Supervised Set-to-Set Hashing in Visual Recognition

I-Hong Jhuo
CODAIT, IBM
ihjhuo@gmail.com

Abstract

Visual data, such as an image or a sequence of video frames, is often naturally represented as a point set. In this paper, we consider the fundamental problem of finding a nearest set from a collection of sets, to a query set. This problem has obvious applications in large-scale visual retrieval and recognition, and also in applied fields beyond computer vision. One challenge stands out in solving the problem—set representation and measure of similarity. Particularly, the query set and the sets in dataset collection can have varying cardinalities. The training collection is large enough such that linear scan is impractical. We propose a simple representation scheme that encodes both statistical and structural information of the sets. The derived representations are integrated in a kernel framework for flexible similarity measurement. For the query set process, we adopt a learning-to-hash pipeline that turns the kernel representations into hash bits based on simple learners, using multiple kernel learning. Experiments on two visual retrieval datasets show unambiguously that our set-to-set hashing framework outperforms prior methods that do not take the set-to-set search setting.

1 Introduction

Searching for similar data samples is a fundamental step in many large-scale applications. As the data size explodes, hashing techniques have emerged as a unique option for approximate nearest neighbor (ANN) search, as it can dramatically reduce both the computational time and the storage space. Successes are seen in areas including computer vision and information retrieval [Kulis *et al.*, 2009; Sun *et al.*, 2017; Wang *et al.*, 2012a; Wang *et al.*, 2016]. Hashing methods perform space partitioning to encode the original high-dimensional data points into binary codes. With the resulting binary hash codes, one can perform extremely rapid ANN search that entails only sublinear search complexity.

Conventional hashing schemes concern point-to-point (P2P) search setting. They either depend on randomization and are data oblivious (represented by the classic Locality Sensitive Hashing – LSH), or are based on advanced machine learning techniques to learn hashing functions that are better tailored to the specific data and/or label distribution. The latter includes unsupervised [Gong and Lazebnik, 2011; Weiss *et al.*, 2008], semi-supervised [Wang *et al.*, 2012a; Wang *et al.*, 2010], and supervised hashing [Kulis and Darrell, 2009; Salakhutdinov and Hinton, 2009; Mu *et al.*, 2010; Liu *et al.*, 2012a; Zhang and Li, 2014; Shen *et al.*, 2015; Li *et al.*, 2016; Weng *et al.*, 2019].

A natural generalization of the point-to-point search is set-to-set (S2S) search. For example, one can pose the facial image recognition problem as one that queries for a nearest subspaces to a given point [Wang *et al.*, 2013]. Indeed, there are several recent attempts, studying point-to-hyperplane search that is useful for active learning [Liu *et al.*, 2012b], or subspace-to-subspace search [Basri *et al.*, 2011] that models S2S search assuming linear structures in the sets.

In this paper, we consider the set-to-set search problem in its full generality. This general setting finds applications ranging from video-based surveillance to 3D face retrieval from collections of 2D images [Berretti *et al.*, 2010; Tuzel *et al.*, 2007; Sivic *et al.*, 2005]. Compared to specialized settings discussed above that come with natural notion of distance, a central challenge here is how to measure the distance/similarity between sets. We propose a similarity measure that captures both the statistical and structural aspects of the sets (Section 3). To learn the hash bits, we adopt *dyadic hypercut* as a weak learner [Moghaddam and Shakhnarovich, 2002] to derive a boosted algorithm that integrates both the structural and statistical similarities. The whole framework is illustrated in Fig. 1. In this paper, we focus on image applications, and hence coin the name Image Set Hashing (ISH). However, the core components of the proposed framework can be extended to generic scenarios.

2 Related Work

In this section, we discuss representative works in randomization-based hashing and learning-based hashing for P2P setting, and recent work on certain restricted S2S setting. Review of recent development of hashing techniques can be found in [Wang *et al.*, 2016; Wang *et al.*, 20114].

LSH hashing and variants are iconic randomization-based hashing schemes. They are simple in theory and efficient in practice, and flexible enough to handle various distance measures [Charikar, 2002; Datar *et al.*, 2004; Kulis and Darrell,

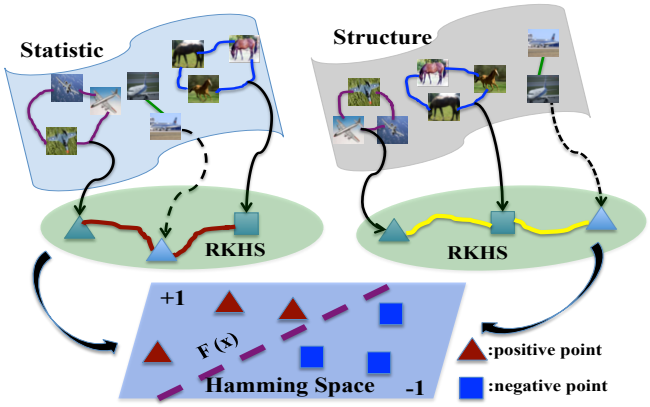

Figure 1: Illustration of the proposed ISH framework. First, statistical and structural information of the image sets are encoded. Second, appropriate kernel mappings are chosen to measure the similarities between image sets. A boosted algorithm based on the two kernels is used to construct the hash function.

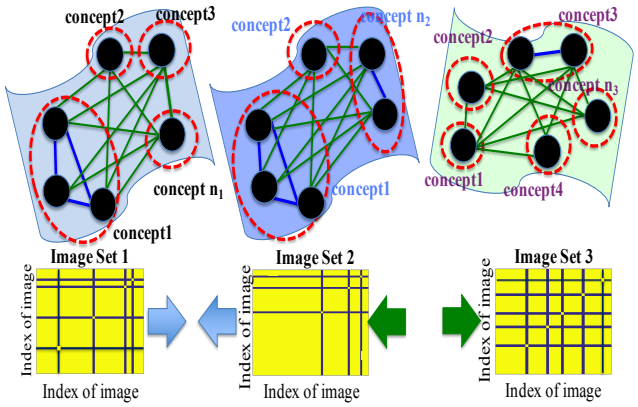

Figure 2: Extracting the structural information and measure the structural similarity. In the top row, graphs are constructed to represented individual sets: nodes are data points and graph weights indicate the similarities. Dense cliques (red circles) are then extracted to reveal the holistic structure in a set. Point sets with similar clique structures (sets 2 and 1 as shown) are assumed to have higher similarity. The bottom row shows the similarity matrices, in which yellow blocks indicate the high similarities among data points in each clique. Best viewed in color.

2009]. However, the data oblivious design of the LSH family methods often causes suboptimal recall-precision trade-off curve. Hence, learning-based hashing schemes have been under active development recently. Generally, both the data and label information is fed into carefully designed learning pipeline to produce more adaptive and efficient hash codes. Depending on whether the label information is in use, these schemes are either unsupervised, e.g., spectral hashing [Weiss *et al.*, 2008], graph hashing, and ITQ (iterative quantization) [Gong and Lazebnik, 2011], or (semi-)supervised hashing, including [Lin *et al.*, 2013; Liu *et al.*, 2012a; Mu *et al.*, 2010; Norouzi and Fleet, 2011; Sablayrolles *et al.*, 2017; Wang *et al.*, 2010]. Particularly, recent efforts have built more powerful hashing schemes on top of deep learning [Salakhutdinov and Hinton, 2009; Salakhutdinov and Hinton, 2007; Liong *et al.*, 2015; Masci *et al.*, 2013]. All these methods only deal with the P2P setting.

Study of the S2S setting started only very recently, and is mostly about image applications. Statistical or geometric assumptions are often made on the sets to facilitate representation. For example, statistical distribution of data points in each set can be assumed, and KL-divergence can be used to measure set similarity [Arandjelovic *et al.*, 2005]. By comparison, point sets can also lie on linear subspaces or more general geometric objects [Cevikalp and Triggs, 2010; Hu *et al.*, 2011; Kim *et al.*, 2007; Liu *et al.*, 2014; Sun *et al.*, 2014; Sun *et al.*, 2015; Wang *et al.*, 2012b]. Among the representation schemes, representation based on covariance matrices has led to superior performances on image sets (video frames) [Wang *et al.*, 2012b; Lu *et al.*, 2013; Tuzel *et al.*, 2007]. For instance, in [Li *et al.*, 2015], covariance matrix is used in such way, and similarity is then measured via kernel mapping and learning. Promising result has been reported, but the framework is restricted to cases when the query is a single point. For image applications specifically, the Set Compression Tree [Relja and Zisserman, 2014] compresses a set of image descriptors jointly (rather than individual descriptors) and achieve a very small memory footprint (as low as 5 bits). However, all existing representation methods do not account the holistic structural information; this may lead to incorrect similarity measurements on highly nonlinear data distributions. Our representation and hashing scheme is a first attempt to directly address the above problems in the general S2S setting.

3 Structural and Statistical Modeling

In this section, we detail how the structural and statistical information is extracted from the point sets, and how similarity between sets is measured.

3.1 Structure via Graph Modeling

The idea here is to use graph for discovering structures within data, and then measure the similarity via appropriate kernel mapping on graphs [Tenenbaum *et al.*, 2000; Gartner, 2003; Zhou *et al.*, 2009], Fig. 2 gives an example.

To model data points within a set, we derive an affinity matrix A based on quantized pairwise distances [Zhou *et al.*, 2009]: if the distance is larger than a predefined threshold μ, the corresponding affinity value in A is set to 0, and 1 otherwise. We use $\mathbf{x}_i$'s to denote the point sets, and A^i's to denote the corresponding affinity matrices thus constructed. With all the A^i's at hand, the point set similarity is defined as:

$$K_g(\mathbf{x}_i, \mathbf{x}_j) = \frac{\sum_{p=1}^{n_i}\sum_{q=1}^{n_j} A_{ip}A_{jq}g(x_{ip},x_{jq})}{\sum_{p=1}^{n_i} A_{ip}\sum_{q=1}^{n_j} A_{jq}}, \qquad (1)$$

where, $A_{ip} = 1/\sum_{u=1}^{n_i} a^i_{pu}$, $A_{jq} = 1/\sum_{v=1}^{n_j} a^j_{qv}$ and $g(x_{ip}, x_{jq}) = \exp(-\gamma_g \parallel x_{ip} - x_{jq} \parallel^2)$. The parameter γ_g is a constant and n_i and n_j are the number of data points in $\mathbf{x}_i$ and $\mathbf{x}_j$, respectively.

To understand the captured structural information, each clique (formed by several 1 elements) in A^i can be regarded as one concept. If A^i is an all-one matrix, all data points in

one set belong to one concept and each point set is considered as one data point. When A^i is an identical matrix, each data point is independent, and no relation can be discovered. When A^i is a clique-based matrix, data can be clustered into cliques and K_g is a clique-based graph kernel. In this way, we leverage the structural information for our set hashing.

3.2 Statistical Information

Covariance matrices have provided effective local region representation for visual recognition and human identification [Tuzel *et al.*, 2007; Liu *et al.*, 2014]. Intuitively, they describe the local image statistics. In this work, we use covariance matrices to depict the statistical variance of images within each set. Given N image sets, $\mathcal{X} = \{(\mathbf{x}_1, l_1), \cdots, (\mathbf{x}_i, l_i), \cdots, (\mathbf{x}_N, l_N)\}$. $\mathbf{x}_i = \{x_{i1}, \cdots, x_{i,n_i}\}$ is an image set, where $x_{i,j} \in \mathbb{R}^d$ represents the jth d-dimensional feature vector in $\mathbf{x}_i$, and the set consists of n_i images. l_i's are the labels of each image set. Each image set is represented with a $d \times d$ covariance matrix:

$$C_i = \frac{1}{n_i} \sum_{j=1}^{n_i} (x_{ij} - \bar{\mathbf{x}}_i)(x_{ij} - \bar{\mathbf{x}}_i)^\top, \tag{2}$$

where $\bar{\mathbf{x}}_i$ is the mean feature vector within the set. The diagonal elements of C_i represent the variance of each individual image feature, and the off-diagonal elements are their respective covariance. In this way, the covariance modeling can provide a desirable statistic for semantic variance among all images for an image set. Moreover, we use Gaussian-logrithm kernel [Jayasumana *et al.*, 2013] to map each covariance matrix of an image set into high dimension space, as follows:

$$\begin{aligned} K_s(\mathbf{x}_i, \mathbf{x}_j) &= \phi(C_i)^\top \phi(C_j) \\ &= \exp(-\left\|\log(C_i) - \log(C_j)\right\|_F^2 / 2\gamma_s^2), \end{aligned} \tag{3}$$

where γ_s is a positive constant, which be set to the mean distances of training points. Kernel matrix K_s, is for the image sets in Riemannian space and $\|\cdot\|_F$ denotes the matrix Frobenius norm. In the end, each pair of image sets $\mathbf{x}_i$ and $\mathbf{x}_j$ are mapped by the K_g and K_s kernel functions into a high dimensional space.

4 Image Set Hashing

4.1 Learning Framework

Suppose we have an image set dataset $\mathcal{X} = \{\mathbf{x}_i, l_i\}$. The goal of hashing is to generate an array of appropriate hash functions $h : \mathbb{R}^d \mapsto \{0, 1\}$ by a designed function Ψ and each bit is constructed by $h(\mathbf{x}) = \text{sign}(\Psi(\mathbf{x}))$. However, there is no straightforward way to generate hash codes for each image set. Inspired by [Vemulapalli *et al.*, 2013; Li *et al.*, 2015; Wang *et al.*, 2012b] , we develop a mapping framework to construct hash codes for image sets in a common Hamming space based on multiple kernels. Kernel methods [Hamm and Lee, 2009; Jayasumana *et al.*, 2013; Vemulapalli *et al.*, 2013; Wang *et al.*, 2012b] are known to capture and unfold rich information in data distribution. After the mapping process, we can generate hash codes in a Hamming space by simultaneously considering the structural and statistical information and iteratively maximize the discriminant margins based on multiple kernels learning.

4.2 Weak Learners with Boosting Algorithm for Hash Functions

Since a multi-class classification problem can always be treated as an array of two-class problems by adopting one-against-one or one-against-all strategies, we design a boosting algorithm to learn binary splits for constructing hash functions. Specifically, we consider *dyadic hypercut* [Moghaddam and Shakhnarovich, 2002] with multiple kernel functions. A dyadic hypercut f is generated by a kernel with a pair of different labels in training samples, where f is parameterized by positive sample $\mathbf{x}_a$, negative sample $\mathbf{x}_b$, and kernel functions $\{K_m\}$,i.e., m indicates statistical kernel (K_s) or structural kernel (K_g), and can be represented as follows:

$$f(\mathbf{x}) = \text{sign}(K_m(\mathbf{x}_a, \mathbf{x}) - K_m(\mathbf{x}_b, \mathbf{x}) + \varepsilon), \tag{4}$$

where $\varepsilon \in \mathbb{R}$ is a threshold. The size of the totally generated weak learner pool is $|f| = M \times n_a \times n_b$, where M, n_a and n_b are the numbers of kernels, positive training samples and negative training samples, respectively. With an efficient boosting process, we iteratively select a subset of weak learners by considering the learning loss.

Note that the learning process may be susceptible to overfitting when $|f|$ is large. To alleviate this issue, we adopt a boosting algorithm [Freund and Schapire, 1995; Moghaddam and Shakhnarovich, 2002] to combine a number of weak splits (weak learners) into a strong one. Specifically, we iteratively select the discriminant weak learners generated from multiple kernels via maintaining a weighted distribution w^t over data. Each iteration t produces a weak hypothesis $f(\mathbf{x}) : \mathbf{x} \to \{+1, -1\}$ and a weighted error δ^t. The learning algorithm is aimed at selecting weak learner f^t for minimizing δ^t followed by updating next distribution w^{t+1}. We adopt *exponential loss* [Freund and Schapire, 1995] and minimize the loss function to select the best weak learner f^t at iteration t and the best weak learner is computed as:

$$f^t = \min_f \sum_{i=1}^{N} w_i^t \exp(-l_i f(\mathbf{x}_i)), \tag{5}$$

where w_i^t indicates the weight of $\mathbf{x}_i$ at iteration t. Once obtaining the best weak learner, we update the data distribution based on weighted errors. The linear combination of weak learners, i.e., a strong split, is computed as follows:

$$F(\mathbf{x}) = \text{sign}(\sum_{t=1}^{T} \lambda^t f^t(\mathbf{x})), \tag{6}$$

where $\lambda^t = \frac{1}{2} \log \frac{1-\delta^t}{\delta^t}$. Each iteration t, $F = \sum_{\tau=1}^{t-1} \lambda^\tau f^\tau$ is a linear combination of the $(t-1)$ weak learners.

4.3 Objective Function

With the designed hash functions, the following are desired properties of the hash codes: (1) Each hash value is independent of the binary representation for each sample. (2) When samples are close to each other in feature space (e.g. with similar distributions), the hash codes should induce similar hash values with a small Hamming distance. (3) In the resulting Hamming space, different contents of samples should have different hash codes, which push different samples of categories as far as possible, meanwhile gather the samples

of the same category close to each other. Based on the criteria, we derive our multiple kernel hashing as follows:

$$\min_{H_q,H_r} \alpha D_s + \beta D_c + \nu_1 \sum_{\substack{r'\in\{1:R\}\\ i\in\{1:N\}}} \Omega$$

$$s.t. \quad H_q^{r'i} = \text{sign}(\sum_{t=1}^{T} \lambda_t f_q^{tr'}(\mathbf{x}_i)), \forall i \in \{1:N\}, \forall r' \in \{1:R\}$$

$$H_r^{r'i} = \text{sign}(\sum_{t=1}^{T} \lambda_t f_r^{tr'}(\mathbf{x}_i)), \forall i \in \{1:N\}, \forall r' \in \{1:R\}, \tag{7}$$

where $H_*^{r'i}$ is the hash value of the ith image set using the r'th strong split, i.e., hash function, and r and q represent the retrieval and query sets in the training process. R is the number of the splits and $\Omega = \sum_{t=1}^{T}(\lambda_q^t f_q^{tr'}(\mathbf{x}_i) + \nu_2 \lambda_r^t f_r^{tr'}(\mathbf{x}_i))$. $f_*^{r'}$ is the r'th strong split generated from the number of T weighted weak learners (in eq. (6)), λ_*^t is the coefficient of weak learners trained via a boosting algorithm, ν_1 and ν_2 are constant parameters.

The minimization of the first two terms, i.e., D_s and D_c, tend to find an optimal difference of the distance between intra- and inter- categories, which capture the discriminative property among all the samples and are defined in eq. (8) and eq. (9), and $d(\cdot,\cdot)$ indicates distance measure in Hamming space. In addition, D_c can refine the hash codes generated from the training image sets (q and r parts) by maximizing the separability of category algorithm in [Rastegari *et al.*, 2012]. Simultaneous consideration of the two distance functions helps to minimize the within-category distances and meanwhile maximize the between-category dis. By formulating the structural and statistical information with multiple kernels for our objective function, the image set hashing becomes more robust and discriminative.

$$D_s = \sum_{(m,n)\in\mathcal{M}} d(H_*^m, H_*^n) - \nu_3 \sum_{(m,n)\in\mathcal{C}} d(H_*^m, H_*^n) \tag{8}$$

$$D_c = \sum_{(m,n)\in\mathcal{M}} d(H_q^m, H_r^n) - \nu_4 \sum_{(m,n)\in\mathcal{C}} d(H_q^m, H_r^n), \tag{9}$$

where $\mathcal{M}$ and $\mathcal{C}$ are represented as intra- and inter-category, H_* can be H_q or H_r, and ν_3 and ν_4 are the pre-computable constant parameters to balance the intra-catetory and inter-category scales.

4.4 Optimization

The objective function optimization problem 7 is a typical nonsmooth, nonconvex multiple variable minimization problem. We derive an iterative block coordinate descent algorithm [Tseng, 2001] for the optimization. Algorithm 1 gives the entire algorithm. Here, we highlight several critical steps in our algorithm. We first compute the kernel matrices K_g and K_s with eq. (1) and eq. (3) for q training and r training image sets. After the kernel computation, we adopt kernel PCA [Scholkopf *et al.*, 1997] for the q and r two parts to obtain the initial hash codes, i.e., H_q and H_r, based on their statistical kernels in Step 1 to Step 3. Next, we update the H_q and H_r codes by optimizing eq. (8) for seeking the discriminability and utilize an efficient subgradient descent algorithm [Rastegari *et al.*, 2012] for the binary optimization (the optimization algorithm gives the generated code with two properties: sample-wise balance and bit-wise balance.). In Step 6, we use the updated hash codes, H_q and H_r, to train R two-class strong splits based on multiple kernels. More specifically, we adopt cross-training strategy [M. Rastegari J. Choi and Davis, 2013] by using the hash codes, H_r, as training labels to train the strong splits with q training image sets and similar process to H_q hash code with r training image sets. After that, we update the current hash codes, H_q and H_r by using the learned strong splits based on multiple kernel learning. In order to improve the discriminability, we combine H_q and H_r together to refine the learned hash codes with eq. (9). The process is then repeated. Convergence typically occurs within few outer iterations. Once we obtained the two strong split models ($\sum_{r'=1}^{R} F_q^{r'}(\mathbf{x})$ and $\sum_{r'=1}^{R} F_r^{r'}(\mathbf{x})$ in Algorithm 1), we can adopt them to generate query and database hash codes in the testing process, respectively.

Algorithm 1 Image Set Hashing

Input: a set of training image sets, $\mathcal{X} = \{\mathbf{x}_i, l_i\}$ is divided into q and r two training image set parts, where $\mathbf{x}_i = \{x_{ij}\}_{j=1}^{n_i} \in \mathbb{R}^d$, $i \in \{1, 2, \cdots, N\}$, $l_i = \{1, \cdots, L\}$.

Initialize: Compute kernel matrices for q training image sets, i.e., $(K_m)_q$, by using the kernel functions (K_g and K_s) according to eq. (1) and eq. (3); Similar to r training image sets, $(K_m)_r$

1: $V_q \in \mathbb{R}^{N\times R}, V_r \in \mathbb{R}^{N\times R} \leftarrow$ kernel PCA with statistical kernels for K_q and K_r, respectively
2: $H_q \leftarrow \text{sign}(V_q^\top (K_s)_q)$
3: $H_r \leftarrow \text{sign}(V_r^\top (K_s)_r)$

Optimization:

4: **while** not converged **do**
5: Optimize H_q, H_r with eq.(8)
6: Train R splits by weak learner selection in e.q.(6) on q kernels $(K_m)_q$ by using H_r as training labels, and inversely train another R splits on r kernels $(K_m)_r$ by using H_q as training labels
7: $H_q \leftarrow \text{sign}(\sum_{r'=1}^{R} F_q^{r'}(\mathbf{x})) \leftarrow (K_m)_q$
8: $H_r \leftarrow \text{sign}(\sum_{r'=1}^{R} F_r^{r'}(\mathbf{x})) \leftarrow (K_m)_r$
9: Optimize $H = [H_q, H_r] \in \{0,1\}^{R\times 2N}$ with eq.(9)
10: Train R splits by weak learner selection in e.q.(6) on q kernels $(K_m)_q$ by using H_r as training labels, and inversely train another R splits on r kernels $(K_m)_r$ by using H_q as training labels
11: Check the convergence condition
12: **end while**
13: **Output:** $\sum_{r'=1}^{R} F_q^{r'}(\mathbf{x})$ and $\sum_{r'=1}^{R} F_r^{r'}(\mathbf{x})$ for query and database encoding in the testing process, respectively.

5 Experiments

We evaluate the effectiveness of the proposed Image Set Hashing (ISH) method on two well-known benchmarks, CIFAR-10 and TV-series, i.e., Big Bang Theory. We also conduct extensive comparison studies with state-of-the-art methods, including Locality Sensitive Hashing (LSH) [Indyk and Motwani, 1998], Spectral Hashing (SH) [Weiss *et al.*, 2008], Kernelized LSH (KLSH) [Kulis and Darrell, 2009]; Semi-Supervised Hashing (SSH) [Wang *et al.*, 2010] and supervised methods, Kernel-Based Supervised Hashing (KSH) [Liu *et al.*, 2012a] and Hashing across Euclidean space and Riemannian manifold (HER) [Li *et al.*, 2015]. For the competing techniques, we adopted the publicly released codes of SH, KLSH, KSH and HER in our experiments.

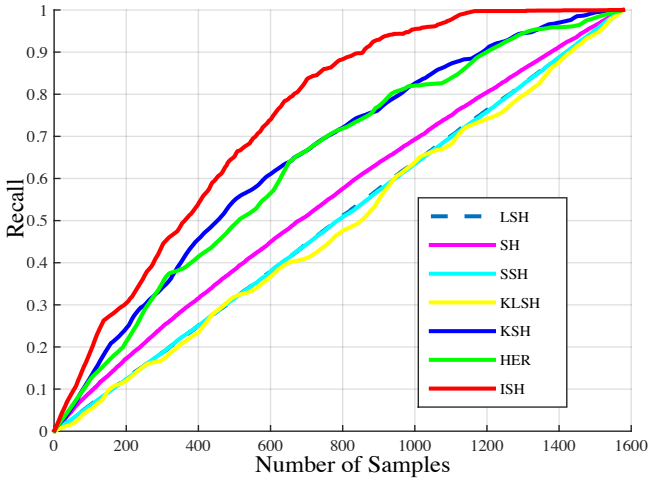

Figure 3: The evaluation results by mean recall curves for Hamming ranking using 24 bits on the CIFAR-10 dataset. The number of retrieved samples is up to 1600.

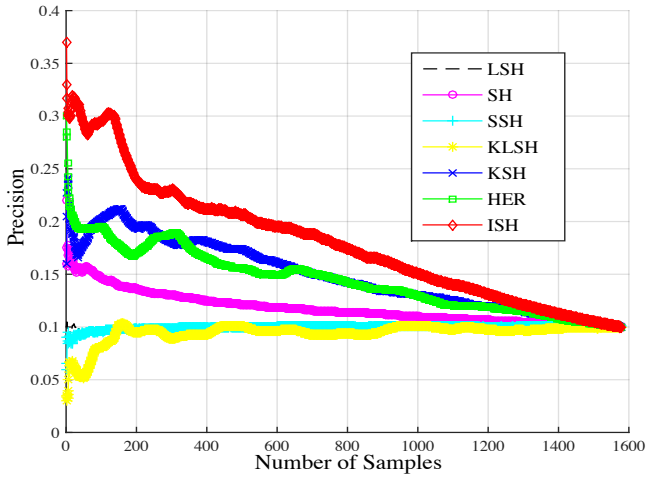

Figure 4: The evaluation results by mean precision curves for Hamming ranking using 24 bits on the CIFAR-10 dataset. The number of retrieved samples is up to 1600.

5.1 Experiment on CIFAR-10

We compare the performance for different hashing techniques on the CIFAR-10 dataset. As a labeled subset of the 80M tiny images, the CIFAR-10 dataset consists of a total of 60K color images, each of which has the size of 32×32 resolution. The dataset contains 6K image samples with ten object categories. To evaluate the performance, we uniformly and randomly sample images from each category to form a total of 195 image sets, each of which contains about $25 \sim 50$ images for the training process (q and r two parts), 100 image sets as query in testing and 1577 image sets for the testing database in KLSH, KSH, HER, ISH methods. To test the LSH, SH and SSH methods, we randomly select 1K images as queries and the remaining as database samples. For feature representation, each image is represented as a 512-dimensional GIST feature vector [Oliva and Torralba, 2001].

We evaluate two test scenarios: Hamming ranking and hash look-up. In Fig. 3 and Fig. 4 show the mean recall and precision curves from different number of returned search samples when using 24-bit hash codes. As we can see, the proposed ISH method produces higher quality of Hamming embedding since it significantly outperforms the competing methods in terms of precisions, recalls, and MAPs. In general, the methods using set information often provide better performance than those based on P2P settings. For instance, the HER method generates the second best MAPs for most of the test cases. The relative performance gains in MAP ranges from 6% to 23.6% compared to the HER method. Such performance gains confirm the value of exploring statistical and clique-based structural information for hash function design. In Fig. 5, we also evaluate the results using the hash look-up table strategy by showing the precision curve within Hamming radius 2 for hash codes from 8-bit to 48-bit. Again the proposed ISH method achieved the best precisions across all the cases.

5.2 Experiment on TV-series

The Big Bang Theory (BBT) video (image set) benchmark [1] was collected by [Bauml *et al.*, 2013] and contains in 3341 face videos from $1 \sim 6$ episodes of season one. The dataset includes around $5 \sim 8$ main cast characters and has multiple characters at the same full-view scene shot. Even though most of the scenes are taken in indoors, it is still extremely challenging since the resolution of faces regions are quite small with an average size of 75 pixels. In the experiments, we use the provided face features, which are extracted from face videos by block Discrete Cosine Transformation (DCT) feature. In this way, each face is represented by a 240-dimensional feature vector.

In the experiments, we have two different settings. For the first setting, we follow the setting used in [Li *et al.*, 2015] and apply still images for the q part in training and query in testing process and denote the setting as ISH^0. The second setting indicated as ISH, we have 150 image sets for q and r two parts in the training process, respectively. For query in testing, we use 100 image sets and the remaining image sets for database. The setting can completely utilize statistical and structural information. For the comparison, the first group of compared methods consists of seven point-to-point (P2P) hash methods, i.e, LSH, ITQ, SH, Discriminative Binary Codes (DBC) [Rastegari *et al.*, 2012] , SSH, MM-NN [Masci *et al.*, 2013], and KSH (point). In addition, we generate kernels for image sets (represented by covariance matrices) and employ them as input for the KLSH, KSH (set), and HER methods. Thus, we have the second group of methods that uses kernels for image sets. Such a modification can be used to further justify the advantage of explore the structural and statistical information of image sets. The performance evaluated by MAPs for the compared methods is shown in Table 1. We vary the number of hash bits from 8

[1] https://cvhci.anthropomatik.kit.edu/ baeuml/datasets.html

		TV drama: the Big Bang Theory			
Method	8 bits	16 bits	32 bits	64 bits	128 bits
LSH [Indyk and Motwani, 1998]	0.2109	0.2086	0.2092	0.1963	0.1994
ITQ [Gong and Lazebnik, 2011]	0.2935	0.3025	0.2989	0.3029	0.3060
SH [Weiss *et al.*, 2008]	0.2377	0.2652	0.2665	0.2623	0.2673
DBC [Rastegari *et al.*, 2012]	0.4489	0.4495	0.4235	0.4005	0.3867
SSH [Wang *et al.*, 2010]	0.2716	0.2855	0.2662	0.2584	0.3003
MM-NN [Masci *et al.*, 2013]	0.3752	0.3955	0.4664	0.5124	0.4922
KLSH [Kulis and Darrell, 2009]	0.2450	0.2498	0.2381	0.2256	0.2325
KSH (point)	0.4090	0.4366	0.4454	0.4567	0.4604
KSH [Liu *et al.*, 2012a] (set)	0.4590	0.4619	0.4534	0.4685	0.4631
HER [Li *et al.*, 2015]	0.4606	0.5049	0.5227	0.5490	0.5539
ISH^0	0.4833	0.5279	0.5359	0.5501	0.5712
ISH	**0.5018**	**0.5592**	**0.5864**	**0.6007**	**0.6280**

Table 1: The evaluation results measured by Mean Average Precision on the on video (image set) benchmark TV-series (BBT). The length of hash codes ranges from 8-bit to 128-bit. Besides the proposed ISH, the first group of compared methods consists of seven P2P hashing methods, i.e., LSH, ITQ, SH, DBC, SSH, MM-NN, and KSH. The second group of compared methods include three modified techniques, i.e., KLSH, KSH, and HER, that use image set information as input.

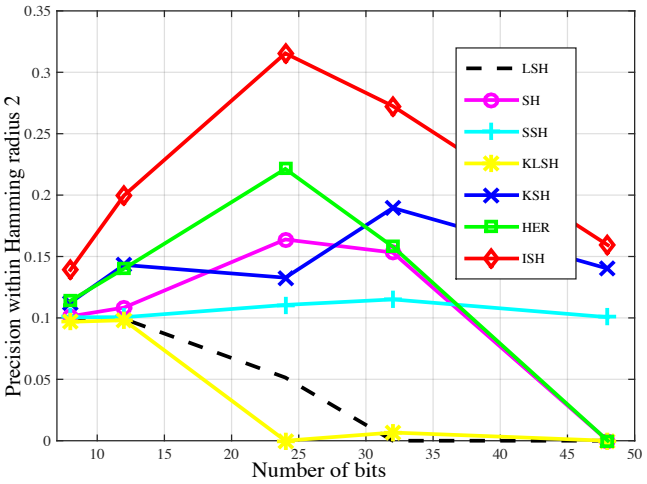

Figure 5: Mean precision curves of Hamming radius 2 in 24 bits on CIFAR-10 dataset. The curves of LSH, SH, SSH are randomly sampled.

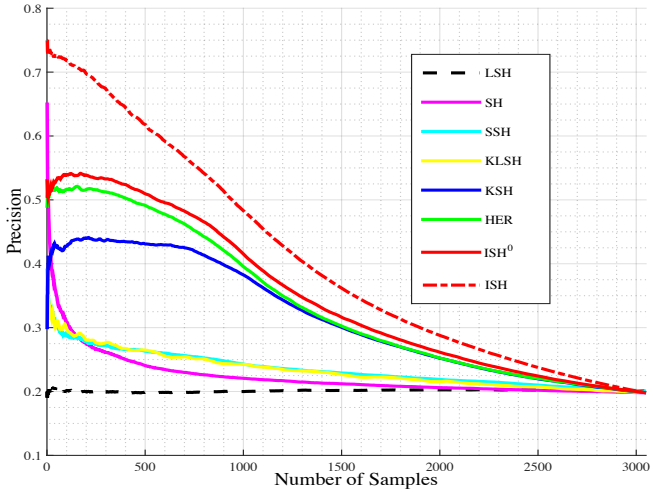

Figure 6: The evaluation results by mean precision curves for Hamming ranking using 128 bits on the BBT video dataset.

to 128 bits. As we can see, the method (ISH^0) generates the second best MAPs when structural information is ignored in the q training and the query testing process. Moreover, ISH method achieved the best performance for all the tested cases when both of statistical and structural information are considered. This evidence shows that the structural information can help to generate more robust and discriminative hash codes.

In addition, we use 128-bit hash codes and show the performance curves for the seven compared methods. Fig. 6, Fig. 7 and Fig. 8 show the precision, recall, and precision-recall curves, respectively. From these results, the proposed ISH achieves the best performance compared to all the baseline techniques, including both P2P methods and modified S2S methods. The underlying reason lies in that the ISH method can simultaneously capture the statistical (covariance matrix) and structural (graph kernel) information, i.e., combination of weak learners, to generate each hash code. Hence, it captures the most intrinsic characteristics within complicated variations of face images for the same subject. Moreover, if we compare the general performance between the P2P and S2S settings for the KSH method, we can observe that the S2S hashing continuously attains higher search accuracy. It further confirms the advantage of using set information. However, simply employing the covariance matrix as inputs limits the performance improvements. In summary, with the S2S setting and fully explore statistical and structural information, the proposed ISH yields significant performance gain across all the experiments.

6 Conclusion

We have presented a set-to-set (S2S) ANN search problem and proposed to learn the optimal hash codes for image sets by simultaneously exploiting the statistical and structural information. The key idea is to transform the image sets into a high dimension space where each of image set can be characterized by a graph kernel and statistical measurement. As a result, the proposed S2S hashing achieves a robust and discriminative representation for searching datapoint sets. The experimental results have demonstrated the effectiveness of the proposed ISH method by showing superior performance over several representative competing approaches.

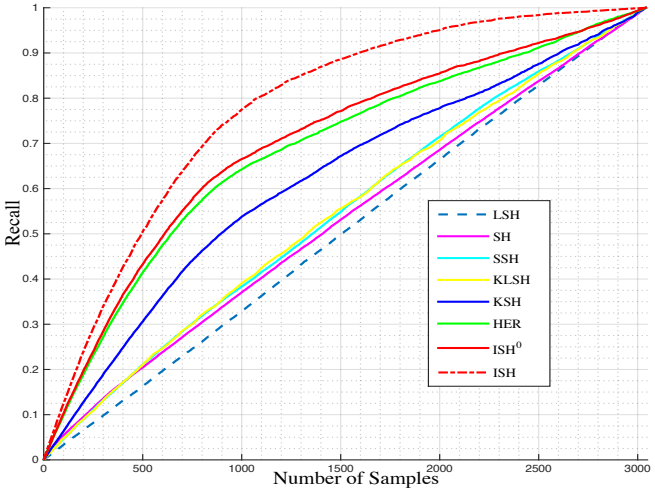

Figure 7: The evaluation results by mean recall curves for Hamming ranking using 128 bits on the BBT dataset.

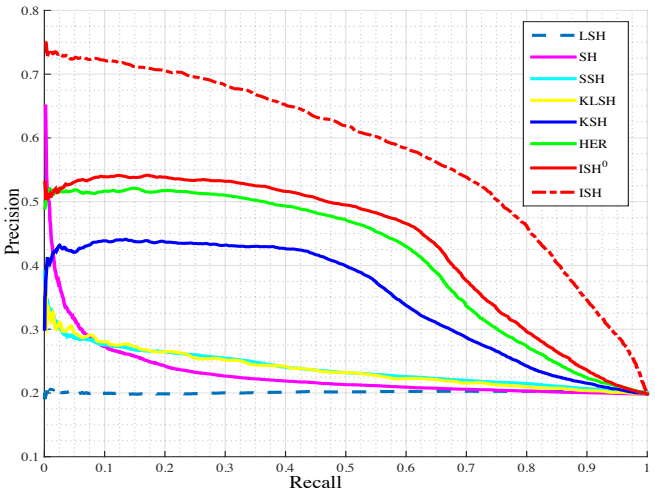

Figure 8: The evaluation results by mean precision-recall curves for Hamming ranking using 128 bits on the BBT dataset.

References

[Arandjelovic *et al.*, 2005] O. Arandjelovic, G. Shakhnarovich, J. Fisher, R. Cipolla, and T. Darrell. Face recognition with image sets using manifold density divergence. In *CVPR*, 2005.

[Basri *et al.*, 2011] R. Basri, T. Hassner, and L. Zelnik-Manor. Approximate nearest subspace search. In *IEEE Trans PAMI*, 2011.

[Bauml *et al.*, 2013] M. Bauml, M. Tapaswi, and R. Stiefelhagen. Semi-supervised learning with constraints for person identification in multimedia data. In *CVPR*, 2013.

[Berretti *et al.*, 2010] S. Berretti, A. D. Bimbo, and P. Pala. 3d face recognition using isogeodesic stripes. In *IEEE Trans. PAMI*, 2010.

[Cevikalp and Triggs, 2010] H. Cevikalp and B. Triggs. Face recognition based on image sets. In *CVPR*, 2010.

[Charikar, 2002] M. S. Charikar. Similarity estimation techniques from rounding algorithms. In *STOC*, 2002.

[Datar *et al.*, 2004] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni. Locality sensitive hashing scheme based on p-stable distributions. In *SCG*, 2004.

[Freund and Schapire, 1995] Y. Freund and R. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *Computational Learning Theory*, 1995.

[Gartner, 2003] T. Gartner. A survey of kernels for structured data. In *SIGKDD*, 2003.

[Gong and Lazebnik, 2011] Y. Gong and S. Lazebnik. Iterative quantization: a procrustean approach to learning binary codes. In *CVPR*, 2011.

[Hamm and Lee, 2009] J. Hamm and D. Lee. Grassmann discriminant analysis: a unifying view on subspace-based learning. In *ICML*, 2009.

[Hu *et al.*, 2011] Y. Hu, A. S. Mian, and R. Owens. Sparse approximated nearest points for image set classification. In *CVPR*, 2011.

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

[Jayasumana *et al.*, 2013] S. Jayasumana, R. Hartley, M. Salzmann, H. Li, and M. Harandi. Kernel methods on the riemannian manifold of symmetric positive definite matrices. In *CVPR*, 2013.

[Kim *et al.*, 2007] T.-K. Kim, J. Kittler, and R. Cipolla. Discriminative learning and recognition of image set classes using canonical correlations. In *IEEE Trans. PAMI*, 2007.

[Kulis and Darrell, 2009] B. Kulis and T. Darrell. Learning to hash with binary reconstructive embeddings. In *NIPS*, 2009.

[Kulis *et al.*, 2009] B. Kulis, P. Jain, and K. Grauman. Fast similarity search for learned metrics. In *IEEE Trans. PAMI*, 2009.

[Li *et al.*, 2015] Y. Li, R. Wang, Z. Huang, S. Shan, and X. Chen. Face video retrieval with image query via hashing across euclidean space and riemannian manifold. In *CVPR*, 2015.

[Li *et al.*, 2016] W.-J. Li, S. Wang, and W.-C. Kang. Feature learning based deep supervised hashing with pairwise labels. In *IJCAI*, 2016.

[Lin *et al.*, 2013] G. Lin, C. Shen, D. Suter, and A. V. D. Hengel. A general two-step approach to learning-based hashing. In *ICCV*, 2013.

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

[Liu *et al.*, 2012a] W. Liu, J. Wang, R. Ji, Y.-G. Jiang, and S.-F. Chang. Supervised hashing with kernels. In *CVPR*, 2012.

[Liu *et al.*, 2012b] W. Liu, J. Wang, Y. Mu, S. Kumar, and S.-F. Chang. Compact hyperplane hashing with bilinear functions. In *ICML*, 2012.

[Liu *et al.*, 2014] M. Liu, S. Shan, R. Wang, and X. Chen. Learning expressionlets on spatio-temporal manifold for dynamic facial expression recognition. In *CVPR*, 2014.

[Lu *et al.*, 2013] J. Lu, G. Wang, and P. Moulin. Image set classification using holistic multiple order statistics features and localized multi-kernel metric learning. In *ICCV*, 2013.

[M. Rastegari J. Choi and Davis, 2013] D. Hal M. Rastegari J. Choi, S. Fakhraei and L. Davis. Predictable dual-view hashing. In *ICML*, 2013.

[Masci *et al.*, 2013] J. Masci, M. Bronstein, A. Bronstein, and J. Schmidhuber. Multimodal similarity-preserving hashing. In *ICLR*, 2013.

[Moghaddam and Shakhnarovich, 2002] B. Moghaddam and G. Shakhnarovich. Boosted dyadic kernel discriminants. In *NIPS*, 2002.

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

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

[Oliva and Torralba, 2001] A. Oliva and A. Torralba. Modeling the shape of the scene: a holistic representation of the spatial envelope. In *IJCV*, 2001.

[Rastegari *et al.*, 2012] M. Rastegari, A. Farhadi, and D. Forsyth. Attribute discovery via predictable discriminative binary codes. In *ECCV*, 2012.

[Relja and Zisserman, 2014] A. Relja and A. Zisserman. Extremely low bit-rate nearest neighbor search using a set compression tree. In *IEEE Trans. PAMI*, 2014.

[Sablayrolles *et al.*, 2017] A. Sablayrolles, M. Douze, H. Ju, and N. Usunier. How should we evaluate supervised hashing? In *ICASSP*, 2017.

[Salakhutdinov and Hinton, 2007] R. Salakhutdinov and G. E. Hinton. Learning a nonlinear embedding by preserving class neighborhood structure. In *ICAIS*, 2007.

[Salakhutdinov and Hinton, 2009] R. Salakhutdinov and G. Hinton. Semantic hashing. In *Inter'l. J. Approximate Reasoning*, 2009.

[Scholkopf *et al.*, 1997] B. Scholkopf, A. Smola, and K.-R. Muller. Kernel principle component analysis. In *ICANN*, 1997.

[Shen *et al.*, 2015] F. Shen, C. Shen, W. Liu, and H.T. Shen. Supervised discrete hashing. In *CVPR*, 2015.

[Sivic *et al.*, 2005] J. Sivic, M. Everingham, and A. Zisserman. Person spotting: video shot retrieval for face sets. In *Image and Video Retrieval*, 2005.

[Sun *et al.*, 2014] X. Sun, Q. Qu, N. M. Nasrabadi, and T. D. Tran. Structured priors for sparse-representation-based hyperspectral image classification. In *IEEE GRSL*, 2014.

[Sun *et al.*, 2015] X. Sun, N. M. Nasrabadi, and T. D. Tran. Task-driven dictionary learning for hyperspectral image classification with structured sparsity constraints. In *IEEE Trans. GRSL*, 2015.

[Sun *et al.*, 2017] X. Sun, N. M. Nasrabadi, and T. D. Tran. Supervised multilayer sparse coding networks for image classification. In *CoRR*, 2017.

[Tenenbaum *et al.*, 2000] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. In *Science*, 2000.

[Tseng, 2001] P. Tseng. Convergence of a block coordinate descent method for nondifferentiable minimization. In *J. of optimization theory and applications*, 2001.

[Tuzel *et al.*, 2007] O. Tuzel, F. Porikli, and P. Meer. Human detection via classification on riemannian manifolds. In *CVPR*, 2007.

[Vemulapalli *et al.*, 2013] R. Vemulapalli, J. K. Pillai, and R. Chellappa. Kernel learning for extrinsic classification of manifold features. In *CVPR*, 2013.

[Wang *et al.*, 2010] J. Wang, S. Kumar, and S.-F. Chang. Sequential projection learning for hashing with compact codes. In *ICML*, 2010.

[Wang *et al.*, 20114] J. Wang, H. T. Shen, J. Song, and J. Ji. Hashing for similarity search: a survey. In *arXiv:1408.2927*, 20114.

[Wang *et al.*, 2012a] J. Wang, S. Kumar, and S.-F. Chang. Semi-supervised hashing for large scale search. In *IEEE Trans. PAMI*, 2012.

[Wang *et al.*, 2012b] R. Wang, H. Guo, L. S. Davis, and Q. Dai. Covariance discriminative learning: a natural and efficient approach to image set classification. In *CVPR*, 2012.

[Wang *et al.*, 2013] X. Wang, S. Atev, J. Wright, and G. Lerman. Fast subspace search via grassmannian based hashing. In *ICCV*, 2013.

[Wang *et al.*, 2016] J. Wang, W. Liu, S. Kumar, and S.-F. Chang. Learning to hash for indexing big data - a survey. In *Proceedings of the IEEE. 104(1):34-57*, 2016.

[Weiss *et al.*, 2008] Y. Weiss, A. Torralba, and R. Fergus. Spectral hashing. In *NIPS*, 2008.

[Weng *et al.*, 2019] L. Weng, I.-H. Jhuo, and W.-H. Cheng. Perceptual hashing for large-scale multimedia search. In *John Wiley Sons*, 2019.

[Zhang and Li, 2014] D. Zhang and W.-J. Li. Large-scale supervised multimodal hashing with semantic correlation maximization. In *AAAI*, 2014.

[Zhou *et al.*, 2009] Z.H. Zhou, Y.-Y. Sun, and Y.-F. Li. Multi-instance learning by treating instances as non-i.i.d. samples. In *ICML*, 2009.