2014 22nd International Conference on Pattern Recognition

Learning Semantic Binary Codes by Encoding Attributes for Image Retrieval

Jianwei Luo and Zhiguo Jiang
Image Processing Center, School of Astronautics
Beihang University, Beijing, 100191, China
Beijing Key Laboratory of Digital Media, Beijing, 100191, China
Email: ljw321521@sa.buaa.edu.cn and jiangzg@buaa.edu.cn

***Abstract*—This paper addresses the problem of learning semantic compact binary codes for efficient retrieval in large-scale image collections. Our contributions are three-fold. Firstly, we introduce semantic codes, of which each bit corresponds to an attribute that describes a property of an object (e.g. dogs have furry). Secondly, we propose to use matrix factorization (MF) to learn the semantic codes by encoding attributes. Unlike traditional PCA-based encoding methods which quantize data into orthogonal bases, MF assumes no constraints on bases, and this scheme is coincided with that attributes are correlated. Finally, to augment semantic codes, MF is extended to encode extra non-semantic codes to preserve similarity in origin data space. Evaluations on a-Pascal dataset show that our method is comparable to the state-of-the-art when using Euclidean distance as ground truth, and even outperforms state-of-the-art when using class label as ground truth. Furthermore, in experiments, our method can retrieve images that share the same semantic properties with the query image, which can be used to other vision tasks, e.g. re-training classifiers.**

Keywords*—*image retrieval; attribute; hashing function; matrix factorization;

I. Introduction

Recently, the vision community has made great progress on the problem of learning similarity-preserving binary codes for representing large-scale image collections [1], [2], [3], [4], [5], [6], [7]. It can greatly improve the efficiency of storing and retrieval of large amounts of images by encoding low-level image features into compact binary codes. For example, given a fixed 64-bit binary codes, it can represent 10^{19} images. Although promising results [2], [7] have been achieved on several large-scale datasets, the problem is far from solved.

Literately, to address the large-scale image retrieval problem, there are two main streams, namely unsupervised and supervised methods. Unsupervised algorithms map images into compact binary codes by making the hamming distance of similar features small in the low-dimensional space without any priors. For example, Locally Sensitive Hashing (LSH) [1] randomly selects several Gaussian functions to project data into low-dimensional space and Spectral Hashing (SH) [6] tries to maximize variance of binary codes by spectral graph analysis method. Unlike unsupervised methods, supervised approaches leverage class label into objective function to maximize the discrimination of different categories, making visually similar objects lying in the nearest neighbors in the low-dimensional space. For example, Gordoa et al. [3] performs Canonical Correlation Analysis (CCA) [8] on both original data and class label, which makes codes consistency of categories. However, both of these two kinds of approaches do not explicitly encode semantic information into their codes.

Attributes have attracted lots of attention in the past few years, and it has been proved that they can benefit a lot of computer vision tasks, such as object recognition [9], [10], zero-shot learning [11], [12], multi-queries image retrieval [13], [14], action recognition in videos [15], fine-grained objects recognition [16] and so on. Semantic attributes are usually descriptions of objects, and can be used to discriminate different categories. For example, people category has properties such as legs, head, torso, eyes, skin, clothes, hairs and so on, and cat category has furry property which can distinguish cat from people. But for visually similar categories which often share the same semantic properties, it is hard to distinguish them only by semantic attributes, e.g. having furry property cannot be used to discriminate cat from dog. So extra non-semantic attributes are usually needed to assist semantic attributes. [9] tries to train many classifiers on random split samples, and considers the outputs of these classifiers as extra discriminative attributes. Classemes [17] built by learning thousands of classifiers on large-collections of images from internet, are considered as non-semantic attributes.

In this paper, we propose to use matrix factorization (MF) to learn semantic compact codes for image retrieval. MF approximates input data as linear combinations of a set of bases. Unlike principle components analysis (PCA) [18] based methods [6], [2], MF assumes no orthogonal constraints on bases, considering the correlation between data. To encode semantic meaning into binary codes, we enforce attributes as constraints so that each bit of the learned semantic codes corresponds to an attribute. To simplify, we denote this method as MF-Attr. The binary codes are constructed according to coefficients. For entries of coefficients which are great than 0, the corresponding bits are set to 1 which denotes existence of an attribute, otherwise bits are assigned as 0. MF-Attr is also extended to augment semantic codes by learning extra non-semantic codes which helps preserve similarity in data space. Furthermore, the model of MF-Attr is easy to deal with. In training phase, bases and coefficients are optimized by alternatively learning one while fixing the other, and given a query image, coefficients can be solved by an analytic equation. Experiments show that our method is comparable to and even outperforms the state-of-the-art during different evaluations. More importantly, the retrieved images of our method can share the same semantic properties with the query image, which can be used to other vision tasks, for example, re-training classifiers to discriminate visually similar categories.

1051-4651/14 $31.00 © 2014 IEEE
DOI 10.1109/ICPR.2014.57

The rest of this paper is organized as follows. Section II introduces related work, and Section III presents our methods. Dateset and features are introduced in Section IV, followed by experiments and results in Section V. Conclusions are given in Section VI.

II. Related Work

LSH [1] is the first hashing method which maps high-dimensional vectors into compact binary codes, but its assumption that data are uniformly distributed is not practical. Semantic Hashing [5] is based on restricted Boltzmann machine, tries to learn high-level features for categories, and then encodes these high-level features into binary codes. Several methods [6], [2] use PCA as initial step to reduce the dimension of data, and perform quantization on the resulted orthogonal axes. For example, Spectral hashing (SH) [6] uses a separable Laplacian eigenfunction formulation to assign more bits to directions along which the variances are big. Iterative quantization (ITQ) [2] improves SH by applying rotation to the PCA resulted orthogonal bases and quantizes the transformed feature to the closest vertex of cube. Unlike our methods, all of the above approaches do not explicitly encode semantic meaning into their codes. Our method is similar in spirit of [7] which incorporates discrimination and learnability into its objective function, and can retrieve objects sharing some same properties with the query image, but our codes have explicit expressive meaning, and we do not use class labels as supervisory information.

Attributes can be seen as mid-level features, and can describe absence or not of properties of an object. Farhadi et al. [9] use attributes to perform object naming, object describing and novelty attribute discovering tasks. In [11], attributes are used to recognize unseen objects which share similar properties with seen categories. Attributes can also be used as queries to retrieve related images [19]. Sharmanska et al. [20] augments semantic attributes with extra discriminative attributes to represent image, but different from ours, they learn classifiers for semantic attributes and discriminative attributes separately, and their formula for extra attributes is learned by a non-linear autoencoder.

Matrix factorization formulates data into a linear combination of a set of bases, and many algorithms can be considered as variants of this framework with different constraints. PCA enforces an orthogonality constraint to reduce the origin data into a very low-dimensional representation with great variance. Vector quantization [21] applies a hard winner-take-all constraint and projects data into the nearest cluster center. Sparse coding [22] can also be considered as matrix factorization with encoding the data by a small set of bases. For non-negative matrix factorization [23], input data is assumed non-negative, and it can be seen as positive linear additive combination [23], [24] of a few bases. Unlike all of the above algorithms, our approach do not assume bases are either orthogonal or non-negative, and our coefficients encode semantic meaning with attributes as constraints.

III. Attribute Constrained Matrix Factorization for Learning Semantic Codes

In this section, we will first introduce the general matrix factorization formulation, and then extend it to our model.

A. Matrix Factorization

Given a data matrix $X = (x_1, x_2, .., x_N) \in R^{d\times N}$, where $x_1, ..., x_N \in R^d$ are features, d is the feature dimension and N is the number of images, MF tries to find matrices U and C to approximate X with $X = UC$, of which $U \in R^{d\times k}$ is a set of bases formed by column vector $U_i \in R^d$, and k is the number of bases. $C \in R^{k\times N}$ is the coefficient matrix consisting of column vectors $C_i \in R^k$. The learning objective function for U and C is

$$\min_{U,C} \parallel X - UC \parallel_F^2, \tag{1}$$

where $\parallel \cdot \parallel_F$ denotes the Frobenius norm of a matrix. As discussed in Section II, with different constraints, Eq. (1) can be applied to different problems. For example, by enforcing bases to be orthogonal, i.e. $U^T U = I$, where I is the identity matrix, MF is transfered to PCA. When the number of bases is over-complete, i.e. great bigger than the dimension of data, and coefficients are sparse achieved by using l_1 norm as regularizer, the problem of Eq. (1) is converted to sparse coding [22].

B. Our Model

Given the data matrix X, the goal of our model is to learn a binary code matrix $B \in \{0, 1\}^c$, where c denotes the code length. For each x_i, the corresponding binary code B_i is encoded by function $B_i = sgn(C_i)$, and B_i, C_i is the column vector of B and C respectively. Given a scalar v, $sgn(v) = 1$ if $v \geq 0$ and 0 otherwise, and for a matrix or a vector, $sgn(\cdot)$ denotes the result of entry-wise operation. Similar to [20], We decompose U and C as $U = [U_A, U_E]$, and $C = [C_A, C_E]$, where $U_A \in R^{d\times m}$, $U_E \in R^{d\times q}$, $C_A \in R^{m\times N}$ and $C_E \in R^{q\times N}$, and m and q are the numbers of semantic codes and extra non-semantic codes respectively. Operator $[\cdot, \cdot]$ represents concatenation of vectors.

Now we define our first algorithm MF[1] with no-attributes constraints as following

$$\begin{aligned} &\min_{U,C} \parallel X - UC \parallel_F^2 \\ &subject\ to\ \forall i, \parallel U_i \parallel_2^2 \leq \zeta, \end{aligned} \tag{2}$$

where ζ is a constant for constraining the norm of bases as in [22], and $\parallel \cdot \parallel_2$ is the 2-norm of a vector.

Introducing prior annotated attributes matrix $A^* \in R^{m\times N}$ as constraint for C_A, Eq. (2) is transferred to

$$\begin{aligned} \min_{U_A,U_E,C_A,C_E} \parallel X - U_A C_A - U_E C_E \parallel_F^2 + \\ \gamma \textstyle\sum_i^N (\parallel C_{Ai} \parallel_2^2 + \parallel C_{Ei} \parallel_2^2) \end{aligned} \tag{3}$$

$$subject\ to\ \ sgn(C_A \odot A^*) = \mathbf{1}, \tag{4}$$

$$\forall i, \parallel U_{Ai} \parallel_2^2 \leq \zeta, \parallel U_{Ei} \parallel_2^2 \leq \zeta, \tag{5}$$

where $\odot$ means element-wise multiplication, $\mathbf{1}$ denotes a matrix of which each element is one, and γ is the trade-off coefficient between regularizer and loss function. In experiments, we tried different $\gamma = 10, 1.0, 0.1, 0.01$, but they made no big difference, and we fixed $\gamma = 1.0$. The constraint Eq. (4) means the sign of entries of coefficient matrix C_A and

[1] We mix the name of this weakly constrained function with the general matrix factorization of Eq. (1)

Algorithm 1 Constrained matrix factorization for joint learning

```
Input: data matrix $X, A^*$,Output: $C_A, C_E, U_A, U_E$
Initialize $C_A > 0, C_E, U_A, andU_E$ randomly. Make $C_A = C_A \odot A^*$, so $C_A$ satisfy Eq. 7
repeat
    Compute $C^t = [{C_A}^t, {C_E}^t]$
    $C^t = -({U^{t-1}}^T U^{t-1} + \gamma I)^{-1} U^{t-1} X$:
    for each entry $C_{Aij}$, if ${C_A}^t_{ij} \odot A^*_{ij} < 0$ (violate Eq.(7),
    then, ${C_A}^t_{ij} = 0.5 * {C_A}^{t-1}_{ij}$, $t$ is the iterate number
    Compute $U^t$ by Lagrange dual Eq.(10) and Eq.(11)
until converged
return $C_A, C_E, U_A, U_E$ from $C$ and $U$
```

ground truth semantic attributes matrix A^* coincides. Taking $U = [U_A, U_E]$ and $C = [C_A, C_E]$ into Eq. (4), we obtain

$$\min_{U,C} \parallel X - UC \parallel_F^2 + \gamma \sum_i^N \parallel C_i \parallel_2^2, \tag{6}$$

$$subject\ to \quad sgn(C_A \odot A^*) = \mathbf{1}, \tag{7}$$

$$\forall j, \parallel U_j \parallel_2^2 \leq \zeta, \tag{8}$$

C. Optimization Methods

The objective functions of Eq. (2) and Eq. (3) are convex in U (while holding C fixed) and convex in C (while holding U fixed), but not convex in both simultaneously. For learning coefficient matrix C, the optimization problem is a general constrained quadratic optimization problem, and C can take analytic solution, i.e. $C = (U^T U)^{-1} UX$ and $C = (U^T U + \gamma I)^{-1} UX$ respectively for Eq. (2) and Eq. (3). For learning the bases U, Eq. (6) is reduced to Eq. (2), and the optimization problem is a constrained least-square problem, which can be solved much more efficiently by lagrange dual method as in [22]. The Lagrangian of Eq. (2) is

$$L(U, \vec{\lambda}) = trace((X - UC)^T (X - UC)) + \sum_{j=1}^N \lambda_j (\sum_{i=1}^k U_{ij}^2 - \zeta), \tag{9}$$

where $\vec{\lambda}$ is a vector of dual variables, and each $\lambda_j \geq 0$. This Lagrangian can be minimized over U analytically, then we obtain the Lagrange dual

$$\begin{aligned} D(\vec{\lambda}) &= \min_U L(U, \vec{\lambda}) \\ &= trace(X^T X - XC^T (CC^T + \Lambda)^{-1} (XC^T)^T - \zeta \Lambda), \end{aligned} \tag{10}$$

where Λ is the diag matrix of λ_j. This Lagrange dual (Eq.(10)) can be optimized by the Newton gradient method as in [22], and after solving $D(\vec{\lambda})$, the optimal basis matrix is as follow

$$U^T = (CC^T + \Lambda)^{-1} (XC^T)^T. \tag{11}$$

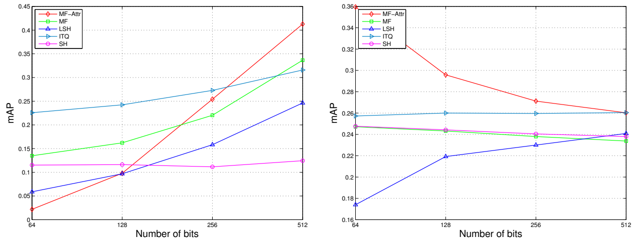
The overall algorithm is described in algorithm (1).

IV. Datasets and Features

We evaluated our methods on the **a-Pascal** dataset [9], which was collected from Pascal VOC2008 dataset. It contains 20 object classes, namely people, bird, cat, cow, dog, horse, sheep, aeroplane, bicycle, boat, bus, car, motorbike, train, bottle, chair, dining table, potted plant, sofa, and tv/monitor. Each category contains about 150 to 1000 objects, except that people category has more than 5000 instances. The authors [9] also collected 64 semantic attribute annotations to describe objects (refer to table I). Note that, for images from the same category, their annotated attributes may be a little different due to variety of natural poses, viewpoints and orientations.

TABLE I. List of Attributes

2D Boxy	3D Boxy	Round	Vert Cyl
Horiz Cyl	Occluded	Tail	Beak
Head	Ear	Snout	Nose
Mouth	Hair	Face	Eye
Torso	Hand	Arm	Leg
Foot/Shoe	Wing	Propeller	Jet engine
Window	Row Wind	Wheel	Door
Headlight	Taillight	Side mirror	Exhaust
Pedal	Handlebars	Engine	Sail
Mast	Text	Label	Furn. Leg
Furn. Back	Furn. Seat	Furn. Arm	Horn
Rein	Saddle	Leaf	Flower
Stem/Trunk	Pot	Screen	Skin
Metal	Plastic	Wood	Cloth
Furry	Glass	Feather	Wool
Clear	Shiny	Vegetation	Leather



(a) Euclidean ground truth

(b) Class label ground truth

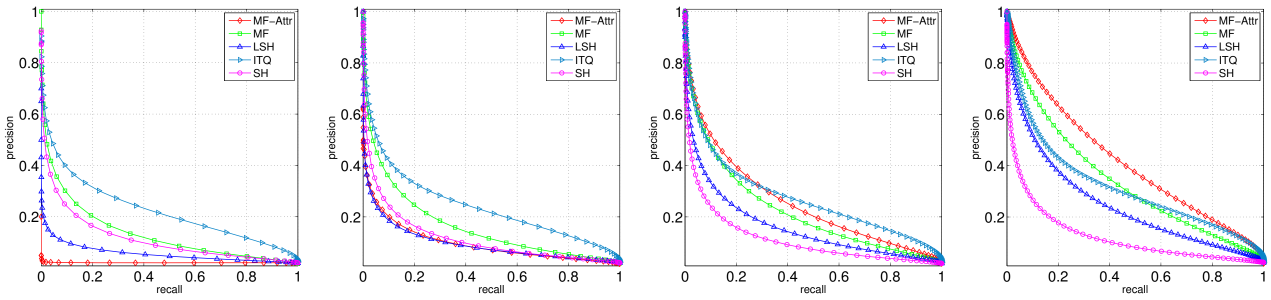
Fig. 1. Comparative evaluation results on a-Pascal dataset with respect to the number of code bits. The number of code bits refers to the sum of the number of semantic code bits and the number of non-semantic code bits, and the bit number of semantic codes is always 64. For 64-bit codes, there are no non-semantic codes. (a) The performance is measured by the mean average precision (mAP) for retrieval using top 50 Euclidean neighbors of each query image as true positives. Refer to Figure 2 for complete recall-precision curves for all the methods. (b) The performance is measured by the averaged precision of the top 100 ranked images for each query, the semantic class labels are considered as ground truth. Refer to Figure 3 for the complete class label precision curves for all the methods.

We use the same features[2] as [9], which are bag of words style features of color, texture, HOG [25] and edges. Texture descriptors are quantized to the nearest 256 k-means centers, HOG descriptors are quantized to 1000 k-means centers, edges are quantized into 8 unsigned bins, and color descriptors are quantized to 128 k-means centers. LAB values are also included in the color descriptor. To represent image better, the box of each object is divided into three vertical and two horizontal blocks, and histograms are computed within each block and the whole box. Then these seven histograms are stacked to form a 9751 dimensional feature.

V. Experiments and Results

We first evaluated our methods on the image retrieval task. Then we evaluated the performance of MF-Attr on the task of learning semantic compact codes, and the precision of retrieved attributes with respect to ground truth attributes is computed.

[2] http://vision.cs.uiuc.edu/attributes/



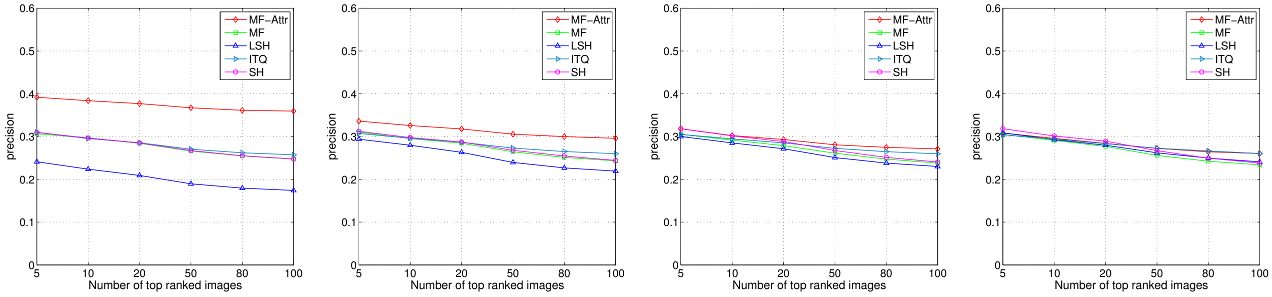
(a) 64bits

(b) 128bits

(c) 256bits

(d) 512bits

Fig. 2. From left to right: Recall precision curves @64bits, @128bits, @256bits and @512bits respectively. Euclidean neighbors are used as ground truth. See figure 1(a) for reference to summary of the mean average precision of all the methods with code size as variable.



(a) 64bits

(b) 128bits

(c) 256bits

(d) 512bits

Fig. 3. From left to right: Label precision curves @64bits, @128bits, @256bits and @512 bits respectively . Class labels are used as ground truth. See figure 1(b) for reference to summary of the mean average precision of all the methods with code size as variable.

A. *Image Retrieval*

Protocols: Our methods were evaluated following two widely used protocols as in [2], [4], [6]. The first one is the nearest neighbor search using Euclidean neighbors as ground truth. The average distance to the 50th nearest neighbor is used as a nominal threshold to determine whether a retrieved image is considered as a true positive for the given query, and the recall-precision curve and the mean average precision (mAP) were computed. Second, we use class label as ground truth to evaluate the semantic consistency of codes, and precision is computed on the top 100 ranked retrieval images.

In experiments, we uniformly divided the a-Pascal dataset into two sets, namely a-Pascal-train dataset and a-Pascal-test dataset, and used a-Pascal-train as training set. During testing, we randomly selected 50 samples from each category of a-Pascal-test as test queries, and the queries are performed on the a-Pascal-train dataset. Each experiment was performed 5 times. We compared MF-Attr with the state-of-the-art ITQ, and three baselines, i.e. LSH, SH, and MF, and results were reported with respect to different code sizes. Bear in mind that, for MF-Attr encoded codes, the semantic codes is of fixed 64-bit length, and the extra remaining bits are non-semantic codes. For 64-bit codes, the size of non-semantic codes is 0.

Results: Figure 1(a) summaries the mAP of MF-Attr, MF, LSH, SH and ITQ as a function of code size, and Euclidean nearest neighbors are considered as ground truth. We can see from figure 1(a) that the performances of all methods improve as code size increases, especially, the mAP of MF-Attr grows up with the biggest slope. MF-Attr does not perform well at code sizes 64 and 128, but as the number of bits increases, it is comparable to the state-of-the-art ITQ at bits 256, and even outperforms the state-of-the-art when the number of bits is 512.

In figure 1(b) where class label is used as ground truth, MF-Attr is superior to all the other methods. However, as the code size increases, the performance of MF-Attr decreases a lot. When the code size is 64, the mAP of MF-Attr is about 0.36, about 0.1 greater than the second best. When the code size is 512, the performance of MF-Attr is almost the same as the state-of-the-art. The performances of MF-Attr are different in figure 1(b) from those in figure 1(a), and we argue this may be caused by our goal of learning semantic codes. Semantic codes enforce its bits to coincide with semantic attributes, so they implicitly discriminate categories. However, since the size of non-semantic codes increases as the total code size increase, more non-semantic codes reduce the performance of semantic codes, but improve the performance of retrieving when using Euclidean distance as ground truth. Semantic codes and non-semantic codes behave like regularizer in terms of that, semantic codes make final codes preserve category-level similarity, and non-semantic codes make final codes preserve similarity in origin data. It's surprising to note that our method MF, which does not encode attribute constraints, also achieves promising results in all the experiments. It may be because MF does not make any assumptions on data.

Corresponding to figure 1(a) and figure 1(b), figure 2 and figure 3 show complete recall-precision and class label precision curves. From figure 2(a) to figure 2(d), the performance curve of MF-Attr moves up, and achieves the best at code size 512, while the others stay in the same order. In figure 3, MF-Attr outperforms other methods, its performance decreases as the code size increases, which coincides with figure 1. It is interesting that, when the number of code bits is 512, the performances of all methods are almost the same, and it may be because that the a-Pascal dataset is really challenging, and the performance of methods cannot improve anymore as the code size increases.

B. Attributes Retrieval

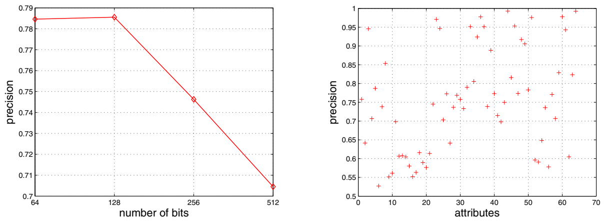
Because our codes explicitly encode semantic attributes, we evaluated MF-Attr on attribute retrieval task, and precisions are computed for total attributes and each attribute respectively. The precision for total attributes retrieval is defined as a ratio of the sum of the number of correct retrieved attributes in each image over the sum of the number of all the attributes in each image, and the precision of each attribute retrieval is defined as a ratio of the sum of the number of correct retrieved images with the attribute over the sum of the number of images with the attribute. Results are reported on the whole a-Pascal-test dataset with respect to different code sizes.

Figure 4(a) shows that the precision curve of MF-Attr on retrieval of all the attributes goes down as the code size increases. It may be because that, similar as figure 1(b), non-semantic codes behave as regularizer for semantic codes and reduce the performance, when the number of code size increases. Figure 4(b) shows the distribution of the precision with respect to the attribute at code size 512. The highest value is near 1.0, while the lowest one is about 0.5. The top 10 retrieval attributes are Horn, Leather, Wool, Sail, Screen, Propeller, Saddle, Pedal, Mast, and Jet engine, while Metal, Arm, Face, Cloth, Leg, Torso, Ear, Eye, Head, and Occluded are the lowest 10 attributes. We can discover from the results that most of the attributes with higher precision are global properties of objects, whereas lower ones are local descriptions. Although the features have localizing functionality, they are still not accurate enough to represent local attributes.

Figure 5 displays bad retrieval examples of all methods, where from top to bottom are MF-Attr, MF, ITQ, LSH, and SH respectively. Given a plane image as query, MF-Attr gets four boat images and one plane image, and even if this result is bad, the retrieved images share the similar semantic meaning with the query image. Unlike ours, other methods do not show semantic relationship between the retrieved images and the query image. For example, ITQ, LSH and SH retrieve more images from people category which shares no similar attributes with plane category. Note that, MF method can also get images sharing semantic property with the query image, but it does not have explicit meaning in its codes. Refer to figure 6 for more comparision examples and figure 7 for more examples of MF-Attr.

VI. Conclusion

In this paper, we have proposed to use matrix factorization to learn semantic binary codes by encoding attributes as constraints, and evaluations have been performed on dataset a-Pascal. Semantic codes really help retrieval performances when using class label as ground truth, and non-semantic codes help



(a) (b)

Fig. 4. Evaluation of method MF-Attr on Retrieval of Attributes. The left figure is the precision curve of total attributes retrieval with respect to code sizes, and the right figure is the precision for each attribute.

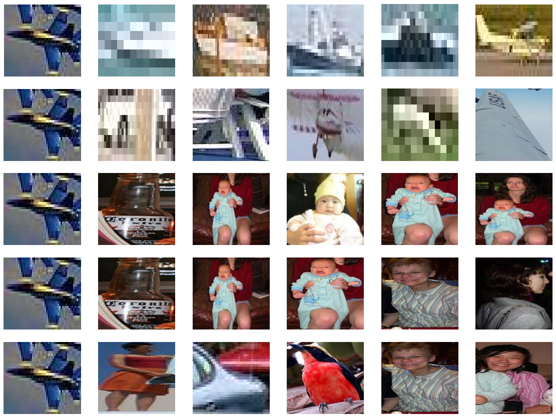


Fig. 5. Comparisons of retrieval results of methods: MF-Attr, MF, ITQ, LSH and SH from top to bottom. The first column is the query image, and among the other columns are the five retrieved images. To note that, these are not the best results. In spite of MF-Attr are not very successful, we can find that the retrieval images of MF-Attr have explicit semantic relation with the query image, whereas the other methods just find feature-similar images not semantic-similar ones. Taking the third column for example, MF-Attr retrieves a boat image, which shares metal property with the given query image, and both of these two are transports. The remaining images for ITQ, LSH, SH are baby, baby, and bird. It's interesting that ITQ, LSH and SH share some retrieved images, it may be because they directly map origin data into low-dimensional space.

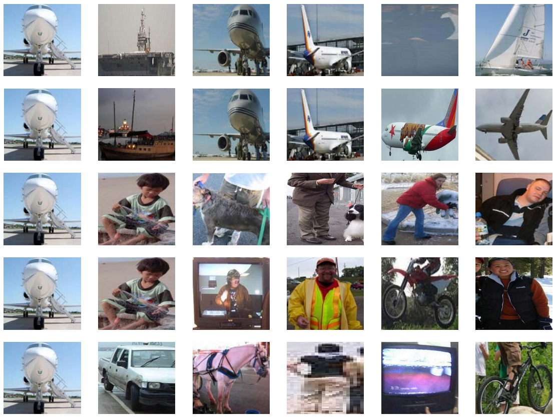


Fig. 6. Another example of comparisons of retrieval results of methods: MF-Attr, MF, ITQ, LSH and SH from top to bottom. The first column is the query image, and among the other columns are the five retrieved images (refer to figure 5 for description).

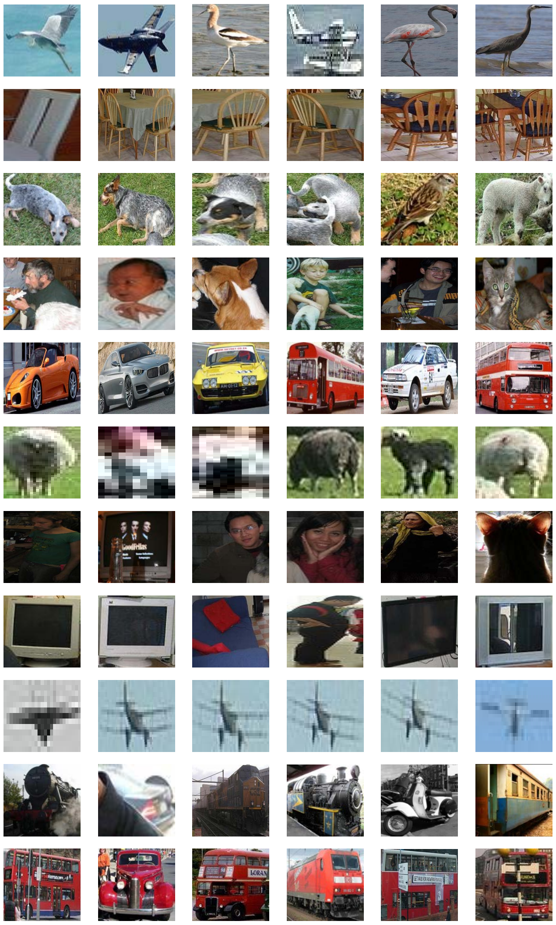


Fig. 7. More examples of MF-Attr methods. The first column is the query image, and the remainings are the top 5 retrieved images. We can see from this figure that, our method can retrieve accurate images within the same category or sharing some semantic attributes.

preserve similarity in origin data. Our framework is simple, and only encodes attributes as constraints, without leveraging class labels for supervision. In the future, we will explore class label information to augment performance.

Acknowledgment

This work was supported by NSFC grant (No. 61371134) and 973 Program (Project No. 2010CB327900) and the authors would like to give warmly thanks to Haopeng Zhang and Jun Shi for their helpful discussions.

References

[1] A. Andoni and P. Indyk, "Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions," in *Foundations of Computer Science, 2006. FOCS'06. 47th Annual IEEE Symposium on*. IEEE, 2006, pp. 459–468.

[2] Y. Gong and S. Lazebnik, "Iterative quantization: A procrustean approach to learning binary codes," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 817–824.

[3] 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.

[4] M. Raginsky and S. Lazebnik, "Locality-sensitive binary codes from shift-invariant kernels," in *Advances in neural information processing systems*, 2009, pp. 1509–1517.

[5] R. Salakhutdinov and G. Hinton, "Semantic hashing," *International Journal of Approximate Reasoning*, vol. 50, no. 7, pp. 969–978, 2009.

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

[7] M. Rastegari, A. Farhadi, and D. Forsyth, "Attribute discovery via predictable discriminative binary codes," in *Computer Vision–ECCV 2012*. Springer, 2012, pp. 876–889.

[8] H. Hotelling, "Relations between two sets of variates," *Biometrika*, vol. 28, no. 3/4, pp. 321–377, 1936.

[9] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth, "Describing objects by their attributes," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 1778–1785.

[10] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar, "Attribute and simile classifiers for face verification," in *Computer Vision, 2009 IEEE 12th International Conference on*. IEEE, 2009, pp. 365–372.

[11] C. H. Lampert, H. Nickisch, and S. Harmeling, "Learning to detect unseen object classes by between-class attribute transfer," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 951–958.

[12] D. Parikh and K. Grauman, "Relative attributes," in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 503–510.

[13] F. X. Yu, L. Cao, R. S. Feris, J. R. Smith, and S.-F. Chang, "Designing category-level attributes for discriminative visual recognition," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*. IEEE, 2013, pp. 771–778.

[14] M. Rastegari, A. Diba, D. Parikh, and A. Farhadi, "Multi-attribute queries: To merge or not to merge?" in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*. IEEE, 2013, pp. 3310–3317.

[15] J. Liu, B. Kuipers, and S. Savarese, "Recognizing human actions by attributes," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 3337–3344.

[16] K. Duan, D. Parikh, D. Crandall, and K. Grauman, "Discovering localized attributes for fine-grained recognition," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 3474–3481.

[17] L. Torresani, M. Szummer, and A. Fitzgibbon, "Efficient object category recognition using classemes," in *Computer Vision–ECCV 2010*. Springer, 2010, pp. 776–789.

[18] I. Jolliffe, *Principal component analysis*. Wiley Online Library, 2005.

[19] B. Siddiquie, R. S. Feris, and L. S. Davis, "Image ranking and retrieval based on multi-attribute queries," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 801–808.

[20] V. Sharmanska, N. Quadrianto, and C. H. Lampert, "Augmented attribute representations," in *Computer Vision–ECCV 2012*. Springer, 2012, pp. 242–255.

[21] A. Gersho and R. M. Gray, *Vector quantization and signal compression*. Springer, 1992.

[22] H. Lee, A. Battle, R. Raina, and A. Ng, "Efficient sparse coding algorithms," in *Advances in neural information processing systems*, 2006, pp. 801–808.

[23] D. Seung and L. Lee, "Algorithms for non-negative matrix factorization," *Advances in neural information processing systems*, vol. 13, pp. 556–562, 2001.

[24] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.

[25] 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.