Journal of Southeast University (English Edition) Vol. 29, No. 4, pp. 372 - 378 Dec. 2013 ISSN 1003—7985

Learning compact binary code based on multiple heterogeneous features

Zuo Xin[1,2] Luo Limin[1] Shen Jifeng[3] Yu Hualong[2]

([1]Laboratory of Image Science and Technology, Southeast University, Nanjing 210096, China)
([2] School of Computer Science and Engineering, Jiangsu University of Science and Technology, Zhenjiang 212003, China)
([3]School of Electrical and Information Engineering, Jiangsu University, Zhenjiang 212013, China)

Abstract: A novel hashing method based on multiple heterogeneous features is proposed to improve the accuracy of the image retrieval system. First, it leverages the imbalanced distribution of the similar and dissimilar samples in the feature space to boost the performance of each weak classifier in the asymmetric boosting framework. Then, the weak classifier based on a novel linear discriminate analysis (LDA) algorithm which is learned from the subspace of heterogeneous features is integrated into the framework. Finally, the proposed method deals with each bit of the code sequentially, which utilizes the samples misclassified in each round in order to learn compact and balanced code. The heterogeneous information from different modalities can be effectively complementary to each other, which leads to much higher performance. The experimental results based on the two public benchmarks demonstrate that this method is superior to many of the state-of-the-art methods. In conclusion, the performance of the retrieval system can be improved with the help of multiple heterogeneous features and the compact hash codes which can be learned by the imbalanced learning method.

Key words: hashing code; linear discriminate analysis; asymmetric boosting; heterogeneous feature

doi: 10.3969/j. issn. 1003 - 7985. 2013. 04. 004

In the last few years, the problem of learning similarity-preserving binary codes for large-scale image retrieval has received much attention in the vision community. Learning compact binary code from high dimensional features can enable significant gains in computational speed and storage cost. The scheme of learning binary codes should have the following properties such as low storage cost, high similarity preserving and computational efficiency. The above constraints need to be satisfied simultaneously, which makes the hashing code learning very challenging.

For the image retrieval problem, it is quite intuitive to use linear search methods, but it is computationally forbidden for large scale datasets. This limitation induces the development of various approximate nearest neighbor (ANN) algorithms. The popular ANN approaches can be divided into two categories, tree-based methods and hashing-based methods. In the former category, there are numerous algorithms such as KD-tree[1], metric tree[2] and so on. It is shown that these methods perform very well in relatively low dimensional data, and the computational complexity will deteriorate into linear time when the dimension of data is increased. The latter is the most popular method due to the merit of significant reduction in storage and sub-linear search time. But the key issue of hashing-based image retrieval methods is how to design a compact code maintaining high precision and recall rate.

In this paper, we focus on the hashing-based methods and our goal is to learn discriminative binary code in the supervised learning framework. Locality-sensitive hashing (LSH)[3] is the most popular hashing method, which is also a data-independent method. The LSH method needs a relatively long code to obtain high recall rate. One solution to this problem is to use multiple tables which encode short codes to achieve high performance, but it also increases the computational burden and storage overhead. Another method to deal with this problem is to use multi-probe LSH[4], which maps similar samples into multiple buckets intelligently. Recently, the LSH has been extended to accommodate kernel similarity[5], which implicitly models the mapping from original space to hamming space by utilizing the kernel trick to compute the similarity between two samples.

In contrast to the above described data-independent hash schemes, recent research aims at data-dependent hashing in order to learn a compact set of discriminative hash codes. ITQ [6] produces the quantization of hash codes in a geometric view and minimizes the quantization error by finding an orthogonally rotation matrix after obtaining the PCA coefficients. Spectral hashing (SH)[7] seeks compact binary code to preserve the affinity among

Received 2013-07-21.

Biographies: Zuo Xin (1980—), female, graduate; Luo Limin (corresponding author), male, doctor, professor, luo. list@ seu. edu. cn.

Foundation items: The National Natural Science Foundation of China (No. 61305058), the Natural Science Foundation of Higher Education Institutions of Jiangsu Province (No. 12KJB520003), the Natural Science Foundation of Jiangsu Province (No. BK20130471), the Scientific Research Foundation for Advanced Talents by Jiangsu University (No. 13JDG093).

Citation: Zuo Xin, Luo Limin, Shen Jifeng, et al. Learning compact binary code based on multiple heterogeneous features. [J]. Journal of Southeast University (English Edition), 2013, 29(4): 372 - 378. [doi: 10.3969/j. issn. 1003 - 7985. 2013. 04. 004]

data points both in hamming space and original space. Shift-invariant kernels hashing (SIKH)[8] uses the shift-invariant kernels to represent the relationship of hamming distance between two points. Semantic hashing[9] makes use of restricted Boltzmann machine which has similar functionality to neural networks to learn compact code. Binary reconstruction embedding (BRE)[10] constructs binary code by minimizing the difference between the original metric and hamming distance, and uses the coordinate-descent algorithm to solve the optimization problem. Minimal loss hashing (MLH)[11] formulates the problem based on the structured learning framework and solves it with a perception-like algorithm to obtain the optimized binary code. Semi-supervised hashing (SSH)[12] learns the code by minimizing the empirical error with an information theoretic regularizer over the labeled set which is solved with a common gradient descent algorithm. Anchor graph hashing (AGH)[13] learns the compact code by utilizing the graph-based hashing method which automatically discovers the neighborhood structure in the original space. LDAHash[14] intends to find a mapping to minimize the expectation of hamming distance in a positive pair set and maximize the hamming distance in a negative pair set.

The above mentioned methods gain great progress in learning compact and discriminative code, but still face many problems which are very difficult to tackle with. In this paper, we propose a novel hashing method based on the asymmetric boosting framework, which considers the imbalanced distribution of both similar and dissimilar samples in the feature space. Our method learns each coding bit step by step, which takes into account the distribution information between similar and dissimilar pairs.

1 Related Work

1.1 Locality-sensitive hashing

The key idea of LSH[15] is mapping similar data points to the same bucket with high probability. We denote $h(\boldsymbol{x})$ as the hashing function from the LSH family. It has the following locality preserving property:

$$P\{h(\boldsymbol{x}) = h(\boldsymbol{y})\} = \mathrm{sim}(\boldsymbol{x}, \boldsymbol{y}) \tag{1}$$

where $h(\boldsymbol{x}) = \mathrm{sign}(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + b)$; $\boldsymbol{w}$ is a random vector from a p-stable distribution and b is a random intercept; $\mathrm{sim}(\boldsymbol{x}, \boldsymbol{y}) = \exp(\parallel \boldsymbol{x} - \boldsymbol{y} \parallel^2/\sigma^2)$; σ is the standard deviation of the Gaussian function.

Suppose that we learn a K-bit binary code denoted by $H(\boldsymbol{x}) = [h_1(\boldsymbol{x}), h_2(\boldsymbol{x}), \ldots, h_K(\boldsymbol{x})]$, and give l hash tables. Then the collision probability[16] of two points is calculated as

$$P(H(\boldsymbol{x}) = H(\boldsymbol{y})) \propto l\left[1 - \frac{\cos^{-1}(\boldsymbol{x}^{\mathrm{T}}\boldsymbol{y})}{\pi}\right]^K \tag{2}$$

There is a tradeoff between parameters l and K. Large K will decrease the collision probability, which reduces false collisions and also affect the similar samples. Large l can reduce this effect, but it will increase the computational complexity.

1.2 Boost similarity sensitive coding (BoostSSC)

BoostSSC[17] is aimed to learn an embedding from the original space to the hamming space. Given that a sample $\boldsymbol{x}$ is represented by a binary code with M bits $\boldsymbol{y}_i = [h_1(\boldsymbol{x}), h_2(\boldsymbol{x}), \ldots, h_M(\boldsymbol{x})]$, the distance between two samples $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ is given by a weighted hamming distance $D(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sum_{n=1}^{M} a_n \mid h_n(\boldsymbol{x}_i) - h_n(\boldsymbol{x}_j) \mid$. The weight α_n and hamming function h_n are learned in the boosting framework. Each weak classifier f_n which minimizes the square loss in the training set is learned in each iteration.

$$f_n = \arg\min\left(\sum_{k=1}^{K} w_n^k (z_k - f_n(\boldsymbol{x}_i^k, \boldsymbol{x}_j^k))^2\right) \tag{3}$$

where z_k is the neighborhood label and w_n^k is the weight for sample k in the n-th iteration.

$$w_n^k = \exp\left(-z_k \sum_{t=1}^{n-1} f_t(\boldsymbol{x}_i^k, \boldsymbol{x}_j^k)\right) \tag{4}$$

where $f_t(\boldsymbol{x}_i, \boldsymbol{x}_j) = \alpha_t[(\boldsymbol{e}_t^{\mathrm{T}}\boldsymbol{x}_i > T_t) = (\boldsymbol{e}_t^{\mathrm{T}}\boldsymbol{x}_j > T_t)] + \beta_t$; α_t and β_t are the coefficients; $\boldsymbol{e}_t$ is the unit vector with only the t-th component equal to 1.

We can see that the each weak classifier only considers a single dimension of the feature to encode signature which is less informative. Our proposed method makes use of a subspace of multiple features to represent each bit and it will be discussed in section 2 in detail.

2 Discriminative Binary Code Learning

2.1 Semi-supervised hashing based on LDA

Traditional hash coding methods focus on learning in an unsupervised way, but using label information can actually boost the retrieval performance in a large margin. In this section, we will introduce a semi-supervised hashing algorithm which intends to minimize the hamming distance in the same class, and maximize the hamming distance in different classes. Given a set of samples $\boldsymbol{\chi} = \{\boldsymbol{x}_i\}_{i=1}^N$, where $\boldsymbol{x}_i \in \mathbf{R}^d$, the purpose of hashing is to find a set of hash function h: $\mathbf{R}^d \rightarrow \{0, 1\}^L$, where each one generates a single hash bit. The objective function is as follows:

$$\min \alpha E\{ \mid \boldsymbol{y} - \boldsymbol{y}' \mid^2 \mid S\} - E\{ \mid \boldsymbol{y} - \boldsymbol{y}' \mid^2 \mid D\} \tag{5}$$

where S and D represent the same sample set and the heterogeneous sample set, respectively; $E\{\cdot\}$ is the expectation with respect to the set S or set D; $\boldsymbol{y} = \mathrm{sign}(\boldsymbol{Px})$, and $\boldsymbol{P}$ is the projection matrix needed to be learned; α is a scalar parameter. Then Eq. (5) can be converted to

$$\min \alpha E\{ \mid \boldsymbol{Px} - \boldsymbol{Px}' \mid^2 \mid S\} - E\{ \mid \boldsymbol{Px} - \boldsymbol{Px}' \mid^2 \mid D\} \tag{6}$$

We can see from Eq. (6) that

$$E\{\,|\boldsymbol{Px}-\boldsymbol{Px}'|^2\,|\,S\}=\mathrm{Tr}\{\boldsymbol{P\Sigma}_S\boldsymbol{P}^{\mathrm{T}}\}$$
$$\boldsymbol{\Sigma}_S=E\{(\boldsymbol{x}-\boldsymbol{x}')(\boldsymbol{x}-\boldsymbol{x}')^{\mathrm{T}}\,|\,S\} \tag{7}$$

Eq. (7) can be reformulated into our final objective function as

$$\boldsymbol{L}=\mathrm{Tr}\{\boldsymbol{P\Sigma}_R\boldsymbol{P}^{\mathrm{T}}\} \tag{8}$$

where $\boldsymbol{\Sigma}_R=\boldsymbol{\Sigma}_S\boldsymbol{\Sigma}_D^{-1}$. Since $\boldsymbol{\Sigma}_R$ is a symmetric positive semi-definite matrix, it can be solved with eigen decomposition $\boldsymbol{\Sigma}_R=\boldsymbol{U\Lambda U}^{\mathrm{T}}$. The matrix $\boldsymbol{P}$ can be obtained by the m smallest eigenvectors of matrix $\boldsymbol{\Sigma}_R$, i. e.,

$$\boldsymbol{P}=((\boldsymbol{\Sigma}_R)_m)^{1/2} \tag{9}$$

The LDA criterion has a good generalizing ability, the effectiveness of which is theoretically proved and also experimentally verified[14]. Fig. 1 shows the projection from original 2D space to 1D line. We can see that the data of different classes can be well separated by choosing an appropriate direction of projection.

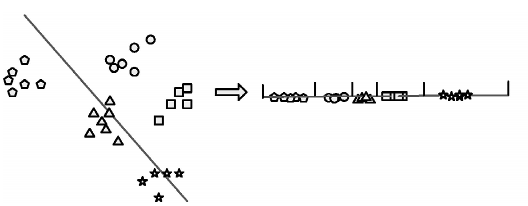

Fig. 1 Binary coding with LDA criterion

2.2 Learning based on asymmetric boosting

In the training stage, the positive samples are pairs of images $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, where $\boldsymbol{x}_j$ is one of the N nearest neighbors of $\boldsymbol{x}_i$, $\boldsymbol{x}_j\in N(\boldsymbol{x}_i)$. Negative samples are pairs of images that are not neighbors. We utilize the asymmetric gentle AdaBoost algorithm for training, and the weak classifier is implemented with a lookup table(LUT) algorithm which is different from the BoostSCC method[17]. Each weak classifier encodes multiple bits instead of a single bit, which is different from the previous methods. At each iteration n, we select a weak classifier f_n which minimizes the loss in the training set, and the objective function is shown in Eq. (5). But the weight w_n^k for sample k in the n-th iteration is formulated as

$$w_n^k=\exp\left(-z_kC_k\sum_{t=1}^{n-1}f_t(\boldsymbol{x}_i^k,\boldsymbol{x}_j^k)\right) \tag{10}$$

where C_k is an asymmetric factor which has a large penalty for misclassified similar pairs. The LUT based on weak classifiers is

$$f_n(\boldsymbol{x}_i,\boldsymbol{x}_j)=\mathrm{LUT}[\,|\boldsymbol{w}^{\mathrm{T}}F(\boldsymbol{x}_i)-\boldsymbol{w}^{\mathrm{T}}F(\boldsymbol{x}_j)|\,] \tag{11}$$

where the LUT function returns a binary code, which is corresponding to the learned projection $\boldsymbol{w}$. The LUT is constructed similarly to Ref. [18], which divides the projected feature space into multiple sub-regions and encodes each region with different signatures. The advantage of using the LUT method is that it can model more complex data distribution[19].

2.3 Fusing multiple heterogeneous features

The framework of our proposed method is shown in Fig. 2. Given a set of training images, we first extract the feature vector, and denote it as $\mathbf{R}^D$. Then we utilize random sampling to select a lower dimensional set $\mathbf{R}^d$, where $d\ll D$. After sampling, all the data are mapped into a lower dimensional space. The optimal projection vector can be learned in a semi-supervised manner which is based on LDA criterion. Each projection can generate one bit for hashing code. We use the boosting algorithm to learn each bit sequentially, which considers the samples' distribution and focuses on the misclassified samples in each round. The motivation of this paper is to utilize heterogeneous features to capture the color, texture, spatial information, etc. In this paper, we use the Gist feature, BoW(bag of words) feature, wavelet texture and color moment to capture the discriminative information among different classes which are complementary to each other.

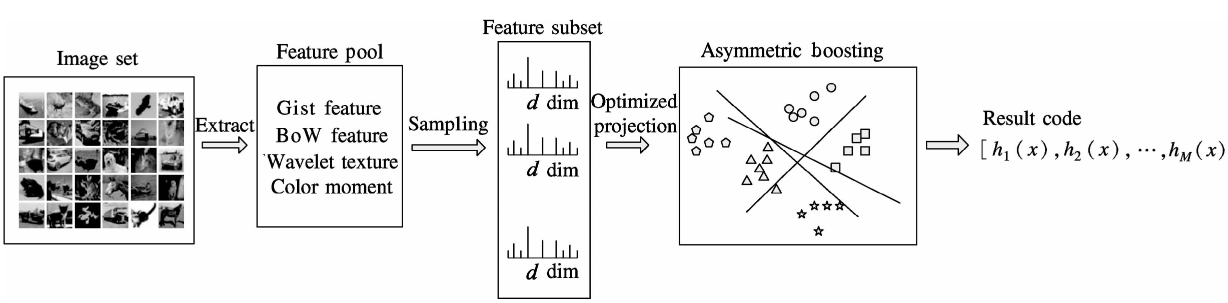

Fig. 2 Binary code learning

After obtaining the projection matrix $\boldsymbol{P}$, the hash code is calculated by formula $\boldsymbol{y}=\mathrm{sign}(\boldsymbol{Px})$. So the computational complexity in the query stage is $O(Md)$, where M is the length of the code and d is the dimension of the sub-feature vector.

3 Image Retrieval Procedure

We build an image retrieval system based on the method described in section 2, which is shown in Fig. 3. Our system consists of two stages, training and testing. The training stage can be learned offline and the testing stage runs online.

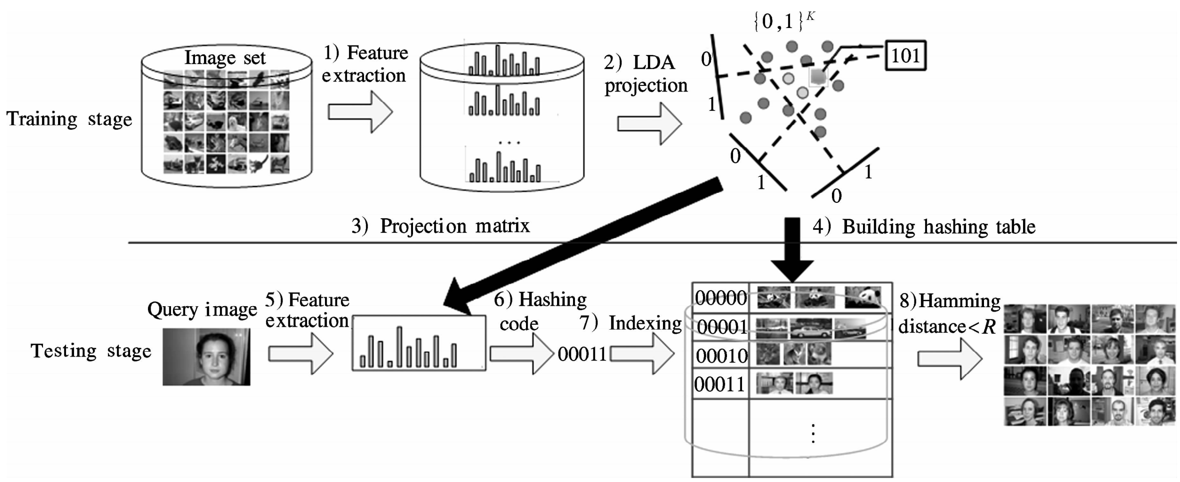

Fig. 3 Image retrieval procedures

In the training stage all the training data extract features from original images (step 1) and learn a set of LDA-based projections (step 2) in order to generate a set of binary codes. All the projection parameters can be represented with a projection matrix (step 3). The mapping of feature space to hamming space can preserve the semantic similarity between images, and similar pairs will have similar hamming codes. In other words, similar images can be mapped into the hashing table buckets within a small hamming distance. After obtaining the projection matrix, the codes of the whole dataset can be calculated (step 4). These codes are loaded into memory during the testing stage.

In the testing stage, each query is converted into a feature vector (step 5) and the binary code is generated (step 6) with the learned projection matrix. After obtaining the binary code of a query image, we can look up the hash table (step 7) and rank the results according to the hamming distance among query codes and dataset codes. There is a threshold R (step 8) which controls the number of returned results. In this manner, we can obtain the results of query in sub-linear time.

4 Experimental Results and Analysis

We evaluate our proposed method on CIFAR-10 dataset and LabelMe22K dataset, which are public in the internet. We compare our method with many state-of-the-art algorithms such as spectral hashing (SH), binary reconstruction embedding (BRE), locality-sensitive hashing (LSH), and iterative quantization (ITQ).

4.1 Dataset

The CIFAR-10 dataset is a labeled subset of an 80 million tiny image collection. It consists of 6×10^4 color images with 32×32 resolutions and is categorized into 10 classes. Each class contains 6 000 samples and each sample has a class label. Some example images from the CIFAR-10 dataset are shown in Fig. 4(a). The LabelMe22K dataset consists of 22 019 images, and they are divided into 20 019 training images and 2 000 test images. Each image extracts the 512D Gist feature, the 300D BoW feature, the 128D wavelet texture and the 225D color moment as features. Some of the example images from the LabelMe22K dataset are shown in Fig. 4(b).

Fig. 4 Sample images. (a) CIFAR-10 dataset; (b) LabelMe22K dataset

In the following experiment, we fix all the parameters. We set $d = 64$ for the dimension of feature subset. The asymmetric factors C_1 and C_2 of asymmetric gentle AdaBoost are set to be 1 and 0.25, respectively. The 256-bin LUT table is used for each weak classifier.

4.2 Results on CIFAR-10 dataset

We evaluate the experimental results by the number of bits vs. precision curve with hamming radius $R \leqslant 3$. Fig. 5 shows the results in the CIFAR-10 dataset. We can see from Fig. 5 that our proposed method is superior than those of the LSH, SH, ITQ and SIKH methods. Our method can obtain a precision of 0.792 4 encoding with only 32 bits which is 6% higher than that of the ITQ method, 12% higher than that of the LSH method. We also find that the SIKH method obtains quite low precision of approximately 6%. Both the ITQ and LSH methods have a very similar precision which is also superior to that of the SH method. The precision-recall curve is demonstrated in Fig. 6, which measures the precision with specified recall rate. From Fig. 6, we can see that our method also performs the best among all the methods. Our method is comparable with the ITQ and SH methods when the recall is greater than 0.6 and has a large margin among all the others when the recall is below 0.4.

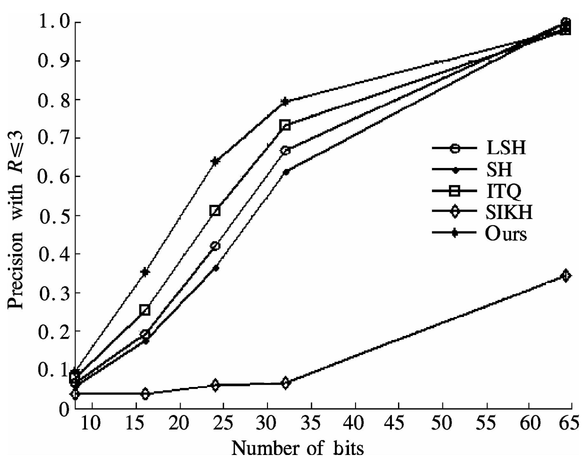

Fig. 5 Number of bits vs. precision curves (CIFAR-10 dataset)

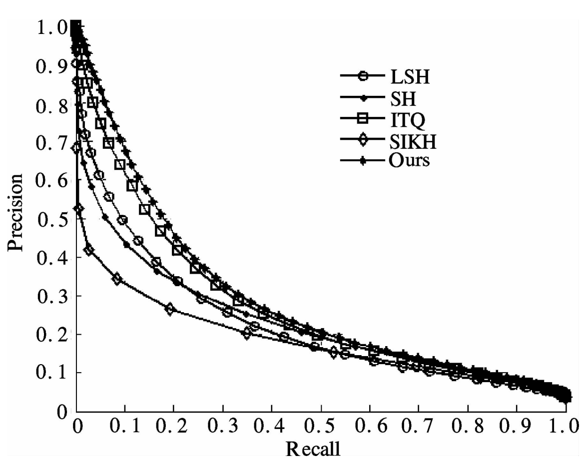

Fig. 6 Precision-recall curves (CIFAR-10 dataset)

Fig. 7 shows some results of a query of horse images with the returned ranking of the top 30 images. The image with the blue rectangle means false positive. We can see that our method is very robust in searching class-category images, but there are some false positive samples, for example, dogs with standing pose are very similar to those of horses.

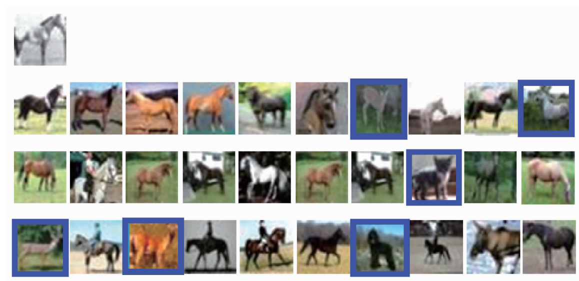

Fig. 7 Sample retrieval results on CIFAR-10 dataset

We also observe that the contour information in this dataset has a predominant effect on feature representation. The false positives with similar shapes can be ranked with high confidence. The color and texture information is not so important in such low resolution datasets that these features have no effect.

4.3 Results on LabelMe22K dataset

The number of bits vs. precision curves are shown in Fig. 8. We can see that our method can obtain a perfect precision of about 0.997 with only 32 bits, which is quite competitive with the other methods. The ITQ method is also very good, with only a small margin (less than 3%) to our methods. The SIKH method still performs the worst. The experimental results are consistent with the results on the CIFAR-10 dataset. We also observe that our method increases the performance with a nearly linear speed which is faster than other methods. Fig. 9 shows the precision-recall curves. We can see that our method is superior to all of the other four methods with a large margin. It is worth mentioning that the precision of our method is 14% higher than that of the ITQ method when the recall rate is equal to 0.6. We also observe that the changing speed of the precision curves is much slower than

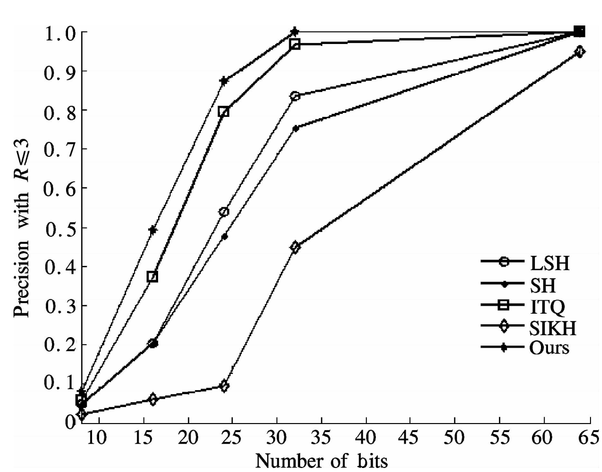

Fig. 8 Number of bits vs. precision curves (LabelMe22K dataset)

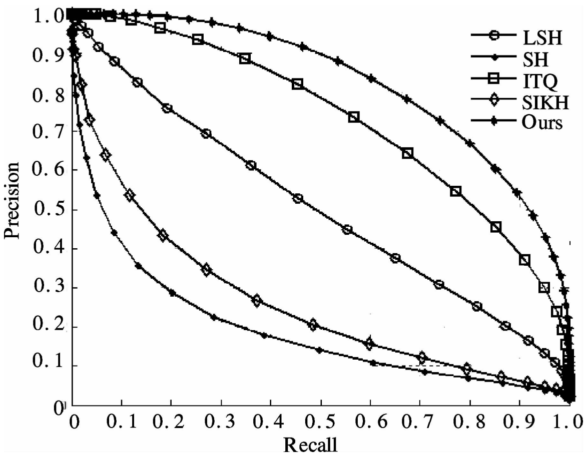

Fig. 9 Precision-recall curves(LabelMe22K dataset)

those of the LSH, SH and SIKH methods. The ITQ method also has a very similar changing speed compared with ours.

Some query results are shown in Fig. 10 with a street view outdoor scene. This scene contains a variety of complex backgrounds such as buildings, cars, trees, etc. The retrieval results are also very good when we see the top 30 ranking images. The images with the blue rectangle mean false positives. We can see from Fig. 10 that the spatial envelopes of images have a predominant effect on the feature due to the low resolution of the image sets. It also suffers from the same problem as in the CIFAR-10 dataset, and many informative features cannot play a role in such a low resolution dataset. One resolution to this problem needs a moderate resolution image set. Furthermore, the training time of our method is little shorter than those of the other methods. In the testing stage, all the methods have the same running time due to the same coding formula.

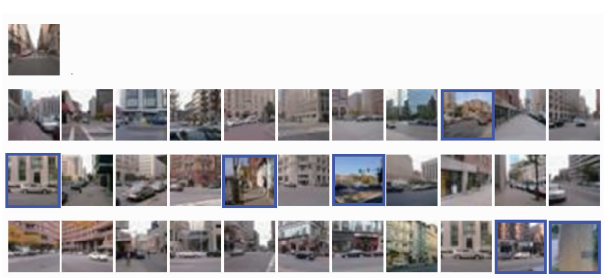

Fig. 10 Sample retrieval results on LabelMe22K dataset

5 Conclusion

This paper shows that learning discriminative binary code has an impact on class-category image retrieval in a supervised learning framework. We use LDA criterion to sequentially learn each bit with the asymmetric gentle boosting algorithm. The compact code learned from our method can obtain improved retrieval speed with reduced storage cost. Experimental results on the CIFAR-10 and LabelMe22K datasets demonstrate that our proposed method is superior to many state-of-the-art algorithms such as the ITQ, SH methods, etc.

Furthermore, there is still a lot of work to do in the future, such as testing the method in different image resolutions, and applying it in more complicated datasets with more categories.

References

[1] Silpa-Anan C, Hartley R. Optimised KD-trees for fast image descriptor matching [C]//*IEEE Conference on Computer Vision and Pattern Recognition*. Anchorage, AK, USA, 2008: 1 − 8.

[2] Uhlmann J K. Satisfying general proximity/similarity queries with metric trees[J]. *Information Processing Letters*, 1991, **40**(4): 175 − 179.

[3] Gionis A, Indyk P, Motwani R. Similarity search in high dimensions via hashing[C]//*Proceedings of the 25th International Conference on Very Large Data Bases*. Edinburgh, Scotland, 1999: 518 − 529.

[4] Lv Q, Josephson W, Wang Z, et al. Multi-probe LSH: efficient indexing for high-dimensional similarity search [C]//*Proceedings of the 33rd International Conference on Very Large Data Bases*. Vienna, Austria, 2007: 950 − 961.

[5] Kulis B, Grauman K. Kernelized locality-sensitive hashing[J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2012, **34**(6): 1092 − 1104.

[6] Gong Y, Lazebnik S. Iterative quantization: a procrustean approach to learning binary codes[C]//*IEEE Conference on Computer Vision and Pattern Recognition*. Providence, RI, USA, 2011: 817 − 824.

[7] Weiss Y, Torralba A, Fergus R. Spectral hashing[C]//*Advances in Neural Information Processing Systems*. British Columbia, Canada, 2008: 1 − 8.

[8] Lazebnik S, Raginsky M. Locality-sensitive binary codes from shift-invariant kernels [C]//*Advances in Neural Information Processing Systems*. British Columbia, Canada, 2009: 1509 − 1517.

[9] Salakhutdinov R, Hinton G. Semantic hashing[J]. *International Journal of Approximate Reasoning*, 2009, **50**(7): 969 − 978.

[10] Kulis B, Darrell T. Learning to hash with binary reconstructive embeddings[C]//*Advances in Neural Information Processing Systems*. Vancouver, Canada, 2009: 1042 − 1050.

[11] Norouzi M, Fleet D. Minimal loss hashing for compact binary codes[C]//*Proceedings of the 28th International Conference on Machine Learning*. Bellevue, WA, USA, 2011: 353 − 360.

[12] Wang J, Kumar S, Chang S F. Semi-supervised hashing for large-scale search[J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2012, **34**(12): 2393 − 2406.

[13] Liu W, Wang J, Kumar S, et al. Hashing with graphs [C]//*Proceedings of the 28th International Conference on Machine Learning*. Bellevue, WA, USA, 2011: 1 − 8.

[14] Strecha C, Bronstein, A M, Bronstein M M, et al. LDAHash: improved matching with smaller descriptors

[J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2012, **34**(1): 66 − 78.
[15] Andoni A, Indyk P. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions[C]// *47th Annual IEEE Symposium on Foundations of Computer Science*. Berkeley, CA, USA, 2006: 459 − 468.
[16] Datar M, Immorlica N, Indyk P, et al. Locality-sensitive hashing scheme based on p-stable distributions[C]//*Proceedings of the 20th Annual Symposium on Computational Geometry*. New York, USA, 2004: 253 − 262.
[17] Shakhnarovich G, Viola P, Darrell T. Fast pose estimation with parameter-sensitive hashing[C]//*Proceedings of the 9th IEEE International Conference on Computer Vision*. Nice, France, 2003: 750 − 757.
[18] Shen J F, Sun C Y, Yang W K, et al. A novel distribution-based feature for rapid object detection[J]. *Neurocomputing*, 2011, **74**(17): 2767 − 2779.
[19] Chang H, Ai H Z, Li Y, et al. High-performance rotation invariant multiview face detection[J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2007, **29**(4): 671 − 686.

基于多源异质特征的紧致二进制编码学习

左　欣[1,2]　罗立民[1]　沈继锋[3]　于化龙[2]

([1] 东南大学影像科学与技术实验室，南京 210096)
([2] 江苏科技大学计算机科学与工程学院，镇江 212003)
([3] 江苏大学电气信息工程学院，镇江 212013)

摘要：为了提高图像检索系统的精度，提出了一种基于多种异质特征的新颖哈希函数学习方法. 该方法首先利用特征空间中相似样本与非相似样本分布的不平衡性来提升每个弱分类器的性能，从而建立非对称的 Boosting 框架；然后将一种基于异质特征子空间学习的线性判别弱分类器融入该框架下，并利用每轮算法中的误判样本的信息来依次学习紧致且平衡的哈希编码. 该方法能有效地融合具有互补功能的不同模态的信息，实现了检索系统的性能提升. 在 2 个公开数据集上的实验结果表明该方法优于其他算法，由此看出增加多源异质特征和利用不平衡性学习紧致哈希编码都可以大大提高图像检索的精度.

关键词：哈希编码；线性判别分析；非对称 boosting；异质特征
中图分类号：TP391.3