EFFICIENT SIMILARITY LEARNING FOR ASYMMETRIC HASHING

Cheng Da[1, 2], *Yang Yang*[1], *Kun Ding*[1, 2], *Chunlei Huo*[1], *Shiming Xiang*[1, 2], *Chunhong Pan*[1]

[1]National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences
[2]School of Computer and Control Engineering, University of Chinese Academy of Sciences
{cheng.da, yang.yang, kding, clhuo, smxiang, chpan}@nlpr.ia.ac.cn

ABSTRACT

Hashing techniques with asymmetric schemes (*e.g.,* only binarizing the database points) have recently attracted wide attention in the circle of image retrieval. In comparison with those methods which binarize simultaneously both of the query and database points, they not only enjoy the storage and search efficiencies, but also provide higher accuracy. Gearing to this line, this paper proposes a metric-embedded asymmetric hashing (MEAH) that learns jointly a bilinear similarity measure and binary codes of database points in an unsupervised manner. Technically, the learned similarity measure is able to bridge the gap between the binary codes and the real-valued codes, which are represented possibly with different dimensions. What is more, this measure is capable of preserving the global structure hidden in the database. Extensive experiments on two public image benchmarks demonstrate the superiority of our approach over the several state-of-the-art unsupervised hashing methods.

***Index Terms*—** Unsupervised hashing, asymmetric hashing, bilinear similarity measure

1. INTRODUCTION

Similarity search aims to find some items whose distances are smallest to a given query item, also known as nearest neighbor search [1, 2]. Technically, similarity search is a fundamental task in some real-world applications such as image retrieval [3, 4]. However, measuring the similarity between high-dimensional data points is costly in large scale datasets. To address it, various hashing techniques [5, 6, 7, 8, 9] have been developed, which encode high-dimensional data points into compact binary ones while preserving the similarity in the original space. Based on these binary codes, the similarity can be calculated with low computation and memory costs.

In general, hashing methods can be divided into symmetric hashing and asymmetric hashing according to the encoding schemes of the query and database. For symmetric hashing, both of the query and database are embedded into binary codes by the same hash function [6, 7, 8, 10, 11, 12]. For asymmetric hashing, different strategies are utilized to encode the query and database [13, 14, 15, 16, 17]. In practice, the tricks with asymmetric hashing have been proven to help improve the performance of retrieval [13].

Among the existing asymmetric hashing methods, there are several types of asymmetric structures. Typically, in SDH [15] and COSDISH [16], explicit hash functions are learned for query, while the binary database points can be directly obtained unrelated to this hash function in training stage. Neyshabur *et al.* [13] learned two distinct hash functions to generate binary codes and demonstrated the power of asymmetry theoretically and experimentally. All the above methods are using binary codes of the query and database. However, Gordo *et al.* [18] argued that binarizing the database points without the query can provide higher accuracy and also enjoy the storage and search efficiencies. For this reason, two asymmetric distances are proposed to measure the similarities between the real-valued codes and binary codes in [18].

Inspired by the above-mentioned asymmetric schemes, we adopt the real-valued query and two distinct hash functions [13, 14]. Among them, two distinct hash functions are employed to generate the real-valued and binary codes, respectively. Since real-valued and binary codes belong to different distribution spaces, it may be inappropriate to directly compute the similarities between asymmetric codes via commonly used distance functions, *e.g.,* Euclidean distance. Accordingly, a bilinear similarity measure is presented to achieve this goal. With the learned bilinear similarity measure, the gap between the binary and real-valued codes can be bridged. Meanwhile, we minimize the fitting error between the learned similarities and the affinities measured in a dimension-reduced space. As such, the global structure in the database can also be preserved.

In consideration of the difficulty in gaining semantic labels in practical applications [6, 8, 10, 19, 20, 21, 22], our approach is learned in an unsupervised manner. Extensive experimental results demonstrate that our approach outperforms the state-of-the-art methods.

2. METRIC EMBEDDED ASYMMETRIC HASHING

2.1. Problem formulation

Given N data points $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_N] \in \mathbb{R}^{D\times N}$, where $\mathbf{x}_i, i = 1, ..., N$ is a D-dimensional vector. In our approach, each data point $\mathbf{x} \in \mathbb{R}^{D\times 1}$ (subscript omitted) is further transformed to a binary code and a real-valued code. For the binary code, a binary hash function $g(\mathbf{x}) = \text{sign}(\mathbf{W}^T\mathbf{x})$ is learned to embed $\mathbf{x}$ into K-bits binary code $\mathbf{b} \in \{-1,1\}^{K\times 1}$, where $\mathbf{W} \in \mathbb{R}^{D\times K}$ is a projection matrix. As for the real-valued code, we apply a dimension reduction method on $\mathbf{x}$ to generate the real-valued code $\tilde{\mathbf{x}} \in \mathbb{R}^{L\times 1}, L < D$. Accordingly, $\mathbf{x}$ will be mapped into L-dimensional real-valued codes by the embedding function $h(\mathbf{x}) = \mathbf{V}^T\mathbf{x}$, where $\mathbf{V} \in \mathbb{R}^{D\times L}$ is a projection matrix [1] . Note that since a real-valued code conveys more information than a binary one with the same length, the optimal dimension of $\tilde{\mathbf{x}}$ (dimension-reduced $\mathbf{x}$) will not be higher than that of $\mathbf{b}$, i.e., $L \le K$.

Furthermore, the similarity between two data points in the original space should be preserved in terms of the learned corresponding binary codes and real-valued codes. Although the similarity in the original space can be defined by many distance functions (*e.g.,* Euclidean distance, cosine distance), directly utilizing these distance functions to compute the similarity between data points by using binary and real-valued codes is nontrivial. This is due to the fact the asymmetry codes often follow two data distributions and even their dimensions may be different, *e.g.,* $L < K$. To address these issues, we exploit a parametric similarity function with a bilinear form [25] to measure the similarity between the asymmetric codes. To be specific, this function is denoted by $s(\mathbf{x}_i, \mathbf{x}_j) = h(\mathbf{x}_i)^T\mathbf{M}g(\mathbf{x}_j)$, where $\mathbf{M} \in \mathbb{R}^{L\times K}$ is a matrix that aims to parameterize the bilinear similarity measure. To preserve the global structure in the original space, we minimizing the fitting error, defined by the following function:

$$\min_{\mathbf{W},\mathbf{M}} \quad \sum_{i,j=1}^{N} (s(\mathbf{x}_i, \mathbf{x}_j) - A_{ij})^2. \tag{1}$$

Where A_{ij} is the affinity between $\mathbf{x}_i$ and $\mathbf{x}_j$ in the original space. As stated in [25, 26], when there is sufficient training data, the matrix $\mathbf{M}$ is not strictly required to be positive semidefinite or symmetric. Considering that our approach is an asymmetric hashing and the bilinear similarity measure (also a metric) $s(\mathbf{x}_i, \mathbf{x}_j)$ is learned to fit the affinity matrix $\mathbf{A}$ in the original space, we name our approach as Metric Embedded Asymmetric Hashing (MEAH).

In this paper, there is no supervised information to reveal the semantic similarity. Alternatively, we adopt the commonly used unsupervised metric denoted by $A_{ij} = e^{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_F^2}{\sigma}} \in (0, 1]$, where σ is a parameter and $\sigma > 0$. Unfortunately, computing all pairwise affinities in $\mathbf{A} \in \mathbb{R}^{N\times N}$ between data points of $\mathbf{X}$ is extremely time-consuming in large scale datasets. Therefore, we compute an approximation of $\mathbf{A}$ by a product of two smaller matrices [10]. In addition, on the basis of chunklets (generated by k-means), relevant component analysis (RCA) can learn a discriminant projection space and unravel the inherent structure of the data [23, 24]. Then, based on the RCA-reduced data points $\tilde{\mathbf{X}}$, the affinity matrix $\mathbf{A}$ can be approximated by $P(\tilde{\mathbf{X}})^T Q(\tilde{\mathbf{X}})$ defined by:

$$P(\tilde{\mathbf{x}}) = [\sqrt{\frac{e^2-1}{e\sigma}} e^{-\frac{\|\tilde{\mathbf{x}}\|_F^2}{\sigma}} \tilde{\mathbf{x}}; \sqrt{\frac{e^2+1}{2e}} e^{-\frac{\|\tilde{\mathbf{x}}\|_F^2}{\sigma}}], \tag{2}$$

$$Q(\tilde{\mathbf{x}}) = [\sqrt{\frac{e^2-1}{e\sigma}} e^{-\frac{\|\tilde{\mathbf{x}}\|_F^2}{\sigma}} \tilde{\mathbf{x}}; \sqrt{\frac{e^2+1}{2e}} e^{-\frac{\|\tilde{\mathbf{x}}\|_F^2}{\sigma}}]. \tag{3}$$

With this approximation, we can avoid the $O(N^2)$ computation complexity when calculating the affinity matrix $\mathbf{A}$.

Technically, with (2) and (3) the objective function (1) can be further reformulated as follows:

$$\min_{\mathbf{W},\mathbf{M}} \quad \|\tilde{\mathbf{X}}^T\mathbf{M}\text{sign}(\mathbf{W}^T\mathbf{X}) - P(\tilde{\mathbf{X}})^T Q(\tilde{\mathbf{X}})\|_F^2. \tag{4}$$

Where $\tilde{\mathbf{X}} = \mathbf{V}^T\mathbf{X}$ and $\|\cdot\|_F$ is the Frobenius norm. Obviously, because of the discrete sign function, it is difficult to solve the above problem (generally NP hard). In order to remove the sign function, a common and effective strategy is to incorporate a regulation [15]. Then, we rewrite the objective function (4) as

$$\begin{aligned} \min_{\mathbf{W},\mathbf{M},\mathbf{B}} \quad & \|\tilde{\mathbf{X}}^T\mathbf{M}\mathbf{B} - P(\tilde{\mathbf{X}})^T Q(\tilde{\mathbf{X}})\|_F^2 \\ & + \lambda\|\mathbf{W}^T\mathbf{X} - \mathbf{B}\|_F^2 \\ \text{s.t.} \quad & \mathbf{B} \in \{-1,1\}^{K\times N}, \end{aligned} \tag{5}$$

where λ is a penalty parameter, balancing the fitting error and the quantization loss. The first term in (5) is a fitting error. By minimizing the fitting error, we can learn a bilinear similarity measure (parameterized by $\mathbf{M}$) to reveal the global structure among the database. Meanwhile, the second term in (5) is employed to the guarantee that the hash function is able to learn binary codes with the minimum quantization loss.

In testing, we can calculate the similarities between one query and all data points by $(\mathbf{V}^T\mathbf{q})^T\mathbf{M}\mathbf{B}$, where $\mathbf{q} \in \mathbb{R}^{D\times 1}$ is a given query. In this way, the retrieval results are returned by sorting these similarities.

2.2. Optimization

Note that problem (5) is a non-convex problem with $\mathbf{W}, \mathbf{M}, \mathbf{B}$ together. Accordingly, we choose to solve $\mathbf{W}$, $\mathbf{M}$ and $\mathbf{B}$ in an alternating fashion, i.e., optimize one variable while keeping the others fixed at each time.

In practice, we initialize the variables $(\mathbf{V}, \mathbf{B}, \mathbf{M})$ at the beginning. Specifically, $\mathbf{V}$ is obtained by performing RCA on the original data points $\mathbf{X}$. Binary codes $\mathbf{B}$ is initialized by ITQ [6]. And the matrix $\mathbf{M}$ is initialized with ones on the main diagonal and zeros elsewhere.

W-Step. When $\mathbf{M}$ and $\mathbf{B}$ are fixed, the problem (5) is a least-square regression problem. Thus, the matrix $\mathbf{W}$ has a closed-form solution:

$$\mathbf{W} = (\mathbf{X}\mathbf{X}^T)^{-1}\mathbf{X}\mathbf{B}^T. \tag{6}$$

[1]The projection matrix $\mathbf{V}$ can be learned by any dimension reduction methods, *e.g.,* (Relevant Component Analysis, RCA) [23, 24], (Principal Component Analysis, PCA). In this paper, we choose RCA.

B-Step. With $\mathbf{W}$ and $\mathbf{M}$ fixed, the problem (5) is reformulated as

$$\begin{aligned} \min_{\mathbf{B}} \quad & \|\mathbf{R}^T\mathbf{B}\|_F^2 - 2\mathrm{Tr}\,(\mathbf{B}^T\mathbf{Z}) \\ \text{s.t.} \quad & \mathbf{B} \in \{-1, 1\}^{K \times N}, \end{aligned} \tag{7}$$

where $\mathbf{R} = \mathbf{M}^T\tilde{\mathbf{X}}$, $\mathbf{Z} = \mathbf{R}\hat{\mathbf{A}} + \lambda\mathbf{W}^T\mathbf{X}$, $\hat{\mathbf{A}} = P(\tilde{\mathbf{X}})^T Q(\tilde{\mathbf{X}})$ and $\mathrm{Tr}\,(\cdot)$ is the trace norm. Obviously, solving the problem (7) is nontrivial, due to the discrete constraints. However, as mentioned in SDH [15], the single row of $\mathbf{B}$ has a closed-form solution with other rows fixed. Based on this idea, we circularly update $\mathbf{B}$ row by row. That is to say, we update only one bit for all training samples with other bits fixed each time. Specifically, the binary codes $\mathbf{B}$ are learned by discrete cyclic coordinate descent method. Suppose that $\mathbf{b}^T$ is one row of $\mathbf{B}$, and the remaining rows of $\mathbf{B}$ is represented as $\mathbf{B}'$. In a similar way, we suppose that $\mathbf{r}^T$, $\mathbf{z}^T$ is one row of $\mathbf{R}$, $\mathbf{Z}$ corresponding to $\mathbf{b}^T$. $\mathbf{R}'$ and $\mathbf{Z}'$ are the matrix of $\mathbf{R}$ and $\mathbf{Z}$ except $\mathbf{r}^T$ and $\mathbf{z}^T$, respectively.

Therefore, we obtain the following problem w.r.t. $\mathbf{b}$:

$$\begin{aligned} \min_{\mathbf{b}} \quad & (\mathbf{r}^T\mathbf{R}'^T\mathbf{B}' - \mathbf{z}^T)\mathbf{b} \\ \text{s.t.} \quad & \mathbf{b} \in \{-1, 1\}. \end{aligned} \tag{8}$$

Clearly, this problem has a closed-form solution:

$$\mathbf{b} = \mathrm{sign}\,(\mathbf{z} - \mathbf{B}'^T\mathbf{R}'\mathbf{r}). \tag{9}$$

M-Step. We update $\mathbf{M}$ with $\mathbf{W}$ and $\mathbf{B}$ fixed. Similar to $\mathbf{W}$ step, the problem (5) turns out be a least-square regression problem, and the matrix $\mathbf{M}$ has a closed-form solution

$$\mathbf{M} = (\tilde{\mathbf{X}}\tilde{\mathbf{X}}^T)^{-1}(\tilde{\mathbf{X}}\hat{\mathbf{A}}\mathbf{B}^T)(\mathbf{B}\mathbf{B}^T)^{-1}. \tag{10}$$

The proposed Metric Embedded Asymmetric Hashing (MEAH) is summarized in Algorithm 1.

Algorithm 1 Metric Embedded Asymmetric Hashing

Input: Training data points $\mathbf{X} \in \mathbb{R}^{D \times N}$; the length of binary codes K; the length of real-valued codes L; maximum iteration number t; penalty parameter λ.

Perform RCA on data points $\mathbf{X}$, obtain matrix $\mathbf{V} \in \mathbb{R}^{D \times L}$ for dimension reduction.
Calculate RCA-projected data $\tilde{\mathbf{X}} = \mathbf{V}^T\mathbf{X} \in \mathbb{R}^{L \times N}$.
Initialize $\mathbf{B} \in \{-1, 1\}^{K \times N}$ by ITQ and $\mathbf{M}$ with ones on the main diagonal and zeros elsewhere.
while not converge or iteration number $< t$ **do**
    W-Step: Update $\mathbf{W}$ using Eqn. (6)
    B-Step: Update $\mathbf{B}$ row by row using Eqn. (9).
    M-Step: Update $\mathbf{M}$ using Eqn. (10).
end while

Output: Bilinear measure matrix $\mathbf{M} \in \mathbb{R}^{L \times K}$; matrix $\mathbf{V} \in \mathbb{R}^{D \times L}$; binary codes $\mathbf{B} \in \{-1, 1\}^{K \times N}$.

3. EXPERIMENTS

3.1. Datasets and Evaluation protocol

We evaluate our approach on two commonly used image datasets: CIFAR-10 and ESP-GAME. CIFAR-10 is a single-label dataset containing 60,000 color images from 10 semantic categories. ESP-GAME is a multi-label dataset consisting of 20,768 images assigned with multiple labels from 268 categories. For both two datasets, the 512-dimensional GIST [27] features are extract to represent these images. In addition, the ground-truth is defined by semantic labels, namely two images are similar if they share at least one label. To evaluate the retrieval performance of our approach, mean average precision (MAP) and top-K precision are chosen as evaluation protocol. Naturally, 1000 samples are selected for testing, and the remaining samples for training models each time.

3.2. Compared Methods

To demonstrate the effectiveness of our proposed approach, we compare MEAH against one data-independent method LSH [5], and some representative unsupervised hashing methods [28], including ITQ [6], SH [8], AGH [19] and SGH [10]. In our experiments, two query strategies are employed for verifying the effectiveness of real-valued codes. Specifically, we search database by $(\mathbf{V}^T\mathbf{q})^T\mathbf{M}\mathbf{B}$. This real-valued query strategy is denoted by MEAH. Additionally, we can also perform query by $\mathrm{sign}\,((\mathbf{V}^T\mathbf{q})^T\mathbf{M}))\mathbf{B}$. This binary query strategy is denoted by MEAH-B.

3.3. Experimental settings and results

Unless otherwise specified, we empirically set the parameters on both two datasets as follows. In (5), λ is taken as 0.5. The maximum iteration number t is set to 10 and the reduced dimension L is set to 30 in (1). As for other compared hashing methods, we set the parameters as the suggestions of the corresponding authors. All experiments are conducted on a PC with 3.5 GHz CPU and 32 GB RAM. The results are reported based on the average of 10 random runs.

The MAP performance of different hashing methods are reported in Table 1. It can be obviously seen that MEAH obtains the best performance on both two datasets, while MEAH-B is superior to other binary hashing methods with high bits (more than 32 bits) in terms of MAP . In comparison with MEAH-B, MEAH shows better search accuracy. For ex-

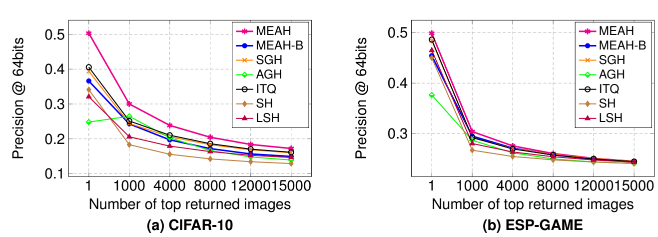


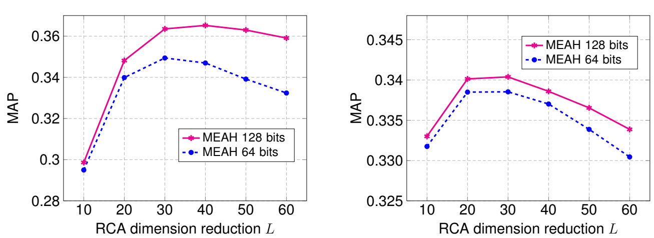
Fig. 1. The top-K precision of different numbers of top returned images on (a) CIFRA-10 and (b) ESP-GMAE.

Table 1. The MAP of different hashing methods are compared on two datasets. The best results are reported in boldface.

Method	CIFRA-10				ESP-GAME			
	32 bits	64 bits	96 bits	128 bits	32bits	64 bits	96 bits	128 bits
MEAH	**0.3120**	**0.3494**	**0.3592**	**0.3630**	**0.3325**	**0.3385**	**0.3398**	**0.3404**
MEAH-B	0.2614	0.3127	0.3302	0.3432	0.3108	0.3252	0.3313	0.3334
SGH	0.2669	0.2906	0.2997	0.3047	0.3125	0.3208	0.3243	0.3257
AGH	0.2941	0.3041	0.3083	0.3105	0.3055	0.3142	0.3168	0.3176
ITQ	0.2757	0.2940	0.3023	0.3082	0.3175	0.3233	0.3259	0.3278
SH	0.2232	0.2285	0.2264	0.2297	0.2912	0.2893	0.2914	0.2895
LSH	0.2077	0.2405	0.2548	0.2670	0.2927	0.3040	0.3115	0.3151

ample, MEAH outperforms MEAH-B by 11.74% and 4.09% with 64 bits on CIFRA-10 and ESP-GMAE, respectively. This observation indicates that utilizing the real-valued codes rather than binary ones can obtain remarkable improvements. Meanwhile, SGH yields favorable performance among the compared methods. It should be also noted that the framework of MEAH-B is similar to SGH. Benefited by asymmetric scheme, our MEAH-B achieves better performance than SGH. Compared to SGH with 64 bits, the MAP of MEAH is clearly higher than SGH with a large margin (over 20.23%) on CIFAR-10. Additionally, on ESP-GAME, ITQ is the second best method which is slightly better than SGH, but worse than MEAH. This shows that MEAH can also deal with multi-label datasets and achieve the best performance.

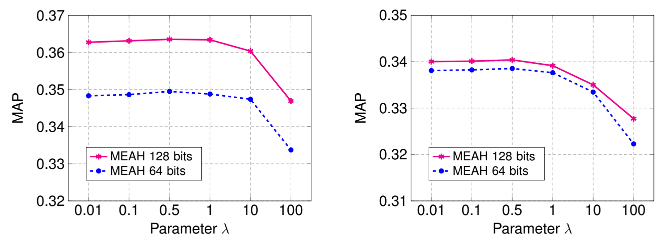
Simultaneously, the top-K precision with 64 bits is illustrated in Figure 1. On CIFAR-10, MEAH surpass other methods with a obvious margin under different number of returned images. While on ESP-GAME, the top-K precision of MEAH is a little higher than other methods (also the best). Consequently, experimental results demonstrate the best search performance of MEAH in terms of MAP and top-K precision.



(a) CIFAR-10 (b) ESP-GAME

Fig. 2. The effect of different RCA-reduced dimensions on MAP performance on (a) CIFRA-10 and (b) ESP-GMAE.

To evaluate the effect of RCA-reduced dimensions, the MAP performance of different values of L is shown in Figure 2. We can see that MAP is sensitive to the RCA dimension reduction. Concretely, when L is small between 20 and 40, MEAH remains a stably promising performance. However, when L becomes smaller or larger, the performance of MEAH degrades significantly. This observation is consistent with our intention that moderately low-dimensional real-valued codes convey sufficient information for representation. Hence, the dimension of two datasets is reduced to 30 in MEAH.



(a) CIFAR-10 (b) ESP-GAME

Fig. 3. The effect of λ on MAP performance on (a) CIFRA-10 and (b) ESP-GMAE.

Figure 3 illustrates the MAP of different parameter λs. Clearly, the MAP of MEAH is not sensitive to λ when $0.1 < \lambda < 1$. While the performance becomes worse in the case of $\lambda \geqslant 1$. The underlying reason is that MEAH mainly focus on minimizing the fitting error, while the quantization error is only a regularization term. Thus, the fitting error term should be more critical than the quantization error term. For this reason, we set the tradeoff parameter λ to 0.5 on two datasets.

4. CONCLUSION

In this paper, we have proposed a novel unsupervised asymmetric hashing method named MEAH, which exploits an asymmetric structure based on binary and real-valued codes. Different from previous binary hashing methods which measure the similarity in Hamming space or Euclidean space, MEAH learns a bilinear similarity function to directly measure the similarity between these asymmetric codes. The learned bilinear similarity measure can bridge the gap between the binary and real-valued codes and also preserve the global structure in the database. By leveraging the underlying information of the real-valued codes, the more precise similarity can be calculated in the query stage. Experimental results show that MEAH achieves superior performance over several state-of-the-art binary hashing methods.

Acknowledgments. This work was supported by the National Natural Science Foundation of China under Grants 91646207, 91438105, 61375024, and 91338202, and the Beijing Natural Science Foundation under Grant (4162064).

5. REFERENCES

[1] Jingdong Wang, Heng Tao Shen, Jingkuan Song, and Jianqiu Ji, "Hashing for similarity search: A survey," *arXiv preprint, arXiv:1408.2927*, pp. 1–29, 2014.

[2] Jun Wang, Wei Liu, Sanjiv Kumar, and Shih-Fu Chang, "Learning to hash for indexing big data - A survey," *Proceedings of the IEEE*, vol. 104, no. 1, pp. 34–57, 2016.

[3] Jun Wang, Ondrej Kumar, and Shih-Fu Chang, "Semi-supervised hashing for scalable image retrieval," in *CVPR*, 2010, pp. 3424–3431.

[4] Rongkai Xia, Yan Pan, Hanjiang Lai, Cong Liu, and Shuicheng Yan, "Supervised hashing for image retrieval via image representation learning," in *AAAI*, 2014, pp. 2156–2162.

[5] Aristides Gionis, Piotr Indyk, and Rajeev Motwani, "Similarity search in high dimensions via hashing," in *VLDB*, 1999, pp. 518–529.

[6] Yunchao Gong, Svetlana Lazebnik, Albert Gordo, and Florent Perronnin, "Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval," *TPAMI*, vol. 35, no. 12, pp. 2916–2929, 2013.

[7] Wei Liu, Jun Wang, Rongrong Ji, Yu-Gang Jiang, and Shih-Fu Chang, "Supervised hashing with kernels," in *CVPR*, 2012, pp. 2074–2081.

[8] Yair Weiss, Antonio Torralba, and Robert Fergus, "Spectral hashing," in *NIPS*, 2008, pp. 1753–1760.

[9] Fumin Shen, Chunhua Shen, Qinfeng Shi, Anton van den Hengel, Zhenmin Tang, and Heng Tao Shen, "Hashing on nonlinear manifolds," *TIP*, vol. 24, no. 6, pp. 1839–1851, 2015.

[10] Qing-Yuan Jiang and Wu-Jun Li, "Scalable graph hashing with feature transformation," in *IJCAI*, 2015, pp. 2248–2254.

[11] Guosheng Lin, Chunhua Shen, and Anton van den Hengel, "Supervised hashing using graph cuts and boosted decision trees," *TPAMI*, vol. 37, no. 11, pp. 2317–2331, 2015.

[12] Kun Ding, Chunlei Huo, Bin Fan, and Chunhong Pan, "knn hashing with factorized neighborhood representation," in *ICCV*, 2015, pp. 1098–1106.

[13] Behnam Neyshabur, Nati Srebro, Ruslan Salakhutdinov, Yury Makarychev, and Payman Yadollahpour, "The power of asymmetry in binary hashing," in *NIPS*, 2013, pp. 2823–2831.

[14] Fumin Shen, Wei Liu, Shaoting Zhang, Yang Yang, and Heng Tao Shen, "Learning binary codes for maximum inner product search," in *ICCV*, 2015, pp. 4148–4156.

[15] Fumin Shen, Chunhua Shen, Wei Liu, and Heng Tao Shen, "Supervised discrete hashing," in *CVPR*, 2015, pp. 37–45.

[16] Wang-Cheng Kang, Wu-Jun Li, and Zhi-Hua Zhou, "Column sampling based discrete supervised hashing," in *AAAI*, 2016, pp. 1230–1236.

[17] Cheng Da, Shibiao Xu, Kun Ding, Gaofeng Meng, Shiming Xiang, and Chunhong Pan, "AMVH: Asymmetric multi-valued hashing," in *CVPR*, 2017.

[18] Albert Gordo, Florent Perronnin, Yunchao Gong, and Svetlana Lazebnik, "Asymmetric distances for binary embeddings," *TPAMI*, vol. 36, no. 1, pp. 33–47, 2014.

[19] Wei Liu, Jun Wang, Sanjiv Kumar, and Shih-Fu Chang, "Hashing with graphs," in *ICML*, 2011, pp. 1–8.

[20] Wei Liu, Cun Mu, Sanjiv Kumar, and Shih-Fu Chang, "Discrete graph hashing," in *NIPS*, 2014, pp. 3419–3427.

[21] Weihao Kong and Wu-Jun Li, "Isotropic hashing," in *NIPS*, 2012, pp. 1655–1663.

[22] Jae-Pil Heo, Youngwoon Lee, Junfeng He, Shih-Fu Chang, and Sung-Eui Yoon, "Spherical hashing," in *CVPR*, 2012, pp. 2957–2964.

[23] Aharon Bar-Hillel, Tomer Hertz, Noam Shental, and Daphna Weinshall, "Learning distance functions using equivalence relations," in *ICML*, 2003, pp. 11–18.

[24] Noam Shental, Tomer Hertz, Daphna Weinshall, and Misha Pavel, "Adjustment learning and relevant component analysis," in *ECCV*, 2002, pp. 776–792.

[25] Gal Chechik, Varun Sharma, Uri Shalit, and Samy Bengio, "Large scale online learning of image similarity through ranking," *JMLR*, vol. 11, pp. 1109–1135, 2010.

[26] Yang Yang, Shengcai Liao, Zhen Lei, and Stan Z. Li, "Large scale similarity learning using similar pairs for person verification," in *AAAI*, 2016, pp. 3655–3661.

[27] Aude Oliva and Antonio Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *IJCV*, vol. 42, no. 3, pp. 145–175, 2001.

[28] Deng Cai, "A revisit of hashing algorithms for approximate nearest neighbor search," *arXiv preprint, arXiv:1612.07545*, pp. 1–11, 2016.