Neighborhood-Preserving Hashing for Large-Scale Cross-Modal Search

Botong Wu
Nat'l Engineering Laboratory for Video Technology
Cooperative Medianet Innovation Center
Key Laboratory of Machine Perception (MoE)
Sch'l of EECS, Peking University, Beijing, 100871, China
botongwu@pku.edu.cn

Yizhou Wang
Nat'l Engineering Laboratory for Video Technology
Cooperative Medianet Innovation Center
Key Laboratory of Machine Perception (MoE)
Sch'l of EECS, Peking University, Beijing, 100871, China
Yizhou.Wang@pku.edu.cn

ABSTRACT

In the literature of cross-modal search, most methods employ linear models to pursue hash codes that preserve data similarity, in terms of Euclidean distance, both within-modal and across-modal. However, data dimensionality can be quite different across modalities. It is known that the behavior of Euclidean distance/similarity between datapoints can be drastically different in linear spaces of different dimensionality. In this paper, we identify this "variation of dimensionality" problem in cross-modal search that may harm most of distance-based methods. We propose a semi-supervised nonlinear probabilistic cross-modal hashing method, namely Neighborhood-Preserving Hashing (NPH), to alleviate the negative effect due to the variation of dimensionality issue. Inspired by tSNE [19], rather than preserve pairwise data distances, we propose to learn hash codes that preserve neighborhood relationship of datapoints via matching their conditional distribution derived from distance to that of datapoints of multi-modalities. Experimental results on three real-world datasets demonstrate that the proposed method outperforms the state-of-the-art distance-based semi-supervised cross-modal hashing methods as well as many fully-supervised ones.

CCS Concepts

•**Information systems → Multimedia and multimodal retrieval;**

Keywords

Cross-Modal; Hashing; Neighborhood-Preserving

1. INTRODUCTION

Nowadays heterogeneous data are ubiquitous. For example, it is common to see an article on a webpage elucidating topics by embedding images, video clips and/or hyperlinks into text. Hence, there emerges a high demand on retrieving one type of data using data of another modality about similar topics, *e.g.*, using text to search for relevant images, or vice versa. Consequently, many cross-modal or cross-view retrieval methods have been proposed to solve the heterogeneous data retrieval problem *e.g.*, [30, 15, 32]. With the explosive growth of data, how to efficiently retrieve from large-scale datasets is a big challenge. Hashing based methods are proposed as an efficient means to tackle the challenge. However, most existing hashing methods (*e.g.*, [22, 12, 4, 24, 31, 9, 14]) are unimodal.

Recently, quite a number of cross-modal hashing methods have been proposed, *e.g.*, [2, 34, 27, 11, 33, 13, 35, 28, 29, 10, 5, 21, 23, 8]. Cross-modal hashing may be divided into supervised methods and semi-supervised ones by whether a method is using semantic information or not. In this paper, we propose a semi-supervised cross-modal hashing method for large-scale search. In the last years, several semi-supervised methods have been proposed. For example, Cross-View Hashing (CVH) [10] extends spectral hashing [24] to the cross-modal setting by preserving within-modal-similarity and cross-modal-similarity. Collective Matrix Factorization Hashing (CMFH) [5] and Semantic Topic Multimodal Hashing (STMH) [21] pursue a single set of hash codes to preserve data distance across modalities. CMFH method learns shared codes by collective matrix factorization with a latent factor model from different modalities. STMH method learns a set of hash codes of the text modality in accordance with the topics learned by a clustering based method. Similar to CVH and CMFH, Partial Multi-Modal Hashing ($\mathbf{PM^2H}$) [23] also learns shared hash codes by linear matrix decomposition to preserve data distance across different modalities. The within-modal-similarity is preserved using graph Laplacian as in CVH. Recently, many deep cross-modal hashing methods [36, 20] have been proposed. Most of them achieve outstanding performance, but they needed a large number of semantic labels. Whereas, in practice it is very hard or laborious to obtain complete and accurate semantic tags for all the objects in large-scale datasets. The only easy-getting cross-modal correlation is the datapoints belonging to the same object. The method, NPH, proposed in this paper falls into this category.

Most cross-modal hashing methods are linear models using techniques such as linear subspace learning and graph Laplacian to pursue hash codes that preserve data similarity both within-modal and across-modal. However, there is an important issue that requires further attention, *i.e.*, the variation of dimensionality across modalities may be drastic. We know that in spaces of different dimensionality the behavior (*e.g.*, distribution) of Euclidean distance between datapoints can be drastically different which prevents many projection-based/distance-based algorithms from being effective. This is a key reason inducing *the curse of dimensionality* [1]. How-

MM '16, October 15-19, 2016, Amsterdam, Netherlands
© 2016 ACM. ISBN 978-1-4503-3603-1/16/10…$15.00
DOI: http://dx.doi.org/10.1145/2964284.2967241

ever, this “variation of dimensionality" issue is largely ignored in the literature of cross-modal search.

In the literature of data visualization, dimensionality reduction and the like, people study methods to map high dimensional data to low dimensional space such that the local/global structure of data is preserved as well as possible. They also confront the above “variation of dimensionality" challenge. Rather than preserving the distance, “stochastic neighborhood embedding" [19] are proposed to place objects in a low dimensional space so as to optimally preserve neighborhood identity.

In this paper, motivated by [19], we propose a nonlinear probabilistic cross-modal hashing method, namely Neighborhood-Preserving Hashing (NPH) to alleviate the negative effect due to a large variation of dimensionality across modalities. Specifically, in order to preserve neighborhood relationship, we first convert the pairwise Euclidean distance between datapoints in each modality into conditional probabilities that represent their similarities, then learn a hashing function for each modality that projects datapoints to a common hash space. In the hash space, the shared hash codes preserve the data similarity, in terms of neighborhood relationship, in multi-modalities jointly. This is realized by minimizing the Kullback-Leibler divergence between the conditional distribution of hash codes and the collective conditional distributions of the datapoints from each modality. As in [19], we observe that taking different types of distributions (*e.g.*, heavy-tailed or light-tailed) can greatly compensate for the variation of dimensionality during matching the distributions across modalities. Hence we propose *distribution selection* when converting pairwise distance of datapoints/hash codes into conditional probabilities. Experiment results demonstrate that the proposed method outperforms the state-of-the-art cross-modal search methods by a notable margin. Especially on a large-scale dataset, CIFAR-580K, the proposed method achieves on average about 7% performance gain over the second best method.

2. NEIGHBORHOOD PRESERVING HASHING

Assuming that there are n objects, each is described by data of M modalities, $\mathbf{o}_i = (\mathbf{x}_i^{(1)}, \mathbf{x}_i^{(2)}, ..., \mathbf{x}_i^{(M)})$, where $i = 1, \cdots, n$, $m = 1, \cdots, M$. $\mathbf{x}_i^{(m)} \in \mathbf{R}^{d_m}$ denotes the datapoint of the m-th modality of the i-th object, and d_m is the dimensionality of the m-th modality. The data of each modality are zero-centered, i.e. $\sum_{i=1}^n \mathbf{x}_i^{(m)} = \mathbf{0}$, $\forall m$.

The goal of cross-modal hashing is to learn M hash functions that map the data of each modality to binary hash codes $\mathbf{B}^{(m)} \in \{-1,1\}^{c\times n}$, where c denotes the length of the hash codes. For example, the mapping/hash function can be defined as $f(\mathbf{x}_i^{(m)}) = \text{sgn}(\mathbf{W}^{(m)T}\mathbf{x}_i^{(m)})$, where $\mathbf{W}^{(m)} \in \mathbf{R}^{d_m\times c}$ denotes a projection matrix.

There are $M(M-1)$ tasks in a M-modal cross-search, *i.e.*, using data of one modality the model should be able to search for data of the other $M-1$ modalities.

The overall procedure of NPH is as follows: (i) In order to preserve neighborhood relationship, the pairwise Euclidean distances between datapoints in each modality are converted into conditional probabilities that represent their similarities in terms of probable neighbors. (ii) Optimal hash codes for training set are learned via minimizing M KL-divergences between the conditional distribution of the data from each modality (denoted as $\mathbf{P}^{(m)}$) and that of the hash codes (denoted as $\mathbf{Q}^{(m)}$). (iii) Learning hash functions to map datapoints to the hash codes for each modality via learning c binary classifiers of kernel logistic regression.

2.1 Formulation

Similar to SNE [6], we convert Euclidean distances between datapoints $\mathbf{x}_j^{(m)}$ and $\mathbf{x}_i^{(m)}$ into conditional probabilities $p_{j|i}^{(m)}$ as follows:

$$p_{j|i}^{(m)} = \frac{T(\mathbf{x}_i^{(m)}, \mathbf{x}_j^{(m)}; \theta)}{\sum_{k\neq i} T(\mathbf{x}_k^{(m)}, \mathbf{x}_i^{(m)}; \theta)}, \tag{1}$$

where $T(\mathbf{x}_i^{(m)}, \mathbf{x}_j^{(m)}; \theta)$ returns the probability from the distance between the two datapoints. θ denotes the parameter. The conditional distribution can be of different types:

(i) For Gaussian distribution,

$$T(\mathbf{x}_i^{(m)}, \mathbf{x}_j^{(m)}; \sigma_i) = \exp(-||\mathbf{x}_i^{(m)} - \mathbf{x}_j^{(m)}||^2/2\sigma_i^2) \tag{2}$$

where σ_i denotes the variance of the Gaussian that is centered on datapoint $\mathbf{x}_i^{(m)}$. The details of computing its value can be found in [6].

(ii) For Student t-distribution with one degree of freedom

$$T(\mathbf{x}_i^{(m)}, \mathbf{x}_j^{(m)}) = (1 + ||\mathbf{x}_i^{(m)} - \mathbf{x}_j^{(m)}||^2)^{-1} \tag{3}$$

We can adopt more type of distributions, *e.g.*, χ^2-distribution. In this paper, we only choose from Gaussian and Student t-distribution to demonstrate the idea. The conditional probability $p_{j|i}$ signifies the probable neighborhood relationship of the datapoints. We discuss distribution selection in Section 2.4. To simplify the problem, we make the relationship symmetric by $p_{ij} = (p_{j|i} + p_{i|j})/2$, also we set $p_{ii}^{(m)} = 0$.

Similarly, the distribution of hash codes $\mathbf{Q}$ can be computed as

$$q_{j|i}^{(m)} = \frac{T(\mathbf{b}_i, \mathbf{b}_j; \theta)}{\sum_{k\neq i} T(\mathbf{b}_k, \mathbf{b}_i; \theta)}, \tag{4}$$

where $\mathbf{b}_i \in \{-1,1\}^c$ and $\mathbf{b}_j$ denote the i-th and the j-th hash codes, T(.,.) returns the probability from their Hamming distance. We set $q_{ii}^{(m)} = 0$.

The objective of NPH is to learn a shared set of hash codes $\mathbf{B} = (\mathbf{b}_1, \cdots, \mathbf{b}_n)$ that can match its distribution $\mathbf{Q}^{(m)}$ to the datapoint distributions of each modality $\mathbf{P}^{(m)}$ jointly as well as possible. This matching optimizes the preservation of the neighborhood relationship between the hash codes and the datapoints in each modality.

We match the two distributions of the m-th modality, $\mathbf{P}^{(m)}$ and $\mathbf{Q}^{(m)}$, by minimizing their Kullback-Leibler(KL) divergence defined as follows

$$KL(\mathbf{P}^{(m)}||\mathbf{Q}^{(m)}(\mathbf{B})) = \sum_{i\neq j} p_{ij}^{(m)} \log \frac{p_{ij}^{(m)}}{q_{ij}^{(m)}} \tag{5}$$

To purse hash codes of m-th modality that preserves the neighborhood relationship in original space, we define the objective function as

$$O_m(\mathbf{B}) = \mathbf{KL}(\mathbf{P^{(m)}}||\mathbf{Q^{(m)}}(\mathbf{B})) \tag{6}$$

Hence, the final objective function which pursues the shared hash codes for all modalities is defined as

$$\begin{aligned} \min O(\mathbf{B}) &= \sum_{m=1}^{M} \alpha_m O_m(\mathbf{B}) \\ s.t.\ \frac{1}{n}\mathbf{B}\mathbf{B}^T &= \mathbf{I}_{c\times c} \end{aligned} \tag{7}$$

where α_m are empirical parameters and $\sum_{m=1}^M \alpha_m = 1$, n is the number of training data. The constrain $\frac{1}{n}\mathbf{B}\mathbf{B}^T = \mathbf{I}_{c\times c}$ requires the hash codes to be uncorrelated. Since $\mathbf{B}$ is binary, optimizing the above objective is NP hard. Hence, we relax $\mathbf{B}$ to be real-value. To make the orthogonal constrain more concise, we denote the relaxed $\mathbf{B}$ as $\hat{\mathbf{B}} = \frac{1}{\sqrt{n}}\mathbf{B}$. The constrain turns to be $\hat{\mathbf{B}}\hat{\mathbf{B}}^T = \mathbf{I}_{c\times c}$.

Task	Method	Wiki					NUS-WIDE					CIFAR-580K				
		c = 16	c = 32	c = 64	c = 96	c = 128	c = 16	c = 32	c = 64	c = 96	c = 128	c = 16	c = 32	c = 64	c = 96	c = 128
Task 1: Modality 1 to Modality 2	CMSSH*	0.2005	0.1938	0.1936	0.1861	0.1931	0.4544	0.5024	0.5024	0.4780	0.4942	0.1832	0.1694	0.1656	0.1652	0.1677
	SCM*	0.2359	0.2445	0.2520	0.2558	0.2553	0.5104	0.5114	0.5084	0.5065	0.5087	0.1723	0.1773	0.1780	0.1793	0.1791
	QCH*	0.2540	0.2566	0.2584	0.2455	0.2217	0.5571	0.5715	0.5713	0.5660	0.5620	0.1770	0.1934	0.1961	0.1950	0.1951
	CVH	0.2153	0.1743	0.1710	0.1767	0.1996	0.4712	0.4532	0.4510	0.4472	0.4461	0.1916	0.2380	0.2831	0.2481	0.2263
	CMFH	0.2458	0.2565	0.2623	0.2622	0.2636	0.5238	**0.5219**	**0.5244**	0.5165	**0.5169**	0.1587	0.1575	0.1565	0.1566	0.1557
	STMH	0.2006	0.2054	0.2343	0.2216	0.2255	**0.5272**	0.4776	0.4581	0.4374	0.4263	0.1819	0.1749	0.1664	0.1599	0.1596
	NPH	**0.2649**	**0.3358**	**0.3990**	**0.4172**	**0.4205**	0.5167	0.5059	0.5115	**0.5176**	0.5072	**0.2055**	**0.2434**	**0.3269**	**0.3850**	**0.4086**
Task 2: Modality 2 to Modality 1	CMSSH*	0.2397	0.2397	0.2486	0.2290	0.2254	0.4767	0.5175	0.5469	0.5278	0.5148	0.1650	0.1726	0.1650	0.1632	0.1697
	SCM*	0.3679	0.4026	0.4370	0.4432	0.4518	0.5624	0.5764	0.5909	0.5949	0.5968	0.1751	0.1766	0.1774	0.1765	0.1783
	QCH*	0.4176	0.4152	0.4034	0.3739	0.2895	0.5668	0.5900	0.5775	0.5708	0.5678	0.1709	0.1862	0.1917	0.1941	0.1940
	CVH	0.3127	0.2513	0.2194	0.2083	0.2341	0.4609	0.4472	0.4465	0.4496	0.4520	0.1911	0.2386	0.2837	0.2473	0.2246
	CMFH	0.6190	0.6461	0.6589	0.6526	0.6641	0.6565	0.7008	0.7218	0.7252	0.7277	0.1595	0.1578	0.1561	0.1566	0.1571
	STMH	0.6036	0.6157	0.6291	0.6496	0.6450	0.6536	0.6849	0.7071	0.7220	0.7252	0.1837	0.1746	0.1678	0.1662	0.1622
	NPH	**0.6715**	**0.6985**	**0.7132**	**0.7140**	**0.7172**	**0.6566**	**0.7421**	**0.7598**	**0.7688**	**0.7696**	**0.2033**	**0.2403**	**0.3193**	**0.3891**	**0.4144**

Table 1: Comparison of the Mean Average Precision (MAP) values of the state-of-the-art cross-modal hashing methods in different code-lengths on three datasets. The fully supervised cross-modal hashing methods are marked by $*$. The best MAP values of all the methods (semi-/full-supervised) are underlined, and the best semi-supervised results are highlighted in black-boldface.

2.2 Optimization

The gradient of the objective function O with respect to $\hat{\mathbf{b}}_i$ is distribution dependent. For Gaussian distribution, the gradient of O_m of the m-th modality is

$$\frac{\partial O_m}{\partial \hat{\mathbf{b}}_i} = 4\alpha_m \sum_j (p_{ij}^{(m)} - q_{ij}^{(m)})(\hat{\mathbf{b}}_i - \hat{\mathbf{b}}_j). \tag{8}$$

For Student t-distribution with one degree of freedom, the gradient

$$\frac{\partial O_m}{\partial \hat{\mathbf{b}}_i} = 4\alpha_m \sum_j (p_{ij}^{(m)} - q_{ij}^{(m)})(\hat{\mathbf{b}}_i - \hat{\mathbf{b}}_j)(1 + ||\hat{\mathbf{b}}_i - \hat{\mathbf{b}}_j||^2)^{-1}. \tag{9}$$

Hence, the gradient of the objective function O can be computed

$$\frac{\partial O}{\partial \hat{\mathbf{b}}_i} = \sum_{m=1}^{M} \alpha_m \sum_j \frac{\partial O_m}{\partial \hat{\mathbf{b}}_i} \tag{10}$$

The Lagrange of our objective function with orthogonal constrain is

$$L(\hat{\mathbf{B}}, \mathbf{\Lambda}) = O(\hat{\mathbf{B}}) - \frac{1}{2}\mathrm{tr}(\mathbf{\Lambda}(\hat{\mathbf{B}}\hat{\mathbf{B}}^{\mathrm{T}} - \mathbf{I})) \tag{11}$$

Then we set the gradient of Eqn.11 w.r.t. $\hat{\mathbf{B}}$ to be $\mathbf{0}$.

$$\frac{\partial L(\hat{\mathbf{B}}, \mathbf{\Lambda})}{\partial \hat{\mathbf{B}}} = \frac{\partial O(\hat{\mathbf{B}})}{\partial \hat{\mathbf{B}}} - \mathbf{\Lambda}\hat{\mathbf{B}} = \mathbf{0} \tag{12}$$

For convenience, denote $\mathbf{G} = \frac{\partial O(\hat{\mathbf{B}})}{\partial \hat{\mathbf{B}}}$. As $\mathbf{\Lambda}$ is symmetric, we can get $\mathbf{\Lambda} = \mathbf{G}\hat{\mathbf{B}}^T = \hat{\mathbf{B}}\mathbf{G}^T$ and $\frac{\partial L(\hat{\mathbf{B}}, \mathbf{\Lambda})}{\partial \hat{\mathbf{B}}} = \mathbf{G} - \hat{\mathbf{B}}\mathbf{G}^T\hat{\mathbf{B}} = \hat{\mathbf{B}}\mathbf{A}$, where $\mathbf{A} = \hat{\mathbf{B}}^T\mathbf{G} - \mathbf{G}^T\hat{\mathbf{B}}$. Then $\hat{\mathbf{B}}$ can be updated by Crank-Nicolson-like method [17] as

$$\hat{\mathbf{B}} = \hat{\mathbf{B}} - \frac{\tau}{2}(\hat{\mathbf{B}} + \hat{\mathbf{B}}\mathbf{Q})\mathbf{A}, \tag{13}$$

where τ is the step size and $\mathbf{Q} = (\mathbf{I} + \frac{\tau}{2}\mathbf{A})^{-1}(\mathbf{I} - \frac{\tau}{2}\mathbf{A})$. We update $\hat{\mathbf{B}}$ with Barzilai-Borwein (BB) method as in [25]. After learning the relaxed hash codes $\hat{\mathbf{B}}$, the median vector of $\hat{\mathbf{B}}$ can be obtained

$$\mathbf{u} = \mathrm{median}(\hat{\mathbf{B}}) \in \mathbf{R}^c. \tag{14}$$

Then the shared binary hash code can be determined by the median vector, where $B_{ij} = 1$ when $B_{ij} \geq u_i$; $B_{ij} = -1$, otherwise. We binarizing $\hat{\mathbf{B}}$ with the median vector rather than directly using the sign function because balancing the labeled data can improve the classification performance when learning the hash function in Section 2.3.

2.3 Learning Hash Functions

After getting the hash codes $\mathbf{B}$ by optimizing Eqn. 7, here we introduce about how to learn a set of efficient hash functions to map data to the hash codes.

For each modality, we train c binary classifiers, each of which maps a datapoint $\mathbf{x}_i^{(m)}$ to the corresponding bit of its hash code. The classifiers are embodied by the Kernel Logistic Regression model [7], which is defined as

$$\mathbf{w}_k^{(m)*} = \arg\min \sum_{i=1}^{n} \log(1 + \exp(-B_{ki} \sum_{j=1}^{s} W_{jk}^{(m)} \cdot \kappa(\mathbf{m}_j^{(m)}, \mathbf{x}_i^{(m)}))) + \lambda ||\mathbf{w}_k^{(m)}||_2^2 \tag{15}$$

where $B_{ki} \in \{-1, 1\}$ denotes the k-th row and the i-th column of $\mathbf{B}$, *i.e.*, the k-th bit of the i-th hash code, $k = 1, \cdots, c$. $\kappa(\mathbf{m}_j^{(m)}, \mathbf{x}_i^{(m)})$ is the kernel function that measures similarity between any pair of data $\mathbf{m}_j^{(m)}$ and $\mathbf{x}_i^{(m)}$, here, we utilize the RBF kernel. In order to reduce the computation in training and testing, $\mathbf{m}_j^{(m)}$ is the j-th cluster center of the training data obtained by k-means. We set $s = 500$ in each modality and set $\lambda = 0.01$. We learn $\mathbf{w}_k^{(m)}$ with the minFunc implemented by M. Schmidt *.

Given a datapoint in the m-th modality, say $\mathbf{x}^{(m)}$, using the learned Logistic function we obtain the probability of a hash bit $b_k^{(m)}$ being 1 and -1. Then the hash bit $b_k^{(m)}$ can be determined as

$$b_k^{(m)} = \mathrm{sign}\big(p(b_k^{(m)} = 1|\mathbf{x}^{(m)}) - p(b_k^{(m)} = -1|\mathbf{x}^{(m)})\big) \tag{16}$$

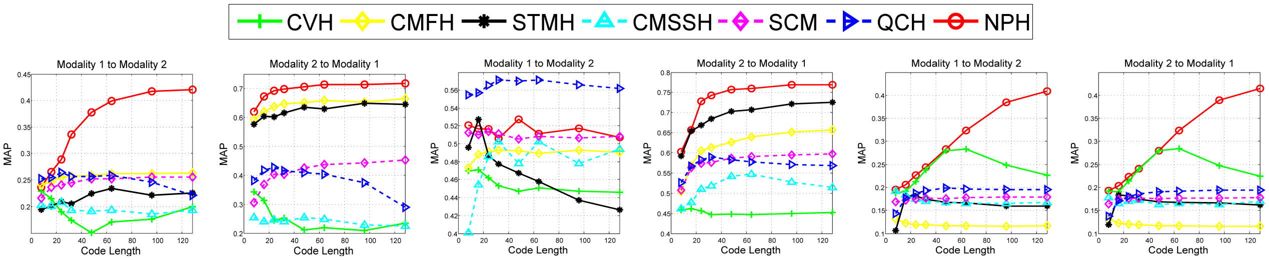
In this way, we can generate the hash code for any given datapoint.

2.4 Distribution Selection

As in [19], we also observe that different types of distributions (e.g., heavy-tailed or light-tailed) possess unique capabilities in depicting the neighborhood relationship of data in different dimensional spaces.

We employ two types of distributions — a heavy-tailed distribution (Student t-distribution) and a light-tailed distribution (Gaussian distribution) — and propose to choose distribution type for matching the neighborhood distributions between hash codes and datapoints of each modality. Notice that even for the same set of hash codes, when matching to different modalities, the type of $\mathbf{Q}^{(m)}$ may be different *w.r.t* its matched modality. During training, we adopt 5-fold cross-validation to choose the best combination of distributions for all modalities data and the hash codes.

*M. Schmidt. minFunc: unconstrained differentiable multivariate optimization in Matlab. http://www.cs.ubc.ca/ schmidtm/Software/minFunc.html, 2005.



(a) On Wiki dataset (b) On NUS-WIDE dataset (c) On CIFAR-580K dataset

Figure 1: MAP results of different methods on three datasets with various hash code length.

Moreover, we find that the proposed NPH can achieve better performance on different tasks with different distribution selection. So for each task we learn a particular set of hash codes. Experiments show that this method greatly improves the performance.

3. EXPERIMENT

In this section, we will compare NPH with some state-of-the-art semi-supervised as well as fully-supervised cross-hashing methods on two widely used cross-modal datasets Wiki [16] and NUS-WIDE [3], and also on a large-scale cross-view dataset CIFAR-580K [26] sampled from 80M-Tiny image dataset [18].

In our experiment, ten runs of independent experiments are performed. In each run, training and testing sets are randomly sampled. The Mean Average Precision (MAP) is reported in this paper. All our experiments are conducted on a workstation with Intel(R) Xeon(R) E5-2620@2.0GHz CPUs, 128 GB RAM and 64-bit Ubuntu system.

3.1 Datasets and Evaluation

The Wiki dataset [16] contains 2,866 objects (which are described as image-text pairs) and ten semantic labels for each object. Each image is represented by a 128-dimensional Bag-of-Visual-Word (BoVW) feature and each text is denoted by a 10-dimensional vector. In our experiment, 80% of the data are randomly sampled for training and the rest are for testing.

The NUS-WIDE dataset [3] consists of 269,648 images and 5,018 raw tags in text. The semantic labels of 81 concepts are provided. We select 186,643 image-tag pairs belonging to 10 largest classes. Each image is denoted by a 500-dimensional Bag-of-Visual-Word (BoVW) vector, and each tag is 1000-dimensional most frequent tags from the raw tags. In our experiment, 1% of the data are randomly selected for testing. For training, the CMFH and STMH use 99% of the data and other algorithms use 5000 data.

The CIFAR-580K dataset consists of 580,804 images with ten class labels. As in [26], in order to construct multi-modal data, from each image a 384-dimensional GIST descriptor is extracted as one view and a 496-dimensional HOG descriptor is extracted as the other view. Similar to the NUS-WIDE dataset, we randomly select 1% data for testing. Due to large-scale, we randomly select 30% data as the training set for CMFH and STMH, and 5000 data as training set for other algorithms.

Since NPH is a semi-supervised method of cross-modal search, it is compared with the state-of-the-art semi-supervised cross-modal hashing methods including CVH [10], CMFH [5] and STMH [21]. Besides, we also compare it to some state-of-the-art fully supervised models including CMSSH [2], SCM [29] and QCH [26].

To evaluate the methods, we adopt widely used criterion, namely Mean Average Precision (MAP) with retrieval range R. In our experiment, we set $R = 50$, the same as [5, 21]. We set $\alpha_1 = \alpha_2 = 0.5$ in Eqn. 7 for all experiments of NPH. For distribution selection, we adopt 5-fold cross-validation to select from $2^{2\times 2} = 16$ distribution combinations in total (see Section 2.4).

Task	Method	Wiki				
		c = 16	c = 32	c = 64	c = 96	c = 128
Modality 1 to Modality 2	NPH-SNE	0.2624	0.3324	0.3997	0.4199	0.4160
	NPH-tSNE	0.2502	0.2497	0.3016	0.3590	0.4005
	NPH	**0.2649**	**0.3358**	**0.3990**	**0.4172**	**0.4205**
Modality 2 to Modality 1	NPH-SNE	0.5294	0.5878	0.6415	0.6594	0.6696
	NPH-tSNE	0.6602	0.6890	0.6946	0.6955	0.6951
	NPH	**0.6715**	**0.6985**	**0.7140**	**0.7172**	**0.7725**

Table 2: MAP values of NPH with different distribution selection. The best values are highlighted in boldface.

3.2 Results on the Three Datasets

From Table 1 and Figure 1(a), it can be seen that on the Wiki dataset NPH outperforms all other methods for all bits on all Tasks. NPH outperforms the second best method CMFH over 10% on Task 1 and over 5% on Task 2. From Figure 1(a), we can observe that on Task 1 the performance of NPH improves considerably with the increase of code length compared to the other methods.

From Table 1 and Figure 1(b), it can be seen that on NUS-WIDE the performance of NPH is comparable to CMFH and STMH on Task 1. However, when code length is over 32, NPH outperforms CMFH (the 2nd best semi-supervised method) by about 3% on Task 1. Meanwhile, NPH outperforms all other cross-modal hashing methods on Task 2 for all code lengths.

CIFAR-580K is a large-scale dataset, which is used to evaluate the scalability of the methods. Table 1 shows that NPH outperforms all the compared methods on both tasks by a large margin - over 7% gain on average of both tasks compared to CVH (the 2nd best). Particularly, the performance gain increases with the code length notably even compared to the supervised methods.

3.3 Effect on Distribution Selection

The results of NPH in Table 1 and Figure 1 are the results of distribution selection for each modality and the hash codes. Here we compare distribution selection to two methods that adopt fixed distributions based on NPH, namely, NPH-SNE which adopts Gaussian distributions for both modality data and hash codes (as in SNE [6] for data visualization) and NPH-tSNE which adopts Gaussian distributions for modality data and Student t-distribution for hash codes (as in tSNE [19]). We report the results on Wiki dataset in Table 2. It can be seen that NPH achieves the best performance with the distribution selection and the strategy of training a particular set of hash codes for each task.

4. CONCLUSION

In this paper, we propose a semi-supervised cross-modal hashing, called Neighborhood Preserving Hashing (NPH). We propose to learn hash codes that preserve data neighborhood relationship via matching their distribution derived from distance to that of datapoints of multi-modalities. We further utilize cross-validation to select distributions to better match the modalities data and hash codes. Experiments show that the proposed model achieves superior performance over state-of-the-art cross-modal hashing methods.

5. ACKNOWLEDGMENTS

This work was supported in part by the following grants 973-2015CB351800, NSFC-61231010, NSFC-61527804, NSFC-61421062 and NSFC-61210005.

6. REFERENCES

[1] R. E. Bellman. *Dynamic Programming*. 1957.

[2] M. M. Bronstein, A. M. Bronstein, F. Michel, and N. Paragios. Data fusion through cross-modality metric learning using similarity-sensitive hashing. In *CVPR*, 2010.

[3] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y. Zheng. Nus-wide: a real-world web image database from national university of singapore. In *CIVR*, 2009.

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

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

[6] G. E. Hinton and S. T. Roweis. Stochastic neighbor embedding. In *NIPS*, pages 833–840, 2002.

[7] S. S. Keerthi, K. Duan, S. K. Shevade, and A. N. Poo. A fast dual algorithm for kernel logistic regression. *JML*, 61(1-3):151–165, 2005.

[8] S. Kim and S. Choi. Multi-view anchor graph hashing. In *ICASSP*, 2013.

[9] B. Kulis and K. Grauman. Kernelized locality-sensitive hashing for scalable image search. In *ICCV*, 2009.

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

[11] Z. Lin, G. Ding, M. Hu, and J. Wang. Semantics-preserving hashing for cross-view retrieval. In *CVPR*, pages 3864–3872, 2015.

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

[13] J. Masci, M. Bronstein, A. Bronstein, and J. Schmidhuber. Multimodal similarity-preserving hashing. *TPAMI*, 2013.

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

[15] N. Quadrianto and C. H. Lampert. Learning multi-view neighborhood preserving projections. In *ICML*, pages 425–432, 2011.

[16] N. Rasiwasia, J. Costa Pereira, E. Coviello, G. Doyle, G. R. Lanckriet, R. Levy, and N. Vasconcelos. A new approach to cross-modal multimedia retrieval. In *ACM MM*, 2010.

[17] G. D. Smith. Numerical solution of partial differential equations. 1965.

[18] A. Torralba, R. Fergus, and W. T. Freeman. 80 million tiny images: A large data set for nonparametric object and scene recognition. *TPAMI*, 30(11):1958–1970, 2008.

[19] L. Van der Maaten and G. Hinton. Visualizing data using t-sne. *JMLR*, 9(2579-2605):85, 2008.

[20] D. Wang, P. Cui, M. Ou, and W. Zhu. Deep multimodal hashing with orthogonal regularization. In *Proceedings of the 24th International Conference on Artificial Intelligence*, pages 2291–2297. AAAI Press, 2015.

[21] D. Wang, X. Gao, X. Wang, and L. He. Semantic topic multimodal hashing for cross-media retrieval. In *IJCAI*, pages 3890–3896, 2015.

[22] J. Wang, S. Kumar, and S.-F. Chang. Semi-supervised hashing for scalable image retrieval. In *CVPR*, 2010.

[23] Q. Wang, L. Si, and B. Shen. Learning to hash on partial multi-modal data. In *IJCAI*, pages 3904–3910. AAAI Press, 2015.

[24] Y. Weiss, A. Torralba, and R. Fergus. Spectral hashing. In *NIPS*, 2009.

[25] Z. Wen and W. Yin. A feasible method for optimization with orthogonality constraints. *Mathematical Programming*, 142(1-2):397–434, 2013.

[26] B. Wu, Q. Yang, W.-S. Zheng, Y. Wang, and J. Wang. Quantized correlation hashing for fast cross-modal search. In *IJCAI*, 2015.

[27] F. Wu, Z. Yu, Y. Yang, S. Tang, Y. Zhang, and Y. Zhuang. Sparse multi modal hashing. *T MultiMedia*, 2014.

[28] D. Zhai, H. Chang, Y. Zhen, X. Liu, X. Chen, and W. Gao. Parametric local multimodal hashing for cross-view similarity search. In *IJCAI*, 2013.

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

[30] L. Zhang, Y. Zhang, R. Hong, and Q. Tian. Full-space local topology extraction for cross-modal retrieval. *TIP*, 24(7):2212–2224, 2015.

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

[32] Y. Zhen, P. Rai, H. Zha, and L. Carin. Cross-modal similarity learning via pairs, preferences, and active supervision. In *AAAI*, 2015.

[33] Y. Zhen and D.-Y. Yeung. Co-regularized hashing for multimodal data. In *NIPS*, 2012.

[34] Y. Zhen and D.-Y. Yeung. A probabilistic model for multimodal hash function learning. In *SIGKDD*, 2012.

[35] X. Zhu, Z. Huang, H. T. Shen, and X. Zhao. Linear cross-modal hashing for efficient multimedia search. In *Proceedings of the 21st ACM international conference on Multimedia*, pages 143–152. ACM, 2013.

[36] Y. Zhuang, Z. Yu, W. Wang, F. Wu, S. Tang, and J. Shao. Cross-media hashing with neural networks. In *ACM MM*, 2014.