Binary Codes Embedding for Fast Image Tagging with Incomplete Labels

Qifan Wang*, Bin Shen*, Shumiao Wang, Liang Li, and Luo Si

Department of Computer Science
Purdue University
West Lafayette, IN 47907-2107, USA
{wang868,bshen,wang845,li900,lsi}@purdue.edu

Abstract. Tags have been popularly utilized for better annotating, organizing and searching for desirable images. Image tagging is the problem of automatically assigning tags to images. One major challenge for image tagging is that the existing/training labels associated with image examples might be incomplete and noisy. Valuable prior work has focused on improving the accuracy of the assigned tags, but very limited work tackles the efficiency issue in image tagging, which is a critical problem in many large scale real world applications. This paper proposes a novel Binary Codes Embedding approach for Fast Image Tagging (BCE-FIT) with incomplete labels. In particular, we construct compact binary codes for both image examples and tags such that the observed tags are consistent with the constructed binary codes. We then formulate the problem of learning binary codes as a discrete optimization problem. An efficient iterative method is developed to solve the relaxation problem, followed by a novel binarization method based on orthogonal transformation to obtain the binary codes from the relaxed solution. Experimental results on two large scale datasets demonstrate that the proposed approach can achieve similar accuracy with state-of-the-art methods while using much less time, which is important for large scale applications.

Keywords: Image Tagging, Binary Codes, Hashing.

1 Introduction

The purpose of image tagging, assigning tags or keywords to images, is to benefit people for managing, organizing and searching desired images from various resources. For example, Flickr has more than 2 billion images with millions of newly uploaded photos per day. Users can better categorize or search desired images based on the tags associated with them. Due to the rapid growth of the Internet, a huge amount of images have been generated and users can only manually tag a very small portion of the images. Therefore, it is a challenging task to automatically assigning accurate tags to images for large scale data.

[*] The two authors contributed equally to this work.

D. Fleet et al. (Eds.): ECCV 2014, Part II, LNCS 8690, pp. 425–439, 2014.

Numerous research have been conducted on improving the accuracy of image tagging, such as automatic image annotation techniques [28,31] and multi-label learning [3,10]. Although these methods generate promising results of effectively assigning tags to image examples, they usually require a large set of training images with clean and complete tags/labels. But for many Web image applications, the annotated tags are incomplete and noisy, making it difficult to directly apply these methods for image tagging. Several tag completion [2,14,30] methods have been recently proposed to deal with incomplete and noisy tags, which achieve better results in terms of tag predicting accuracy by modeling global tag consistency. However, most existing methods only focus on the effectiveness without paying much attention to efficiency. In real world applications, the data size grows explosively and there are often a large number of possible tags and thus it is a practical and important research problem to design efficient methods for large scale image tagging.

This paper proposes a novel Binary Codes Embedding approach for Fast Image Tagging (BCE-FIT) by designing compact binary codes for both image examples and tags. In particular, each image example is represented by a C-bit binary code and each tag is also represented using a C-bit binary code. Our key ideas of constructing the binary codes are that *(1) if a tag is associated to an image, then the Hamming distance between their corresponding binary codes should be small; (2) two similar images should have similar codes; (3) the codes of two semantically similar tags should also be similar.* We then formulate the problem of learning binary codes as a discrete optimization problem by simultaneously ensuring the observed tags to be consistent with the constructed binary codes and preserving the similarities between image examples and tags. An iterative optimization method together with a novel binarization method is proposed to obtain the optimal binary codes. In tag predicting process, we calculate the Hamming distances between the code of a query image and the codes of all possible tags, and choose those tags within small Hamming distance to the query image. The Hamming distances between the binary codes of images and tags can be efficiently calculated using the bitwise XOR operation. In this way, assigning tags to images can be efficiently conducted.

We summarize the contributions in this work as follows: (1) To our best knowledge, we propose the first research work to learn compact binary codes for both images and tags in order to efficiently assigning tags to image examples. (2) We propose a learning framework to obtain the optimal binary codes and develop an efficient coordinate descent method as the optimization procedure. (3) We prove the orthogonal invariant property of the optimal relaxed solution and learn an orthogonal matrix to further improve the code performance.

2 Related Work

2.1 Image Tagging

Image tagging can be viewed as a multi-label learning problem where each image is associated with multiple tags. Numerous work have been proposed on multi-label learning for automatic image annotation and classification by exploiting the dependence among tags [1,3,10,28]. Desai *et al.* [6] propose a discriminative structured prediction model for multi-label object recognition. Hariharan *et al.* [10] introduce a max-margin framework for large scale multi-label classification. In [3], Chen *et al.* propose an efficient multi-label classification method using hypergraph regularization. Bao *et al.* [1] formulate a scalable multi-label propagation framework for image annotation. Liu et al. [17] propose a constrained nonnegative matrix factorization method for multi-label learning.

Besides the multi-label learning methods, several machine learning approaches have been proposed for image tagging, including tag propagation [9,19], distance metric learning [12] and tag recommendation [23]. Li *et al.* [12] propose a neighbor voting algorithm for social tagging which accurately and efficiently learns tag relevance by accumulating votes from visual neighbors. A tag propagation (TagProp) method has been proposed in [9] which propagates tag information from the labeled examples to the unlabeled examples via a weighted nearest neighbor graph. Makadia *et al.* [19] propose a widely-used annotation baseline denoted as JEC, which is a straightforward but sophisticated greedy algorithm propagating labels from nearest visual neighbors to the target image. Zhou *et al.* [32] develop a hybrid probabilistic model for unified collaborative and content based image tagging.

Image tag completion [2,14,24,30] methods have been recently proposed for image tagging task by recovering the missing entries in the tag matrix. Cabral *et al.* [2] propose two convex algorithms for matrix completion based on a rank minimization criterion. Wu *et al.* [30] introduce a direct tag matrix completion algorithm by ensuring the completed tag matrix to be consistent with both the observed tags and the visual similarity. Lin *et al.* [14] propose a image-specific and tag-specific linear sparse reconstruction model for automatic image tag completion. Although existing image tagging methods generate promising results, very limited prior research addresses the efficiency problem, which is a practical and critical issue in many large scale real world applications.

2.2 Learning Binary Codes

Extensive research on learning binary codes for fast similarity search [5,8,25,26] have been proposed in recent years. Locality Sensitive Hashing (LSH) [5] method utilizes random linear projections to map data examples from a high-dimensional Euclidean space to a low-dimensional one. The work in [21] uses stacked Restricted Boltzman Machine (RBM) to generate compact binary hashing codes for fast similarity search of documents. The PCA Hashing (PCAH) [13] method projects each example to the top principal components of the training set, and then binarizes the coefficients by setting a bit to 1 when its value is larger than the median value seen for the training set, and -1 otherwise.

Recently, Spectral Hashing (SH) [29] is proposed to learn compact binary codes that preserve the similarity between data examples by balancing the binary codes. The work in [15] proposes a graph-based hashing method to automatically discover the neighborhood structure inherent in the data to learn appropriate compact codes. A Canonical Correlation Analysis with Iterative Quantization (CCA-ITQ) method has been proposed in [7,8] which treats the image features and tags as two different views. The hashing function is then learned by extracting a common space from these two views. More recently, a bit selection method [16] has been proposed to select the most informative hashing bits from a pool of candidate bits generated from different hashing methods.

Existing hashing methods focus on constructing binary codes on images for fast similarity search and can not be directly applied for fast assigning tags to images. The reason is that image tagging requires to design compact binary codes for both image examples and tags. Therefore, different from prior work, we propose a binary codes embedding approach for fast image tagging which learns binary codes for both image examples and tags simultaneously.

3 Binary Codes Embedding for Fast Image Tagging

3.1 Problem Setting and Overview

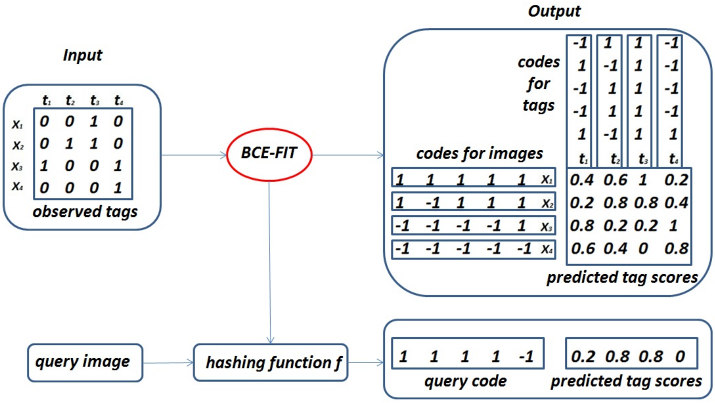
We first introduce the problem of BCE-FIT. Assume there are total n training images in the dataset, denoted as: $x_i, i \in \{1, 2, \ldots, n\}$, where x_i is the d-dimensional feature of the *i-th* image. There are total m possible tags denoted as: $t_j, j \in \{1, 2, \ldots, m\}$. Denote the observed tag matrix as: $T \in \{0, 1\}^{n \times m}$, where a label $T_{ij} = 1$ means the *j-th* tag is assigned to the *i-th* image, and $T_{ij} = 0$ means a missing tag or the *i-th* image is not associated with the *j-th* tag. Note that the *i-th* row of T is the tag vector associated with image x_i. In our approach, the training tags could be noisy and incomplete, which is the case in real world applications. The main purpose of BCE-FIT is to obtain optimal binary codes $y_i \in \{-1, 1\}^{C \times 1}, i \in \{1, 2, \ldots, n\}$ for the training images and $z_j \in \{-1, 1\}^{C \times 1}, j \in \{1, 2, \ldots, m\}$ for all possible tags, where C is the code length. We also want to learn a hashing function $f : \boldsymbol{R}^d \rightarrow \{-1, 1\}^C$, which maps each image x_i to its binary code y_i (i.e., $y_i = f(x_i)$).

The proposed BCE-FIT approach is a general learning framework and we first describe the problem formulation of how to construct the objective function. Then we represent the optimization method to obtain the optimal binary codes and the hashing function. Fig.1 shows an example of the proposed approach.

3.2 Problem Formulation

The goal of image tagging is to automatically assign tags to both training images and query images. Three main ingredients of constructing the compact binary codes are: (1) if a tag t_j is assigned to an image x_i, then their corresponding binary codes z_j and y_i should be similar; (2) visually similar images x_i and x_j should have similar codes y_i and y_j; and (3) semantically similar tags, e.g. 'human' vs. 'people', t_i and t_j should also have similar codes z_i and z_j. The similarity between two binary codes can be measured based on their normalized Hamming distance as follows:

$$s(y_i, z_j) = 1 - \frac{1}{C} dist_{Ham}(y_i, z_j) = \frac{1}{2} + \frac{y_i^T z_j}{2C} \tag{1}$$



	t_1	t_2	t_3	t_4
x_1	0	0	1	0
x_2	0	1	1	0
x_3	1	0	0	1
x_4	0	0	0	1

t_1	t_2	t_3	t_4
-1	1	1	-1
1	-1	1	-1
-1	1	1	-1
-1	1	1	-1
1	-1	1	1

						t_1	t_2	t_3	t_4
1	1	1	1	1	x_1	0.4	0.6	1	0.2
1	-1	1	1	1	x_2	0.2	0.8	0.8	0.4
-1	-1	-1	-1	1	x_3	0.8	0.2	0.2	1
-1	-1	-1	-1	-1	x_4	0.6	0.4	0	0.8

query code					predicted tag scores			
1	1	1	1	-1	0.2	0.8	0.8	0

Fig. 1. An example of the proposed BCE-FIT. In this example, there are 4 training images ($n = 4$) with 4 possible tags ($m = 4$). 5 bits are used to represent the binary codes ($C = 5$). The predicted tag score is the similarity between the binary code of an image and a tag, which is calculated based on the normalized Hamming distance in Eqn.1. For the query image in this example, we will assign tags t_2 and t_3 to the query image since the corresponding tag scores are relatively high (0.8).

where $dist_{Ham}$ is the Hamming distance between two binary codes, which is just the number of bits that they differ. It can be seen from Eqn.1 that the smaller the Hamming distance is, the more similar their binary codes become. Note that the similarity between two binary codes is a real value between 0 and 1.

The first key problem in designing binary codes is to ensure the consistency between the observed tags and the constructed binary codes. Specifically, we propose to minimize the squared loss of the observed tags and the similarity estimated from the binary codes, which is a commonly used loss function in many machine learning applications.

$$\sum_{i=1}^{n}\sum_{j=1}^{m}(T_{ij} - s(y_i, z_j))^2 \tag{2}$$

As discussed before, $T_{ij} = 0$ can be interpreted in two ways that tag T_{ij} is missing or the *i-th* image is not related to the *j-th* tag, which indicates that $T_{ij} = 1$ contains more useful information than a tag with value 0. Therefore, an importance matrix $I \in \boldsymbol{R}^{n\times m}$ is introduced to denote the confidence of how we trust tag information in tag matrix T. We set I_{ij} to a higher value when $T_{ij} = 1$ than $T_{ij} = 0$ as follows:

$$I_{ij} = \begin{cases} a, & if\ T_{ij} = 1 \\ b, & if\ T_{ij} = 0 \end{cases} \tag{3}$$

where a and b are parameters satisfying $a > b > 0$.[1] Then the square loss term becomes:

$$\sum_{i=1}^{n}\sum_{j=1}^{m} I_{ij}(T_{ij} - s(y_i, z_j))^2 \tag{4}$$

Substituting Eqn.1 into Eqn.4 we have:

$$\sum_{i=1}^{n}\sum_{j=1}^{m} I_{ij}(T_{ij} - \frac{1}{2} - \frac{y_i^T z_j}{2C})^2 \tag{5}$$

The second key problem in designing binary codes is similarity preserving, which indicates that visually similar images should be mapped to similar binary codes within a short Hamming distance. The pairwise visual similarity, S_{ij}, between two images x_i and x_j can be pre-calculated as:

$$S_{ij} = e^{-\frac{\|x_i - x_j\|^2}{\sigma^2}} \tag{6}$$

where σ^2 is the bandwidth parameter. Note that we use the Gaussian function/kernel to calculate the similarity in this work due to its popularity in many hashing methods [29,27], but other similarity criteria may also be used, such as cosine similarity or inner product similarity. To measure the similarity between images represented by the binary codes, one natural way is to minimize the follow quantity:

$$\sum_{i,j=1}^{n} (s(y_i, y_j) - S_{ij})^2 \tag{7}$$

The third criteria in designing binary codes is to ensure that semantically similar tags have similar codes. For example, we wish that the binary hashing codes for tags 'car' and 'automobile' be as close as possible since these two tags represent similar semantic meaning. In the extreme case, if two tags t_i and t_j appear in exactly the same set of images, i.e. the column i and j of tag matrix are identical, their binary codes should also be identical. However, since the tag information might be incomplete, we only assume that semantically similar tags tend to appear in the same image. Therefore, in order to measure the semantical similarity between two tags t_i and t_j, we use the number of images that are commonly shared by both tags, which can be calculated as: $\frac{T_i^T T_j}{m}$. Here T_i is the *i-th* column of tag matrix T. Dividing m is to normalize this quantity from 0 to 1. Then the similarity preservation between tags represented by the binary codes can be measured as:

$$\sum_{i,j=1}^{m} (s(z_i, z_j) - \frac{T_i^T T_j}{m})^2 \tag{8}$$

[1] In our experiments, we set the importance parameters a=1 and b=0.01 consistently throughout all experiments.

The entire objective function of the proposed BCE-FIT approach consists of three components: the square loss of tag consistency term in Eqn.4 and two similarity preservation term given in Eqn.7 and 8 as follows:

$$\begin{aligned}\min_{y,z} \quad & \sum_{i=1}^{n}\sum_{j=1}^{m} I_{ij}(T_{ij} - s(y_i, z_j))^2 \\ & +\alpha \sum_{i,j=1}^{n} (s(y_i, y_j) - S_{ij})^2 + \beta \sum_{i,j=1}^{m} (s(z_i, z_j) - \frac{T_i^T T_j}{m})^2 \\ s.t. \quad & y_i, z_j \in \{-1, 1\}^{C\times 1}, \quad \sum_{i=1}^{n} y_i = 0 \quad \sum_{j=1}^{m} z_j = 0\end{aligned} \tag{9}$$

where α and β are trade-off parameters. The constraints $\sum_{i=1}^{n} y_i = 0$ and $\sum_{j=1}^{m} z_j = 0$ are the bit balance constraints, which are equivalent to maximizing the entropy of each bit of the binary codes to ensure each bit carrying as much information as possible.

3.3 Optimization Algorithm

Relaxation. Directly minimizing the objective function in Eqn.9 is intractable since it is a constrained discrete optimization problem which is NP-hard to solve [29]. Therefore, we propose to relax the balance constraints into soft penalty terms and then relaxing the space of solution to $[-1, 1]^{C\times 1}$. Then the relaxed objective function becomes:

$$\begin{aligned}\min_{\tilde{y},\tilde{z}} \quad & \sum_{i=1}^{n}\sum_{j=1}^{m} I_{ij}(T_{ij} - \frac{1}{2} - \frac{\tilde{y_i}^T \tilde{z_j}}{2C})^2 \\ & +\alpha \sum_{i,j=1}^{n} (\frac{1}{2} + \frac{\tilde{y_i}^T \tilde{y_j}}{2C} - S_{ij})^2 + \beta \sum_{i,j=1}^{m} (\frac{1}{2} + \frac{\tilde{z_i}^T \tilde{z_j}}{2C} - \frac{T_i^T T_j}{m})^2 \\ & +\gamma(\|\sum_{i=1}^{n} \tilde{y_i}\|^2 + \|\sum_{j=1}^{m} \tilde{z_j}\|^2) \\ s.t. \quad & \tilde{y_i}, \tilde{z_j} \in [-1, 1]^{C\times 1}\end{aligned} \tag{10}$$

where γ is a trade-off parameter. $\|\sum_{i=1}^{n} \tilde{y_i}\|^2$ and $\|\sum_{j=1}^{m} \tilde{z_j}\|^2$ are soft penalty terms converted from the bit balance constraints. However, even after the relaxation, the objective function is still non-convex with respect to $\tilde{y}$ and $\tilde{z}$ jointly, which makes it difficult to optimize. Fortunately, this relaxed problem is differentiable with respect to either one of the two sets of parameters when the other one is fixed, and therefore we propose to solve the problem by coordinate descent method. In particular, we alternatively update $\tilde{y}$ and $\tilde{z}$ while fixing the other set of parameters by doing the following two steps until convergence.

Step 1: Fix $\tilde{y}$, Optimize $\tilde{z}$. By taking the partial derivative of Eqn.10 with respect to $\tilde{z_j}$, we can obtain the gradient and LBFGS method is then applied for solving this optimization problem to obtain optimal $\tilde{z}$.

Step 2: Fix $\tilde{z}$, Optimize $\tilde{y}$. Similar to step 1, we use LBFGS method to solve for the optimal $\tilde{y}$ using the gradient of Eqn.10 with respect to $\tilde{y_i}$.

Due to the space limitation, we will provide the two gradients in supplemental material. We alternate the process of updating $\tilde{y}$ and $\tilde{z}$ for several iterations to find a locally optimal solution. In practice, we have found that a reasonable small number of iterations can achieve good performance.

Binarization. After obtaining the optimal real value solution $\tilde{y}$ and $\tilde{z}$ for the relax problem, we need to binarize them to obtain binary hashing codes y and z. A direct binarization method is to obtain binary codes y and z that are closest to $\tilde{y}$ and $\tilde{z}$. In particular, we seek to minimize the quantization error between the binary codes and the relaxed solution as follow:

$$\begin{aligned} &\min_{y,z} \sum_i \|y_i - \tilde{y_i}\|^2 + \sum_j \|z_j - \tilde{z_j}\|^2 \\ &s.t. \quad y_i, z_j \in \{-1,1\}^{C\times 1} \end{aligned} \tag{11}$$

which leads to the close form solution:

$$y_i = sgn(\tilde{y_i}),\ z_j = sgn(\tilde{z_j}) \tag{12}$$

where $sgn()$ is the signum function of a real value vector.

In this work, we propose a novel binarization method that improves the quantization error through an orthogonal transformation by making use of the structure of the relaxed solution. We first prove the following theorem.

Theorem 1. *Assume Q is a $C \times C$ orthogonal matrix, i.e., $Q^TQ = I$. If $\tilde{y}$ and $\tilde{z}$ are an optimal solution to the relaxed problem Eqn.10, then $Q\tilde{y}$ and $Q\tilde{z}$ are also an optimal solution.*

Proof. By substituting $Q\tilde{y}$ and $Q\tilde{z}$ into Eqn.10, we have $\sum_{i,j} I_{ij}(T_{ij} - \frac{1}{2} - \frac{(Q\tilde{y_i})^T Q\tilde{z_j}}{2C})^2 = \sum_{i,j} I_{ij}(T_{ij} - \frac{1}{2} - \frac{\tilde{y_i}^T\tilde{z_j}}{2C})^2$. Similarly, the value of the second and third terms will also not change. $\|\sum_i Q\tilde{y_i}\|^2 = \|Q\sum_i \tilde{y_i}\|^2 = \|\sum_i \tilde{y_i}\|^2$ and $\|\sum_j Q\tilde{z_j}\|^2 = \|Q\sum_j \tilde{z_j}\|^2 = \|\sum_j \tilde{z_j}\|^2$. We also have that $Q\tilde{y}, Q\tilde{z} \in [-1,1]^{C\times 1}$. Thus, the value of the objective function in Eqn.10 does not change by the orthogonal transformation.

Based on the above observation, we propose to binarize $\tilde{y}$ and $\tilde{z}$ by minimizing the quantization error between the binary hashing codes and the orthogonal transformation of the relaxed solution as follow:

$$\begin{aligned} &\min_{y,z,Q} \sum_i \|y_i - Q\tilde{y_i}\|^2 + \sum_j \|z_j - Q\tilde{z_j}\|^2 \\ &s.t. \quad y_i, z_j \in \{-1,1\}^{C\times 1}, \quad Q^TQ = I \end{aligned} \tag{13}$$

Note that the direct binarization method can be achieved by simply setting $Q = I$. The intuitive idea behind this method is that the orthogonal transformation not only preserves the optimality of the relaxed solution but also provides us more flexibility to achieve more effective hashing codes with low quantization error. Similar ideas have also been investigated in other applications such as [8] for applying orthogonal transformation for only images in similarity search. However, our new research not only applies orthogonal transformation for images but also for tags. The above optimization problem can be solved by minimizing Eqn.13 with respect to y, z and Q alternatively.

Fix Q, update y and z. The close form solution can be expressed as:

$$y_i = sgn(Q\tilde{y_i}), \;\; z_j = sgn(Q\tilde{z_j}) \tag{14}$$

Fix y and z, update Q. The objective function becomes:

$$\min_{Q^TQ=I} \sum_i \|y_i - Q\tilde{y_i}\|^2 + \sum_j \|z_j - Q\tilde{z_j}\|^2 \tag{15}$$

Let $Y = [y_1, y_2, \ldots, y_n]$ and $Z = [z_1, z_2, \ldots, z_m]$. Then the above objective function can be rewritten to:

$$\begin{aligned} &\min_{Q^TQ=I} \|Y - Q\tilde{Y}\|_F^2 + \|Z - Q\tilde{Z}\|_F^2 \\ &= \|Y\|_F^2 + \|\tilde{Y}\|_F^2 + \|Z\|_F^2 + \|\tilde{Z}\|_F^2 \\ &-trace((Y\tilde{Y}^T + Z\tilde{Z}^T)Q^T) \end{aligned} \tag{16}$$

which is equivalent to:

$$\max_{Q^TQ=I} trace((Y\tilde{Y}^T + Z\tilde{Z}^T)Q^T) \tag{17}$$

here $trace()$ is the matrix trace function and $\|\|_F$ is the matrix $Frobenius$ norm. In this case, the objective function is essentially a variant of classic Orthogonal Procrustes problem [22], which can be solved efficiently by singular value decomposition using the following theorem (we refer to [22] for the detailed proof).

Theorem 2. *Let $U\Lambda V^T$ be the singular value decomposition of $Y\tilde{Y}^T + Z\tilde{Z}^T$. Then $Q = UV^T$ minimizes the objective function in Eqn.15.*

We then perform the above two steps alternatively to obtain the optimal binary codes y and z. After obtaining the binary codes, we can assign tags to images by calculating the predicted tag score using Eqn.1 (see figure 1).

In order to deal with the out-of-example problem in image tagging, where we need to generate binary codes for query images. A linear hashing function is used to map the image examples into binary codes as:

$$y_i = f(x_i) = sgn(Hx_i) \tag{18}$$

Algorithm 1. Binary Codes Embedding for Fast Image Tagging (BCE-FIT)

```
Input: Images $X$, Observed tag matrix $T$
Output: Hashing codes $y$ and $z$, Hashing function $H$
Initialize $\tilde{y}$ and $Q$
repeat
    Update $\tilde{z}$ in Step 1.
    Update $\tilde{y}$ in Step 2.
until the solution converges
repeat
    Update $y$ and $z$ using Eqn.14
    Update $Q = UV^T$ according to Theorem 2.
until the solution converges
Obtain hashing function $H$ using Eqn.18.
```

where H is a $C \times d$ parameter matrix representing the hashing function. Then the optimal hashing function can be directly obtained by minimizing $\sum_i (\tilde{y}_i - Hx_i)^2 + \lambda \|H\|_F^2$, where λ is a weight parameter for the regularization term to avoid overfitting and $X = [x_1, x_2, \ldots, x_n]$ is the data feature matrix. The full learning algorithm is described in Algorithm 1.

3.4 Analysis

The optimization algorithm of (BCE-FIT) consists of two main loops. In the first loop, we iteratively optimize over $\tilde{z}$ and $\tilde{y}$ to obtain the optimal relaxed solution, where the time complexities for updating $\tilde{z}$ and $\tilde{y}$ are bounded by $O(nC^2 + nmC)$ and $O(nmC + nC)$ respectively. The second loop iteratively optimizes the binary hashing codes and the orthogonal transformation matrix, where the time complexities for updating y, z and Q are bounded by $O(nC^2 + mC^2 + C^3)$. Thus, the total time complexity of the learning algorithm is bounded by $O(nmC + nC + nC^2 + mC^2 + C^3)$, which scales linearly with n given $n \gg m > C$. For each query, the time for obtaining its binary code is constant $O(Cd)$.

4 Experiments

4.1 Datasets and Implementation

We conduct our experiments on two large scale datasets, $Flickr1m$ [11] and *NUS-WIDE* [4]. $Flickr1m$ is collected from Flicker images for image annotation and retrieval tasks. This benchmark contains 1 million image examples associated with more than $7k$ unique tags. A subset of $250k$ image examples with the most common $2k$ tags is used in our experiment by filtering out those images with less than 10 tags. 512-dimensional GIST descriptors [20] are used as image features. We randomly choose $240k$ image examples as training set and $10k$ for testing. *NUS-WIDE* [4] is created by NUS lab, which contains $270k$ images associated with $5k$ unique tags. We use the most common $2k$ tags in our experiment. We

Table 1. Performance of different algorithms with varying number of training tags on both datasets with 32 hashing bits

*Flickr*1*m*	AP@10					AP@20				
training tags	2	4	6	8	10	2	4	6	8	10
BCE-FIT	65.4	**68.9**	71.1	73.4	77.4	65.2	66.3	**70.4**	71.6	74.5
LSR[14]	**66.3**	68.6	**71.7**	**76.2**	**79.5**	**65.4**	**66.9**	69.3	**72.1**	**75.3**
TMC[30]	62.9	64.1	66.8	71.7	73.4	57.2	61.8	62.7	66.4	70.1
LM3L[10]	60.4	65.8	68.3	71.6	74.7	58.5	62.0	65.8	68.7	70.8
CCA-ITQ[7,8]	55.2	57.5	59.7	61.1	64.6	53.3	55.2	56.3	57.8	60.2
SH[29]	53.7	55.3	57.5	58.4	60.7	52.4	53.8	55.1	55.6	57.5

NUS-WIDE	AP@10					AP@20				
training tags	2	4	6	8	10	2	4	6	8	10
BCE-FIT	51.1	56.2	63.4	71.7	74.5	48.4	54.2	**61.8**	**70.2**	75.1
LSR[14]	**51.7**	**56.5**	**66.4**	**72.5**	**76.7**	**49.2**	**54.6**	59.4	67.5	**76.4**
TMC[30]	48.3	53.1	61.4	72.0	73.6	46.6	51.7	58.4	62.9	67.7
LM3L[10]	47.6	53.4	59.1	70.6	74.0	47.2	52.0	58.1	60.5	64.8
CCA-ITQ[7,8]	46.8	51.5	57.7	61.4	65.2	44.3	47.1	50.6	55.8	59.0
SH[29]	43.2	47.0	52.9	56.8	58.3	40.7	43.8	47.2	51.5	56.1

also filter out those images with less than 10 tags, resulting in a subset of 110k image examples. 500-dimensional visual features are extracted using a bag-of-visual-word model with local SIFT descriptor [18]. We randomly partition this dataset into two parts, 10k for testing and around 100k for training.

We implement our method using Matlab on a PC with Intel Duo Core i5-2400 CPU 3.1GHz and 8GB RAM. The parameters α, β and γ are tuned by cross validation on the training set and we will discuss how they will affect the performance of our approach later in detail. We repeat each experiment 10 times and report the result based on the average over the 10 runs. Each run adopts a random separation of the dataset.

4.2 Results and Discussion

The proposed BCE-FIT approach is compared with five state-of-the-art methods, including three non-hashing methods TMC [30], LM3L [10] and LSR [14], and two hashing methods CCA-ITQ [7,8] and SH [29]. For LM3L, we use linear kernels in this method to obtain fair comparison. For CCA-ITQ, the tags are treated as a different view and a common space is then learned between tags and images to form the hashing codes. For SH, the observed labels are viewed as the similarities between images and tags and a bipartite graph is constructed between nodes representing images and tags. Then, spectral hashing is applied to obtain binary codes for images and tags based on this graph. Four sets of experiments are conducted on both datasets to evaluate the effectiveness and efficiency of the proposed BCE-FIT for image tagging.

Table 2. Training time and testing time (sec) for different methods on both datasets. The length of hashing code is fix to 32 for all hashing methods.

	Flickr1m		*NUS-WIDE*	
method	training	testing	training	testing
BCE-FIT	232	1.23	86.45	0.38
LSR[14]	337	24.31	108	7.39
TMC[30]	837	5.36	528	2.57
LM3L[10]	489	23.52	154	7.86
CCA-ITQ[7,8]	254	1.23	91.83	0.37
SH[29]	198	1.22	79.44	0.38

In the first set of experiments, we evaluate the performance of different algorithms by varying the number of training tags. In particular, we vary the number of training tags for each image from $\{2, 4, 6, 8, 10\}$. We then rank the tags based on their relevance scores (Eqn.1) and return the top K ranked tags. We use the average precision (AP@K) of top 10 and 20 ranked tags as the evaluation metric. Table 1 summarizes the results for different methods. Note that for all hashing methods in this set of experiments, we fix the length of hashing codes to be 32. It is not surprising to see that the performance of all methods improve with the increasing number of training tags. From these comparison results, we can also see that BCE-FIT achieves similar or comparable accuracy results to the non-hashing methods and substantially outperforms the other hashing methods. Our hypothesis is that both CCA-ITQ and SH only focus on encoding the consistency of the binary codes to the observed tags without preserving the visual similarities among the image examples and the semantical similarities among tags, which tend to over fit. On the other hand, the proposed BCE-FIT constructs binary codes by simultaneously ensuring the learned codes to be consistent with observed tags and preserving the similarity between images and

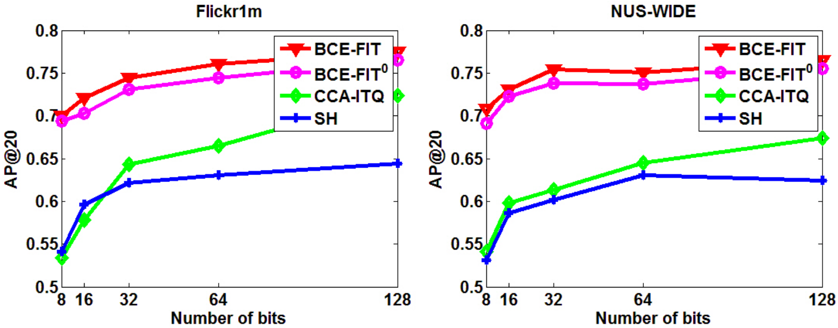


Fig. 2. Results of image tagging by varying number of hashing bits on two datasets

tags, which indicates that BCE-FIT generates more effective codes and predicts tags accurately. We also evaluate the precision and recall behavior of different methods. Due to the space limitation, we will include the precision and recall results in supplemental material.

In the second set of experiments, we evaluate the efficiency of different methods on both datasets. The training time and tag prediction time are reported in Table 2. We also fix the hashing bits to be 32 for all hashing methods. From the reported results, it is clear that image tagging process of hashing methods is 20 to 25 times faster than multi-label learning method LM3L, tag sparse reconstruction method LSR and tag matrix completion method TMC. The reason is that hashing methods use binary codes to calculate the tag relevance scores, which only involves efficient bit-wise operations XOR, while these non-hashing methods need to deal with real value vectors to compute the tag scores. We also observe that the training time of our method is comparable with other hashing methods and is much faster than TMC since the learning algorithm of TMC is quite involved with multiple terms.

In the third set of experiments, we evaluate the effectiveness of all hashing methods on both datasets by varying the number of hashing bits. We fix the number of training tags to be 10 in our experiments. We also compare our BCE-FIT with direct binarization method from Eqn.12 and call this BCE-FIT^0. The comparison results are reported in Fig.2. It is clear that the proposed BCE-FIT substantially outperforms other hashing methods on all different number of hashing bits. We can also observe that the binarization method with orthogonal transformation is consistently better than directly binarizing method. This is because BCE-FIT generates more effective hashing codes with lower quantization error than BCE-FIT^0 through orthogonal transformation, which preserves the optimality of the relaxed solution.

The fourth set of experiments study the performance of BCE-FIT with respect to the parameters α, β and γ. To prove the robustness of the proposed method, we conduct parameter sensitivity experiments on both datasets. In each

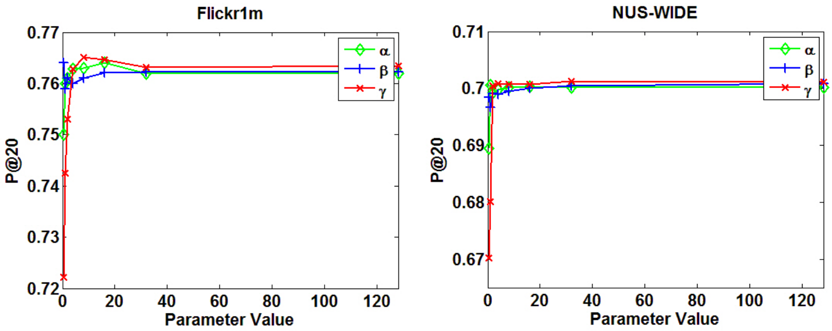


Fig. 3. Parameter Sensitivity for α, β and γ. Results of average precision with 32 hashing bits.

experiment, we tune only one parameter from $\{0.5, 1, 2, 4, 8, 16, 32, 128\}$, while fixing the other two to the optimal values obtained from the first set of experiments. We report the results on *Flickr1m* and *NUS-WIDE* in Fig.3. It is clear from these experimental results that the performance of BCE-FIT is relatively stable with respect to α, β and γ.

5 Conclusion

This paper proposes a novel Binary Codes Embedding approach for Fast Image Tagging (BCE-FIT) by designing compact binary hashing codes for both images and tags. We formulate the problem of learning binary hashing codes as a discrete optimization problem by simultaneously ensuring the observed tags to be consistent with the constructed hashing codes and preserving the similarities between images and tags. An efficient coordinate descent method is developed as the optimization procedure. Extensive experiments on two large scale datasets demonstrate that the proposed approach can achieve comparable performance with state-of-the-art methods while using much less time. There are several possible directions to explore in the future research. For example, we plan to apply some sequential learning approach to accelerate the training speed of our method.

Acknowledgments. This work is partially supported by NSF research grants IIS-0746830, DRL-0822296, CNS-1012208, IIS-1017837, CNS-1314688 and a research grant from Office of Naval Research (ONR-11627465). This work is also partially supported by the Center for Science of Information (CSoI), an NSF Science and Technology Center, under grant agreement CCF-0939370.

References

1. Bao, B.K., Ni, B., Mu, Y., Yan, S.: Efficient region-aware large graph construction towards scalable multi-label propagation. Pattern Recognition 44(3), 598–606 (2011)
2. Cabral, R.S., la Torre, F.D., Costeira, J.P., Bernardino, A.: Matrix completion for multi-label image classification. In: NIPS, pp. 190–198 (2011)
3. Chen, G., Zhang, J., Wang, F., Zhang, C., Gao, Y.: Efficient multi-label classification with hypergraph regularization. In: CVPR, pp. 1658–1665 (2009)
4. Chua, T.S., Tang, J., Hong, R., Li, H., Luo, Z., Zheng, Y.: Nus-wide: a real-world web image database from national university of singapore. In: CIVR (2009)
5. Datar, M., Immorlica, N., Indyk, P., Mirrokni, V.S.: Locality-sensitive hashing scheme based on p-stable distributions. In: Symposium on Computational Geometry, pp. 253–262 (2004)
6. Desai, C., Ramanan, D., Fowlkes, C.: Discriminative models for multi-class object layout. In: ICCV, pp. 229–236 (2009)
7. Gong, Y., Ke, Q., Isard, M., Lazebnik, S.: A multi-view embedding space for modeling internet images, tags, and their semantics. International Journal of Computer Vision 106(2), 210–233 (2014)
8. Gong, Y., Lazebnik, S., Gordo, A., Perronnin, F.: Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval. IEEE TPAMI (2012)

9. Guillaumin, M., Mensink, T., Verbeek, J.J., Schmid, C.: Tagprop: Discriminative metric learning in nearest neighbor models for image auto-annotation. In: ICCV, pp. 309–316 (2009)
10. Hariharan, B., Zelnik-Manor, L., Vishwanathan, S.V.N., Varma, M.: Large scale max-margin multi-label classification with priors. In: ICML, pp. 423–430 (2010)
11. Huiskes, M.J., Thomee, B., Lew, M.S.: New trends and ideas in visual concept detection: the mir flickr retrieval evaluation initiative. In: Multimedia Information Retrieval, pp. 527–536 (2010)
12. Li, X., Snoek, C.G.M., Worring, M.: Learning social tag relevance by neighbor voting. IEEE Transactions on Multimedia 11(7), 1310–1322 (2009)
13. Lin, R.S., Ross, D.A., Yagnik, J.: Spec hashing: Similarity preserving algorithm for entropy-based coding. In: CVPR, pp. 848–854 (2010)
14. Lin, Z., Ding, G., Hu, M., Wang, J., Ye, X.: Image tag completion via image-specific and tag-specific linear sparse reconstructions. In: CVPR, pp. 1618–1625 (2013)
15. Liu, W., Wang, J., Kumar, S., Chang, S.F.: Hashing with graphs. In: ICML, pp. 1–8 (2011)
16. Liu, X., He, J., Lang, B., Chang, S.F.: Hash bit selection: A unified solution for selection problems in hashing. In: CVPR, pp. 1570–1577 (2013)
17. Liu, Y., Jin, R., Yang, L.: Semi-supervised multi-label learning by constrained non-negative matrix factorization. In: AAAI, pp. 421–426 (2006)
18. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. IJCV 60(2), 91–110 (2004)
19. Makadia, A., Pavlovic, V., Kumar, S.: A new baseline for image annotation. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part III. LNCS, vol. 5304, pp. 316–329. Springer, Heidelberg (2008)
20. Oliva, A., Torralba, A.: Modeling the shape of the scene: A holistic representation of the spatial envelope. IJCV 42(3), 145–175 (2001)
21. Salakhutdinov, R., Hinton, G.E.: Semantic hashing. Int. J. Approx. Reasoning 50(7), 969–978 (2009)
22. Schonemann, P.: A generalized solution of the orthogonal procrustes problem. Psychometrika 31(1), 1–10 (1966)
23. Toderici, G., Aradhye, H., Pasca, M., Sbaiz, L., Yagnik, J.: Finding meaning on youtube: Tag recommendation and category discovery. In: CVPR, pp. 3447–3454 (2010)
24. Wang, Q., Ruan, L., Zhang, Z., Si, L.: Learning compact hashing codes for efficient tag completion and prediction. In: CIKM, pp. 1789–1794 (2013)
25. Wang, Q., Si, L., Zhang, D.: Learning to hash with partial tags: Exploring correlation between tags and hashing bits for large scale image retrieval. In: ECCV (2014)
26. Wang, Q., Si, L., Zhang, Z., Zhang, N.: Active hashing with joint data example and tag selection. In: SIGIR (2014)
27. Wang, Q., Zhang, D., Si, L.: Semantic hashing using tags and topic modeling. In: SIGIR, pp. 213–222 (2013)
28. Wang, S., Jiang, S., Huang, Q., Tian, Q.: Multi-feature metric learning with knowledge transfer among semantics and social tagging. In: CVPR, pp. 2240–2247 (2012)
29. Weiss, Y., Torralba, A., Fergus, R.: Spectral hashing. In: NIPS, pp. 1753–1760 (2008)
30. Wu, L., Jin, R., Jain, A.K.: Tag completion for image retrieval. IEEE Trans. Pattern Anal. Mach. Intell. 35(3), 716–727 (2013)
31. Zheng, J., Jiang, Z.: Tag taxonomy aware dictionary learning for region tagging. In: CVPR, pp. 369–376 (2013)
32. Zhou, N., Cheung, W.K., Qiu, G., Xue, X.: A hybrid probabilistic model for unified collaborative and content-based image tagging. IEEE Trans. Pattern Anal. Mach. Intell. 33(7), 1281–1294 (2011)