

Multi-kernel Hashing with Semantic Correlation Maximization for Cross-Modal Retrieval

Guangfei Yang, Huanghui Miao, Jun Tang, Dong Liang, and Nian Wang^(✉)

School of Electronics and Information Engineering, Anhui University, Hefei, China
yanggfahu@163.com, miaohhahu@163.com, tangjunahu@163.com,
dliang@ahu.edu.cn, wangnianahu@126.com

Abstract. Cross-modal hashing aims to facilitate approximate nearest neighbor search by embedding multimedia data represented in high-dimensional space into a common low-dimensional Hamming space, which serves as a key part in multimedia retrieval. In recent years, kernel-based hashing methods have achieved impressive success in cross-modal hashing. Enlightened by this, we present a novel multiple kernel hashing method, where hash functions are learned in the kernelized space using a sequential optimization strategy. Experimental results on two benchmark datasets verify that the proposed method significantly outperforms some state-of-the-art methods.

Keywords: Cross-modal hashing · Multimedia retrieval
Kernel-based hashing · Multiple kernels

1 Introduction

Nowadays, the amount of multimedia data grows explosively with the rapid development of information technology, consequently making the hashing based approximate nearest neighbor (ANN) search technique in great demand. The basic idea of hashing methods is to embed original high-dimensional data into compact binary codes, which can lead to fast computation of Hamming distances by hardware accelerated bit-wise XOR operation.

Most previous hashing methods focused on single-modal data. One of the most well-known work is locality sensitive hashing (LSH) [3], which projects data samples from original feature space to Hamming feature space while preserving their similarity as much as possible. To achieve better retrieval performance, various extensions of LSH were proposed to design more compact hashing, such as PCA based hashing [14], manifold learning based hashing [7], and kernel learning based hashing [5, 6, 13, 19]. Spectral Hashing (SH) [16] generates hash codes by thresholding a subset of eigenvectors of graph Laplacian constructed on data samples. Co-Regularized Hashing (CRH) [22] presents a boosted co-regularized framework to learn hashing functions for each bit. Supervised Hashing with Pseudo Labels (SHPL) [10] uses the cluster centers of training data to generate pseudo labels, which is utilized to enhance the discrimination of hash codes.

Supervised Hashing with Kernels (KSH) [6] try to construct the hash functions by optimizing the code inner products.

These aforementioned methods are only applicable for single modality. However, most data in real-world applications are in the form of multiple modalities. For instance, a web page may contain both images and text and a YouTube video often has relevant tags. Consequently, more and more research interest has been devoted to cross-modal hashing. CMFH [2] was among the first to learn cross-modal hash functions using collective matrix factorization, and it aims to generate unified hash codes for each instance. In [20], Zhang and Li proposed an algorithm with linear-time complexity to learn hash functions, which can be used for large-scale data. Quantized Correlation Hashing (QCH) [17] aims to jointly learn binary codes learning and minimize the quantization loss. Kernelized Cross-Modal Hashing for Multimedia Retrieval (KCH) [11] maps data from different modalities into a common kernel space by canonical correlation analysis. Notably, multi-kernel learning has emerged as an effective approach to cross-modal hashing, as the utilization of multiple kernels can explore the complementary property of each single kernel. In [24], Zhou et al. proposed a kernelized cross-modal hashing algorithm embedded in boosting framework, but it only utilizes single kernel. Boosting Multi-kernel Locality-Sensitive Hashing (BMKLSH) [18] uses multi-kernel learning to produce hash codes, and the experimental results show its superiority over KLSH [5] based on single kernel learning.

Motivated by the great success of multi-kernel learning, we propose a supervised cross-modal hashing approach based on multi-kernel learning, which is named Multiple Kernel with Semantic Correlation Hashing (MKSH). Unlike the existing single-kernel methods [4, 6, 21, 23], we aim to learn multi-kernel hash functions. Moreover, differing from the existing multi-kernel hashing approaches [15] that assign the same weight to each kernel in a brute-force way, we utilize an alternated optimization strategy to simultaneously learn the kernel combination coefficients and hash functions that can lead to higher retrieval accuracy. Our contributions are summarized as follows:

- We propose a novel cross-modal hashing algorithm utilizing multi-kernel learning.
- In order to find the optimal allocation of different kernels, we propose an iterative method to solve the objective function.
- To further enhance the algorithm performance, we utilize a sequential strategy to learn hash functions.

2 Proposed Algorithm

In this section, we detail the procedure of our hashing approach. Let $O = \{o_i\}_{i=1}^n$ denote a set of multi-view samples and $\mathbf{X} = \{x_i\}_{i=1}^n$, $\mathbf{Y} = \{y_j\}_{j=1}^n$ represent two different views of O , where $\mathbf{X} \in \mathbb{R}^{d_x}$ and $\mathbf{Y} \in \mathbb{R}^{d_y}$. The goal of MKSH is to learn two hash functions for each modality respectively: $f(x) = [f_{(1)}(x), f_{(2)}(x), \dots, f_{(k)}(x)] : \mathbb{R}^{d_x} \rightarrow \{-1, 1\}^k$ and $g(y) =$

$[g_{(1)}(y), g_{(2)}(y), \dots, g_{(k)}(y)] : \mathfrak{R}^{d_y} \rightarrow \{-1, 1\}^k$, where k denotes the length of hash codes.

2.1 Learning Hash Functions

We use multiple kernels to define the mapping function in each modality as:

$$\begin{cases} K(x_i) = [\mu_1 K_1(x_i^{(1)}) + \mu_2 K_2(x_i^{(2)}) \dots \mu_M K_M(x_i^{(M)})] \\ K(y_j) = [\mu_1 K_1(y_j^{(1)}) + \mu_2 K_2(y_j^{(2)}) \dots \mu_M K_M(y_j^{(M)})] \end{cases} \quad (1)$$

where M indicates the number of kernels, and $K_M(x_i^{(M)})$ is defined as $K_M(x_i^{(M)}) = k_M(\bar{x}_i, x_j)$ ($K_M(y_j^{(M)}) = k_M(\bar{y}_i, y_j)$), and $\bar{x} \in \mathbf{X}$ ($\bar{y} \in \mathbf{Y}$) are landmarks. We can use clustering methods to obtain landmarks. Then we define a prediction function with kernel as follows:

$$p(x) = \sum_{j=1}^m K(x_j) w_j - b \quad (2)$$

where m is the number of landmarks, and $b \in \mathfrak{R}$ is the bias, $w_i \in \mathfrak{R}$ is the coefficient. As a fast alternative to the median, following [6], we set $b = \frac{1}{n} \sum_{i=1}^n \sum_{j=1}^m K(x_j) w_j$. Then we have:

$$\begin{aligned} p(x) &= \sum_{j=1}^m \left(K(x_j) - \frac{1}{n} \sum_{i=1}^n K(x_j) \right) w_j \\ &= \mathbf{W}^T K(x). \end{aligned} \quad (3)$$

The hashing functions are defined as follows:

$$\begin{cases} f(x) = \text{sgn} \left(\mathbf{W}_x^T K(x) \right) \\ g(y) = \text{sgn} \left(\mathbf{W}_y^T K(y) \right) \end{cases} \quad (4)$$

where $\text{sgn}(u)$ is set to 1 if $u > 0$, otherwise -1 , and $\mathbf{W}_x \in \mathfrak{R}^{d_x \times k}$ represent the projection matrices. We utilize the cosine similarity between the semantic label vectors to construct the pairwise semantic similarity $\tilde{\mathbf{S}}_{ij}$, where $\tilde{\mathbf{S}}_{ij} = (l_i \cdot l_j) / (\|l_i\|_2 \|l_j\|_2)$, l_i and l_j are label vectors. We also use L to store label information, with $L_{ij} = l_{i,j} / \|l_i\|_2$, where L_{ij} denotes the element at the i th row and the j th column in the matrix \mathbf{L} , then we write $\tilde{\mathbf{S}}_{ij} = \mathbf{L} * \mathbf{L}^T$, finally, we perform element wise linear transformation on $\tilde{\mathbf{S}}_{ij}$ to get semantic similarity matrix \mathbf{S}_{ij} as follows:

$$\mathbf{S}_{ij} = 2\mathbf{L} * \mathbf{L}^T - \mathbf{1}_n \mathbf{1}_n^T. \quad (5)$$

where $\mathbf{S}_{ij} \in [-1, 1]$ is the semantic similarity matrix, and $\mathbf{1}_n$ is an all-one column vector. Then we define the objective function minimizing the squared error as follows:

$$\min_{f,g} \sum_{i,j} (f(x_i)^T g(y_j) - \mathbf{S}_{ij})^2 \quad (6)$$

Eq. (6) can be rewritten as:

$$\min_{\mathbf{W}_x, \mathbf{W}_y} \left\| \text{sgn}(K(x)\mathbf{W}_x) \text{sgn}(K(y)\mathbf{W}_y)^T - \mathbf{S}_{ij} \right\|_F^2. \quad (7)$$

2.2 Learning Projection Matrices

The problem described in Eq. (7) is NP hard. However, we can use spectral relaxation to obtain a close-formed solution. We rewrite Eq. 7 as follows:

$$\begin{aligned} \min_{\mathbf{W}_x, \mathbf{W}_y} & \left\| K(x)\mathbf{W}_x (K(y)\mathbf{W}_y)^T - \mathbf{S}_{ij} \right\|_F^2 \\ \text{s.t.} & \begin{cases} \mathbf{W}_x^T K(x)^T K(x)\mathbf{W}_x = n\mathbf{I}_c \\ \mathbf{W}_y^T K(y)^T K(y)\mathbf{W}_y = n\mathbf{I}_c \end{cases} \end{aligned} \quad (8)$$

Removing the constant, then we have:

$$\begin{aligned} \max_{\mathbf{W}_x, \mathbf{W}_y} & \text{tr} \left(\mathbf{W}_x^T K(x)^T \mathbf{S}_{ij} K(y)\mathbf{W}_y \right) \\ \text{s.t.} & \begin{cases} \mathbf{W}_x^T K(x)^T K(x)\mathbf{W}_x = n\mathbf{I}_c \\ \mathbf{W}_y^T K(y)^T K(y)\mathbf{W}_y = n\mathbf{I}_c \end{cases} \end{aligned} \quad (9)$$

In Eq. (9), \mathbf{I}_c denotes an identity matrix of size $c \times c$, the term $K(x)^T \mathbf{S} K(y)$ can be regarded as to weigh the relationship between two different modalities. If we define $C_{xy} = K(x)^T \mathbf{S} K(y)$ and $C_{xx} = K(x)^T \mathbf{S} K(x)$ and $C_{yy} = K(y)^T \mathbf{S} K(y)$, then the problem (9) can be viewed as a generalized eigenvalue problem. Consequently, we can get the optimal value of W_x and W_y by eigen-decomposition.

Some literatures have experimentally verified that orthogonal constraints are helpless to produce discriminative hash codes [14]. Following the idea in [20], we turn to use a sequential optimization strategy to learn hash functions. Suppose that the latter projection is related to the former, we solve hashing functions by defining a residue. The residue matrix \mathbf{V}_t is denoted by:

$$\mathbf{V}_t = \mathbf{S} - \sum_{k=1}^{t-1} \text{sgn} \left(K(x)\mathbf{W}_x^{(k)} \right) \text{sgn} \left(K(y)\mathbf{W}_y^{(k)} \right)^T \quad (10)$$

Algorithm 1. Sequential Learning Algorithm for MKSH

Require: The kernel matrices in different views $K_m(x), K_m(y), m = 1, \dots, M$ and the similarity matrix \mathbf{S} .

Ensure: The kernel weights vector μ , the projection matrices \mathbf{W}_x and \mathbf{W}_y .

- 1: Initialize μ with equal weights, and utilize Eq. (1) to calculate the initial kernel matrices $K(x)$ and $K(y)$.
- 2: **repeat**
- 3: Update $\mathbf{W}_x, \mathbf{W}_y$ by Eq. (11).
- 4: Update μ by Eq. (12).
- 5: Update $K(x)$ and $K(y)$ according to Eq. (1).
- 6: **until** convergence.
- 7: $f(x) = \text{sgn}(\mathbf{W}_x^T K(x)), g(y) = \text{sgn}(\mathbf{W}_y^T K(y))$.

Then \mathbf{C}_{xy} can be computed by:

$$\begin{aligned} \mathbf{C}_{xy}^t &= K(x)^T \mathbf{V}_t K(y) \\ &= K(x)^T \mathbf{S} K(y) - \sum_{k=1}^{t-1} K(x)^T \text{sgn}\left(K(x) \mathbf{W}_x^{(k)}\right) \text{sgn}\left(K(y) \mathbf{W}_y^{(k)}\right)^T K(y) \\ &= \mathbf{C}_{xy}^{(t-1)} - K(x)^T \text{sgn}\left(K(x) \mathbf{W}_x^{(t-1)}\right) \text{sgn}\left(K(y) \mathbf{W}_y^{(t-1)}\right)^T K(y) \end{aligned}$$

We rewrite Eq. (8) as follows:

$$\max_{\mathbf{W}_x, \mathbf{W}_y} \left\| \left(K(x) \mathbf{W}_x^{(t)} \right) \left(K(y) \mathbf{W}_y^{(t)} \right)^T - \mathbf{V}_t \right\|_F^2 \quad (11)$$

Once the optimal value of Eq. (11) is obtained we can get the projections of two modalities \mathbf{W}_x and \mathbf{W}_y .

2.3 Optimizing the Weights of Multiple Kernels

The objective function is written as:

$$\begin{aligned} \mathbb{L}(\mathbf{S}, \mathbf{W}_x, \mathbf{W}_y, \mu) &= \frac{1}{2} \mu^T F \mu \\ \text{s.t.} \quad &\sum_{m=1}^M \mu_m = 1, \mu_m \geq 0 \end{aligned} \quad (12)$$

where $F = \text{tr}\left(\mathbf{W}_x^T K(x)^T \mathbf{S} K(y) \mathbf{W}_y\right)$. If \mathbf{W}_x and \mathbf{W}_y are available, Eq. (12) can be regarded as a quadratic programming problem.

The overall algorithm is summarized in Algorithm 1.

3 Experiments

In this section, we conduct experiments on two benchmark datasets to verify the effectiveness of our approach.

3.1 Datasets

The used datasets are the Wiki dataset [8] and the NUS-WIDE dataset [1].

The Wiki dataset contains 2866 image-text pairs. Each image is represented by a SIFT feature vector with 1000-dimensional Bag-of-Visual-Words SIFT histogram, and each text is represented by an index vector of the top 5000 most frequent tags. There are 10 categories in the Wiki dataset.

The NUS-WIDE dataset contains 269648 images collected from Flickr. Following the experimental protocol in [12], we choose a subset comprising the most frequently-used 10 classes. Each image is represented by a 500-dimensional bag-of-visual-words SIFT histogram, and each text is represented by a Bag-of-Words feature vector with top 1000 most frequent tags. In the subset, we randomly choose 5000 image-tag pairs as the training set, and randomly choose 1866 image-text pairs from the remaining documents as the test set. Table 1 shows the details of the evaluated datasets in our experiments.

Table 1. The details of the evaluated datasets

Dataset	Wiki	NUS-WIDE
Image modality	BoVW(1000-D)	BoVW(500-D)
Text modality	BoW(5000-D)	BoW(1000-D)
Dataset size	2866	186577
Training set size	2173	5000
Testing set size	693	1866
Num. of categories	10	10

3.2 Experimental Setup

We perform two cross-modal retrieval tasks on the NUS-WIDE and the Wiki datasets respectively, i.e., ‘img to text’ and ‘text to img’. We compare MKSH to six state-of-the-art cross-modal hashing methods, i.e., LCMH [25], LSSH [23], SCM-Seq [20], CMFH [2], RCMH [9], and KSH-CV [24]. We employ the mean Average Precision (mAP) to evaluate the retrieval performance. The average precision is defined as: $AP = \frac{1}{N} \sum_{i=1}^R P(i) \times \delta(i)$, where $P(i)$ means the retrieval accuracy of top i retrieved documents, and $\delta(i)$ is an indicator function, if the i -th rank is a relevant instance, $\delta(i) = 1$, otherwise $\delta(i) = 0$. N is the number of relevant instances in the training set.

In our experiment, we choose the Gaussian RBF kernel $K(x, y) = \exp(-\frac{\|x-y\|^2}{2\epsilon^2})$, the sigmoid kernel $K(x, y) = \tanh(\alpha xy + c)$ and the exponential kernel $K(x, y) = \exp(-\frac{\|x-y\|}{2\lambda^2})$ as kernel functions, and set $R = 50$.

3.3 Experimental Results

We compare the mAP values of all the methods on the Wiki and NUS-WIDE datasets, and the code length ranges from 16 to 64. The detailed results are reported in Tables 2 and 3. Figure 1 shows the precision-recall curves of two query tasks on the Wiki dataset. We also compare the performance of our method using multiple kernels and single kernel respectively, and the results are plotted in Figs. 2 and 3.

Table 2. mAP results on Wiki dataset.

Task	Methods	Hash code length			
		16 bits	32 bits	48 bits	64 bits
Img to text	LCMH	0.1166	0.1354	0.1440	0.1489
	LSSH	0.1959	0.1801	0.1854	0.1821
	SCM-Seq	0.1837	0.1902	0.1751	0.1880
	CMFH	0.1627	0.1627	0.1626	0.1635
	RCMH	0.1584	0.1600	0.1603	0.1670
	KSH-CV	0.1674	0.1765	0.1718	0.1717
	MKSH	0.2192	0.2125	0.2249	0.2161
Text to img	LCMH	0.1296	0.1365	0.1378	0.1457
	LSSH	0.1625	0.1584	0.1524	0.1609
	SCM-Seq	0.1711	0.1708	0.1673	0.1705
	CMFH	0.1577	0.1580	0.1560	0.1561
	RCMH	0.1535	0.1572	0.1618	0.1571
	KSH-CV	0.1923	0.1901	0.1885	0.1825
	MKSH	0.2048	0.2079	0.2184	0.2348

We can draw two conclusions from the aforementioned experimental results. Firstly, MKSH outperforms the alternatives, which shows its superiority over the compared methods. Secondly, MKSH shows its consistent advantage when the length of hash codes become longer, which can be owed to its sequential optimization strategy.

From Fig. 1 we also have two observations. Firstly, MKSH outperforms the compared methods. Secondly, we can find that RCMH and LCMH are not applicable for large-scale cross-modal retrieval due to their poor performance.

3.4 Parameters Sensitivity Study

According to our experimental study, the four parameters, including ε , α , c and λ , have a slight influence on the performance, so we set $\varepsilon = 0.6$, $\alpha = 9$,

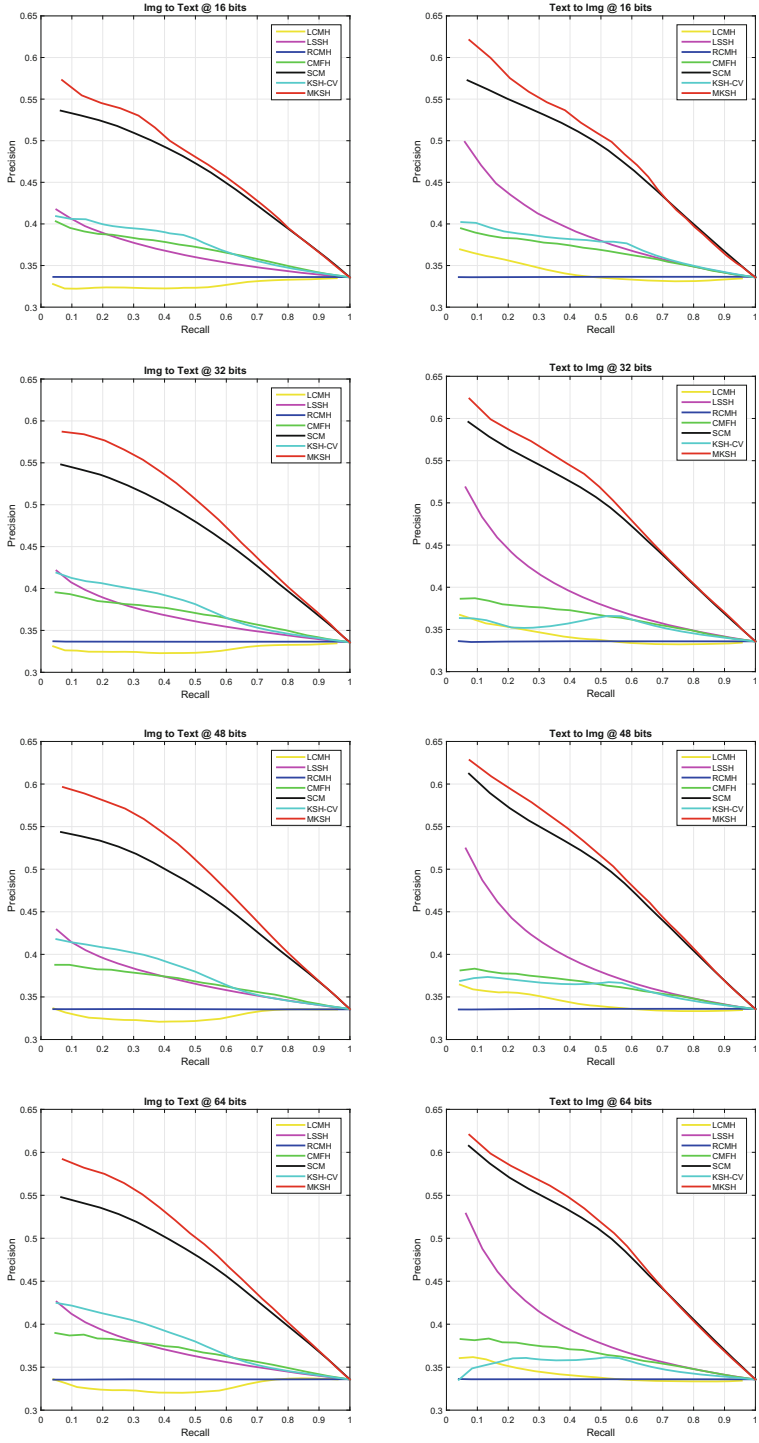
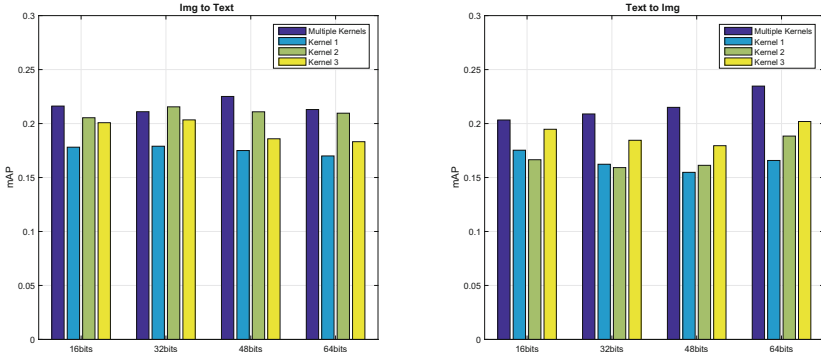


Fig. 1. PR-curves on the NUS-WIDE dataset varying code length

Table 3. mAP results on NUS-WIDE dataset.

Task	Methods	Hash code length			
		16bits	32bits	48bits	64bits
Img to text	LCMH	0.3376	0.3763	0.3783	0.3745
	LSSH	0.4746	0.4920	0.4903	0.4915
	SCM-Seq	0.5739	0.5941	0.5920	0.5962
	CMFH	0.4741	0.4886	0.4871	0.4853
	RCMH	0.3817	0.3820	0.3847	0.3817
	KSH-CV	0.4609	0.4678	0.4648	0.4692
	MKSH	0.6075	0.6303	0.6537	0.6487
Text to img	LCMH	0.4108	0.4183	0.3964	0.3877
	LSSH	0.5827	0.6142	0.6125	0.6119
	SCM-Seq	0.6159	0.6501	0.6668	0.6658
	CMFH	0.4636	0.4781	0.4839	0.4884
	RCMH	0.3822	0.3835	0.3806	0.3807
	KSH-CV	0.4220	0.4177	0.4111	0.4022
	MKSH	0.6890	0.7003	0.7135	0.6684

**Fig. 2.** Compare mAP on multiple kernels and single kernels (Wiki)

$c = -0.1$ and $\lambda = 0.8$. The generation of the kernel matrix depends on the number of landmarks. Figure 4 shows the performance when varying the number of landmarks on the WIKI and NUS-WIDE datasets respectively. We can observe that the precision almost remain the same with the variation of the number of landmarks. Therefore, we can learn that the number of landmarks is not a sensitive parameter.

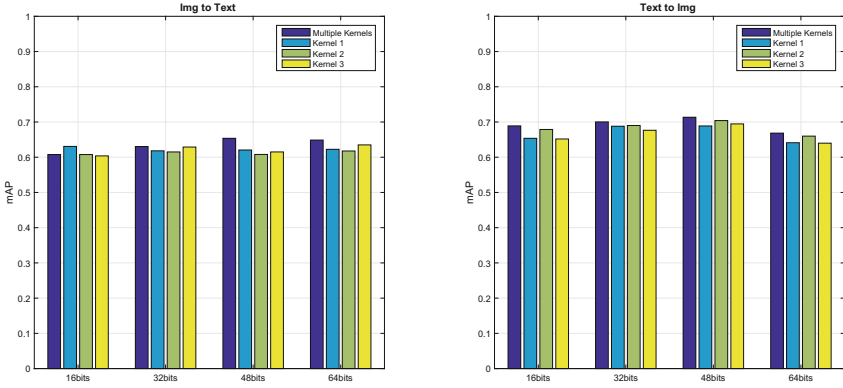


Fig. 3. Compare mAP on multiple kernels and single kernels (NUS-WIDE)

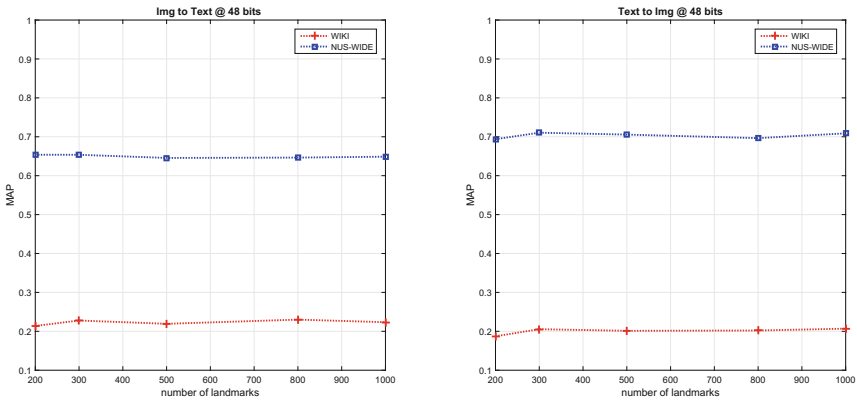


Fig. 4. The effect of landmarks on MKSH

4 Conclusions

In this paper, we have proposed a novel algorithm for cross-modal hashing named MKSH. Multi-kernel learning and a sequential optimization strategy are used to achieve better performance. Experimental results on the Wiki and the NUS-WIDE datasets show that our method outperforms several state-of-the-art methods.

Acknowledgements. This work is supported by the Key Projects of Outstanding Youth Talent Support Program of Anhui Provincial Universities under Grant gxyqZD2016012, and the Natural Science Foundation of China under Grant 61672032.

References

1. Chua, T.S., Tang, J., Hong, R., Li, H., Luo, Z., Zheng, Y.: NUS-WIDE: a real-world web image database from national university of Singapore. In: Proceedings of the ACM International Conference on Image and Video Retrieval, p. 48. ACM (2009)
2. Ding, G., Guo, Y., Zhou, J.: Collective matrix factorization hashing for multimodal data. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2075–2082 (2014)
3. Gan, J., Feng, J., Fang, Q., Ng, W.: Locality-sensitive hashing scheme based on dynamic collision counting. In: Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data, pp. 541–552. ACM (2012)
4. Jiang, K., Que, Q., Kulis, B.: Revisiting kernelized locality-sensitive hashing for improved large-scale image retrieval. In: Computer Vision and Pattern Recognition, pp. 4933–4941 (2014)
5. Kulis, B., Grauman, K.: Kernelized locality-sensitive hashing for scalable image search. In: 2009 IEEE 12th International Conference on Computer Vision, pp. 2130–2137. IEEE (2009)
6. Liu, W., Wang, J., Ji, R., Jiang, Y.G., Chang, S.F.: Supervised hashing with kernels. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2074–2081. IEEE (2012)
7. Liu, W., Wang, J., Kumar, S., Chang, S.F.: Hashing with graphs. In: Proceedings of the 28th International Conference on Machine Learning (ICML 2011), pp. 1–8 (2011)
8. Lu, X., Wu, F., Tang, S., Zhang, Z., He, X., Zhuang, Y.: A low rank structural large margin method for cross-modal ranking. In: International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 433–442 (2013)
9. Moran, S., Lavrenko, V.: Regularised cross-modal hashing. In: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 907–910. ACM (2015)
10. Song, J., Gao, L., Yan, Y., Zhang, D., Sebe, N.: Supervised hashing with pseudo labels for scalable multimedia retrieval, pp. 827–830 (2015)
11. Tan, S., Hu, L., Wang-Xu, A., Tang, J., Jia, Z.: Kernelized cross-modal hashing for multimedia retrieval. In: 2016 12th World Congress on Intelligent Control and Automation (WCICA), pp. 1224–1228. IEEE (2016)
12. Tang, J., Wang, K., Shao, L.: Supervised matrix factorization hashing for cross-modal retrieval. *IEEE Trans. Image Process.* **25**(7), 3157–3166 (2016)
13. Tian, J., Liu, C., Li, Y.Q., Qin, B., Zha, Y.F.: Kernelized supervised context hashing. *IET Image Process.* **10**(12), 986–995 (2016)
14. Wang, J., Kumar, S., Chang, S.F.: Semi-supervised hashing for scalable image retrieval. In: 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3424–3431. IEEE (2010)
15. Wang, S., Jiang, S., Huang, Q., Tian, Q.: S3MKL: scalable semi-supervised multiple kernel learning for image data mining. In: Proceedings of the 18th ACM International Conference on Multimedia, pp. 163–172. ACM (2010)
16. Weiss, Y., Torralba, A., Fergus, R.: Spectral hashing. In: Advances in Neural Information Processing Systems, pp. 1753–1760 (2009)
17. Wu, B., Yang, Q., Zheng, W.S., Wang, Y., Wang, J.: Quantized correlation hashing for fast cross-modal search. In: IJCAI, pp. 3946–3952 (2015)

18. Xia, H., Wu, P., Hoi, S.C., Jin, R.: Boosting multi-kernel locality-sensitive hashing for scalable image retrieval. In: Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 55–64. ACM (2012)
19. Xie, B., Zheng, S.: Kernelized locality-sensitive hashing for semi-supervised agglomerative clustering. *Comput. Sci.* (2013)
20. Zhang, D., Li, W.J.: Large-scale supervised multimodal hashing with semantic correlation maximization. In: Twenty-Eighth AAAI Conference on Artificial Intelligence, pp. 2177–2183 (2014)
21. Zhang, Y., Lu, W., Liu, Y., Wu, F.: Kernelized sparse hashing for scalable image retrieval. *Neurocomputing* **172**(C), 207–214 (2016)
22. Zhen, Y., Yeung, D.Y.: Co-regularized hashing for multimodal data. In: Advances in Neural Information Processing Systems, pp. 1376–1384 (2012)
23. Zhou, J., Ding, G., Guo, Y.: Latent semantic sparse hashing for cross-modal similarity search. In: Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval, pp. 415–424. ACM (2014)
24. Zhou, J., Ding, G., Guo, Y., Liu, Q., Dong, X.: Kernel-based supervised hashing for cross-view similarity search. In: 2014 IEEE International Conference on Multimedia and Expo (ICME), pp. 1–6. IEEE (2014)
25. Zhu, X., Huang, Z., Shen, H.T., Zhao, X.: Linear cross-modal hashing for efficient multimedia search. In: Proceedings of the 21st ACM International Conference on Multimedia, pp. 143–152. ACM (2013)