Tag-based Weakly-supervised Hashing for Image Retrieval

Ziyu Guan[1], Fei Xie[1], Wanqing Zhao*[1], Xiaopeng Wang[1], Long Chen[1], Wei Zhao[2], Jinye Peng[1]
[1]School of Information and Technology, Northwest University of China
[2]School of Computer Science and Technology, Xidian University
{ziyuguan@, xie.xiefei@stumail., zhaowq@, wangxiaopeng@stumail., longchen@stumail.}nwu.edu.cn
ywzhao@mail.xidian.edu.cn, pjy@nwu.edu.cn

Abstract

We are concerned with using user-tagged images to learn proper hashing functions for image retrieval. The benefits are two-fold: (1) we could obtain abundant training data for deep hashing models; (2) tagging data possesses richer semantic information which could help better characterize similarity relationships between images. However, tagging data suffers from noises, vagueness and incompleteness. Different from previous unsupervised or supervised hashing learning, we propose a novel weakly-supervised deep hashing framework which consists of two stages: weakly-supervised pre-training and supervised fine-tuning. The second stage is as usual. In the first stage, rather than performing supervision on tags, the framework introduces a semantic embedding vector (sem-vector) for each image and performs learning of hashing and sem-vectors jointly. By carefully designing the optimization problem, it can well leverage tagging information and image content for hashing learning. The framework is general and does not depend on specific deep hashing methods. Empirical results on real world datasets show that when it is integrated with state-of-art deep hashing methods, the performance increases by 8-10%.

1 Introduction

In recent years, many learning-based hashing schemes [Weiss *et al.*, 2008; Irie *et al.*, 2014; Liu *et al.*, 2012; Zhao *et al.*, 2015; Zhang *et al.*, 2015] have been proposed for image retrieval. They target at learning a compact and similarity preserving representation such that similar images are mapped to nearby binary hash codes in the Hamming space. Among them, the supervised approaches based on deep models have achieved the state-of-the-art results with the help of manually labeled images. However, they still suffer from two issues: (1) training an effective and generalized hashing model requires a large quantity of labeled images, whereas the labeling work is very tedious and expensive; (2) the controlled label set usually only includes coarse-grained concept labels in images, which cannot well characterize the fine-grained similarity relationships between images.

Figure 1: Examples of user tags and manual labels on Flickr images. Black boxes indicate noises.

With the remarkable growth in the popularity of social media websites, more and more users upload, share and tag their images therein. This forms vast collections of images attached with user tags. Such collections could be useful for hashing model training from the following two aspects: (1) tags reflect semantic information of images to some degree, so we could obtain abundant training data for deep hashing models. (2) compared to controlled manual labels, tagging data possesses richer semantic information which could help better characterize similarity relationships between images. Figure 1 shows some example images with both user tags and manual labels in the Flickr dataset we use for experiments. As can be seen, users tend to tag images in a more fine-grained fashion, e.g. "jump jet", "spider". It is hard for a controlled label set to capture fine-grained semantics comprehensively. Tagging data offers the opportunity to train hashing models to capture fine-grained similarity relationships between images.

However, challenges always come with opportunities. Tagging data suffers from noises, vagueness and incompleteness. For example, in Figure 1 "2016" is a noise tag that does not describe image content. Vagueness mainly refer to synonyms and polysemous words. Incompleteness means people often do not tag images comprehensively. These issues hinder supervised training using tagging data.

To this end, a novel weakly-supervised deep hashing framework is proposed for hashing learning using tagging data. The framework consists of two stages: weakly-

*Corresponding author

supervised pre-training and supervised fine-tuning. In the first stage, rather than performing direct supervision on tags, we introduce a semantic embedding vector (sem-vector) for each image and perform learning of hashing and sem-vectors jointly. Specifically, we first project tags into an embedding space by word2vec [Mikolov *et al.*, 2013]. Then we perform sparse coding to extract the underlying "semantic words" (sem-words) which define the space for sem-vectors. Sem-words could represent atoms of semantics so that vagueness of tags can be alleviated. We further use average pooling to obtain image level semantic representations from the attached tags. This can also alleviate the influence from noise and polysemy tags (Pre-filtering can also be used to remove noise tags). To overcome the incompleteness problem, we exploit the co-occurrence information of tags at the sem-word level and also consider image content features. We formulate different criteria into a joint optimization framework for hashing model pre-training. In the second stage, any state-of-art supervised hashing training methods can be adopted.

The main contributions of this work are summarized as follows. 1) First, we propose to help improve image retrieval performance by using images tagged by Web users to train deep hashing models. We identify the benefits and validate them by empirical evaluation. 2) We develop a novel weakly-supervised hashing learning framework. It can well leverage tagging information and image content for deep hashing model training. 3) Third, we conduct experiments on two benchmark datasets to demonstrate that the state-of-art supervised hashing learning approaches can be significantly improved by the proposed framework.

2 Related Work

Considering the high dimensionality of images, one critical challenge in content-based image retrieval (CBIR) is how to efficiently generate search results. Recently, hashing learning is recognized as an important technique for fast approximate similarity search. Hashing learning methods focus on how to learn compact hash codes from the training data. Generally speaking, hashing methods can be categorized into two classes: unsupervised and supervised methods. Unsupervised hashing methods generate compact hash codes by using random projection or training on unlabeled data [Gionis *et al.*, 2000; Weiss *et al.*, 2008]. The most representative one is Locality-Sensitive Hashing (LSH) [Gionis *et al.*, 2000], which aims at maximizing the probability that similar data instances are mapped to similar binary codes. Recent studies have shown that using supervised information can boost the performance of binary hash codes. Supervised hashing methods [Kulis and Darrell, 2009; Liu *et al.*, 2012] usually use label information to guide pairwise similarity estimation for training effective hashing functions. For example, Chang *et al.* [Liu *et al.*, 2012] introduced Kernel-based Supervised Hashing (KSH) which maps images to binary hash codes by maximizing the separability of code inner products between similar and dissimilar pairs.

Recently, deep convolutional neural networks (CNNs) have shown promising results for hashing learning. Several state-of-art CNN-based hashing methods (e.g., [Lin *et al.*, 2016; Zhang *et al.*, 2015; Yan *et al.*, 2017]) were proposed to learn binary image hash codes. Lin *et al.* [Lin *et al.*, 2016] proposed an unsupervised CNN-based hashing approach called DeepBit which learned a set of non-linear projection functions to compute compact binary codes. Liu *et al.* introduced the Deep Supervised Hashing (DSH) method [Liu *et al.*, 2016] which took pairs of images (similar/dissimilar) as training inputs and encouraged the output of each image to approximate binary values. Another supervised deep hashing method, Deep Regularized Similarity Comparison Hashing (DRSCH), was proposed by Zhang *et al.* [Zhang *et al.*, 2015]. They utilized CNN with a triplet-based objective function for hashing function learning.

The most similar work to ours is Weakly-supervised Multimodal Hashing (WMH) [Tang and Li, 2017], which was also concerned with hashing learning using tags. However, they proposed a concrete hashing method with linear hashing functions, while we propose a general weakly-supervised deep hashing framework that can incorporate any state-of-art deep hashing models. Moreover, WMH directly used tags to calculate image similarity (on the textual modal), which would be more vulnerable to the noise and vagueness issues. In experiments we will integrate our weakly-supervised learning framework with representative and state-of-art hashing models to show our framework can indeed help boost their performance as compared to pure supervised learning.

In the multimedia research community, user-tagged images have received considerable attention. Related research works generally fall into two directions: tag refinement and learning with social images. For the former direction, the tag ranking problem was proposed in [Liu *et al.*, 2009] and improved in [Zhuang and Hoi, 2011] which aimed at distilling tags relevant to image content. Sun *et al.* [Sun *et al.*, 2011] tried to recommend relevant tags to images by exploiting co-occurrences of tags. Qian *et al.* [Qian *et al.*, 2014] further considered the diversity of the recommended tags. These works are orthogonal to ours in that these techniques could be used to pre-process tagging data in our problem. Nevertheless, in this work we only perform simply pre-filtering (details in experiments) in order to test the robustness of our weakly-supervised learning framework. Researchers have explored using user-tagged images for learning. In [Tang *et al.*, 2009], a sparse graph-based semi-supervised learning approach was presented to infer concepts of images using tags. Niu *et al.* [Niu *et al.*, 2015] developed a weakly-supervised matrix factorization method for user-tagged image parsing. Although these learning methods can well handle noisy tagging data, they were focused on coarse-grained concepts/categories and ignored rich fine-grained semantic information in tagging data which are important for similarity search. A more related problem is the tag-based image retrieval (TBIR) problem [Wu *et al.*, 2013; Xia *et al.*, 2015; Li and Tang, 2017]. However, since we are concerned with CBIR, methods for TBIR are not directly applicable.

3 The Framework

Figure 2 illustrates the flow chart of our proposed weakly-supervised hashing learning framework. In the first stage,

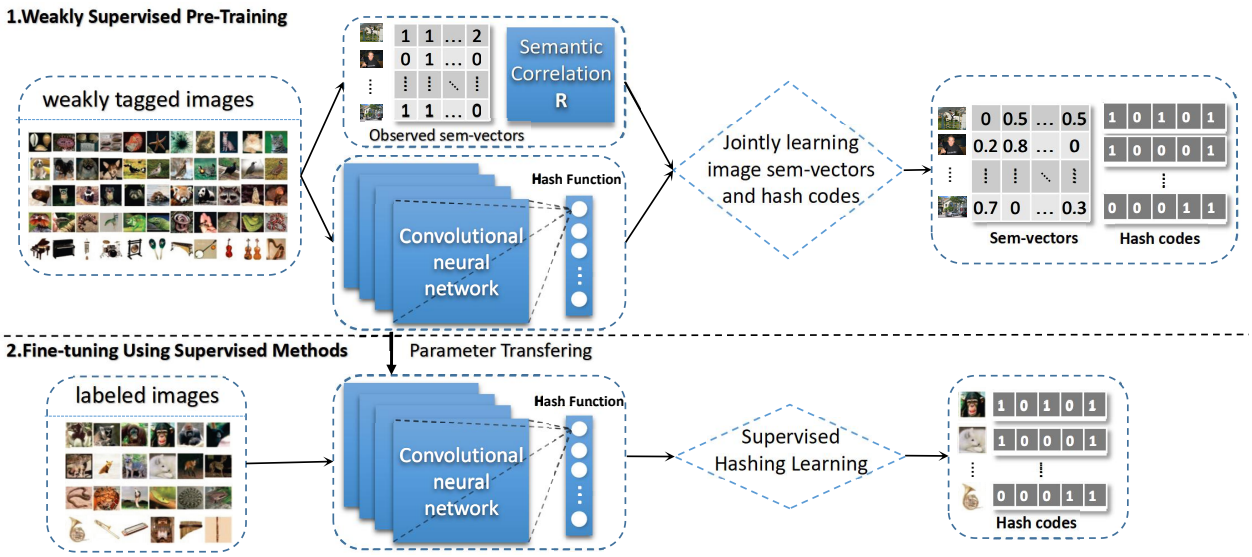

Figure 2: The proposed weakly-supervised deep hashing framework. In the first stage, we pre-train a CNN hashing model by jointly learning image sem-vectors and hash codes. It is weakly-supervised since the training of the hashing model is guided by the more robust sem-vectors which get useful information from tagging data and image content. The second stage performs supervised fine-tuning where any existing supervised deep hashing training method can be used.

we pre-train a CNN hashing model by jointly learning image semantic embedding vectors (sem-vectors) and hash codes. First, we propose a sparse coding scheme for mining "semantic words" (sem-words) from tagging data. The sem-words define the space for sem-vectors. Since sem-words can be regarded as semantic atoms, the tag vagueness problem can be significantly alleviated. Then, a unified optimization problem is established where we synthesize tagging information and image content for learning sem-vectors and hash codes, with consideration of alleviating the noise and incompleteness issues. After pre-training, we perform supervised fine-tuning of the CNN model. This is a general framework since any supervised hashing training method can be used for fine-tuning. It is called a weakly-supervised framework since in the first stage the training of the hashing model is guided indirectly by the more robust sem-vectors, rather than tags. In the next, we detail our framework.

3.1 Semantic Space Construction

As aforementioned, the motivation for creating a semantic space is to alleviate the vagueness issue of tags. We want to mine the sem-words hidden in tagging data so that images can be well represented in terms of their semantic information. Firstly, all the tags are converted to their vector representations through the word2vec tool [Mikolov *et al.*, 2013]. Although the tag vectors can capture semantic relationships between tags, they are dense vectors where each dimension does not exhibit a specific meaning. Hence, we propose to use sparse coding to mine sem-words based on tag vectors.

Formally, we are given a set of N images, and a set of L tags which users use to annotate the N images. The set of tag vectors obtained from word2vec is denoted by $\mathcal{T} = \{\mathbf{t}_1, \ldots, \mathbf{t}_L\}$ where $\mathbf{t}_j \in \mathbb{R}^D$. Each image i is annotated by a set of C_i tags $\{t_{i1}, \ldots, t_{iC_i}\}$. With a little abuse of notation, we also use t_{ij} to denote the index of tag t_{ij}. For example, $\mathbf{t}_{t_{ij}}$ refers to the tag vector of t_{ij}. Let $\mathbf{B} = [\mathbf{b}_1 \mathbf{b}_2 \ldots \mathbf{b}_M] \in \mathbb{R}^{D \times M}$ be the dictionary matrix of M sem-words to be learned. Each tag vector is converted to a sparse representation under the sem-words as follows:

$$\min_{\mathbf{B},\mathcal{U}} \sum_{j=1}^{L} \|\mathbf{t}_j - \mathbf{B}\mathbf{u}_j\|^2 + \lambda \|\mathbf{u}_j\|_1 \qquad (1)$$

where $\mathcal{U} = \{\mathbf{u}_1, \ldots, \mathbf{u}_L\}$ is the set of new representations of the L tags under $\mathbf{B}$, and $\|\cdot\|_1$ is the ℓ_1 regularization term. The dictionary $\mathbf{B}$ can be learned from user tags adaptively. When the number of sem-words M is large enough, each sem-word could be regarded as representing an atom of semantics. In this way, synonym tags tend to be assigned to the same sem-words, while polysemous tags would be connected to multiple sem-words with less intensities and consequently their negative influence can be reduced. For noise tags, they often appear randomly in different contexts, so the intensities are also low. The optimization problem can be solved with typical sparse coding algorithms. In this paper, we use the homotopy method [Donoho and Tsaig, 2008] since $\mathbf{u}_j$'s are expected to be highly sparse.

The learned sem-words define the constructed semantic space. We then compute the image level semantic representation $\mathbf{z}_i$ of image i from the attached tags by average pooling:

$$\mathbf{z}_i = \frac{1}{C_i} \sum_{j=1}^{C_i} \mathbf{u}_{t_{ij}} \qquad (2)$$

Each dimension k of $\mathbf{z}_i$ can be interpreted as measuring the intensity of sem-word $\mathbf{b}_k$ in image i. $\mathbf{z}_i$ is called *observed sem-vector* since it is directly obtained from the observed tags on image i. Average pooling can also suppress noise and polysemous tags due to the normalization term.

3.2 Joint Learning of Sem-vectors and Hash Codes

Even if the observed sem-vector $\mathbf{z}_i$ of image i alleviates the noise and vagueness issues of user tags, we still need to attack the incompleteness issue. Let $\hat{\mathbf{z}}_i$ be the sem-vector of image i to be learned. Firstly, $\hat{\mathbf{z}}_i$ should not deviate too much from the observed sem-vector $\mathbf{z}_i$. This is achieved by a least square loss

$$\min \|\hat{\mathbf{z}}_i - \mathbf{z}_i\|^2 \tag{3}$$

Second, we exploit tag correlation to cope with the incompleteness issue [Wu *et al.*, 2013]. Since tags are noisy and vague, we propose to perform correlation analysis at the sem-word level. We define a sem-word correlation matrix $\mathbf{R} \in \mathbb{R}^{M \times M}$, where R_{kq} represents the correlation between sem-word $\mathbf{b}_k$ and sem-word $\mathbf{b}_q$ as measured by their co-occurrence information as follows

$$R_{kq} = \frac{f_{k,q}}{f_k + f_q - f_{k,q}} \tag{4}$$

where f_k denotes the number of occurrences of $\mathbf{b}_k$ in the set of observed sem-vectors $\mathcal{Z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_N\}$[1], and $f_{k,q}$ is the number of co-occurrences of $\mathbf{b}_k$ and $\mathbf{b}_q$ in $\mathcal{Z}$. For image i, we expect the sem-vector $\hat{\mathbf{z}}_i$ to be consistent with the correlation matrix $\mathbf{R}$:

$$\min \frac{1}{2} \sum_{k=1}^{M} \sum_{q=1}^{M} R_{kq} (\hat{z}_{ik} - \hat{z}_{iq})^2 = \hat{\mathbf{z}}_i^\top \mathbf{L} \hat{\mathbf{z}}_i \tag{5}$$

Here $\hat{z}_{ik}$ is the k-th entry of $\hat{\mathbf{z}}_i$. $\mathbf{L} = \mathbf{D}^R - \mathbf{R}$ is the positive semi-definite Laplacian matrix, in which $\mathbf{D}^R$ is a diagonal matrix with $D^R_{kk} = \sum_{q=1}^{M} R_{kq}$. Eq. (5) means if sem-words $\mathbf{b}_k$ and $\mathbf{b}_q$ are highly correlated (large R_{kq}), their intensity values should be similar in $\hat{\mathbf{z}}_i$. This smoothing technique can help alleviate the incompleteness issue.

Finally, image content can also reflect semantic information in images to some degree [Liu *et al.*, 2009]. A straightforward idea could be to force the similarities of images estimated using sem-vectors to comply with those obtained from low-level image features. However, this would amplify the semantic gap problem of low-level features. Hence, it is better to use high-level feature output generated from a CNN network. On the other hand, image hash codes are also generated by binarizing the output of a CNN network. Let $\mathcal{H} = \{\mathbf{h}_1, \ldots, \mathbf{h}_N\}$ be the set of un-binarized hash codes for our image set. $\mathbf{h}_i$ is computed by $\mathbf{h_i} = \mathcal{F}(\mathbf{x}_i; \mathcal{W})$, where $\mathbf{x}_i$ represents the input of image i (e.g. pixel values), and $\mathcal{F}(\cdot; \mathcal{W})$ abstracts the computation of a CNN network where any state-of-art CNN-based hashing models can be used to instantiate it. In the weakly-supervised stage, we require image similarities estimated via (un-binarized) hash codes to comply with those estimated via sem-vectors. Hence, we combine the two criteria efficiently into one objective term:

$$\min \sum_{i,j} \|\hat{\mathbf{z}}_i^\top \hat{\mathbf{z}}_j - \mathbf{h}_i^\top \mathbf{h}_j\|^2 \tag{6}$$

where sem-vectors and (un-binarized) hash codes reinforce each other to achieve better similarity estimation of images. Since both $\hat{\mathbf{z}}$'s and $\mathbf{h}$'s are variables, this also acts as a soft constraint which is suitable for weakly-supervised learning.

By incorporating (3), (5) and (6), the overall loss function is:

$$\begin{aligned} \mathcal{L}(\{\hat{\mathbf{z}}\}, \{\mathbf{h}\}) =& \frac{1}{2} \sum_{i=1}^{N} \|\hat{\mathbf{z}}_i - \mathbf{z}_i\|^2 + \frac{\beta}{2} \sum_{i=1}^{N} \hat{\mathbf{z}}_i^\top \mathbf{L} \hat{\mathbf{z}}_i \\ &+ \frac{\delta}{2} \sum_{i,j} \left\| \hat{\mathbf{z}}_i^\top \hat{\mathbf{z}}_j - \mathbf{h}_i^\top \mathbf{h}_j \right\|^2 \end{aligned} \tag{7}$$

where $\beta > 0$ and $\delta > 0$ are trade-off parameters. Note that compared to coarse-grained manual labels, sem-vectors can provide more fine-grained estimation of image similarities, which could lead to a better hashing model.

3.3 Optimization

The objective function defined in Eq. (7) is non-convex with respect to $\{\hat{\mathbf{z}}\}$ and $\{\mathbf{h}\}$ jointly, which makes it difficult to optimize. Fortunately, this problem is differentiable with respect to either one of the two sets of parameters when the other set is fixed. Therefore, we solve the problem by coordinate descent. When updating either $\{\hat{\mathbf{z}}\}$ or $\{\mathbf{h}\}$, mini-batch stochastic gradient descent is used. We set mini-batch size to 64. In particular, we alternatively update $\{\hat{\mathbf{z}}\}$ and $\{\mathbf{h}\}$ by the following two steps until convergence.

Step 1: Fix $\{\mathbf{h}\}$, Update $\{\hat{\mathbf{z}}\}$. We sample a mini-batch $\mathcal{S}$ of training images. By taking the partial derivative of Eq. (7) with respect to each $\hat{\mathbf{z}}_i$ under the mini-batch $\mathcal{S}$, we can obtain the gradient in Eq. (8) and update $\{\hat{\mathbf{z}}\}$ accordingly:

$$\nabla_{\hat{\mathbf{z}}_i} \mathcal{L} = (\hat{\mathbf{z}}_i - \mathbf{z}_i) + \beta \mathbf{L} \hat{\mathbf{z}}_i + \delta \sum_{j \in \mathcal{S}} (\hat{\mathbf{z}}_i^\top \hat{\mathbf{z}}_j - \mathbf{h}_i^\top \mathbf{h}_j) \hat{\mathbf{z}}_\mathbf{j} \tag{8}$$

Step 2: Fix $\{\hat{\mathbf{z}}\}$, Update $\{\mathbf{h}\}$. Similar to step 1, we use the stochastic gradient descent and back-propagate the error into the underlying CNN network to optimize the hashing model. Given a mini-batch $\mathcal{S}$, the gradient with respect to $\mathbf{h}_i$ is:

$$\nabla_{\mathbf{h}_i} \mathcal{L} = \delta \sum_{j \in \mathcal{S}} (\mathbf{h}_i^\top \mathbf{h}_j - \hat{\mathbf{z}}_i^\top \hat{\mathbf{z}}_j) \mathbf{h}_j \tag{9}$$

Algorithm 1 Weakly-supervised hashing learning

Input: A set of user-tagged training images
  Parameters: λ, β
Output: Pre-trained hashing model
  Compute the observed sem-vectors $\{\mathbf{z}\}$ for the images.
  Initialize $\{\hat{\mathbf{z}}\}$ and $\{\mathbf{h}\}$ (CNN model).
  repeat
    Set the updating frequencies: p, m
    for 1,...,p **do**
      Update $\{\hat{\mathbf{z}}\}$ according to Step 1.
    end for
    for 1,...,m **do**
      Update $\{\mathbf{h}\}$ (CNN model) according to Step 2.
    end for
  until the solution converges

[1] A sem-word $\mathbf{b}_k$ is deemed to be present in $\mathbf{z}_i$ if the k-th entry of $\mathbf{z}_i$ is nonzero.

	NUS-WIDE	MIR Flickr-1M
# Images	269,648	897,500
# Tags	5018	1000
Avg./max # T/I	7.9/201	12.7/76
Avg./max # I/T	677.1/20,140	416.5/76,890
# of manually labeled images	269,648	25,000
# manual labels	81	38

Table 1: Statistics for the datasets used in the experiments. "T/I" and "I/T" mean "Tags per image" and "Images per tag" respectively.

The weakly-supervised training algorithm is summarized in Algorithm 1, where p and m are updating frequencies for the two steps. To accelerate the optimization, we dynamically set p and m. At the beginning, the ratio $p : m$ is empirically set to 10:1, which means that $\{\hat{\mathbf{z}}\}$ will be optimized preferentially. This is intuitive since initially we want to first incorporate useful content information into sem-vectors. Afterwards, the ratio will gradually turn to 1:10 to emphasize hashing model training. The CNN model is initialized by a pre-trained network and each $\hat{\mathbf{z}}_i$ is initially set to $\mathbf{z}_i$.

For supervised fine-tuning, the optimization follows the respective supervised hashing methods for the hashing models.

4 Experiments

4.1 Datasets and Evaluation Metrics

NUS-WIDE [Chua *et al.*, 2009]: it is a large-scale community contributed benchmark image dataset for evaluating multimedia tasks. There are 269,648 images associated with 5,018 tags annotated by amateur users. We do not pre-filter noise tags since it has already been done. The dataset provides 81 ground-truth concept labels which are used for performance evaluation. All of the 269,648 images are manually labeled, so this dataset can be used to test whether providing additional tagging data for labeled images is of value. We randomly sample 5000 query images and use the rest for training and retrieval.

MIR Flickr-1M [Huiskes *et al.*, 2010]: it consists of one million images collected from Flickr that are annotated by more than 10,000 user tags. We pre-filter tags to simply retain the first 1000 most popular tags in the dataset. We do not use complicated tag refinement methods in order to show that such simple rule works for our method. This reduces the number of images with user tags to 897,500. In this dataset, only 25,000 images have ground-truth manual labels. Among the 25,000 labeled images, 2000 images are randomly selected as test queries and the remaining images are used for training and retrieval.

Table 1 summarizes some statistics of these datasets. Regarding evaluation metrics, we employ mean average precision (MAP), precision@k and precision curves, where a retrieved image is deemed relevant to the query if it shares at least one common label with the query.

4.2 Compared Algorithms and Parameter Setting

To evaluate the proposed weakly-supervised hashing learning framework, we apply it to several state-of-the-art supervised hashing methods, including KSH[2] [Liu *et al.*, 2012], DSH [Liu *et al.*, 2016] and DRSCH [Zhang *et al.*, 2015]. The Weakly-supervised Pre-trained (WP) versions of these methods are named as WP-KSH, WP-DSH and WP-DRSCH respectively. In addition, we also employ two representative unsupervised hashing methods as baselines, i.e., LSH [Gionis *et al.*, 2000] and DeepBit [Lin *et al.*, 2016]. For fair comparison, we use pre-trained VGG16-net [Simonyan and Zisserman, 2015] on the ImageNet dataset as the base net for CNN-based methods (i.e., DeepBit, DSH and DRSCH and their WP versions). For other hashing methods, the output of last fully-connected layer of the pre-trained VGG16-net will be treated as input features of images.

For the proposed framework, we empirically fix the tag vector size D to 200 and the sem-word number M to 512 (by preliminary experiments we find the performance do not change much when they are large enough). The parameter λ in Eq. (1) controls sparsity of the solution; A larger λ will lead to a sparser solution. Empirically, we find that keeping the sparsity to be around 10% of the whole size yields good results, which corresponds to $\lambda = 0.4$. The parameters β and δ in Eq. (7) are set according to cross validation on training data. For the other methods, we adopt the best parameter configurations as described in their papers.

4.3 Experimental Results on NUS-WIDE

Fig. 3 shows the results of different methods on the NUS-WIDE dataset. Fig. 3(a) and (b) report the performance measured by precision within Hamming distance 2 and Precision@500, respectively, for different lengths of hash codes. Fig. 3(c) illustrates precision curves utilizing 64-bit codes. It can be seen that supervised methods always achieve better performance than the unsupervised hashing methods. WP-KSH, WP-DSH and WP-DRSCH outperform their respective supervised counterparts. Recall that in this dataset all the images are with manual labels. The results indicate that our weakly-supervised framework can still boost the retrieval performance by leveraging weak tagging information for a fully labeled image dataset. This verifies our analysis that tagging data can provide richer semantic information which would help better characterize the fine-grained similarity relationships between images. Our framework captures such information by the objective term in Eq. (6). In order to further demonstrate the effectiveness of our weakly-supervised hashing learning algorithm (Algorithm 1), we concentrate on learning using tagging data only. We remove the fine-tuning process for WP-KSH, WP-DSH and WP-DRSCH and call them WP-KSH$^-$, WP-DSH$^-$ and WP-DRSCH$^-$ respectively. We then apply KSH, DSH and DRSCH on tagging data directly (i.e. treat tags as manual labels) for comparison. The corresponding schemes are called KSH$^-$, DSH$^-$ and DRSCH$^-$ respectively. The results measured by MAP with different code lengths are listed in Table 2. We find that the performance of WP-KSH$^-$, WP-DSH$^-$ and WP-DRSCH$^-$ increases by at least 27% compared to the supervised methods. The results show that the traditional supervised hash-

[2]Actually, a hashing method can be integrated into our framework as long as it can be trained by gradient descent.

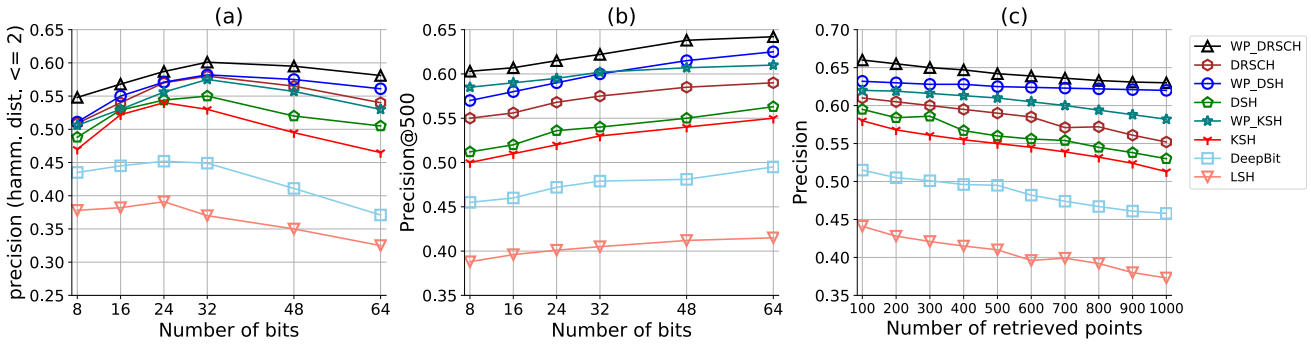

Figure 3: Performance comparison on the NUS-WIDE dataset. (a) Precision within Hamming radius 2; (b) Precision@500; (c) Precision curves with 64 hash bits.

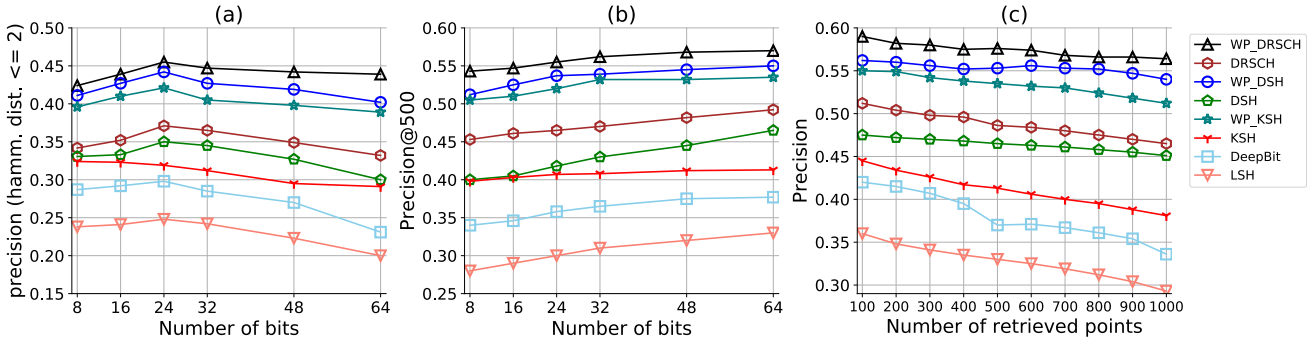

Figure 4: Performance comparison on the MIR Flickr-1M dataset. (a) Precision within Hamming radius 2; (b) Precision@500; (c) Precision curves with 64 hash bits.

Method	Hamming ranking (MAP, %)			
	8 bits	16 bits	32 bits	64 bits
KSH$^-$	12.63	15.32	17.11	18.32
WP-KSH$^-$	**40.64**	**43.55**	**46.76**	**49.34**
DSH$^-$	15.12	18.45	20.15	23.34
WP-DSH$^-$	**44.02**	**47.72**	**49.76**	**52.96**
DRSCH$^-$	18.87	21.56	25.45	28.97
WP-DRSCH$^-$	**50.09**	**53.21**	**55.50**	**56.43**

Table 2: Comparison of the proposed weakly-supervised hashing learning algorithm to supervised methods on training with tagging data only. The minus sign means the fine-tuning process is removed (for WP- methods) or we apply the method directly on tagging data (for supervised methods).

Figure 5: Retrieval results obtained by (a) WP-DRSCH and (b) DRSCH in MIR Flickr-1M for two queries. Each query is shown on the left, with top ranked images shown to the right.

ing methods are not directly applicable on weak tags. The proposed weakly-supervised hashing learning algorithm can better handle user tags.

4.4 Experimental Results on MIR Flickr-1M

For this dataset, weakly-supervised pre-training is performed on all the 897,500 images, while supervised fine-tuning can only utilize 23,000 images. Therefore, we can use this dataset to test whether providing more images with weak tags is beneficial. The results are shown in Fig. 4. We find that the performance of WP-KSH, WP-DSH and WP-DRSCH increases by 10% on average compared to their supervised counterparts. This demonstrates that pre-training on much larger collections of user-tagged images can be very beneficial for hashing learning. Fig. 5 shows the top ranked images for two query images obtained by WP-DRSCH and DRSCH respectively (with 64-bit hash codes). We see that WP-DRSCH can better capture fine-grained similarities than DRSCH, e.g. "truck" vs. "vehicle" in the first example.

5 Conclusion

In this work, we explored using user generated tagging data for hashing learning and image retrieval. We proposed a novel weakly-supervised deep hashing framework. The key

idea is to learn a semantic vector for each image and use it to guide hashing learning from tagging data. The framework is general and can be combined with any supervised hashing learning methods based on deep neural networks. Extensive experiments demonstrated its effectiveness. There are several interesting directions along which we intend to extend this work. The first is to leverage more semantic information for diversity search. Another one is to improve the optimization strategy to accelerate the training speed of our method.

Acknowledgments

This research was supported by the National Natural Science Foundation of China (Grant Nos. 61522206, 61373118, 61672409), the Major Basic Research Project of Shaanxi Province (Grant No. 2017ZDJC-31), the Science and Technology Plan Program in Shaanxi Province of China (Grant No. 2017KJXX-80) and the Changjiang Scholars and Innovative Research Team in University (IRT_17R87).

References

[Chua *et al.*, 2009] Tatseng Chua, Jinhui Tang, Richang Hong, Haojie Li, Zhiping Luo, and Yantao Zheng. Nus-wide: a real-world web image database from national university of singapore. In *CIVR*, page 48, 2009.

[Donoho and Tsaig, 2008] David L. Donoho and Yaakov Tsaig. Fast solution of ℓ_1 -norm minimization problems when the solution may be sparse. *IEEE TIP*, 54(11):4789–4812, 2008.

[Gionis *et al.*, 2000] Aristides Gionis, Piotr Indyk, Rajeev Motwani, et al. Similarity search in high dimensions via hashing. In *VLDB*, pages 518–529, 2000.

[Huiskes *et al.*, 2010] Mark J. Huiskes, Bart Thomee, and Michael S. Lew. New trends and ideas in visual concept detection: The mir flickr retrieval evaluation initiative. In *ACM MIR*, pages 527–536, 2010.

[Irie *et al.*, 2014] Go Irie, Zhenguo Li, Xiaoming Wu, and Shihfu Chang. Locally linear hashing for extracting non-linear manifolds. In *CVPR*, pages 2115–2122, 2014.

[Kulis and Darrell, 2009] Brian Kulis and Trevor Darrell. Learning to hash with binary reconstructive embeddings. In *NIPS*, pages 1042–1050, 2009.

[Li and Tang, 2017] Zechao Li and Jinhui Tang. Weakly supervised deep matrix factorization for social image understanding. *IEEE TIP*, 26(1):276–288, 2017.

[Lin *et al.*, 2016] Kevin Lin, Jiwen Lu, Chu Song Chen, and Jie Zhou. Learning compact binary descriptors with unsupervised deep neural networks. In *CVPR*, pages 1183–1192, 2016.

[Liu *et al.*, 2009] Dong Liu, Xiansheng Hua, Linjun Yang, Meng Wang, and Hongjiang Zhang. Tag ranking. In *WWW*, pages 351–360, 2009.

[Liu *et al.*, 2012] Wei Liu, Jun Wang, Rongrong Ji, Yugang Jiang, and Shihfu Chang. Supervised hashing with kernels. In *CVPR*, pages 2074–2081, 2012.

[Liu *et al.*, 2016] Haomiao Liu, Ruiping Wang, Shiguang Shan, and Xilin Chen. Deep supervised hashing for fast image retrieval. In *CVPR*, pages 2064–2072, 2016.

[Mikolov *et al.*, 2013] Tomas Mikolov, Greg Corrado, Kai Chen, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *ICLR*, pages 1–12, 2013.

[Niu *et al.*, 2015] Yulei Niu, Zhiwu Lu, Songfang Huang, Peng Han, and Jirong Wen. Weakly supervised matrix factorization for noisily tagged image parsing. In *IJCAI*, pages 3749–3755, 2015.

[Qian *et al.*, 2014] Xueming Qian, Xiansheng Hua, Yuanyan Tang, and Tao Mei. Social image tagging with diverse semantics. *IEEE cybernetics*, 44(12):2493–2508, 2014.

[Simonyan and Zisserman, 2015] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *NIPS*, 2015.

[Sun *et al.*, 2011] Aixin Sun, SouravS. Bhowmick, and Junan Chong. Social image tag recommendation by concept matching. In *ACM MM*, pages 1181–1184, 2011.

[Tang and Li, 2017] Jinhui Tang and Zechao Li. Weakly-supervised multimodal hashing for scalable social image retrieval. *IEEE TCSVT*, PP(99):1–1, 2017.

[Tang *et al.*, 2009] Jinhui Tang, Shuicheng Yan, Richang Hong, Guo Jun Qi, and Tat Seng Chua. Inferring semantic concepts from community-contributed images and noisy tags. In *ACM MM*, pages 223–232, 2009.

[Weiss *et al.*, 2008] Yair Weiss, Antonio Torralba, and Rob Fergus. Spectral hashing. In *NIPS*, pages 1753–1760, 2008.

[Wu *et al.*, 2013] Lei Wu, Rong Jin, and Anil K. Jain. Tag completion for image retrieval. *IEEE TPAMI*, 35(3):716–27, 2013.

[Xia *et al.*, 2015] Zhaoqiang Xia, Xiaoyi Feng, Jinye Peng, Jun Wu, and Jianping Fan. A regularized optimization framework for tag completion and image retrieval. *Neurocomputing*, 147(1):500–508, 2015.

[Yan *et al.*, 2017] Xinyu Yan, Lijun Zhang, and Wujun Li. Semi-supervised deep hashing with a bipartite graph. In *IJCAI*, pages 3238–3244, 2017.

[Zhang *et al.*, 2015] Ruimao Zhang, Liang Lin, Rui Zhang, Wangmeng Zuo, and Lei Zhang. Bit-scalable deep hashing with regularized similarity learning for image retrieval and person re-identification. *IEEE TIP*, 24(12):4766–4779, 2015.

[Zhao *et al.*, 2015] Fang Zhao, Yongzhen Huang, Liang Wang, and Tieniu Tan. Deep semantic ranking based hashing for multi-label image retrieval. In *CVPR*, pages 1556–1564, 2015.

[Zhuang and Hoi, 2011] Jinfeng Zhuang and Steven C.H. Hoi. A two-view learning approach for image tag ranking. In *WSDM*, pages 625–634, 2011.