Efficient Dimension Reduction Of Global Signature With Sparse Projectors For Image Near Duplicate Retrieval

Romain Negrel, David Picard, Philippe-Henri Gosselin

Efficient Metric Learning Based Dimension Reduction Using Sparse Projectors For Image Near Duplicate Retrieval

Romain Negrel, David Picard
[1]ETIS/ENSEA - University of Cergy-Pontoise
CNRS, UMR 8051
6, avenue du Ponceau, BP44
F95014 Cergy-Pontoise, France
{romain.negrel,picard}@ensea.fr

Philippe-Henri Gosselin
TexMex Project, Inria, Rennes, France
ETIS/ENSEA - University of Cergy-Pontoise
CNRS, UMR 8051
6, avenue du Ponceau, BP44
F95014 Cergy-Pontoise, France
gosselin@ensea.fr

***Abstract*—In this paper, we tackle the storage and computational cost of linear projections used in dimensionality reduction for near duplicate image retrieval. We propose a new method based on metric learning with a lower training cost than existing methods. Moreover, by adding a sparsity constraint, we obtain a projection matrix with a low storage and projection cost. We carry out experiments on a well known near duplicate image dataset and show our algorithm behaves correctly. Retrieval performances are shown to be promising when compared to the memory footprint and the projection cost of the obtained sparse matrix.**

I. Introduction

In this paper, we focus on dimensionality reduction methods for near duplicate image retrieval (NDIR). The goal of NDIR is to retrieve images matching a given query (*e.g.*, images of the same building) from a large set of images. The most successful methods in this area are based on the extraction of local visual descriptors which are then aggregated into an image signature [1]. As the most efficient signatures are high dimensional, a projection into a small subspace is needed when dealing with large scale datasets [2]. Even when current dimensionality reduction methods provide low dimensional signatures with good retrieval accuracies, they suffer a high projection cost. Both the memory needed to store the projectors and the computational cost to perform the projection are often prohibitive for such methods, and are what we investigate in this paper.

More specifically, we propose a new method to dramatically reduce the memory footprint and the computational cost of such projections. This method is based on metric learning, and learns a sparse projection matrix. Our two main contributions are:

- A new metric learning based method to obtain the projection matrix with a much lower training cost than existing methods,
- The introduction of a sparsity constraint on the projectors to obtain a low storage and computational projection cost.

The rest of the paper is organized as follows: First, we give an overview of the related work in dimensionality reduction techniques used in image retrieval. Then we present metric learning methods on which our proposal is based. We explain our proposed method in section IV. In section V, we analyze the convergence of our learning method and we show results using state of the art signatures on the well known **INRIA Holidays** [3] dataset, before we conclude.

II. Related Work

In this section we present current methods for the reduction of visual features, as well as their main drawbacks. More specifically, we focus on methods that compute linear projectors in Hilbert spaces. The choice of linear projectors can be explained by their simplicity and their ability to deal with large datasets. Each method has its own strategy for computing the projectors, but the projection itself remains the same or very similar. We can classify these approaches into two categories: unsupervised and supervised learning.

Unsupervised approaches learn projectors using a training set of images, usually randomly sampled. The best example in this case is Principal Component Analysis (PCA) which selects components of largest variance. Furthermore, the resulting projectors are orthogonal, and projected data can be whitened [4]. Many methods are then based on these approaches, such as PCA Embedding [5], Semi-Supervised Hashing [6], Spectral Hashing [7] or Tranform Coding [8]. In the context of image retrieval, these methods provide high dimensional reduction with low information lost. For example, visual signatures of hundreds of thousands of dimensions can be reduced to few hundreds, and with similar retrieval performance [9].

Supervised approaches learn projectors using a labeled training set. The first propositions of such methods compute semantic attributes by training a classifier for each semantic concept [10]. In this case, projectors are the classification functions, and are as numerous as the number of semantic concepts. Other approaches are based on metric learning that aims to learn a similarity function between two visual features. Other methods are based on the combination of kernel functions, a.k.a. Multiple Kernel Learning [11]. All these methods are proposed in a context of image categorization, and their effectiveness for image retrieval is yet to be shown. However, recent approaches in this scope have been proposed for image

retrieval. A first one is based on semantic attributes [12] and a second one is based a joint subspace and classification learning [13]. In both cases, these methods outperforms unsupervised approaches for near duplicate image retrieval.

Current methods suffer from a major drawback: the size of projectors is as large as the size of visual features. Consequently, even if the size of reduced visual features is small and scalable, the projection itself is not scalable. Actually, since the size of best visual features is at least hundreds of thousands of dimensions, the corresponding projection matrix quickly becomes very large. Such matrix is thus difficult to spread on a computational grid, or simply too large to fit in memory.

Let us note the projection matrix could be compressed using techniques like Product Quantization [1], like it is often the case for the output signatures. However, the projection would then become non-linear and have a higher computational cost, which is already prohibitive with the linear projection due to its size.

III. Metric learning

As metric learning approaches offer promising results [13], we discuss their key concepts and drawbacks in this section.

The general idea is to learn a parametric similarity function using a training set $\mathcal{I} = \{1, \dots, N\}$ for which a groundtruth is available. The groundtruth is formed by a set of queries $\mathcal{Q} \subset \mathcal{I}$. For each query $q \in \mathcal{Q}$, we have a set $\mathcal{P}_q \subset \mathcal{I}$ of positive images (images similar to the query) and a set $\mathcal{N}_q = \mathcal{I} \setminus \mathcal{P}_q$ of negatives images (images dissimilar to the query). The learning task is to find the optimal parameters of the similarity function such that the obtained similarity is as close as possible to the groundtruth. The most popular parametric similarity function is

$$\mathrm{d}_{\mathbf{W}}(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^\top \mathbf{W} \mathbf{x}_j, \tag{1}$$

with $\mathbf{x}_i$, $\mathbf{x}_j$ two vectors in $\mathbb{R}^D$ and $\mathbf{W} \in \mathbb{R}^{D \times D}$. Note that $\mathrm{d}_{\mathbf{I}}(\mathbf{x}_i, \mathbf{x}_j)$ corresponds to a dot-product in some linear subspace if $\mathbf{W}$ is positive definite.

The main problem of this parametric similarity function is the number of parameters that increases quadratically with the dimension of the input space. Thus, for high dimensional vectors the computational cost to learn $\mathbf{W}$ is prohibitive. In [14] Bai *et al.* proposed to solve this problem by decomposing $\mathbf{W} = \mathbf{U}\mathbf{U}^\top$ with $U \in \mathbb{R}^{D \times R}$ and $R < D$. Eq.(1) then becomes:

$$\mathrm{d}_{\mathbf{U}}(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^\top \mathbf{U}\mathbf{U}^\top \mathbf{x}_j = \left(\mathbf{U}^\top \mathbf{x}_i\right)^\top \mathbf{U}^\top \mathbf{x}_j. \tag{2}$$

In this form, and with R fixed, the number of parameters increases linearly with the dimension of the the input space. Eq. (2) provides a linear projection in a subspace of dimension R:

$$\mathbf{y}_i = \mathbf{U}^\top \mathbf{x}_i. \tag{3}$$

$\mathbf{y}_i$ is then a low dimensional vector giving the same similarity when used with the standard dot product as $\mathrm{d}_{\mathbf{U}}$.

In [14] the authors proposed a method for learning $\mathbf{U}$ so as to preserve the ranking between positive and negative samples. They state a good similarity measure should satisfy the following sets of constraints:

$$\{\mathrm{d}_{\mathbf{U}}(\mathbf{x}_q, \mathbf{x}_i) > \mathrm{d}_{\mathbf{U}}(\mathbf{x}_q, \mathbf{x}_j)\}_{q \in \mathcal{Q}, i \in \mathcal{P}_q, j \in \mathcal{N}_q}. \tag{4}$$

Then, they propose an objective function to minimize, with a loss function that measures how much these constraints are violated:

$$\sum_{q \in \mathcal{Q}} \sum_{i \in \mathcal{P}_q} \sum_{j \in \mathcal{N}_q} \left[1 - \mathrm{d}_{\mathbf{U}}(\mathbf{x}_q, \mathbf{x}_i) + \mathrm{d}_{\mathbf{U}}(\mathbf{x}_q, \mathbf{x}_j)\right]_+, \tag{5}$$

with $[x]_+ = \max(0, x)$. As it is impossible to go through all the triplets $(q, i, j) \in \mathcal{Q} \times \mathcal{P}_q \times \mathcal{N}_q$, the authors propose to perform a stochastic gradient descent (SGD) to optimize the objective. In [15], the authors argue that the triplet groundtruth is sometimes difficult to obtain. They propose to generalize the triplets with quadruplets $\{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4\}$ for which $\mathrm{d}_{\mathbf{U}}(\mathbf{x}_1, \mathbf{x}_2) > \mathrm{d}_{\mathbf{U}}(\mathbf{x}_3, \mathbf{x}_4)$, and find it more convenient to optimize. However, the number of constraints grows to the power 4, which makes such methods impracticable for large datasets.

All the methods presented here share the same drawbacks when dealing with image retrieval. First they take into account a huge number of constraints. In particular, the training set has to be of a sufficient size to cope with the wide variety of images, and consequently the learning procedure has to scale with such large training sets. Secondly, the concerns over the size of the projection matrix stated in the previous section still holds for these methods. Both of these drawbacks make such methods unsuitable for large scale image retrieval.

IV. Proposed method

Int this section, we present our main contribution: a new projection matrix for significantly reducing the size of large signatures with a low storage cost and computational cost. Our projection matrix is based on metric learning approaches for ranking problems with a sparsity constraint.

We aim at a projection matrix $\mathbf{U}$ that maximizes the *mean Average Precision* (mAP) used in NDIR. However, the mAP is difficult to optimize directly. On the contrary, it is easy and sufficient to define a set of constraints on the similarity function such that the mAP is maximal: For each query q, the scores of positive images have to be greater than the scores of negative images, *i.e.*, the constraints described in Eq. 4. As already stated in the previous section, the number of such constraints is very large ($\sum_{q \in \mathcal{Q}} \mathrm{card}(\mathcal{P}_q)\,\mathrm{card}(\mathcal{N}_q)$), and the learning cost of such metric becomes prohibitive.

1) Learning with pivot: To reduce the number of constraints, we introduce for each query q a pivot image $p_q \in \mathcal{N}_q$ belonging to the negative images set $\mathcal{N}_q$. We thus obtain a new set of constraints:

$$\begin{cases} \{\mathrm{d}_{\mathbf{U}}(\mathbf{x}_q, \mathbf{x}_i) > \mathrm{d}_{\mathbf{U}}(\mathbf{x}_q, \mathbf{x}_{p_q})\}_{i \in \mathcal{P}_q}, \\ \{\mathrm{d}_{\mathbf{U}}(\mathbf{x}_q, \mathbf{x}_{p_q}) \geq \mathrm{d}_{\mathbf{U}}(\mathbf{x}_q, \mathbf{x}_j)\}_{j \in \mathcal{N}_q \setminus p_q}. \end{cases} \tag{6}$$

The number of constraints is now $\sum_{q \in \mathcal{Q}} (\mathrm{card}(\mathcal{P}_q) + \mathrm{card}(\mathcal{N}_q) - 1)$ which is far less than the original set of constraints (4). It is easy to show that if this set of constraints is respected the mAP is maximal. Any image in $\mathcal{N}_q$ can be

selected as pivot. In order to stay as close a possible to the original signatures, we propose to select the negative image with the largest score using the original signatures.

Then, to learn the metric $\mathrm{d}_{\mathbf{U}}$ that aims at this set of constraints, we define for each query q an objective function to minimize using the squared loss:

$$\begin{aligned}\mathrm{f}_q(\mathbf{U}) = \quad & \alpha_q \sum_{i\in\mathcal{P}_q} [\varepsilon + \Delta\mathrm{d}_{\mathbf{U}}(q,i,p_q)]_+^2 \\ & +\beta_q \sum_{j\in\mathcal{N}_q\setminus p_q} [\Delta\mathrm{d}_{\mathbf{U}}(q,p_q,j)]_+^2\end{aligned} \tag{7}$$

with

$$\alpha_q = \frac{1}{\sum_{i\in\mathcal{P}_q} \mathrm{h}(\varepsilon + \Delta\mathrm{d}_{\mathbf{U}}(q,i,p_q))}, \tag{8}$$

$$\beta_q = \frac{1}{\sum_{j\in\mathcal{N}_q\setminus p_q} \mathrm{h}(\Delta\mathrm{d}_{\mathbf{U}}(q,p_q,j))}, \tag{9}$$

$$\Delta\mathrm{d}_{\mathbf{U}}(q,i,j) = \mathrm{d}_{\mathbf{U}}(\mathbf{x}_q,\mathbf{x}_i) - \mathrm{d}_{\mathbf{U}}(\mathbf{x}_q,\mathbf{x}_j), \tag{10}$$

$$= \mathrm{d}_{\mathbf{U}}(\mathbf{x}_q,\Delta\mathbf{x}_{ij}), \tag{11}$$

$$\Delta\mathbf{x}_{ij} = \mathbf{x}_i - \mathbf{x}_j, \tag{12}$$

and h the Heaviside step function and ε a margin (*e.g.*, $\varepsilon = 10^{-6}$). Note that depending on the margin, all constraints can be achieved before the objective function reaches 0. The squared loss gives more importance to very poorly sorted images, *i.e.*, the ones having the biggest influence on the mAP. The global optimization problem can be written as follows:

$$\mathbf{U}^* = \arg\min_{\mathbf{U}} \sum_{q\in\mathcal{Q}} \mathrm{f}_q(\mathbf{U}) \tag{13}$$

To solve this problem, we propose to use a Stochastic Gradient Descent (SGD) [16], [17] improved by using active learning. The gradient for one query q is

$$\begin{aligned}\frac{\partial \mathrm{f}_q(\mathbf{U})}{\partial\mathbf{U}} = & \alpha_q \sum_{i\in\mathcal{P}_q} \mathrm{h}(\varepsilon + \Delta\mathrm{d}_{\mathbf{U}}(q,i,p_q))(\varepsilon + \Delta\mathrm{d}_{\mathbf{U}}(q,i,p_q))\mathbf{A}_{qi}\mathbf{U} \\ & + \beta_q \sum_{j\in\mathcal{N}_q} \mathrm{h}(\Delta\mathrm{d}_{\mathbf{U}}(q,p_q,j))\,\Delta\mathrm{d}_{\mathbf{U}}(q,p_q,j)\mathbf{A}_{qj}\mathbf{U},\end{aligned} \tag{14}$$

with $\mathbf{A}_{qi} = \mathbf{x}_q\Delta\mathbf{x}_{p_qi}^\top + \Delta\mathbf{x}_{p_qi}\mathbf{x}_q^\top$. The training procedure consists in repeating the two following steps:

1) we select a subset $\tilde{\mathcal{Q}}_t$ of P queries,
2) perform the gradient update on $\tilde{\mathcal{Q}}_t$.

To select a subset $\tilde{\mathcal{Q}}_t$ of P queries, we select the half among the higher value of $\{\mathrm{f}_q(\mathbf{U}_t)\}_{q\in\mathcal{Q}}$ (active learning part) and the other half by random sampling (stochastic part). The gradient update is then

$$\mathbf{U}_{t+1} = \mathbf{U}_t + \mu_t \sum_{q\in\tilde{\mathcal{Q}}_t} \frac{\partial \mathrm{f}_q(\mathbf{U}_t)}{\partial\mathbf{U}} \tag{15}$$

with μ_t the learning rate computed by golden section search [18].

In [14] the authors show that good results are obtained by initializing the values of $\mathbf{U}$ randomly. We propose to reduce the convergence time by initializing $\mathbf{U}$ with the unsupervised projectors proposed in [9]. We stop the algorithm when the value of objective function is near the numerical precision (*e.g.*, 10^{-16}) or when the maximum number of iterations is reached (*e.g.*, 2000).

2) Learning with sparsity constraint: Furthermore, we propose to add a sparsity constraint on $\mathbf{U}$ to obtain a projection matrix with a low storage cost and a low projection cost. For this, we add a ℓ_0 norm constraint on each column $\mathbf{u}_i$ of matrix $\mathbf{U}$. We can then rewrite the global optimization problem (13) as follows:

$$\begin{aligned}\mathbf{U}^* = & \arg\min_{\mathbf{U}} \sum_{q\in\mathcal{Q}} \mathrm{f}_q(\mathbf{U}) \\ & \text{s.t. } ||\mathbf{u}_i||_0 = M, \forall i;\end{aligned} \tag{16}$$

with $||\cdot||_0$ the ℓ_0 norm and M the number of non-zero entries by columns of matrix $\mathbf{U}$.

Algorithm 1 Rank optimized projectors with ℓ_0 constraint

$t \leftarrow 1$.
Initialize the matrix $\mathbf{U}_1$ with the projection provided with unsupervised algothims of paper [9].
Project $\mathbf{U}_1$ onto the $||\mathbf{u}_i||_0 = M$ constraint.
repeat
  Select a subset $\tilde{\mathcal{Q}}_t$ of P query in $\mathcal{Q}$.
  Compute the gradient on the subset $\tilde{\mathcal{Q}}_t$.
  Compute the optimal learning rate μ_t.
  Perform the gradient update to compute $\mathbf{U}_{t+1}$
  Project $\mathbf{U}_{t+1}$ onto the $||\mathbf{u}_i||_0 = M$ constraint.
  $t \leftarrow t+1$.
until objective function $< 10^{-12}$ **or** $t >$ maxIter

As described in Algorithm 1, we use the same algorithm as presented in the previous section to solve the sparse problem, only adding a step of projection onto the ℓ_0 norm constraint. The projection of $\mathbf{u}_i$ onto the $||\mathbf{u}_i||_0 = M$ constraint is performed by thresholding the smaller absolute values:

$$\hat{u}_{ki} = u_{ki}\,\mathrm{h}(|u_{ik}| - \nabla_i), \forall k \tag{17}$$

with $\nabla_i \in \mathbb{R}^+$ the threshold selected to comply with the sparsity constraint.

Although such operation is non-smooth and might hinder the convergence of the algorithm, we empirically show that our algorithm behaves correctly in the experiments.

It is sometimes more useful to define the sparsity constraint independently of the dimension of the input vectors. To this end, we note by τ the rate of zero values in a sparse matrix:

$$\tau(\mathbf{U}) = \frac{\text{Number of zero values in } \mathbf{U}}{\text{Number of values in } \mathbf{U}}. \tag{18}$$

In the case of our matrix $\mathbf{U}$ constrained by ℓ_0 norm, we have the following relation between M and τ:

$$\tau(\mathbf{U}) = \frac{D-M}{D}. \tag{19}$$

In the experiments, we use τ instead of M to measure the impact of the sparsity constraint on our algorithm.

Fig. 1. Images from Holidays dataset [3].

Sign.	Dim.	Full	Reduc. without sparsity [9]			
			32	64	128	256
VLAD-64	5k	86.23	76.90	80.87	83.18	**84.25**
VLAD-128	10k	87.59	76.86	81.52	83.23	**84.82**
VLAD-256	20k	87.52	75.36	80.80	83.16	**84.59**
VLAD-512	41k	87.75	72.96	79.39	82.19	**83.99**
FV-64	10k	85.17	78.78	81.64	84.41	**85.00**
FV-128	20k	86.15	78.34	82.26	84.05	**85.41**
FV-256	41k	88.10	78.39	82.57	84.84	**86.38**
FV-512	82k	87.09	77.76	81.62	83.85	**85.82**

TABLE I. MAP (IN %) OF VLAD AND FV SIGNATURES AND REDUCED SIGNATURES USING [9] ON THE TESTING SET.

V. EXPERIMENTS

In this section, we present our experimental protocol and the evaluation of our proposed method.

A. Dataset and Experimental Protocol

To evaluate our method, we use the well known benchmark **INRIA Holidays** [3]. Holidays dataset is a set of images (typically personal holiday photographs) ; it contains 1,491 images gathered in 500 groups. Each group is composed of one query image and up to 4 images of correct retrieval results. Evaluation on this dataset is obtained by computing the *mean average precision* (mAP) over all queries.

Since we need a training set to learn our projectors, we split the image groups of Holidays dataset in two separate sets: a training set and a testing set. The two sets are composed of 250 groups randomly and independently sampled. We use the same evaluation protocol as for the whole Holidays dataset to evaluate both the training and testing subsets.

B. Signatures

For all our experiments, we perform a two-step pre-processing on all images: (a) image resizing (to a maximum width of 512 pixels); (b) histogram equalization. We use the HOG local descriptors (128-dimensional) [19], extracted on a regular dense grid of 3 pixels at 4 scales. We use this descriptors to compute "Vector Aggregating Local Descriptors" (VLAD) [1] and "Fisher Vector" (FV) [2] signatures with a codebook of 64, 128, 256 and 512 codewords trained only using the training set. We compute the VLAD signature with a Principal Component Analysis (PCA) cluster-wise [20] which preserves 80 dimensions by cluster. For the FV signature, we perform a PCA on local descriptors that preserves 80 dimensions. Finally, we perform a power-normalization (for all experiments set to 0.1) and ℓ_2-normalization on both signatures.

As a baseline, Table I shows the mAP on the testing set obtained by the original signatures and by the reduced signatures using the unsupervised method proposed in [9]. Rows are the differents signatures (*e.g.*, VLAD-64 is VLAD signature with a codebook of 64 clusters). The second column is the size of original signatures, the third column is the mAP for the original signatures, and from the fourth to the seventh column are the mAP of reduced signatures for different values of the output dimension R. We can see that for original signatures, Fisher Vectors provide better results than VLAD, the best result being obtain with FV-256 (88.1% of mAP). For the reduced signatures, the best results are always obtained with the maximum output dimension $R = 256$.

C. Convergence Analysis

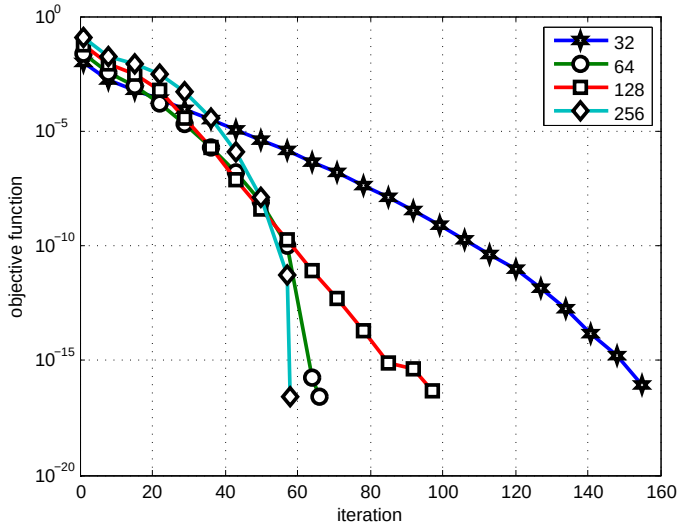
In this section we analyze the convergence of the proposed algorithm with and without the sparsity constraint. For all experiments in this section we use Fisher Vectors with a codebook of 512 clusters, *i.e.*, an original size 82k dimensions.

Figure 2(a) shows the evolution of the objective function without the sparsity constraint for different values of R. We see that the proposed algorithm minimizes the objective function. Furthermore, the convergence rate seems to be exponential. We note that the proposed algorithm needs more iterations to converge with small values of R. However, even for very small value of R, our algorithm converges, meaning all constraints are respected.

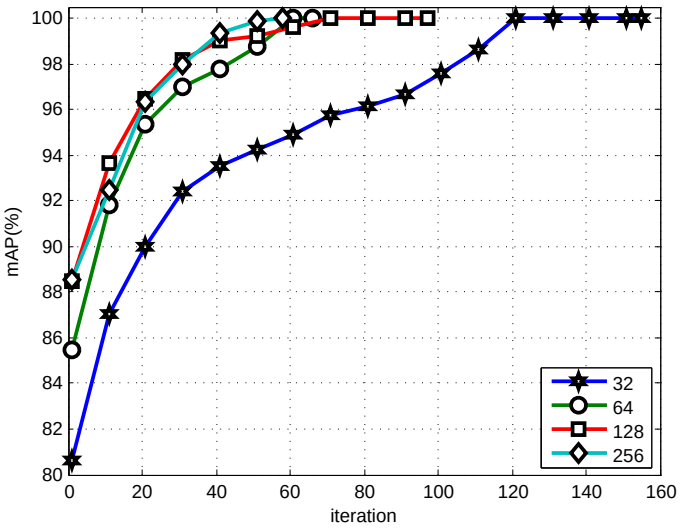
Figure 2(b) shows the evolution of mAP on the training set without the sparsity constraint for different values of R. We see the mAP increases up to 100% (perfect sorting of images for all the queries), which is consistent with the objective function converging to 0.

Figure 3(a) shows the evolution of the objective function with the sparsity constraint for different values of $\tau(\mathbf{U})$ (sparsity rate of matrix $\mathbf{U}$). We see that the proposed algorithm always decreases the objective function, albeit with a slower the convergence rate. We note that the higher the sparsity, the slower the convergence.

Figure 3(b) shows the evolution of mAP on Train set with the sparsity constraint for different values of $\tau(\mathbf{U})$. Again, we see our algorithm is able to increase the mAP up to

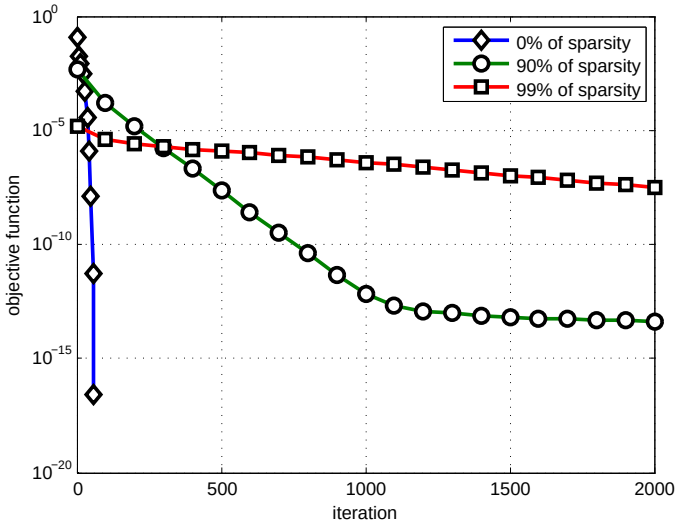


(a) Objective function evolution

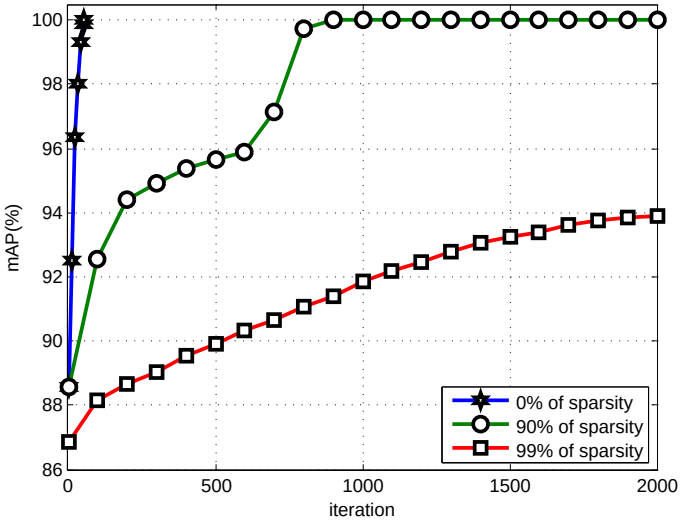


(b) mAP evolution

Fig. 2. Objective function and mAP evolution on the training set.



(a) Objective function evolution



(b) mAP evolution

Fig. 3. Objective function and mAP evolution on the training set for various sparsity rates.

100%, provided the sparsity constraint does not hinder the convergence speed enough to achieve the optimal objective value in reasonable time.

D. Retrieval performances

Now, we focus on the generalization capability of our method by training the projectors on the training set and evaluating the mAP on the testing set.

Figure 4 shows the evolution of mAP on the training set and the testing set with the sparsity constraint fixed at τ =99% for different values of R. We see our algorithm increases the mAP on the testing set. This increase is even bigger when the number of projectors is small (*e.g.*, 22% mAP increase with $R = 32$). Note that the mAP quickly stabilizes on the testing set, even if the algorithm has not fully converged.

Table II shows the mAP on the testing set obtained by the reduced signatures using our sparse projectors. Rows are the different signatures. The second to the fifth columns are reported the mAP with $\tau = 90\%$ for different values of R, and from the sixth to the ninth columns are reported the mAP with $\tau = 99\%$ for different values of R. We can see the mAP also increases with R.

Sign.	Reduc. with 90% of sparsity				Reduc. with 99% of sparsity			
	32	64	128	256	32	64	128	256
VLAD-64	75.87	80.17	82.21	**82.97**	70.36	75.67	77.67	**79.37**
VLAD-128	75.95	80.32	82.36	**82.96**	72.03	77.32	79.78	**81.08**
VLAD-256	74.43	78.74	80.99	**82.78**	71.05	76.33	78.36	**80.53**
VLAD-512	71.97	76.99	79.70	**81.02**	71.61	75.16	78.10	**79.45**
FV-64	78.39	80.90	82.95	**84.36**	74.17	78.46	81.51	**82.72**
FV-128	77.99	80.94	82.78	**84.31**	74.22	78.62	80.71	**82.47**
FV-256	76.24	81.19	83.28	**84.31**	72.73	78.13	81.50	**83.21**
FV-512	76.04	79.95	81.59	**83.12**	75.24	78.38	80.16	**80.90**

TABLE II. PERFORMANCES OF REDUCE SIGNATURE WITH SPARSITY ON THE TESTING SET (MAP IN %).

If we compare the performance of the obtained signatures and the performance of signatures obtained with the method presented in [9] (Table I), we can see that at equivalent final

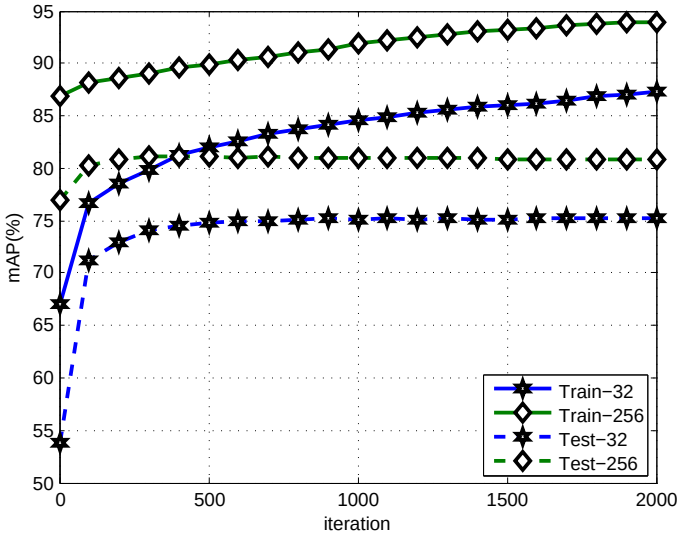


Fig. 4. mAP on training set and testing set vs algorithm iteration

dimension, the signatures of [9] provide better results (*e.g.*, for $R = 256$ the FV-256 provides 86.38% of mAP with [9], 84.31% of mAP with our sparse projectors at $\tau = 90\%$, and 83.21% of mAP with our sparse projectors at $\tau = 99\%$). However, note that 256 projectors with $\tau = 99\%$ have the same storage cost that 2,6 full projectors. For the FV-256 signatures the mAP with 256 sparse projectors at $\tau = 99\%$ is 83.21%, whereas it is 78.39% when using 32 full projectors that are still 12,5 times more costly to store and use.

VI. Conclusion

In this paper, we tackle the storage and computational cost of dimensionality reduction techniques for near duplicate image retrieval. We proposed a new method based on metric learning to train a sparse projection matrix. To learn the matrix, we propose an efficient algorithm based on stochastic gradient descent coupled with an active learning strategy. Using state of the art input signatures, we carried out a study of the convergence of our method on the INRIA Holidays dataset. By evaluating the performance of our projectors, we show promising results with a much lower storage and computational projection cost than existing methods.

References

[1] H. Jégou, M. Douze, C. Schmid, and P. Pérez, "Aggregating local descriptors into a compact image representation," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 3304–3311.

[2] F. Perronnin, Y. Liu, J. Sánchez, and H. Poirier, "Large-scale image retrieval with compressed fisher vectors," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 3384–3391.

[3] H. Jegou, M. Douze, and C. Schmid, "Hamming embedding and weak geometric consistency for large scale image search," in *Computer Vision–ECCV 2008*. Springer, 2008, pp. 304–317.

[4] H. Jégou and O. Chum, "Negative evidences and co-occurences in image retrieval: The benefit of pca and whitening," in *Computer Vision–ECCV 2012*. Springer, 2012, pp. 774–787.

[5] A. Gordo and F. Perronnin, "Asymmetric distances for binary embeddings," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 729–736.

[6] J. Wang, S. Kumar, and S. Chang, "Semi-supervised hashing for large scale search," 2012.

[7] Y. Weiss, A. Torralba, and R. Fergus, "Spectral hashing," in *Advances in neural information processing systems*, 2008, pp. 1753–1760.

[8] J. Brandt, "Transform coding for fast approximate nearest neighbor search in high dimensions," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 1815–1822.

[9] R. Negrel, D. Picard, and P.-H. Gosselin, "Web-scale image retrieval using compact tensor aggregation of visual descriptors," *MultiMedia, IEEE*, vol. 20, no. 3, pp. 24–33, 2013.

[10] Y. Su, M. Allan, and F. Jurie, "Improving object classification using semantic attributes." in *BMVC*, 2010, pp. 1–10.

[11] F. R. Bach, G. R. Lanckriet, and M. I. Jordan, "Multiple kernel learning, conic duality, and the smo algorithm," in *Proceedings of the twenty-first international conference on Machine learning*. ACM, 2004, p. 6.

[12] M. Douze, A. Ramisa, and C. Schmid, "Combining attributes and fisher vectors for efficient image retrieval," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 745–752.

[13] A. Gordoa, J. A. Rodríguez-Serrano, F. Perronnin, and E. Valveny, "Leveraging category-level labels for instance-level image retrieval," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 3045–3052.

[14] B. Bai, J. Weston, D. Grangier, R. Collobert, K. Sadamasa, Y. Qi, O. Chapelle, and K. Weinberger, "Supervised semantic indexing," in *Proceedings of the 18th ACM conference on Information and knowledge management*. ACM, 2009, pp. 187–196.

[15] M. T. Law, N. Thome, and M. Cord, "Quadruplet-wise image similarity learning," in *The IEEE International Conference on Computer Vision (ICCV)*, December 2013.

[16] L. Bottou, "Stochastic learning," in *Advanced lectures on machine learning*. Springer, 2004, pp. 146–168.

[17] S. Shalev-Shwartz, Y. Singer, and N. Srebro, "Pegasos: Primal estimated sub-gradient solver for svm," in *Proceedings of the 24th International Conference on Machine Learning*, ser. ICML '07. New York, NY, USA: ACM, 2007, pp. 807–814. [Online]. Available: http://doi.acm.org/10.1145/1273496.1273598

[18] M. Avriel and D. Wilde, "Optimality proof for the symmetric fibonacci search technique." *The Fibonacci Quarterly*, vol. 4, pp. 265–269, 1966.

[19] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1. IEEE, 2005, pp. 886–893.

[20] R. Negrel, D. Picard, and P.-H. Gosselin, "Using spatial pyramids with compacted vlat for image categorization," in *Pattern Recognition (ICPR), 2012 21st International Conference on*. IEEE, 2012, pp. 2460–2463.