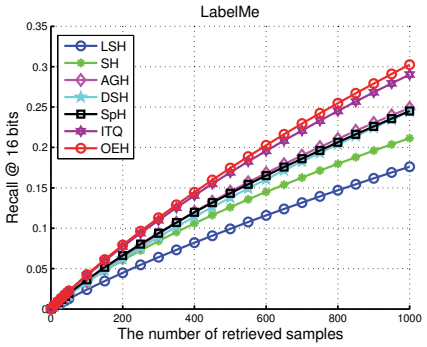
(1) ANN search performance on CIFAR10.



(d) Precision-Recall curve on LabelMe.



(e) Precision@K on LabelMe.



(f) Recall@K on LabelMe.

(2) ANN search performance on LabelMe.

Figure 2: ANN search of performance of different hashing methods on both datasets using 16-bit codes (ours in red).

for training. Since this dataset is fully annotated, we evaluate the performance by the ground-truth semantic labels.

The **LabelMe** dataset consists of 22,019 images, each of which is represented by a 512-dimensional GIST feature as well. We randomly selected 2,000 images from the dataset as the test set, leaving remaining images for training. Since label information is unavailable, neighbors in the Euclidean space are defined by a threshold as pseudo labels. That is, for a given query, the top 5% ranking items with Euclidean distances are defined as with the same label of the query.

The corpora **MIR-QBSH** in MIREX for QBH is used as the evaluation dataset[2], which is composed of 48 MIDI music files and 4431 humming queries. All the queries are hummed from the beginning of the MIDI songs. We also add 2000 noise MIDI files from the $5000+$ Essen Collection[3].

Compared Methods

The proposed OEH method is compared with six unsupervised hashing methods, including Locality Sensitive Hashing (LSH) (Raginsky and Lazebnik 2009), Spectral Hashing (SH) (Weiss, Torralba, and Fergus 2009), Anchor Graph Hashing (AGH) (Liu et al. 2011), Spherical Hashing (SpH) (Heo et al. 2012), Density Sensitive Hashing (DSH) (Jin

[2]http://www.music-ir.org/mirex/wiki/2015:Query_by_Singing/

Humming

[3]http://www.esac-data.org/

et al. 2014) and Iterative Quantization (ITQ) (Gong et al. 2013). [4] We implement our OEH hashing using Matlab on a PC with Intel Duo Core i7-3412 3.4GHz and 16G RAM. We repeat each experiment 10 times and report the average performance over all runs. In each run, training and testing sets are randomly split with the ratios reported above.

In particular, W is randomly initialized with the Gaussian distribution of mean 0 and standard deviation 1, which follows the traditional settings. The learning rate η is set as 0.3 in all experiments. For the constraints in Eq. (8), the landmarks are consisted of 500 features obtained by K-means clustering. We will also give experimental analysis on whether the setting of k-nearest neighbor affects the ordinal graph. It is worth to note that landmarks generated by K-means reflect the distribution and structure of data points. Therefore, the order information among landmarks is considered as vital as the sample to landmarks. Therefore the parameter α is set as 1 as a balance parameter. Following (Bartlett and Wegkamp 2008), we set the regularization parameter λ as 0.001, and the parameter β as 1 in all our experiments for better search performance.

Evaluation Protocols

To evaluate the performance of different hashing methods, mean Average Precision (*m*AP) is employed as the evaluate protocol, which is an overall evaluation of both precision and recall defined by mean of AP as: $mAP = \frac{1}{|Q|}\sum_{i=1}^{|Q|} AP(q_i)$, where $|Q|$ is the number of query points. $AP = \frac{1}{l}\sum_{i=1}^{n} p(i)\triangle(i)$, where l is the number of ground-truth neighbors of the *i*-th query, *p(i)* denotes the precision at the cutoff *i* for the ranking list, and $\triangle(i) = 1$ if the *i*-th retrieved result is a truth points, otherwise $\triangle(i) = 0$. We also consider other three evaluation protocols, *i.e.*, precision at top-K positions (Precision@K), recall at top-K positions (Recall@K) and Precision-Recall curve. For the experiments on query by humming, the evluation measurements are Top-10 Hit Rate and the average retrieval time for each query.

Quantitative Results on Image Retrieval

We compare OEH with the state-of-the-art hashing methods listed above on both **CIFAR10** and **LabelMe** datasets. As shown in Table 1 and Figure 2 with hash bits varied from 8 to 64, we observe that OEH consistently achieves the superior *m*AP results comparing to all baselines, especially when the hash bit is low.

For in depth analysis, previous hashing methods mainly reduce the quantization errors between the real-valued Euclidean distance and the Hamming distance. Instead, in our scheme the constraint of the order information among samples is much more comprehensive than that in Euclidean distance, which therefore results in more accurate quantization. For example, given three samples x_1, x_2, x_3, suppose the Euclidean distance between x_1 and x_2 is much larger than that of x_1 and x_3, the traditional hashing methods constrain

[4]The source codes of all the compared methods are available publicly.

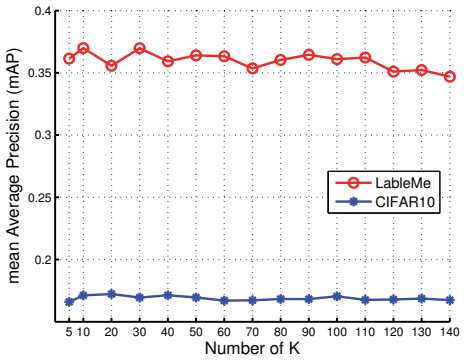


Figure 3: *m*AP curves @16 bits vs the number of K.

that the Hamming distance between $pairs(x_1, x_2)$ is always larger than that between $pairs(x_1, x_3)$, which is not easy to hold during optimization, since Hamming distance is discrete and discontinue. In contrast, for the order comparison constraints in OEH, we just need to preserve the order relation in Hamming space, *e.g.* Hamming distance $D_H(x_1, x_2)$ is larger than that of $D_H(x_1, x_2)$, which can be easily represented in the Hamming space. As a result, the corresponding optimization is much more efficient and accurate, which results in much higher *m*AP comparing to all baselines. Specially, OEH gets higher in **CIFAR10** than all baselines with lower hash bits, which can better reflect the label similarity in the Hamming space.

We further investigate the performance of the proposed OEH algorithm using the protocols of Precision@K, Recall@K and Precision-Recall curve, with a fixed hash bit of 16. The result is shown in Figure 2. With the increasing number of retrieved data points, the precision and recall always increase for all hashing algorithms, among which OEH increases the most. This indicates that the discrimination degree of Hamming distance is not evident comparing to that of Euclidean distance. Therefore, the proposed OEH significantly improves the accuracy of approximated nearest neighbor search. This also suggests that the iterative stochastic gradient descent algorithm proposed in this paper works well in finding the optimal hash functions.

At last, we discuss the influence of both landmarks and the number of k-nearest neighbours. Note that the landmarks are generated by K-means clustering, and it's not necessary that the clustering process always converges. And in an extreme case, the landmarks can be obtained by uniform sampling from the training set. As shown in Figure 3, the *m*AP value is relatively better when the number of k increases from 10 to 20. Therefore, we use a function $k = 2 \times \log(L)$ to set the relative parameter k, where L is the number of landmarks.

The experiments on Query by Humming

The traditional QBH problems focus on the two crucial problems of feature extraction and melody matching. For the problem of pitch detecter, F0 tracking (Molina et al. 2014) is always used to detect pitch sequence as the melody representation, which have been used in many QBH systems (Guo et

Figure 4: Framework of the Query by Humming.

al. 2013). In this part, we mainly focus on melody matching. Dynamic time warping (DTW) is a popular matching method for QBH, which finds a optimal matching path by a dynamic programming. However, DTW is not a suitable solution to match humming query with large-scale song database, due to its higher time complexity. Therefore, many methods have been proposed to solve this problem by dividing retrieval system into two steps. Some quickly matching algorithms, *e.g.* Earth Moverpsilas Distance (EMD) (Wang et al. 2008), are used as the first step to select song candidates, which can be viewed as the fuzzy search. During the second step, DTW algorithm will match query with candidate songs for accurate searching. So, we propose a method for QBH system, which uses OEH as fuzzy search and DTW method as accurate search. The framework of the query by humming system using OEH is shown in Figure 4.

The experiment results on MIR-QBSH dataset is shown in Table 2. The first three row of Table 2, we compared the OEH algorithm with two time seiries matching methods, *e.g.,* EMD and DTW. From the last four rows in Table 2, we compared the proposed QBH system, *a.k.a,* OEH+DTW, with the well known Query-by-Humming system EMD+DTW [5], which has been proposed in (Wang et al. 2008). We also replace the OEH part in the proposed method with other two hashing methods, *e.g.,* LSH and ITQ. For the EMD+DTW, we use the EMD algorithm to do the first stage, and use the top-300 ranking results as the candidate, which will be reranked in the second stage by DTW matching. For the hashing based QBSH, we select Top-500 Hamming ranking results as the candidates. We transform each MIDI file to a numerical pitch sequence, and seperate each sequence into 7 sub-sequence by holding the beginning of the song and moving the end at different frames. The length of sub-sequence is less than 66% of the original sequence length, and more than 33% at the same time. We also transform all the sub-sequence and query sequence to a fixed length by linear interpolation.

[5]The source codes of the EMD+DTW is available publicly.

Table 2: The experiment result on MIR-QBSH.

Methods	Top-10 Hit Rate	Query Time (s)
EMD	0.804	1.523
DTW	0.925	17.071
OEH	0.630	**0.121**
EMD+DTW	0.914	3.572
LSH+DTW	0.586	0.574
ITQ+DTW	0.872	0.787
OEH+DTW	0.907	**0.695**

As shown in Table 2, the proposed OEH+DTW can get the similar retrieval performance compared with classic EMD+DTW, with reducing 80% of searching time. In spite of well search perfomance, the average time of each query is too slow which can't be tolerant in real system. LSH+DTW and ITQ+DTW have worse performance for these application, due to using Euclidean distance to approximate the similarity between time sequence features. But for our OEH, we construct the ordinal graph according to the more accurate DTW metric, that is to say our proposed OEH can be independent of any metric space. As a result, it can filter out most of the nosiy music, and DTW re-ranking can futher improve the searching performance. In general, the proposed method can quickly and accurately complete the task of query by humming, which can also be used in large-scale music database. And it has been integrated into Tencent QQ Music product now.

Conclusion and Future Work

In this paper, we proposed a novel unsupervised hashing approach dubbed Ordinal Embedding Hashing (OEH) for large-scale image retrieval. Unlike most previous unsupervised hashing methods, the proposed approach exploits the ordinal information between sample points and landmarks, and embeds the order relations among data points into a Hamming space. The core idea of OEH is to minimize the number of wrong order comparisons in the Hamming space for a set of predefined ordinal information, which was modeled as an ordinal relation graph. Due to a large number of order comparisons in the ordinal relation graph, we further introduced a landmark-based order embedding to reduce the training scale without significantly decreasing the retrieval accuracy. In optimization, an iterative stochastic gradient descent algorithm was developed. Extensive experiments on two benchmark datasets demonstrated that the proposed OEH approach achieves the best performance in contrast with various state-of-the-art hashing methods. We also extend the OEH to the application of query by humming, and the experiment shows our proposed method has better performance of music search by user's humming or singing. In our future work, we would further extend our proposed method to larger-scale image retrieval dataset and investigate the possibility to conduct large-scale optimization in the binary code learning process.

Acknowledgement

This work is supported by the Special Fund for Earthquake Research in the Public Interest No.201508025, the Nature Science Foundation of China (No. 61402388, No. 61422210 and No. 61373076), the Fundamental Research Funds for the Central Universities (No. 20720150080 and No.2013121026), and the CCF-Tencent Open Research Fund.

References

Arias-Castro, E. 2015. Some theory for ordinal embedding. *arXiv preprint arXiv:1501.02861*.

Bartlett, P. L., and Wegkamp, M. H. 2008. Classification with a reject option using a hinge loss. *The Journal of Machine Learning Research* 9:1823–1840.

Dean, T.; Ruzon, M.; Segal, M.; Shlens, J.; Vijayanarasimhan, S.; and Yagnik, J. 2013. Fast, accurate detection of 100,000 object classes on a single machine. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*.

Gionis, A.; Indyk, P.; Motwani, R.; et al. 1999. Similarity search in high dimensions via hashing. In *Proceedings of VLDB*.

Gong, Y.; Lazebnik, S.; Gordo, A.; and Perronnin, F. 2013. Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35(12):2916–2929.

Grauman, K., and Fergus, R. 2013. Learning binary hash codes for large-scale image search. In *Machine learning for computer vision*. Springer. 49–87.

Guo, Z.; Wang, Q.; Liu, G.; and Guo, J. 2013. A query by humming system based on locality sensitive hashing indexes. *Signal Processing* 93(8):22292243.

Heo, J.-P.; Lee, Y.; He, J.; Chang, S.-F.; and Yoon, S.-E. 2012. Spherical hashing. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*.

Jiang, Q.-Y., and Li, W.-J. 2015. Scalable graph hashing with feature transformation. Proceedings of IJCAI.

Jin, Z.; Li, C.; Lin, Y.; and Cai, D. 2014. Density sensitive hashing. *IEEE Transactions on Cybernetics* 44(8):1362–1371.

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

Kulis, B., and Darrell, T. 2009. Learning to hash with binary reconstructive embeddings. In *Proceedings of Advances in neural information processing systems*.

Kulis, B., and Grauman, K. 2012. Kernelized locality-sensitive hashing. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34(6):1092–1104.

Li, P.; Shrivastava, A.; Moore, J. L.; and König, A. C. 2011. Hashing algorithms for large-scale learning. In *Proceedings of Advances in neural information processing systems*.

Liu, W.; Wang, J.; Kumar, S.; and Chang, S.-F. 2011. Hashing with graphs. In *Proceedings of the 28th international conference on machine learning*.

Liu, W.; Wang, J.; Ji, R.; Jiang, Y.-G.; and Chang, S.-F. 2012. Supervised hashing with kernels. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*.

Liu, W.; Mu, C.; Kumar, S.; and Chang, S.-F. 2014. Discrete graph hashing. In *Proceedings of Advances in neural information processing systems*.

McFee, B., and Lanckriet, G. 2011. Learning multi-modal similarity. *The Journal of Machine Learning Research* 12:491–523.

Molina, E.; Tardón, L. J.; Barbancho, I.; and Barbancho, A. M. 2014. The importance of f0 tracking in query-by-singing-humming.

Norouzi, M., and Blei, D. M. 2011. Minimal loss hashing for compact binary codes. In *Proceedings of the 28th international conference on machine learning*.

Norouzi, M.; Blei, D. M.; and Salakhutdinov, R. R. 2012. Hamming distance metric learning. In *Proceedings of Advances in neural information processing systems*.

Oliva, A., and Torralba, A. 2001. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International journal of computer vision* 42(3):145–175.

Raginsky, M., and Lazebnik, S. 2009. Locality-sensitive binary codes from shift-invariant kernels. In *Proceedings of Advances in neural information processing systems*.

Shen, F.; Shen, C.; Liu, W.; and Shen, H. T. 2015. Supervised discrete hashing. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*.

Shepard, R. N. 1962. The analysis of proximities: Multidimensional scaling with an unknown distance function. i. *Psychometrika* 27(2):125–140.

Terada, Y., and Luxburg, U. V. 2014. Local ordinal embedding. In *Proceedings of the 31st International Conference on Machine Learning*.

Torralba, A.; Fergus, R.; and Weiss, Y. 2008. Small codes and large image databases for recognition. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*.

von Luxburg, U., et al. 2014. Uniqueness of ordinal embedding. In *Proceedings of The 27th Conference on Learning Theory*.

Wang, L.; Huang, S.; Hu, S.; Liang, J.; and Xu, B. 2008. An effective and efficient method for query by humming system based on multi-similarity measurement fusion. In *Audio, Language and Image Processing, 2008. ICALIP 2008. International Conference on*, 471–475. IEEE.

Weiss, Y.; Torralba, A.; and Fergus, R. 2009. Spectral hashing. In *Proceedings of Advances in neural information processing systems*.

Young, F. W. 2013. *Multidimensional scaling: History, theory, and applications*. Psychology Press.