Sparse Similarity Matrix Learning for Visual Object Retrieval

Zhicheng Yan and Yizhou Yu

Abstract— **Tf-idf weighting scheme is adopted by state-of-the-art object retrieval systems to reflect the difference in discriminability between visual words. However, we argue it is only suboptimal by noting that tf-idf weighting scheme does not take quantization error into account and exploit word correlation. We view tf-idf weights as an example of diagonal Mahalanobis-type similarity matrix and generalize it into a sparse one by selectively activating off-diagonal elements. Our goal is to separate similarity of relevant images from that of irrelevant ones by a safe margin. We satisfy such similarity constraints by learning an optimal similarity metric from labeled data. An effective scheme is developed to collect training data with an emphasis on cases where the tf-idf weights violates the relative relevance constraints. Experimental results on benchmark datasets indicate the learnt similarity metric consistently and significantly outperforms the tf-idf weighting scheme.**

I. Introduction

Recently, object retrieval systems based on bags of visual words (BoVW) have achieved both good retrieval precision and high scalability [23][7][1][22]. In BoVW model [8], image features are quantized into "visual words" in a learnt vocabulary. To reflect the difference in discriminability between visual words, tf-idf (Term Frequency Inverse Document Frequency) weighting scheme assigns larger weights to infrequent visual words across images by simply assuming they are more discriminative.

Although tf-idf weights has been used to achieve state-of-the-art performance[1], we argue that it is only suboptimal. First, existing works [4][11][15][5] on image classification and retrieval show better performance can be obtained by supervised learning of similarity metric. Given labeled image, we can improve tf-idf weight such that images with same label are more similar to each other than differently labeled images. Second, it is well known that quantization error from the use of a large-sized vocabulary, which is the case for object retrieval, can significantly hurt retrieval performance [24]. We consider the tf-idf weights as a special example of a Mahalanobis-type similarity matrix, where the only non-zero elements are the diagonal ones. Thus, tf-idf weighting scheme fails to take quantization error into account. Soft word assignment [24] was used to compensate quantization error while we achieve similar goal by selectively activating off-diagonal elements of similarity matrix during learning. Third, diagonal matrix of tf-idf weights also ignores correlation between visual words, which can also be exploited to improve retrieval performance.

To this end, we propose an image similarity learning approach for object retrieval. We generalize diagonal similarity matrix to take into consideration correlation among visual words as well as quantization error from hard word assignment. Our approach is different from previous work [5] on metric learning for image classification which learns a full Mahalanobis distance matrix for a small vocabulary (e.g. d=10K). Due to the large vocabulary used for object retrieval (e.g. d=500K), a full similarity matrix in our setting would make similarity learning intractable since the training time would grow by at least tens of thousands of orders of magnitude and a huge number of training images would be needed. Therefore, we selectively activate only a small number of off-diagonal elements in the similarity matrix which are shown to be sufficient for compensating quantization error and exploiting word correlation. We demonstrate that the resulting sparse similarity matrix can be optimally learnt through supervised training and retrieval performance can significantly benefit from such learning. Our learning procedure makes use of labeled images as training data. We explicitly distinguish relevant images from irrelevant ones and formulate this task as a constrained optimization problem. To balance the goals of learning an optimal similarity and making learning tractable, we develop an effective scheme for collecting training data with a particular emphasis on failure cases of tf-idf weights. We evaluate our method on benchmark datasets. Experimental results indicate the learnt similarity metric consistently and significantly outperforms the tf-idf weighting scheme.

II. Related Work

A. Large-Scale Object Retrieval

Recently, many efforts have been made to improve object retrieval.[22] proposes hierarchical vocabulary tree for the use of large vocabulary and efficient retrieval. Alternatively, approximate k-means clustering can produce a flat vocabulary with similar computational complexity but can achieve better performance [23]. Geometric constraints are also exploited [23] to improve similarity computation. Geometrically verified images can be used to perform reliable query expansion [7][6]. [30] further improves retrieval performance by grouping local features and imposing weak geometric constraints.

In addition, many techniques have been proposed to improve image similarity measure. Conventionally, Euclidean distance

Zhicheng Yan is with Department of Computer Science at University of Illinois at Urbana-Champaign (email: zyan3@illinois.edu). Yizhou Yu is with Department of Computer Science at The University of Hong Kong, Hong Kong (email: yizhouy@acm.org).

is used for comparing patch-based feature descriptors (e.g. SIFT [18]). However, it is arguable it is not the best choice for such a task by noting that SIFT essentially is a histogram of gradients. Using Euclidean distance for feature quantization results in a loss in retrieval performance which can be alleviated by soft word assignment [24]. Alternatively, [16] combines traditional feature quantization and novel Hamming Embedding (HE) to reduce quantization error. In [21], probabilistic relation between words in a fine vocabulary is learnt to improve the similarity measure of images. Recently, Hellinger kernel [1] has been found to better describe distance between SIFT descriptors and can replace Euclidean distance at no additional cost. Distance metric for descriptors can also be learnt in a principled way which achieves both better discrimination and lower dimensionality [25][3][28][19][13].

B. Distance Metric Learning for Image Classification/Retrieval

Distance metric learning has been extensively studied in machine learning community [31][27][2][29][12]. An introduction on this topic can be found in [32]. Due to the limited space, we only mention the most relevant ones to our work. For image classification/retrieval, distance metric learning techniques are used to obtain a better similarity measure than simple L^2 distance. In [11][10], local distance metric for individual images are learnt in a globally consistent manner. In the framework of information-theoretic metric learning [9], a full learnt Mahalanobis distance metric is integrated into a set of locality-sensitive hashing functions for fast retrieval [15]. As data scale grows, online image similarity learning becomes more appealing to match the scenario where training data is streamed into a learning algorithm [14][5].

Our work to some extent is similar to [4] where weights associated with vocabulary words are learnt in a supervised manner. However, we distinguish our work from theirs in two aspects. First, [4] uses densely sampled features and image similarity is defined by SPMK [17], which is widely used for scene classification. However, for object retrieval, state-of-the-art performance is achieved using sparse features and dissimilarity is measured by L^2 distance between image BoVW vectors [22][23]. It is easy to see that such dissimilarity is equivalent to a similarity metric where cross similarity between two images is further normalized by self-similarities of two images. This gives rise to a more complex optimization problem in the learning stage. Second, [4] only learns a small diagonal similarity matrix while we learn a large sparse one. This is because in their application only a small vocabulary (e.g. 2000 words) is required and quantization error is negligible. However, for object retrieval, a large vocabulary (e.g. 500k) is essential for high retrieval performance [23] but meanwhile introduces significant quantization error [24]. Therefore, we learn both diagonal and sparse off-diagonal elements in the similarity matrix to maximize performance improvements.

The rest of this paper is organized as follows. In Section III, we formulate our similarity learning task as a constrained minimization problem. Section IV elaborates on the details of our implementation. We show our results and compare them to related work in Section V. Section VI concludes this paper.

III. Learning Similarity Metric

A. Bag-of-visual-words Object Retrieval

For object retrieval, dissimilarity D_{ij} between two images is defined as Euclidean distance between L^2 normalized image BoVW vectors.

$$D_{ij} = \left\| \hat{\mathbf{I}}_i - \hat{\mathbf{I}}_j \right\|_2^2 = 2 - 2\hat{\mathbf{I}}_i^T \hat{\mathbf{I}}_j \tag{1}$$

$$\hat{\mathbf{I}}_i = \frac{\mathbf{w} * \mathbf{I_i}}{\|\mathbf{w} * \mathbf{I_i}\|_2} \tag{2}$$

where $\mathbf{w}$ is a weighting vector which is usually assigned with tf-idf weights. $\mathbf{I_i}$ is a L^1 normalized term frequency vector and $\hat{\mathbf{I}}_i$ a L^2 normalized BoVW vector for image i. Operator $*$ in Equation (2) denotes element-wise product. For the rest of this paper, we define a normalized similarity metric which is equivalent to the above dissimilarity measure. That is, a pair of images with the maximum normalized similarity have the minimum dissimilarity between them.

$$\hat{S}_{ij} = (\hat{\mathbf{I}}_i^T \hat{\mathbf{I}}_j)^2 = \frac{(\mathbf{I}_i^T \mathbf{M} \mathbf{I}_j)^2}{(\mathbf{I}_i^T \mathbf{M} \mathbf{I}_i)(\mathbf{I}_j^T \mathbf{M} \mathbf{I}_j)} = \frac{S_{ij}^2}{S_{ii} S_{jj}} \tag{3}$$

where $\mathbf{M} = diag([w_1^2, ..., w_n^2]^T)$ is a Mahalanobis-type similarity matrix. S_{ij} can be viewed as unnormalized cross similarity between a pair of images (i, j). We can see that $\hat{S}_{ij}$ is computed by normalizing S_{ij} using self-similarities S_{ii} and S_{jj}. In a retrieval session, database images are sorted according to their normalized similarity with a query image.

B. Generalized Image Similarity Metric

The similarity matrix $\mathbf{M}$ in Equation (3) can be generalized to include non-zero off-diagonal elements for the reasons explained in section I . Activating off-diagonal elements allows the co-occurrence of two different visual words across two images to contribute to their similarity score. However, including all off-diagonal elements in learning stage is not feasible due to the huge number of unknowns in the full similarity matrix. Besides, it is well known that in the event of a large vocabulary generated by approximate k-means clustering, quantization error occurs when matching features are quantized to different visual words due to descriptor noise. When choosing the size of a vocabulary, we need to consider the trade-off between descriptor variation and quantization granularity. According to this observation, soft word assignment outperforms hard word assignment by distributing fractional weights to *k*-nearest words (*k*-NN) to alleviate quantization error [24]. In the same spirit, we activate only those off-diagonal elements that belong to the *k*-NN of a diagonal word. We assume pairwise interaction between other types of words is negligible. Thus the generalized similarity matrix is still sparse and the number of

non-zero weights to be learnt is increased only by a constant factor.

Formally, given a vocabulary $V = \{V_i\}$ of size n, we first define a binary matrix mask $\mathbf{B^d}$ of size $n \times n$, where diagonal elements have value 1 and off-diagonal elements have value 0. This represents only diagonal elements in $\mathbf{M}$ are allowed to be non-zero. Then, for each row i in $\mathbf{B^d}$, we set the corresponding elements of the k-NN of word V_i to be 1. This defines a sparse matrix mask, $\mathbf{B^n}$. Last, since the nearest-neighbor relation is in general not symmetric, $\mathbf{B^n}$ is usually not symmetric. We further set $\mathbf{B^s} = \mathbf{B^n}|(\mathbf{B^n})^\mathbf{T}$ to obtain a symmetric matrix mask, which facilitates learning a symmetric similarity metric. In the following sections, we will use the notation $\mathbf{M^s}$ to denote the similarity matrix we aim to learn.

C. Supervised Similarity Metric Learning

Our similarity metric learning is inspired by relative comparison [27]. Given a training set of image triplets (i, j, l) and prior knowledge such as image i is more similar to image j than image l, we learn a similarity matrix $\mathbf{M^s}$ to satisfy $S_{ij} \geq S_{il} + \gamma$, where γ is a positive safe margin separating S_{ij} and S_{il}. Formally, we solve the following constrained minimization.

$$\mathbf{M^s} = \arg\min \sum_{(i,j,l)\in \boldsymbol{T}} w_{(i,j,l)}[\hat{S}_{il} - \hat{S}_{ij} + \gamma]_+ \quad (4)$$

$$s.t. \quad \begin{cases} \mathbf{M^s}(u,v) \geq 0, & \text{if } \mathbf{B^s}(u,v) = 1; \\ \mathbf{M^s}(u,v) = 0, & \text{if } \mathbf{B^s}(u,v) = 0, \end{cases} \quad (5)$$

where $\boldsymbol{T}$ denotes the training set of image triplets, $\hat{S}_{ij}$ is defined in Equation (3), $[z]_+ = max(0, z)$ is the hinge loss function and $w_{(i,j,l)}$ is triplet weight. The weighting scheme of training triplets will be detailed in section IV-B.

The constraints in the above minimization require both diagonal and off-diagonal elements of the similarity matrix to be non-negative and ensure a non-negative similarity metric by noting that elements of the normalized image BoVW vector $\hat{\mathbf{I}}_i$ are all non-negative. Requiring diagonal elements to be non-negative is a natural choice as all visual words can positively contribute to similarity but to different extents. Non-negative off-diagonal elements not only compensate quantization errors by allowing differently quantized matching features to contribute to image similarity but also exploit correlation between words.

The constrained minimization problem in (4) is in general non-convex. The converged solution can be only locally optimal. However, since our main goal is to improve retrieval performance over tf-idf weights, we use tf-idf weights to initialize the diagonal elements of $\mathbf{M^s}$. Starting from such an initialization, we are guaranteed to converge to a solution better than tf-idf weights. This is also confirmed by our experimental results in section V.

IV. Implementation

We solve our minimization problem in (4) using the constrained optimization solver in Matlab. There exist several issues to be addressed. i) How can we collect useful training triplets when the number of enumerable triplets is extremely large? ii) How can we efficiently evaluate the value of the objective function and its gradient when the number of training triplets is large? iii) How can we avoid overfitting when the number of free variables in $\mathbf{M^s}$ is large? Our solutions to these questions are discussed below.

TABLE I
SPECIFICATION OF DATASETS

Dataset	# images	# objects	# *Positive* images
Oxford5K (D1)	5.1K	11	571
Flickr100K (D2)	100.5K	0	0
Paris6K (D3)	6.4K	11	1791
Flickr95K (D4)	94.7K	0	0

A. Datasets

We will use benchmark datasets *Oxford5K* (*D1*) [23],*Flickr100K* (*D2*) [23], *Paris6K* (*D3*) [24] and a home-made dataset *Flickr95K* (*D4*) for evaluation purpose. *Oxford5K* and *Flickr100K* datasets together form *Oxford105K* dataset. *Paris6K* and *Flickr95K* datasets together form *Paris101K* dataset. Both *Flickr100K* and *Flickr95K* datasets serve the purpose of background images. In *Oxford5K* and *Paris6K* datasets, images are assigned with one of four possible labels for each landmark, namely 1) *Good* 2) *Ok* 3) *Junk* 4) *Absent*. We consider images with *Good* or *Ok* label as *Positive* images. Thus *Oxford5K* and *Paris6K* consist of *Positive*, *Junk* and background images. See Table I for dataset details.

B. Collecting Training Triplets

The first step of similarity learning is to collect a set of training image triplets. On one hand, exhaustively enumerating all possible triplets even for a moderately large dataset is prohibitively expensive. Assume a dataset of 6k images including 600 images of interest (e.g. *Positive* images) and 5.4k background images. The 600 *Positive* images include 10 objects of interest, each of which has 60 images. A triplet can be obtained by first choosing a pair of *Positive* images (i, j) of the same object and then choosing the third image l to be a *Positive* image of a different object or a background image. The number of all possible triplets is $60 * 59 * 5940 * 10 = 210M$ which would make the optimization prohibitively slow. On the other hand, the number of unknowns in $\mathbf{M^s}$ is usually large. To avoid overfitting, we still need to collect a sufficiently large number of useful training triplets.

We adopt an effective strategy to sample a small set of triplets and still achieve good learning results. With the goal of improving tf-idf weights in mind, we can first collect a set of triplets (i, j, l) whose similarity constraints cannot be satisfied in retrieval results obtained using tf-idf weights. During the training stage, we randomly choose to issue a few *Positive* images (at most 80 images per object) in the training set to query the training database. For each *Positive*

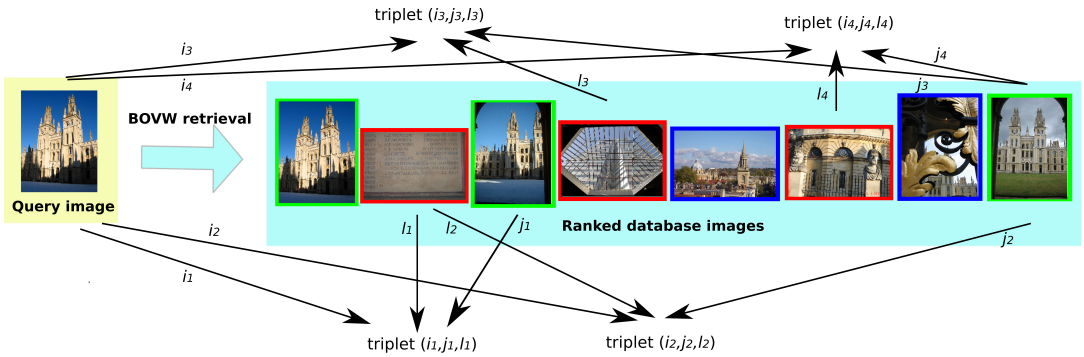


Fig. 1. Training triplet collection. Given a query image, the database images are sorted according to similarity scores calculated using tf-idf weights. On the right, *Positive* images of the query have a green border while junk images of the query have a blue border. Images of other objects have a red border. In this example, four triplets can be collected for training.

image i, the database images are ranked in a descending order according to their similarities with the query image using tf-idf weights. We check such a ranked list from the top to the end. Every time we find a false positive image l (e.g. an image of a different object or a background image), we record its ID and also find out all (or a random subset of) true positive images ranked lower than image l. Junk images of the query object are not used in this collection scheme. See Figure 1 for an example of triplet collection. For such a triplet, similarity score $S_{il} > S_{ij}$, which suggests tf-idf weights have failed. We refer to triplets collected in this way as hard triplets. In the case that the distribution of the number of *Positive* images for individual objects is uneven, the set of hard triplets could be dominated by those triplets obtained from queries of objects which have a large number of *Positive* images. This is especially true for *Oxford5K* dataset where the largest and the smallest number of *Positive* images among all landmarks are 221 and 6, respectively. An uniform weighting scheme for training triplets could make similarity constraints from landmarks with few *Positive* images ignored during learning. Therefore, we adopt the following triplet weighting scheme. Let $N(i)$ denote the number of *Positive* images we issue for the object in query i. Triplet weight is assigned as $w_{(i,j,l)} = \frac{max_t N(t)}{N(i)}$, which approximately guarantees the learnt similarity metric respects similarity constraints of all objects.

In addition, we collect a number of random triplets according to their labels for avoiding overfitting. To obtain a random triplet (i, j, k), we first randomly pick a positive image i of any object, then randomly choose another positive image j of the same object. Finally, we either choose a third positive image k of a different object or a background image k. Random triplets are weighted in a similar way as hard triplets. The final training set $\boldsymbol{T}$ includes both hard and random triplets.

C. Accelerating Optimization

Due to the relatively large number of unknowns (in $\mathbf{M^s}$) being optimized, we usually collect a few million triplets for training. As required by most constrained optimization solvers, we efficiently evaluate both objective function value and gradient by using the techniques below.

First, during the optimization, we maintain a set of active triplets whose hinge loss is non-zero. In each iteration, we only evaluate the active triplets. For every N iterations, we perform an evaluation of all training triplets. Since the number of active triplets decrease rapidly as the optimization progresses, the total computational cost can be significantly reduced. Second, as triplets are independent of each other, we can evaluate their contributions to the function value and gradient in parallel. We have implemented a parallel program using Intel Threading Building Blocks. In practice, we have observed a near-linear speedup that is proportional to the number of processors available. Third, as an image BoVW vector $\mathbf{I_i}$ is a sparse high-dimensional vector, the gradient $\frac{\partial S_{ij}}{\partial \mathbf{M^s}}$ is also sparse. Such gradients are used repeatedly for function and gradient evaluations in every iteration. Thus, we precompute such gradients for all possible image pair (i, j).

D. Reducing Overfitting

Although we collect a large number of training triplets, we still have the risk of overfitting due to the large number of unknowns. To reduce risk, we hold out a few *Positive* images per object (typically 4 images per object if available) from the training data. This independent set of hold-out images are used to evaluate the retrieval performance of the intermediate similarity metric at each iteration. We always keep the similarity metric with the best performance to avoid overfitting.

V. Results and Discussion

In this section, we evaluate by comparing retrieval performance of tf-idf weights and our learnt metric. We also compare our method against existing state-of-the-art techniques.

We use Hessian-Affine feature detector [20] and SIFT/RootSIFT feature descriptors. Approximate k-means clustering is used to construct 4 visual vocabularies of size $500K$. Two of them are constructed from *Oxford5K* dataset using SIFT and RootSIFT descriptor, respectively. Similarly, the remaining two are constructed from *Paris6K* dataset

TABLE II

SUMMARY OF RETRIEVAL PERFORMANCE. COLUMN *FD* INDICATES THE TYPE OF FEATURE DESCRIPTOR: *S* FOR STANDARD SIFT AND *R* FOR ROOTSIFT. COLUMN *SR/QE* INDICATES IF SPATIAL RERANKING FOR THE TOP 200 IMAGES [24] AND AVERAGE QUERY EXPANSION [7] ARE USED. FOR EVALUATIONS ON DATASETS *Oxford5K* AND *Oxford105K*, WE USE A $500K$ VOCABULARY TRAINED ON *Oxford5K*. SIMILARLY, FOR DATASETS *Paris6K* AND *Paris101K*, WE USE A $500K$ VOCABULARY TRAINED ON *Paris6K*. THE LEARNT SIMILARITY MATRIX HAS NON-ZERO OFF-DIAGONAL ELEMENTS DETERMINED BY 2-NN.

	Dataset	FD	SR/QE	mAP		
				tf-idf	learnt	gain
a	ox5K	S		0.637	0.713 ± 0.003	11.9%
b	ox5K	S	√	0.837	0.867 ± 0.005	3.6%
c	ox105K	S		0.515	0.553 ± 0.004	7.4%
d	ox105K	S	√	0.738	0.782 ± 0.005	5.9%
e	ox5K	R		0.680	0.786 ± 0.006	15.6%
f	ox5K	R	√	0.846	0.888 ± 0.006	5.0%
g	ox105K	R		0.559	0.637 ± 0.007	14.0%
h	ox105K	R	√	0.766	0.815 ± 0.003	6.4%
i	Paris6K	S		0.657	0.790 ± 0.013	20.2%
j	Paris6K	S	√	0.783	0.882 ± 0.019	12.6%
k	Paris101K	S		0.537	0.605 ± 0.010	12.7%
l	Paris101K	S	√	0.652	0.739 ± 0.015	13.3%
m	Paris6K	R		0.693	0.823 ± 0.010	18.8%
n	Paris6K	R	√	0.816	0.908 ± 0.006	11.2%
o	Paris101K	R		0.585	0.654 ± 0.011	11.8%
p	Paris101K	R	√	0.693	0.783 ± 0.011	13.0%

using SIFT and RootSIFT descriptor, respectively.

Both *Oxford5K* and *Paris6K* datasets have a standard set of 55 queries for evaluation purposes. We use the standard evaluation protocol in [23]. The average of mean average precision (mAP) scores is reported as the final performance. All the average of mAP scores reported are averaged over multiple runs with randomly chosen training and test sets.

We first evaluate our approach on *Oxford5K* dataset. We divide the dataset into training and test set. The standard 55 queries are always in the test set. The remaining $571 - 55 = 516$ *Positive* images are randomly distributed into training and test set in the ratio $4:1$. The junk images and background images are also randomly distributed into training and test sets in the same ratio. During the training stage, we first hold out a few *Positive* images for testing over-fitting as described in Section IV-D. 4 *Positive* images per object are held out. Objects *Cornmarket*, *Keble* and *Pitt Rivers* have too few *Positive* images and they do not have hold-out *Positive* images. Next, the rest of the *Positive* images in the training set are issued to query the training database and training triplets are collected as described in Section IV-B. During the test stage, the standard set of 55 images are issued to query the *Oxford5K* or *Oxford105K* dataset.

Table II shows the mAPs obtained using tf-idf weights and our learnt metric in various settings. The mean and standard deviation of final mAPs using our learnt metric are shown. We have implemented spatial reranking (SR) [24] and average query expansion (QE) [7]. We also report mAPs with SR and QE enabled as post-processing steps.

In setting (a) shown in Table II, we evaluate on the *Oxford5K* dataset using the SIFT descriptor. The mAP is improved by 11.9% (0.637 vs 0.713) with SR/QE disabled. Once SR/QE are enabled in (b), the mAP gap between tf-idf weights and our learnt metric is decreased but we still improve the mAP by 3.6% (0.837 vs 0.867). We also evaluate the generalization performance of our learnt metric on the *Oxford105K* dataset. In (c) where SR/QE are disabled, our learnt metric outperforms tf-idf weights by 7.4% (0.515 vs 0.553). In (d) where SR/QE are enabled, the mAP is still improved by 5.9% (0.738 vs 0.782).

RootSIFT [1] employs Hellinger kernel to measure SIFT descriptor similarity and has been proven to be able to boost retrieval performance at no additional cost. Here we are interested in the degree of improvement we can gain from a learnt metric when the baseline performance with RootSIFT is already high. We performed similar experiments in (e)-(h) by replacing SIFT with RootSIFT. Again, on the *Oxford5K* dataset, the mAP is improved by 15.6% (0.680 vs 0.786) and 5.0% (0.846 vs 0.888) before and after SR/QE are enabled. For the test on *Oxford105K* dataset, we observe the mAP increases by 14.0% (0.559 vs 0.637) and 6.4% (0.766 vs 0.815) before and after SR/QE are enabled.

We also evaluated our method on *Paris6K* and *Paris101K* datasets in (i)-(p) with similar settings. When SIFT descriptor is used, the mAP is improved by 20.2% (0.657 vs 0.790) in (i) and 12.6% (0.783 vs 0.882) in (j) before and after SR/QE are enabled. The generalization performance of our learnt metric on the *Paris101K* dataset is improved by 12.7% (0.537 vs 0.605) in (k) and 13.3% (0.652 vs 0.739) in (l) before and after SR/QE are enabled. When RootSIFT is used, our learnt metric improves the mAP by 18.8% (0.693 vs 0.823) in (m) and 11.2% (0.816 vs 0.908) in (n). On *Paris101K*, we observe that the mAP increases by 11.8% (0.585 vs 0.654) in (o) and 13.0% (0.693 vs 0.783) in (p) before and after SR/QE are enabled. We notice that the improvements of mAP on *Paris6K* and *Paris101K* are more significant than those on *Oxford5K* and *Oxford105K* datasets. This is probably because there are more *Positive* images available in the *Paris6K* dataset (see Table I). This allows us to collect a more comprehensive set of triplets for training.

Figure 2 compares average precision (AP) of individual landmarks [1] using tf-idf weight and the learnt similarity metrics. The metrics in (a) and (b) are learnt from *Oxford5K* and *Paris6K* datasets, respectively. In (a), the learnt metric improves performance for most landmarks, especially for those with a large number of *Positive* images. For landmarks with very few *Positive* images such as *Cornmarket* which has only 3 *Positive* images in the training set, the AP is not improved due to the lack of training triplets. In (b) where

[1] 11 landmarks in *Oxford5K* dataset are *All Souls*, *Ashmolean*, *Balliol*, *Bodleian*, *Christ Church*, *Cornmarket*, *Hertford*, *Keble*, *Magdalen*, *Pitt Rivers* and *Radcliffe*. For *Paris6K* dataset, 11 landmarks are *La Defense*, *Eiffel Tower*, *Hotel des Invalides*, *Louvre* , *Moulin Rouge*, *Musee d'Orsay*, *Notre Dame*, *Pantheon*, *Pompidou*, *Sacre Coeur* and *Arc de Triomphe*.

TABLE III

COMPARISON TO THE STATE OF THE ART. COLUMN *FD* INDICATES THE TYPE OF FEATURE DESCRIPTOR: *S* FOR SIFT AND *R* FOR ROOTSIFT. COLUMN *V* INDICATES IF THE VOCABULARY IS DEPENDENT ON THE TEST DATASET. DEPENDENT VOCABULARY IS PRODUCED FROM THE TEST DATASET WHILE INDEPENDENT VOCABULARY IS PRODUCED FROM A DIFFERENT DATASET. FOR EXAMPLE, TEST ON *Oxford5K* USES AN INDEPENDENT VOCABULARY OBTAINED FROM *Paris6K* AND VICE VERSA. COLUMN *SA* INDICATES IF SOFT ASSIGNMENT IS USED.

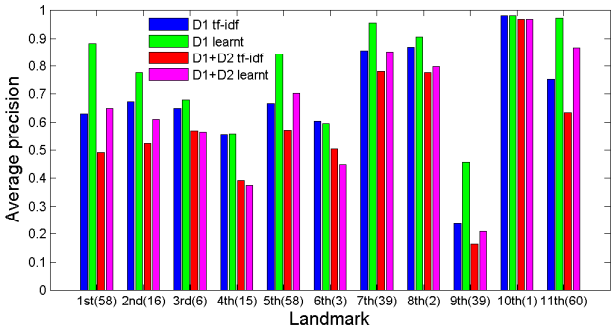
	Approach	FD	V	SA	mAP		
					D1	D1+D2	D3
a	[24]	S	D	√	0.825	0.718	N/A
b	[26]	S	D		0.814	0.767	0.803
c	This paper	S	D		**0.867**	**0.782**	**0.882**
d	[24]	S	I	√	0.719	**0.605**	N/A
e	This paper	S	I		**0.769**	0.588	0.730
f	[1]	R	D		0.881	**0.823**	0.850
g	This paper	R	D		**0.888**	0.815	**0.908**
h	[1]	R	I		0.714	0.602	0.660
i	This paper	R	I		**0.803**	**0.658**	**0.792**

all landmarks have a sufficiently large number of *Positive* images, the learnt metric improves AP for all landmarks. In summary, our learnt similarity metric outperforms tf-idf weights in all the settings discussed above. The mAP difference between tf-idf weights and the learnt metric is often reduced after SR/QE are enabled. Nevertheless, most of the time, even the reduced performance gap is well above 5%.

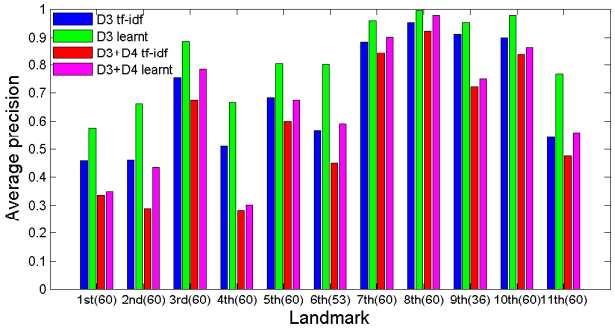
A. Performance Comparison to State-of-the-Art

In this section, we compare our method against state-of-the-art object retrieval systems. Table III summarizes the performance of our method and other existing systems in various settings. First, in settings (a)-(c) where SIFT descriptor and a dependent vocabulary are used, our method outperforms previous work [24] and [26]. In [24], quantization error is compensated by soft word assignment while we achieve this by selectively activating off-diagonal elements of the similarity matrix. We improve the best results in previous work on *Oxford5K*, *Oxford105K* and *Paris6K* by 5.1% (0.825 vs 0.867), 2.0% (0.767 vs 0.782) and 9.8% (0.803 vs 0.882), respectively. If an independent vocabulary is used as in settings (d) and (e), performance of both our method and previous approaches [24] decreases. Our method outperforms previous work on *Oxford5K* dataset but fails to improve on *Oxford105K* dataset. After examining average precision for individual landmarks, we notice we do not improve over tf-idf weights for landmarks with very few *Positive* images, such as *Pitt River*, *Cornmarket* and *Keble*. This suggests the success of our learning method relies on a sufficiently large number of *Positive* images for each object.

Second, in settings (f) and (g), SIFT is replaced with RootSIFT, which generally gives rise to better retrieval performance. Results in [1] were achieved with a combination of discriminative query expansion and spatial database-side feature augmentation. Despite this, our method achieves similar performance on *Oxford5K* and *Oxford105* datasets. we also improves the mAP by 6.8% (0.850 vs 0.908) on *Paris6K* dataset. If an independent vocabulary is used as in settings (h) and (i), the improvement is significant. We improves the mAP on *Oxford5K*, *Oxford105K*and *Paris6K* by 12.5% (0.714 vs 0.803), 9.3% (0.602 vs 0.658) and 20.0% (0.660 vs 0.792), respectively.



(a) Oxford



(b) Paris

Fig. 2. Retrieval performance comparison for individual landmarks. Average precisions are reported with SR/QE disabled. The number of *Positive* images used during training for each landmark is shown in parentheses.

B. Pairwise Image Similarity Distribution

Figure 3 shows distributions of similarity scores for two different groups of image pairs under tf-idf weighting and our learnt metric, respectively. The first group (green) consists of pairs of matching images and the second group (blue) consists of pairs of non-matching images. Matching images refer to a pair of *Positive* images of the same object while non-matching images could be 1) two *Positive* images of two different objects or 2) one *Positive* image and one background image. We expect the overlap between the similarity distributions of these two groups shrinks after a similarity metric is learnt. Such a change is confirmed in Figure 3. As a result, matching images are more likely to have a higher rank than non-matching images, in which case retrieval performance can be improved.

C. Impact of Off-Diagonal Elements

First, we investigate how the number of activated off-diagonal elements in each row of the similarity matrix affects the quality of learnt metric. The impact of off-diagonal elements on retrieval performance is shown in Figure 4.

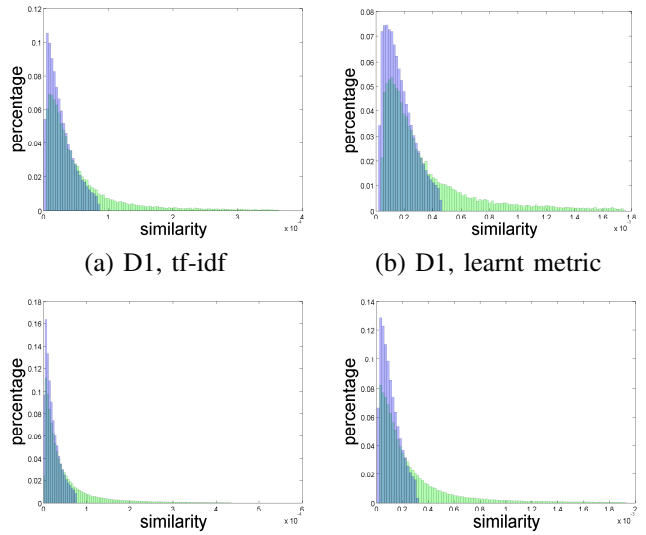


Fig. 3. Distributions of similarity scores for groups of matching (green) and non-matching (blue) image pairs using tf-idf weights and our learnt metric. Two rows show such distributions on the *Oxford5K* and *Paris6K* datasets, respectively. In each row, the left and right plots show distributions using tf-idf weights and our learnt metric, respectively. The largest 8% and smallest 8% similarity scores in each group are removed for clarity. Every distribution is normalized by the number of pairs in the corresponding group.

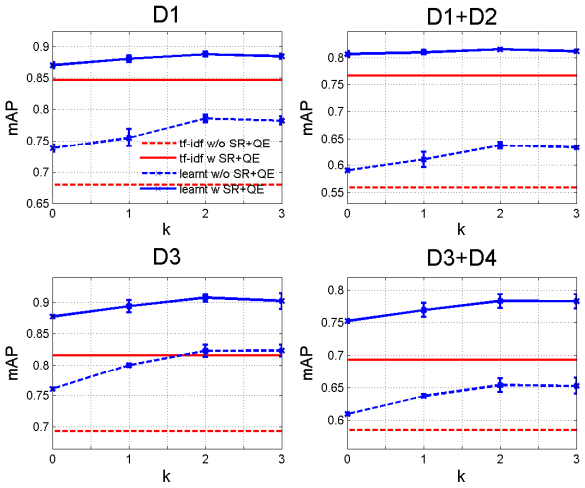


Fig. 4. Impact of the number of off-diagonal elements on mean average precision (mAP). In (a) and (b), similarity metrics are learnt using dataset *OXford5K* with a $500k$ vocabulary generated from *Oxford5K*, and results on *Oxford5K* and *Oxford105K* are shown, respectively. Similarly, in (c) and (d), similarity metrics are learnt using dataset *Paris6K* with a $500k$ vocabulary generated from *Paris6K*, and results on *Paris6K* and *Paris101K*are shown, respectively. RootSIFT is the feature descriptor used for obtaining these results.

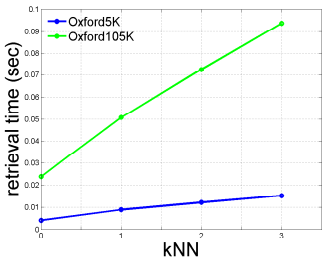


Fig. 5. Impact of the number of off-diagonal elements on index traversal time.

When the number of nearest neighbors, k, increases from 0 to 1, we observe an increase of test mAP, especially when spatial re-ranking and query expansion are disabled. This confirms the benefit of activating off-diagonal elements in the similarity matrix. The retrieval performance saturates when k grows to 2. Therefore, we report our results when k is set to 2.

Second, we investigate the impact of off-diagonal elements on retrieval time. Retrieval time consists of feature extraction and quantization for the query image, inverted index traversal and post-processing steps (SR/QE). Applying a learnt metric only incurs an increase in the inverted index traversal time linearly proportional to the number of nearest neighbors, k. This can be seen by noting that for each visual word in the query image, we need to traverse k more entries in the inverted index. Figure 5 shows the average index traversal time of the 55 standard queries for *Oxford5K* and *Oxford105K* under different values of k.

D. Handling Over-fitting

Due to the large number of unknowns in $\mathbf{M^s}$ being optimized, we have the risk of overfitting even when a large number of triplets are used for training. Figure 6 shows test mAP (w/o SR+QE) varies as the learning procedure proceeds when we learn a similarity metric on the *Oxford5K* dataset using RootSIFT. A 500K vocabulary was trained on *Oxford5K* and 2-NN was adopted for the similarity matrix. Spatial re-ranking and query expansion were disabled. As we can see, the test mAP first increases until it reaches the peak. Then it starts to decline possibly because of overfitting. As described in Section IV-D, a small set of *Positive* images are held out to query the training database using the intermediate similarity metric obtained during each iteration and the metric with the best mAP is chosen as the final metric.

E. Statistics

Table IV reports statistics of our metric learning method using SIFT descriptor. As we can see, the number of free variables in similarity matrix $\mathbf{M^s}$ grows linearly with the number of nearest neighbors when we activate more off-diagonal elements. The average learning time was measured on a desktop with two Intel Xeon E5-2620 processors. We collected on average $19.1M$ and $13.8M$ training triplets from *Oxford5K* and *Paris6K* datasets, respectively.

VI. Conclusion

We have presented a method to learn an optimal sparse similarity metric for object retrieval. By selectively activating off-diagonal elements in a similarity matrix, we generalize diagonal similarity matrix with only a linear increase of training complexity. Evaluations on benchmark datasets have confirmed the advantages of a learnt similarity metric over

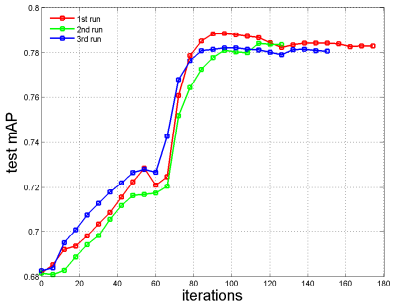


Fig. 6. Test mAP versus the number of iterations during optimization.

TABLE IV

STATISTICS OF SIMILARITY METRIC LEARNING. COLUMN k-NN INDICATES THE NUMBER OF ACTIVE OFF-DIAGONAL ELEMENTS PARTICIPATING IN METRIC LEARNING. COLUMN $\mathbf{M^s}$ INDICATES THE NUMBER OF UNKNOWNS IN THE SIMILARITY MATRIX. COLUMN *Time* REPORTS THE AVERAGE TRAINING TIME IN HOURS.

Dataset	k-NN	$\mathbf{M^s}$	Time
Oxford5K	0	$500K \pm 0$	2.2 ± 0.4
Oxford5K	1	$934K \pm 5K$	4.5 ± 0.5
Oxford5K	2	$1308K \pm 7K$	7.0 ± 0.9
Oxford5K	3	$1687K \pm 9K$	9.1 ± 0.9
Paris6K	0	$500K \pm 0$	2.5 ± 0.5
Paris6K	1	$922K \pm 4K$	5.2 ± 0.6
Paris6K	2	$1323K \pm 6K$	7.4 ± 0.6
Paris6K	3	$1732K \pm 10K$	10.5 ± 0.7

the tf-idf weighting scheme and the advantages of a generalized sparse similarity matrix over a diagonal one.
As future work, our method can be extended to be suitable for online learning scenario. In this case, stochastic gradient descent method can be used to solve similarity metric learning problem in a scalable way.

REFERENCES

[1] R. Arandjelović and A. Zisserman. "Three things everyone should know to improve object retrieval," *IEEE Conference on Computer Vision and Pattern Recognition* , 2012.

[2] A. Bar-Hillel, T. Hertz, N. Shental, and D. Weinshall. "Learning a mahalanobis metric from equivalence constraints." *Journal of Machine Learning Research* , 6(1):937, 2006.

[3] M. Brown, G. Hua, and S. Winder. "Discriminative learning of local image descriptors." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(1):43-57, 2011.

[4] H. Cai, F. Yan, and K. Mikolajczyk. "Learning weights for codebook in image classification and retrieval." *In Computer Vision and Pattern Recognition*, pp 2320-2327. IEEE, 2010.

[5] G. Chechik, V. Sharma, U. Shalit, and S. Bengio. "An online algorithm for large scale image similarity learning." *In Proc. NIPS*, volume 1. Citeseer, 2009.

[6] O. Chum, A. Mikulík, M. Perdoch, and J. Matas. "Total recall ii: Query expansion revisited." *In Computer Vision and Pattern Recognition*, pp. 889-896. IEEE, 2011.

[7] O. Chum, J. Philbin, J. Sivic, M. Isard, and A. Zisserman. "Total recall: Automatic query expansion with a generative feature model for object retrieval." *In IEEE International Conference on Computer Vision*, 2007.

[8] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray. "Visual categorization with bags of keypoints." *In Workshop on Statistical Learning in Computer Vision, ECCV*, pp 1-22, 2004.

[9] J. Davis, B. Kulis, P. Jain, S. Sra, and I. Dhillon. "Information-theoretic metric learning." *In Proceedings of the 24th international conference on Machine learning*, pages 209-216. ACM, 2007.

[10] A. Frome, Y. Singer, and J. Malik. "Image retrieval and classification using local distance functions". *In Advances in Neural Information Processing Systems 19*, volume 19, page 417. MIT Press, 2007.

[11] A. Frome, Y. Singer, F. Sha, and J. Malik. "Learning globally-consistent local distance functions for shape-based image retrieval and classification." *International Conference on Computer Vision*, pp 1-8. IEEE, 2007.

[12] A. Globerson and S. Roweis. "Metric learning by collapsing classes." *Advances in neural information processing systems*, 18:451, 2006.

[13] G. Hua, M. Brown, and S. Winder. "Discriminant embedding for local image descriptors." *International Conference on Computer Vision*, pp 1-8. IEEE, 2007.

[14] P.Jain B.Kulis, I. Dhillon, and K. Grauman. "Online metric learning and fast similarity search." *Advances in Neural Information Processing Systems*, 22, 2008.

[15] P.Jain B.Kulis and K. Grauman. "Fast image search for learned metrics." *In Computer Vision and Pattern Recognition, IEEE*, pp 1-8, 2008.

[16] H.Jegou, M. Douze, and C. Schmid. "Hamming embedding and weak geometric consistency for large scale image search." *ECCV 2008*, pages 304-317, 2008.

[17] S.Lazebnik, C. Schmid, and J. Ponce. "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories." *In Computer Vision and Pattern Recognition* , volume 2, pp 2169-2178. IEEE, 2006.

[18] D.Lowe. "Distinctive image features from scale-invariant keypoints. " *International journal of computer vision*, 60(2):91-110, 2004.

[19] K. Mikolajczyk and J.Matas. "Improving descriptors for fast tree matching by optimal linear projection." *IEEE 11th International Conference on Computer Vision*, pp 1-8, 2007.

[20] K. Mikolajczyk and C.Schmid. "Scale and affine invariant interest point detectors." *International Journal of Computer Vision*, 60(1):63-86,2004

[21] A.Mikulík, M.Perdoch, O. Chum, and J. Matas. "Learning a fine vocabulary." *ECCV 2010*, 2010.

[22] D.Nister and H. Stewenius. "Scalable recognition with a vocabulary tree." *In IEEE Conference on Computer Vision and Pattern Recognition*, 2006.

[23] J. Philbin,O. Chum, M. Isard, J. Sivic, and A. Zisserman. "Object retrieval with large vocabularies and fast spatial matching." *IEEE Conference on Computer Vision and Pattern Recognition* , 2007.

[24] J. Philbin, O.Chum, M. Isard, J. Sivic, and A. Zisserman. "Lost in quantization: Improving particular object retrieval in large scale image databases." *In IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[25] J. Philbin, M. Isard, J. Sivic, and A. Zisserman. "Descriptor learning for efficient retrieval". *In European Conference on Computer Vision*, 2010.

[26] D. Qin, S.Gammeter, L. Bossard, T. Quack, and L. Van Gool. "Hello neighbor: accurate object retrieval with k-reciprocal nearest neighbors." *IEEE Computer Vision and Pattern Recognition (CVPR)*, pp 777-784, 2011.

[27] M. Schultz and T. Joachims. "Learning a distance metric from relative comparisons." *Advances in Neural Information Processing Systems (NIPS)*, page 41, 2004.

[28] K. Simonyan, A. Vedaldi, and A. Zisserman. "Descriptor learning using convex optimisation." *European Conference on Computer Vision*, 2012

[29] K. Weinberger and L. Saul. "Distance metric learning for large margin nearest neighbor classification." *The Journal of Machine Learning Research*, 10:207-244, 2009.

[30] Z. Wu, Q. Ke, M. Isard, and J. Sun. "Bundling features for large scale partial-duplicate web image search." *IEEE Computer Vision and Pattern Recognition*, pages 25-32. IEEE, 2009.

[31] E. Xing, A. Ng, M. Jordan, and S.Russell. "Distance metric learning, with application to clustering with side-information." *Advances in neural information processing systems*, 15:505-512, 2002.

[32] L. Yang and R. Jin. "Distance metric learning: A comprehensive survey." *Michigan State Universiy*, pp 1-51, 2006.