Deep LDA Hashing

Abstract

The conventional supervised hashing methods based on classification do not entirely meet the requirements of hashing technique, but *Linear Discriminant Analysis* (LDA) does. In this paper, we propose to perform a revised LDA objective over deep networks to learn efficient hashing codes in a truly end-to-end fashion. However, the complicated eigenvalue decomposition within each mini-batch in every epoch has to be faced with when simply optimizing the deep network w.r.t. the LDA objective. In this work, the revised LDA objective is transformed into a simple least square problem, which naturally overcomes the intractable problem-s and can be easily solved by the off-the-shelf optimizer. Such deep extension can also overcome the weakness of LDA Hashing in the limited linear projection and feature learning. Amounts of experiments are conducted on three benchmark datasets. The proposed Deep LDA Hashing shows nearly 70 points improvement over the conventional one on the CIFAR-10 dataset. It also beats several state-of-the-art methods on various metrics.

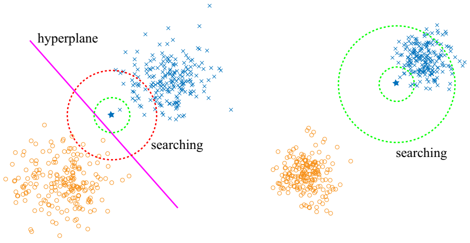
1 Introduction

In the era of big data, the amount of data is growing rapidly along with the popularization of multiple kinds of digital recording devices, especially the visual, audio, and text data. To satisfy the required huge storage space, and organization, learning capacity in dealing with such big data, hashing technique has been widely employed to learn effective binary representation in multiple tasks [10, 37, 21], especially the image retrieval task [22]. This is because the binary representations take advantage of the coding property of existing digital recording manner, i.e., we can take one Byte to encode one specific image of several MBytes.

Hashing is mainly developed to map the image or document into short binary code sequence while p-reserving the similarity structure among the original data. Most works focus on the data-independent methods in the earlier studies, as such methods can enjoy low computation complexity via random projection. The most representative work is *Locality Sensitive Hashing* (LSH) [1]. Several extensions based on LSH are also proposed, e.g., kernel LSH [15] and p-norm LSH [5]. However, the obtained hashing codes suffer from the inefficiency of random projection, which therefore requires longer codes to achieve better performance [23].

To generate more compact codes, more attention are turned on to the data-dependent methods. D-ifferent from the random projection, these methods propose to learn effective hashing functions via exploring the intrinsic data structure or depending on the labels. Hence there are two major categories among them, one is unsupervised hashing, and the other is supervised hashing. Specifically, the former focuses on how to maintain the data structure and make the generated codes more informative. For example, Iterative Quantization (ITQ) [8] adopts PCA projection to maximize the variance of hash functions, while Isotropic Hashing [11] targets to make the projected dimensions into equal variance. The reconstruction strategy is also considered to generate effective low-dimensional representation [14, 3]. In addition, the Laplacian eigenmap is employed to learn the intrinsic manifold structure and nonlinearly embed the data into proper codes, e.g., Spectral Hashing (SH) [32], spectral rotation [22].

Preprint. Work in progress.



(a) Classification (b) LDA

Figure 1: An illustration of the classification and LDA projection. The projected data by LDA have greater distance between class centers than the ones by classification. Obviously, the nearest neighbors are not in the same category in (a).

Unlike the unsupervised methods, supervised hashing directly resorts to the labels that indicate similar or dissimilar items. Hence, it is natural to formulate the hashing problem as a classification task, where the projected codes of the same category are classified as the similar items. CCA-ITQ [8] extends the ITQ into the supervised scenario by maximizing the correlation between features and corresponding labels, then the learned projection matrix is employed for generating codes. *Supervised Discrete Hashing* (SDH) [28] and *Fast SDH* (FSDH) [12] directly generate the hash codes within the linear multi-class classification framework, which shows considerable performance compared with before. However, these methods suffer from the weakness of shallow model in nonlinear modeling. Recently, as the deep networks, especially *Convolution Neural Network* (CNN), show effectiveness in feature learning and nonlinear modeling, more attention are paid to the deep supervised hashing. To be specific, the early works usually employ the siamese network (i.e., two stream networks but share the same parameters) [4] to learn if two items are similar or not [20, 38]. Considering that the pair-loss is the indirect supervision over networks, why not directly utilizing the label of class information? Yang et al.[33] formulate the deep hashing learning as a multi-class classification by minimizing the cross-entropy error. Li et al. [19] combine the pair-label and direct class label to supervise the hashing network, similar framework can also be found in [34].

Actually, if we rethink the meaning of the original hashing objective for similarity retrieval, we can find that the supervised classification fashion does not entirely meet the hashing requirements. Hashing hopes to retrieve similar points in the Hamming space, which means the similar items should be close to each other but away from the dissimilar ones. Although the nearby points of different categories are successfully classified by the learned hyperplane, they are still considered as the similar items according to the nearest neighbor searching. Fortunately, LDA exactly fits the hashing objective, which aims to minimize intra-class covariance and maximize inter-class covariance simultaneously[26]. A simple illustration of such projection can be found in Fig. 1. It is easy to find that LDA is more suitable for retrieving similar points in the hashing task compared with the classical classification objective. However, the existing *LDA Hashing* (LDAH) [29] relies on the linear projection over hand-craft features, which limits its capacity in learning effective hashing function.

In this paper, to overcome the weaknesses of LDAH, we propose to learn hashing function under a kind of revised LDA objective but based on the effective deep feature learning. That is, CNNs show noticeable effectiveness in learning semantic features, especially for the image messages [16]. Meanwhile, the efficient nonlinear projection of deep models also provide possibilities to encode the original features into proper distribution in the low dimensional space [27]. These two properties exactly remedy the defects of the original LDA hashing. However, simply performing the LDA objective over the networks is difficult to solve, as it has to be faced with the complicated eigenvalue decomposition for each mini-batch data in every epoch. In this work, the revised LDA objective is transformed into a simple least square problem, which can be easily solved by the off-the-shelf optimizer, such as stochastic gradient descent. To directly generate the hashing codes without additional binarization, a truly end-to-end hashing network is proposed by utilizing an adaptive binary activation function. The proposed method is evaluated on three large-scale benchmark datasets, and beats several state-of-the-art methods on various metrics.

2 LDA Hashing Revisited

LDA hashing targets to project the high dimensional data $x_i \in R^d$ into the r-bits ($d \gg r$) short binary codes, while jointly minimizing the intra-class covariance and maximizing the inter-class covariance [29]. To be specific, given the training dataset $\xi = \left\{ (x_i, c_i) \,|x_i \in R^d, i = 1, 2, ..., n \right\}$, where each data point x_i corresponds to one class label $c_i \in \{1, 2, ..., c\}$, and let $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ and

$\bar{x}_i = \frac{1}{n_i}\sum_{x_j \in \chi_i} x_j$ denote the global mean and specific-class mean (e.g., the i-class), respectively. Then the intra-class scatter matrix S_w, the inter-class scatter matrix S_b, and the total-class scatter matrix S_t can be written as

$$\left\{\begin{array}{l} S_w = \sum\limits_{i=1}^{c}\sum\limits_{x\in\chi_i}(x-\bar{x}_i)(x-\bar{x}_i)^T \\ S_b = \sum\limits_{i=1}^{c} n_i(\bar{x}_i-\bar{x})(\bar{x}_i-\bar{x})^T \\ S_t = \sum\limits_{i=1}^{n}(x_i-\bar{x})(\bar{x}_i-\bar{x})^T \end{array}\right. , \tag{1}$$

where $S_t = S_w + S_b$.

Up to this point, LDA hopes to learn an optimal W that makes the projected data points satisfy the above objective, i.e., minimizing the intra-class and maximizing the inter-class covariance. Hence, the classical LDA objective can be formulated as [9],

$$\max_W Tr\left(\left(W^T S_w W\right)^{-1} W^T S_b W\right). \tag{2}$$

Generalized eigenvalue decomposition can be used to efficiently solve this problem, i.e., $S_b w_k = \lambda_k S_w w_k$, where λ_k and w_k are the k-th largest generalized eigenvalue and corresponding eigenvector. Then, the projection matrix W can be constituted by the eigenvectors with maximal r eigenvalues. Hence, the projected low dimensional representation for data x_i can be obtained by

$$h_i = W^T \cdot x_i. \tag{3}$$

However, due to the required binary constraint, hashing is obviously different from traditional dimensionality reduction. Directly imposing the binary constraint into the LDA objective will make it become a NP-hard problem. A common relaxation strategy is performing the binarization based on threshold selection [29],

$$b_i = \mathrm{sign}(W^T \cdot x_i + t), \tag{4}$$

where sign is the sign function and t is the threshold.

Similar with the classical LDA method, LDA hashing also suffers from the so-called "small sample size " (SSS) problem [24]. That is, the original limited data points usually lie in high-dimensional space (e.g., 128-D in SIFT space, 4096-D in deep space), which results in sparse space when the data points are not enough for exploring the data structure as expected. Hence, the intra or inter-class covariance are usually not invertible when solving Eq. 2. The common strategy is adding a regularization term to the original LDA objective as follows [7],

$$\max_W Tr\left(\left(W^T(S_w+\mu I)W\right)^{-1} W^T S_b W\right), \tag{5}$$

where μ is the regularization parameter. Although the LDA objective exactly meets the requirements of hashing technique, the complicated feature distribution makes it difficult to learn effective low-dimensional representation by employing simple linear projection. Hence, LDA hashing has not shown noticeable performance before. Even so, how about performing nonlinear feature learning to make it more discriminative and adaptable to the LDA objective?

3 Deep LDA Hashing

To overcome the weakness of original linear LDAH, *Deep LDA Hashing* (DLDAH) actually does not learn the projection matrix over features but aims to directly learn efficient binary representations from raw image via multiple layers of nonlinear projection (i.e., CNN), which should satisfy the LDA objective. Hence, the LDA objective is directly performed over the final layers of CNN networks. Formally, let $X \in R^{d\times n}$ and $Y \in \{0,1\}^{n\times c}$ denote the final feature representation and corresponding one-hot labels, respectively. Then the output feature X of network f should be encouraged to shrink the intra-class covariance and enlarge the inter-class covariance. For better expression, define $A^t = \frac{1}{n}11^T$ and $A^w_{ij} = \left\{\begin{array}{ll} \frac{1}{n_{c_i}} & c_i = c_j \\ 0 & \text{otherwise} \end{array}\right.$, then the intra-class scatter matrix S_w and the total-class scatter matrix S_t can be written as

$$S_w = X\left(I - A^w\right)\left(I - A^w\right)^T X^T, \tag{6}$$

and

$$S_t = X\left(I - A^t\right)\left(I - A^t\right)^T X^T. \tag{7}$$

As $S_t = S_b + S_w$, then the inter-class scatter matrix S_b can be derived as follows,

$$\begin{aligned}
&S_b = S_t - S_w \\
&{=}X\left(I - A^t\right)\left(I - A^t\right)X^T - X\left(I - A^w\right)\left(I - A^w\right)X^T \\
&{=}X\left(I - A^t\right)X^T - X\left(I - A^w\right)X^T \\
&{=}X\left(A^w - A^t\right)X^T \\
&{=}XH\left(A^w - A^t\right)HX^T \\
&{=}XHA^wHX^T \\
&{=}XHY\left(Y^TY\right)^{-1}Y^THX^T
\end{aligned} \tag{8}$$

where $H = \left(I - A^t\right)$ is the centering matrix. Hence, the objective of DLDAH can be written as

$$\max_f Tr\left(S_t^{-1}S_b\right), \tag{9}$$

where f stands for the deep networks with multiple layers of nonlinear projection. Note that, as the total-class scatter matrix $S_t = S_b + S_w$, Eq.9 is equivalent to maximizing S_b and minimizing S_w [35]. Compared with Eq.2, the linear projection matrix W is **replaced** with multiple layers of nonlinear projection f, which means the proposed model has stronger ability in regularizing the learned features to the hashing objective. However, the matrix S_t may be not invertible due to the SSS problem, an additional regularization term is considered and Eq.9 becomes,

$$\max_f Tr\left(\left(S_t + \mu I\right)^{-1}S_b\right). \tag{10}$$

Although the revised objective of deep network currently meets the requirements of hashing technique, we have to be faced with two intractable problems when training the deep networks with Eq.10. On the one hand, due to the huge training data and amounts of trainable parameters, network optimization is iteratively performed with small batches of the database. However, such optimization strategy is not entirely suitable for the revised LDA objective, i.e., Eq.10. This is because all the scatter matrices are computed based on the whole training data, only a small batch of them can not exactly reflect the complex distribution of feature points. On the other hand, as introduced in Sec.2, generalized eigenvalue decomposition is usually adopted to solve the LDA objective, whose time complexity is $O\left(d^3\right)$. When performing such optimization within the batch-strategy of deep networks, the complexity becomes $O\left(d^3 \cdot m \cdot iter\right)$, where d is the feature dimension, m is the number of batches of each epoch and $iter$ is the number of epoches. It becomes impractical to perform eigenvalue decomposition over the deep features under such huge time complexity. Hence, these two intractable problems make it difficult to directly optimize Eq.10. Fortunately, according to Theorem 1, we can convert the original problem into a well-developed framework and solve the two problems simultaneously.

Theorem 1. *Minimizing the linear regression of least square is equivalent to maximizing the LDA objective in Eq.10*[1].

Hence, Eq.10 can be transformed in,

$$\min_{f,W,b}\left\|X^TW + 1b^T - \widetilde{Y}\right\|_F^2 + \mu\left\|W\right\|_F^2, \tag{11}$$

where $\widetilde{Y} = Y\left(Y^TY\right)^{-1/2}$, W and b are the newly introduced regression matrix and bias term, respectively. It is easy to find that the aforementioned problems are solved automatically. The least square error does not extremely suffer from the problem of optimization based on mini-batch, and the time complexity for the regression layer is the same as standard fully connected layers. Hence, it becomes feasible to optimize the deep network with the proposed LDA objective.

Although the previous works [35, 30, 18] have shown the equivalence between multivariate linear regression and classical LDA objective, i.e., Eq.2, such equivalence is obviously different from the conditions in Theorem 1 and cannot be applied here. On the one hand, the classical LDA objective

[1] The proof is available in the supplementary material.

has been revised for supervising deep models, where the linear projection matrix is replaced with multi-layers of nonlinear transformation. Such modification leads to the disparate least square objective, i.e., the newly introduced variable and different indicator matrix $\widetilde{Y}$. On the other hand, the previous equivalence [35] requires rank(S_b) + rank(S_w) = rank(S_t), which does not always hold for the high-dimensional and under-sampled data [36], i.e., the SSS condition. In contrast, our proof takes no assumption about the scatter matrix, therefore it is comfortable for deep supervision.

3.1 Controllable Projection

By comparing Eq.10 and Eq.11, we can find that the original one-hot label Y is changed into $\widetilde{Y}$ that considers the imbalance between different categories via $\left(Y^TY\right)^{-1/2}$. But how such variation affects the proposed LDA objective, and how about directly employing the original label Y without the modification? Hence, Eq.11 is further explored to answer these questions.

To simplify the analysis, the original label Y is adopted instead of $\widetilde{Y}$. Then, Eq.8 becomes

$$S_b{=}XHYY^THX^T. \tag{12}$$

Substituting Eq.12 into Eq.10,

$$\begin{aligned} & Tr\left((S_t+\mu I)^{-1}S_b\right) \\ & = Tr\left[(S_t+\mu I)^{-1/2}(S_t+\mu I)^{-1/2}XHYY^THX^T\right] \\ & = Tr\left[(S_t+\mu I)^{-1/2}XHY\cdot Y^THX^T(S_t+\mu I)^{-1/2}\right] \\ & = \left\|(S_t+\mu I)^{-1/2}XHY\right\|_F^2. \end{aligned} \tag{13}$$

The original ratio trace becomes the Frobenius norm of the projected centering feature matrix. It can be further written w.r.t. specific-classes as follows,

$$\begin{aligned} & \left\|(S_t+\mu I)^{-1/2}XHY\right\|_F^2 \\ & = \left\|(S_t+\mu I)^{-1/2}\left[n_1\left(\bar{x}_1-\bar{x}\right),...,n_c\left(\bar{x}_c-\bar{x}\right)\right]\right\|_F^2 . \\ & = \sum_{i=1}^{c} n_i^2\left\|(S_t+\mu I)^{-1/2}\bar{x}_i-(S_t+\mu I)^{-1/2}\bar{x}\right\|_2^2 \end{aligned} \tag{14}$$

Eq.14 means that the class mean and the global mean are normalized by the total-class scatter matrix, and the inter-class covariance is maximized by enlarging their Euclidean distance. Note that the label Y performs as a parameter to amplify the inter-class distance according to the number of items of each class, which prompts these class-centers to be as far as possible. When Y is replaced with $\widetilde{Y}$, Eq. 10 can be re-written as

$$\sum_{i=1}^{c} n_i\left\|(S_t+\mu I)^{-1/2}\bar{x}_i-(S_t+\mu I)^{-1/2}\bar{x}\right\|_2^2. \tag{15}$$

Compared with Eq.14, the parameter of n_i^2 becomes n_i in Eq.15. Then, the inter-class covariance becomes smaller than the original one. Hence, for simplicity and enlarging the inter-class distance, the label matrix Y is adopted instead of $\widetilde{Y}$ in Eq.11 in this paper.

3.2 End-to-end hashing network

Hashing technique requires the low-dimensional representation to be binary. As discussed in Sec.2, it becomes a NP-hard problem when solving the LDA objective with the binary constraint. The common strategy is to perform the binarization after projecting the original data into the low-dimensional space [29]. However, such separated two-stage methods may destroy the learned space and result in sub-optimal hashing codes [21]. Although the recent deep hashing methods propose to reduce the quantization loss when optimizing the networks, the activations of the hashing layer are still not binary. This is because the sign function have no gradient, and most works have to use the tanh function as the approximated binary activation within the network [38, 19].

Inspired by [21], a learnable parameter is introduced into the original tanh function, which can avoid the tiny gradient when faced with large input to some extent and project the input into the discrete domain of $\{1, -1\}$. Such activation function is named as *Adaptive Tanh* (ATanh) and written as

$$b_i = \tanh(\alpha x_i) + \nu \left\|\alpha^{-1}\right\|_2^2, \tag{16}$$

where ν is the regularization constant. By minimizing the regularization term of $\left\|\alpha^{-1}\right\|_2^2$, α can gradually increase so that the final activations approach the sign function and have the ability to generate binary codes. To directly perform the LDA objective over the binary representation and generate efficient hashing codes within DLDAH, the ATanh function is performed over the last layer but before the regression layer of LDA, whose activations constitute the hashing layer. Different from the previous unimodal hashing network, there is no extra quantization loss within such end-to-end hashing net, hence it shows stronger capacity in learning efficient codes. Meanwhile, as the LDA regression, i.e. Eq. 10, is directly performed over the binary values, the generated hashing codes within the same class enjoy smaller hamming distance while different classes take larger distance.

4 Experiments

4.1 Dataset

MNIST [17] consists of 70,000 28 × 28 grey-scale images associated with digits from 0 to 9. For the unsupervised methods, we randomly select 100 samples for each digit as the query images and all the rest ones constitute the training set. For the supervised methods, 500 images per class are selected as the training set. For all the methods, all the samples except the testing ones construct the retrieval gallery set.

CIFAR-10 [13] consists of 60,000 32×32 RGB images that are divided into 10 object categories and 6,000 images per category. Following the settings in [31], 100 image samples per class are randomly selected as the testing items and the training set consists of the remaining samples for the unsupervised methods. As for the supervised ones, 500 samples are uniformly selected from each category.

ImageNet [6] is a benchmark dataset built for *Large Scale Visual Recognition Challenge* (ILSVRC). In this paper, the ILSVRC2012 version is chosen for evaluation. It consists of about 1.3 million images that are labeled into 1,000 categories, where one image only corresponds to one label. For efficiency, we select the most frequent 100 categories. Following [2], 100 images per category form the training set, and the 50 validation images of each class constitute the testing set. While for the unsupervised methods, we employ all the images of about 130K for training. And all of the images excluding query sets are also treated as the retrieval gallery.

4.2 Evaluation

In order to effectively evaluate the proposed model, the representative data-independent and data-dependent hashing methods are considered. They are the unsupervised methods LSH, SH, and ITQ, the conventional supervised methods LDAH, SDH, and FSDH, and the state-of-the-art deep hashing methods DHN [38], DTSH [31], HashNet [2], and DSDH [19]. To comprehensively compare all these methods, both the global and local evaluation are considered, i.e., Hamming ranking and hash lookup. Specifically, *Mean Average Precision* (MAP) is employed for computing the quality of the whole retrieved sequence according to the Hamming distance to a query. While for the hash lookup, only the top retrieved samples are considered, i.e., the $\{Recall, F-measure\}$ score is calculated based on a Hamming ball of radius 2 to a query [23, 22].

4.3 Set up

As the proposed model is an extension of the classical LDA on deep models, it is necessary to compare with other deep hashing methods with the same basic network for fairness. Following [31, 19], we adopt the VGG-F model[2] pretrained on the ImageNet as the deep architecture, so are

[2] http://www.vlfeat.org/matconvnet/pretrained/

Table 1: The comparison results of different hashing methods in MAP on CIFAR-10 and ImageNet.

Dataset	CIFAR-10					ImageNet			
Code #bits	8	16	32	64	128	8	16	32	48
LSH	0.1374	0.1403	0.1693	0.1643	0.2062	0.0211	0.0260	0.0475	0.0815
SH	0.1920	0.1810	0.1700	0.1690	0.1696	0.0815	0.1118	0.1578	0.1869
ITQ	0.2308	0.2480	0.2540	0.2750	0.2892	0.1163	0.1883	0.2723	0.3152
LDAH	0.1677	0.1408	0.1243	0.1162	0.1120	0.0659	0.1122	0.1946	0.2540
SDH	0.4881	0.5516	0.5807	0.5925	0.6067	0.1840	0.3172	0.4088	0.4514
FSDH	N/A	0.5616	0.5616	0.5616	0.5616	N/A	N/A	N/A	N/A
DHN	0.5750	0.6688	0.7004	0.7027	0.7078	0.2388	0.3650	0.4475	0.4884
HashNet	0.5813	0.6733	0.7128	0.6809	0.7074	<u>0.2439</u>	<u>0.3819</u>	0.4756	0.5262
DTSH	0.6214	0.7094	0.7613	0.7743	0.7213	0.2118	0.3497	0.4433	0.5035
DSDH	<u>0.6774</u>	<u>0.7261</u>	<u>0.7623</u>	<u>0.7905</u>	<u>0.7502</u>	0.2313	0.3720	<u>0.4810</u>	<u>0.5326</u>
DLDAH	**0.7006**	**0.7500**	**0.7724**	**0.7964**	**0.8002**	**0.2930**	**0.3953**	**0.5208**	**0.5624**

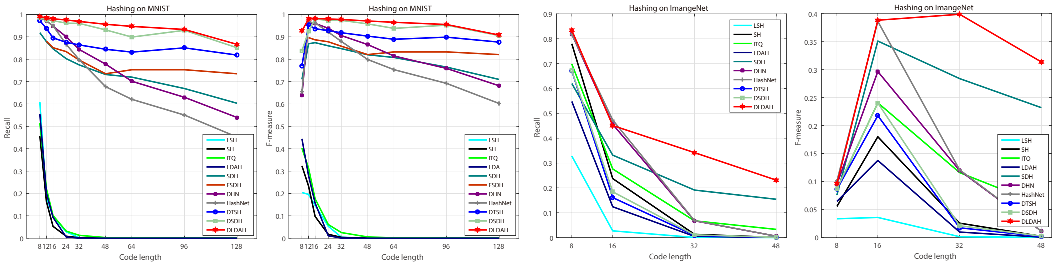


Figure 2: The hash lookup performance of different hashing methods on MNIST and ImageNet dataset with varying code lengths.

the other deep models. The provided results are obtained by running the source code provided by the authors. As the high-level CNN representations better reflect the semantic of images, they are also employed as the features for all the conventional shallow models. And the hype-parameter ν of ATanh follows the empirical value of 0.001 in [21], and μ is set to 0.0005.

4.4 Results and analysis

Results on MNIST. The hash lookup results in F-measure and Recall are shown in Fig. 2 and the proposed DLDAH outperforms all the other methods on all the code lengths. In both Recall and F-measure, the performance of all the methods decrease with the increasing code length, especially the unsupervised ones. Such phenomenon results from the more sparse hamming space [22]. However, although DLDAH also suffers from the same problem, it still remains stable and substantial superiority over the other ones.

Results on CIFAR-10. The Hamming ranking performance is shown in Table 1, and the proposed DLDAH shows the best results, especially better than the classical LDAH by 70 points. Besides, HashNet is an extension of the deep model of DHN, which employs a fixed sequence of parameterized tanh as the activation function for training the network step by step. As it can directly generate the binary codes after training the network, the codes become more efficient. By contrast, DLDAH adopts the learnable parameterized tanh instead of the fixed tanh in HashNet, which could adaptively learn more effective hashing function within the end-to-end network, hence, it shows much improvement than the other deep models.

Fig. 3 shows the hash lookup performance in terms of F-measure and recall. In Fig. 3, the conventional shallow models of SDH and FSDH outperform some deep hashing methods on both metrics, especially when the code is longer than 32 bits. The generated hashing codes by deep models are not as efficient as expected. This is because it is difficult to optimize deep hashing networks under complex code constraints, while it is easy for the shallow linear models. In contrast, due to the efficient LDA supervision, the proposed DLDAH still enjoys reliable codes when the codes become longer. On the other hand, all of these methods suffer from the similar conditions on MNIST but even worse, however, DLDAH remains stable when the code is longer than 64 bits. Besides, although

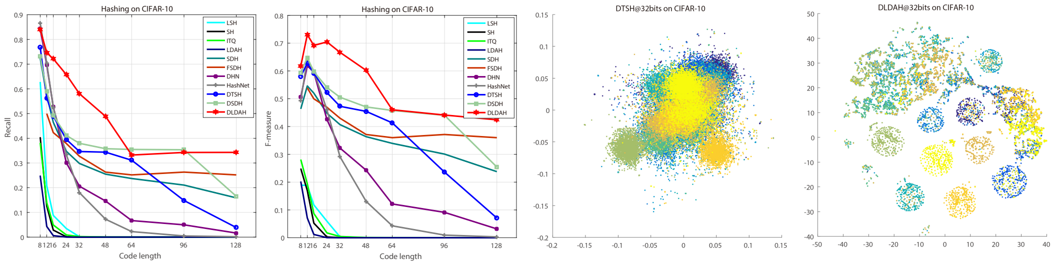


Figure 3: The hash lookup performance and t-SNE visualization on CIFAR-10 dataset with varying code lengths.

DLDAH performs a litter worse than DSDH at 64 bits in recall, it shows much better F-measure and outperforms DSDH on all the lengths.

Different from the quantitative evaluation above, we also visualize the learned hashing codes by embedding them into 2D space via t-SNE [25]. The state-of-the-art method of deep hashing based on classification, i.e., DTSH, is chosen for comparison. DTSH aims to make the query close to the positive items but away from the dissimilar ones. As shown in Fig. 3, most of the hashing codes learned by DTSH are actually mixed together, although some categories are successfully classified. Yet, the codes of DLDAH show distinctly discriminative structure for efficient similarity retrieval, which benefit from the efficient supervision of LDA. Such structure provides possibilities to achieve both high Precision and Recall, as shown above. Even so, some samples are still not successfully separated from the other ones by DLDAH in Fig. 3. This could be because the deep features of different categories have similar semantic, which will be considered for further study.

Results on ImageNet. As FSDH requires that the number of bits should be greater than the number of classes but there is 100 categories in the ImageNet dataset, its results are not shown in this part. Meanwhile, due to the time constraint, we only show the results with limited code lengths. As shown in Table 1, DLDAH outperforms all the other methods by 1-5 points in MAP. Further, we also show the hash lookup performance in Fig. 2. To maintain the performance when faced with much more categories in the ImageNet dataset, the hashing codes should be more efficient within specific Hamming ball. However, the hashing function are not effectively learned by most deep methods as expected, whose performances decrease rapidly with the increasing code length. Such phenomenon also comes from the inefficient supervision of classification, as shown in Fig. 3. However, DLDAH can separate these different categories in the Hamming space to some extent and learn discriminative hashing codes for efficient retrieval.

5 Conclusion

In this paper, we show that the conventional supervised hashing methods based on classification are not entirely suitable for hashing learning. To perform efficient similarity retrieval, a revised LDA objective is proposed to perform over the learned features by deep network, which encourages the similar samples to have small covariance while the dissimilar ones to have large covariance. And the deep hashing network with adaptive binary activation is optimized w.r.t the simple least square regression, which is proved to be equivalent to the original LDA objective. Such model provides a brand new viewpoint in analyzing, learning, and transforming the data structure, and the projected binary codes show better discriminative structure for hashing retrieval. Several deep hashing methods based on classification are defeated and considerable improvements are confirmed on three benchmark datasets.

References

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

[2] Z. Cao, M. Long, J. Wang, and P. S. Yu. Hashnet: Deep learning to hash by continuation. *arXiv preprint arXiv:1702.00758*, 2017.

[3] M. A. Carreira-Perpinán and R. Raziperchikolaei. Hashing with binary autoencoders. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 557–566, 2015.

[4] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 539–546. IEEE, 2005.

[5] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In *Proceedings of the twentieth annual symposium on Computational geometry*, pages 253–262. ACM, 2004.

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.

[7] J. H. Friedman. Regularized discriminant analysis. *Journal of the American statistical association*, 84(405):165–175, 1989.

[8] Y. Gong, S. Lazebnik, A. Gordo, and F. Perronnin. Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(12):2916–2929, 2013.

[9] Y. Jia, F. Nie, and C. Zhang. Trace ratio problem revisited. *IEEE Transactions on Neural Networks*, 20(4):729–735, 2009.

[10] M. Kafai, K. Eshghi, and B. Bhanu. Discrete cosine transform locality-sensitive hashes for face retrieval. *IEEE Transactions on multimedia*, 16(4):1090–1103, 2014.

[11] W. Kong and W.-J. Li. Isotropic hashing. In *Advances in Neural Information Processing Systems*, pages 1646–1654, 2012.

[12] G. Koutaki, K. Shirai, and M. Ambai. Fast supervised discrete hashing and its analysis. *arXiv preprint arXiv:1611.10017*, 2016.

[13] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. 2009.

[14] B. Kulis and T. Darrell. Learning to hash with binary reconstructive embeddings. In *Advances in neural information processing systems*, pages 1042–1050, 2009.

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

[16] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *nature*, 521(7553):436, 2015.

[17] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[18] K. Lee and J. Kim. On the equivalence of linear discriminant analysis and least squares. In *AAAI*, pages 2736–2742, 2015.

[19] Q. Li, Z. Sun, R. He, and T. Tan. Deep supervised discrete hashing. In *Advances in Neural Information Processing Systems*, pages 2479–2488, 2017.

[20] W.-J. Li, S. Wang, and W.-C. Kang. Feature learning based deep supervised hashing with pairwise labels. *arXiv preprint arXiv:1511.03855*, 2015.

[21] X. Li, D. Hu, and F. Nie. Deep binary reconstruction for cross-modal hashing. In *Proceedings of the 2017 ACM on Multimedia Conference*, pages 1398–1406. ACM, 2017.

[22] X. Li, D. Hu, and F. Nie. Large graph hashing with spectral rotation. 2017.

[23] W. Liu, C. Mu, S. Kumar, and S.-F. Chang. Discrete graph hashing. In *Advances in Neural Information Processing Systems*, pages 3419–3427, 2014.

[24] J. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos. Regularization studies of linear discriminant analysis in small sample size scenarios with application to face recognition. *Pattern Recognition Letters*, 26(2):181–191, 2005.

[25] L. v. d. Maaten and G. Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[26] S. Mika, G. Ratsch, J. Weston, B. Scholkopf, and K.-R. Mullers. Fisher discriminant analysis with kernels. In *Neural networks for signal processing IX, 1999. Proceedings of the 1999 IEEE signal processing society workshop.*, pages 41–48. Ieee, 1999.

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

[28] F. Shen, C. Shen, W. Liu, and H. T. Shen. Supervised discrete hashing. In *CVPR*, volume 2, page 5, 2015.

[29] C. Strecha, A. Bronstein, M. Bronstein, and P. Fua. Ldahash: Improved matching with smaller descriptors. *IEEE transactions on pattern analysis and machine intelligence*, 34(1):66–78, 2012.

[30] L. Sun, B. Ceran, and J. Ye. A scalable two-stage approach for a class of dimensionality reduction techniques. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 313–322. ACM, 2010.

[31] X. Wang, Y. Shi, and K. M. Kitani. Deep supervised hashing with triplet labels. In *Asian Conference on Computer Vision*, pages 70–84. Springer, 2016.

[32] Y. Weiss, A. Torralba, and R. Fergus. Spectral hashing. In *Advances in neural information processing systems*, pages 1753–1760, 2009.

[33] H.-F. Yang, K. Lin, and C.-S. Chen. Supervised learning of semantics-preserving hash via a deep convolutional neural networks. *IEEE transactions on pattern analysis and machine intelligence*, 40(2):437–451, 2018.

[34] T. Yao, F. Long, T. Mei, and Y. Rui. Deep semantic-preserving and ranking-based hashing for image retrieval. In *IJCAI*, pages 3931–3937, 2016.

[35] J. Ye. Least squares linear discriminant analysis. In *Proceedings of the 24th international conference on Machine learning*, pages 1087–1093. ACM, 2007.

[36] J. Ye and T. Xiong. Computational and theoretical analysis of null space and orthogonal linear discriminant analysis. *Journal of Machine Learning Research*, 7(Jul):1183–1204, 2006.

[37] R. Zhang, L. Lin, R. Zhang, W. Zuo, and L. Zhang. Bit-scalable deep hashing with regularized similarity learning for image retrieval and person re-identification. *IEEE Transactions on Image Processing*, 24(12):4766–4779, 2015.

[38] H. Zhu, M. Long, J. Wang, and Y. Cao. Deep hashing network for efficient similarity retrieval. In *AAAI*, pages 2415–2421, 2016.