
AI translation • View original & related papers at
chinarxiv.org/items/chinaxiv-202205.00044

Postprint: An Online Hashing Algorithm with Balanced Label Prediction Capability

Authors: He Shuo, Xie Liang

Date: 2022-05-11T10:48:43+00:00

Abstract

To address the issues of time-consuming training, large memory consumption, and difficulty in model updating associated with traditional offline hashing algorithms, as well as the phenomenon of substantial label loss in real-world image datasets, we propose an online hashing algorithm for balanced label prediction (balanced label prediction for online hashing, BLPOH). BLPOH generates predicted labels through a label prediction module and fuses them with incomplete ground-truth labels, thereby effectively alleviating model performance degradation caused by label loss. Observing the distribution imbalance of labels, we propose a label category similarity balancing algorithm and apply it to the label prediction module to enhance label prediction accuracy. By incorporating information from old data into the online update process of the hash function, the compatibility of the model with old data is improved. Experiments on two widely used datasets and comparisons with several current state-of-the-art algorithms confirm the superiority of BLPOH.

Full Text

Preamble

Vol. 39 No. 9

Application Research of Computers
ChinaXiv Partner Journal

A Balanced Label Prediction Online Hashing Algorithm

He Shuo, Xie Liang†

(School of Faculty of Science, Wuhan University of Technology, Wuhan 430070, China)

Abstract: Traditional offline hashing algorithms suffer from time-consuming model training, large memory consumption, and difficulty in model updating.

Moreover, real-world image datasets often exhibit significant label loss. To address these issues, this paper proposes a Balanced Label Prediction for Online Hashing (BLPOH) algorithm. BLPOH generates predicted labels through a label prediction module and fuses them with incomplete real labels, effectively mitigating performance degradation caused by label loss. Observing the phenomenon of imbalanced label distribution, we propose a label category similarity balancing algorithm applied to the label prediction module to improve label prediction accuracy. By incorporating information from old data into the online update process of the hash function, the model's compatibility with historical data is enhanced. Extensive experiments on two widely used datasets, compared with several state-of-the-art algorithms, confirm the superiority of BLPOH.

Keywords: online hashing; multi-label; label prediction; image retrieval

0 Introduction

Hashing algorithms have long attracted significant attention in image retrieval due to their efficient search capabilities and low storage requirements. These algorithms map high-dimensional image features into a low-dimensional binary space [1], producing compact hash codes that preserve similarity information between images [2]. Because of these properties, image retrieval tasks can be performed efficiently in a low-dimensional space, making the quality of hash codes a critical factor affecting retrieval performance.

Researchers have dedicated considerable effort to learning effective hash functions that produce high-quality hash codes. Gionis et al. [3] proposed Locality-Sensitive Hashing (LSH), an unsupervised hashing algorithm that randomly maps original data into different hash buckets and obtains hash codes through a sign function. Weiss et al. [4] introduced Spectral Hashing (SH), which applies image segmentation concepts to solve the encoding problem for image features. Wang et al. [5] developed Semi-Supervised Hashing (SSH) for image retrieval, while Liu et al. [6] proposed Hashing with Graphs (AGH). AGH shares many concepts with SH but approximates the nearest neighbor graph between sample points through a nearest neighbor graph between data cluster centers and sample points. Shen et al. [7] presented Supervised Discrete Hashing (SDH), which solves hash codes bit by bit through a Discrete Cyclic Coordinate Descent (DCC) algorithm to avoid suboptimal solutions caused by relaxation.

However, all these hashing algorithms employ offline learning [8], training hash functions from all data at once. Faced with increasingly large image datasets, these offline hashing algorithms not only consume substantial memory space but also require considerable training time. Moreover, whenever new data arrives, offline hashing algorithms must retrain the entire model from scratch. Online hashing algorithms effectively address these challenges by processing data in a streaming fashion [9], training and updating models from a single data stream with minimal space requirements, fast training, and easy model updates. In

recent years, researchers have proposed various online hashing algorithms, including Online Kernel Hashing (OKH) [10], Online Supervised Hashing (OSH) [11], Adaptive Hashing (AdaptHash) [12], Sketching Hashing (SketchHash) [13], Online Hashing with Mutual Information (MIHASH) [14], Balanced Similarity for Online Discrete Hashing (BSODH) [15], Online Hashing with Efficient Updating of Binary Codes (OHWEU) [16], Hadamard Matrix Online Hashing (HMOH) [17], and Online Hashing via Hadamard Codebook (HCOH) [18].

Although these online hashing algorithms solve the problems of training and updating models on massive datasets, they still suffer from certain drawbacks. Real-world image datasets inevitably suffer from missing labels due to annotation errors [19], yet current online hashing algorithms assume complete label information, leading to degraded model accuracy. Taking BSODH as an example, which considers only multi-label data and determines image similarity based on whether two images share at least one label [20], incomplete labels result in inaccurate construction of the image label similarity matrix, ultimately causing model performance degradation.

The main contributions of this paper are as follows:

- a) We introduce an online hashing algorithm that addresses the challenges of training and updating models on massive datasets. By incorporating a label prediction module, we mitigate performance degradation caused by missing labels in real-world image data.
- b) Considering the problem of imbalanced label category distribution, we design a label category similarity balancing algorithm within the label prediction module to improve prediction accuracy.
- c) We establish a similarity equivalence relationship between old and new data through a degenerate form of Hamming distance, enabling information from old data to supervise the online update of the hash function. This solves the problem of performance degradation on old data in online hashing and improves the compatibility of the hash function with historical data.

This paper proposes BLPOH, an online hashing algorithm capable of balanced label prediction. By adopting online hashing, we avoid the time-consuming training, large memory consumption, and retraining requirements of offline hashing algorithms on massive image datasets. To address the issue of missing labels in real-world images, we introduce a label prediction module [21] that predicts labels for neighboring images based on image similarity and label category similarity, thereby combating performance loss caused by label incompleteness. Since most label categories are dissimilar while only a few are similar, label categories exhibit inherent imbalance. Therefore, we design a novel label category similarity balancing algorithm within the label prediction module to enhance prediction accuracy. Additionally, we cleverly measure distances between hash codes and labels of old and new data using a degenerate form of Hamming distance [22], allowing information from old data to participate in the online update of the

hash function. We conduct extensive experiments on MIR Flickr25k and NUS-WIDE datasets and compare BLPOH with several state-of-the-art algorithms to demonstrate its superiority.

1 Algorithm Framework

Figure 1 illustrates the overall framework of BLPOH. Whenever new data arrives in the data stream, we compute an image similarity matrix by finding approximate nearest neighbors and apply this matrix to label prediction. Then, we calculate an image label similarity matrix using the predicted labels and establish connections with old data, enabling information from historical data to participate in the online update process of the hash function. Finally, hash codes for the image data are computed through the hash function.

1.1 Problem Definition

Consider a multi-label dataset where $X = [x_1, \dots, x_n] \in \mathbb{R}^{d \times n}$ represents image features, with $x_i \in \mathbb{R}^d$ being the i -th image instance and $l_i \in \mathbb{R}^u$ its corresponding label. Here, n denotes the number of instances, u the number of label categories, and d the dimensionality of image features. The hash function maps image feature data to $B = [b_1, \dots, b_n] \in \{-1, 1\}^{k \times n}$, where b_i is the binary hash code vector for image instance x_i and k is the number of hash bits. Typically, hash functions adopt linear hash mapping similar to BSODH. To suit our data processing approach, we define the hash function as:

$$B = \text{sgn}(W^T X)$$

where W is the projection matrix and $\text{sgn}(\cdot)$ is the sign function that returns 1 if the variable is greater than 0 and -1 otherwise.

Table 1 introduces the main variables and their definitions used in this paper.

Table 1. Introduction to Key Variables

Variable	Definition
X_t^s	Data stream at time t
X_t^e	Existing old data at time t
L_t^s	Predicted label matrix for data stream X_t^s at time t
L_t^e	Predicted label matrix for old data X_t^e at time t
B_t^s	Hash code matrix for data stream X_t^s at time t
B_t^e	Hash code matrix for old data X_t^e at time t
W_t	Projection matrix for hash function at time t
V_t	Projection matrix between X_t^s and X_t^e at time t

We represent the data stream in online hashing algorithms as $X_t^s = [x_{t,1}^s, \dots, x_{t,n_t}^s] \in \mathbb{R}^{d \times n_t}$ and $X_t^e = [x_{t,1}^e, \dots, x_{t,m_t}^e] \in \mathbb{R}^{d \times m_t}$, where X_t^s denotes new data arriving at time t and X_t^e represents existing old data at time t . Here, n_t indicates the data volume in the stream at time t , and m_t denotes the volume of existing data at time t . Correspondingly, we represent the hash code matrix for new data in the time t data stream as $B_t^s = \text{sgn}(W_t^T X_t^s) = [b_{t,1}^s, \dots, b_{t,n_t}^s] \in \{-1, 1\}^{k \times n_t}$, and the hash code matrix for existing old data at time t as $B_t^e = \text{sgn}(W_t^T X_t^e) = [b_{t,1}^e, \dots, b_{t,m_t}^e] \in \{-1, 1\}^{k \times m_t}$. In our online hashing algorithm, the projection matrix W_t for hash codes at time t is updated in an online manner as new data X_t^s arrives. We denote the predicted label matrix for the time t data stream as $Y_t^s = [y_{t,1}^s, \dots, y_{t,n_t}^s] \in \{-1, 1\}^{u \times n_t}$, and the predicted label matrix for existing old data as $Y_t^e = [y_{t,1}^e, \dots, y_{t,m_t}^e] \in \{-1, 1\}^{u \times m_t}$, where -1 indicates the absence of a label.

1.2.1 Label Prediction and Label Category Similarity Balancing

As previously mentioned, real-world image datasets inevitably suffer from missing labels due to annotation errors. To address this, we establish a label prediction module to combat model accuracy degradation caused by label loss. Inspired by [21], assuming the label category similarity matrix K is known, an image instance's labels can be approximately represented by its nearest neighbors. Similarly, an image instance's labels can also be approximated by the nearest neighbors of that label. Following [21], we establish a label prediction regularization term, yielding the following relational expression at time t :

$$\min_{Y_t^s} \|Y_t^s Q_t - S_t^s Y_t^s\|_F^2 \quad \text{s.t.} \quad Y_t^s \in \{-1, 1\}^{u \times n_t}$$

where S_t^s is the similarity matrix between image instances computed through approximate nearest neighbors at time t , and $\|\cdot\|_F$ denotes the Frobenius norm.

Unlike [21], we innovatively propose a label category similarity balancing algorithm. This algorithm addresses the phenomenon of imbalanced label category distribution: the vast majority of label pairs are dissimilar, while only a small minority are similar. To solve this problem, we divide the label category similarity matrix into Q_1 and Q_2 , representing similar and dissimilar label pairs, respectively. We assign a smaller weight to Q_1 and a larger weight to Q_2 , thereby shifting the label category similarity toward the direction of dissimilar label pairs. For models without this balancing algorithm, achieving comparable accuracy requires more iterations, resulting in lower prediction accuracy and slower convergence speed under the same number of iterations. We define the label category similarity balancing algorithm as follows:

$$Q_{ij} = \begin{cases} \eta_s & \text{if } Q_{ij} > 0 \\ \eta_d & \text{if } Q_{ij} < 0 \end{cases}$$

where η_s represents the smaller weight and η_d represents the larger weight.

1.2.2 Mapping Relationship and Loss Function Definition

Additionally, we establish a linear mapping relationship from X_t^s to predicted labels Y_t^s , represented by V_t , and define the Frobenius norm loss at time t between X_t^s and Y_t^s :

$$\min_{V_t} \|V_t^T X_t^s - Y_t^s\|_F^2$$

The goal of hashing algorithms is to learn a hash function. According to Equation (1), we minimize the error between the linear hash function W_t and the corresponding hash codes B_t^s , establishing:

$$\min_{W_t} \|W_t^T X_t^s - B_t^s\|_F^2$$

We therefore obtain the closed-form solution for Equation (10):

$$W_t = (X_t^s (X_t^s)^T + \lambda I)^{-1} X_t^s (B_t^s)^T$$

where I is a $d \times d$ identity matrix.

Update V_t : Similar to updating W_t , we fix W_t , B_t^s , B_t^e , Y_t^s , and Y_t^e , then learn V_t . The sub-problem becomes:

$$\min_{V_t} \|V_t^T X_t^s - Y_t^s\|_F^2 + \lambda \|V_t\|_F^2$$

Thus, we obtain the closed-form solution for V_t :

$$V_t = (X_t^s (X_t^s)^T + \lambda I)^{-1} X_t^s (Y_t^s)^T$$

Update B_t^s : Similarly, fixing all other variables, we update B_t^s . The sub-problem is:

$$\min_{B_t^s} \|W_t^T X_t^s - B_t^s\|_F^2 + \alpha \|B_t^s Q_t - S_t^s Y_t^s\|_F^2 \quad \text{s.t.} \quad B_t^s \in \{-1, 1\}^{k \times n_t}$$

Following [23], Equation (14) can be reduced to solving:

$$\min_{B_t^s} \|B_t^s - \text{sgn}(W_t^T X_t^s)\|_F^2 \quad \text{s.t.} \quad B_t^s \in \{-1, 1\}^{k \times n_t}$$

Solving Equation (15) yields:

$$B_t^s = \text{sgn}(W_t^T X_t^s)$$

Update B_t^e : Fixing all other variables, we learn B_t^e . The sub-problem is:

$$\min_{B_t^e} \|W_t^T X_t^e - B_t^e\|_F^2 + \alpha \|B_t^e Q_t - S_t^e Y_t^e\|_F^2 \quad \text{s.t.} \quad B_t^e \in \{-1, 1\}^{k \times m_t}$$

Expanding Equation (17) gives:

$$\min_{B_t^e} \|B_t^e - \text{sgn}(W_t^T X_t^e)\|_F^2 + \alpha \|B_t^e Q_t - S_t^e Y_t^e\|_F^2 \quad \text{s.t.} \quad B_t^e \in \{-1, 1\}^{k \times m_t}$$

1.2.3 Incorporating Old Data Information into Online Hash Function Updates

Hamming distance measures the distance between two vectors of equal dimensionality. The more similar two data points are, the smaller the Hamming distance between their corresponding binary hash codes, and vice versa. When training on new data and updating the model, the resulting model should remain applicable to old data. Therefore, when training models from data streams, information from old data must be considered. If data x_i and x_j are similar, they should have similar label matrices, and by the property of hash functions, hash codes preserve similarity relationships between data. Consequently, we can establish a similarity equivalence relationship between new and old data through hash codes and label data. When hash codes are represented by -1 and 1, the Hamming distance $h(b_i, b_j)$ between hash codes b_i and b_j can be expressed as:

$$h(b_i, b_j) = \frac{1}{2}(k - \langle b_i, b_j \rangle)$$

Equation (18) can be simplified to:

$$\langle b_i, b_j \rangle = k - 2h(b_i, b_j)$$

Similar to BSODH, let $\langle b_i, b_j \rangle$ represent the inner product of b_i and b_j . Let b_i^s , b_i^e , y_i^s , and y_i^e denote the i -th rows of B_t^s , B_t^e , Y_t^s , and Y_t^e , respectively, while $B_{t \setminus i}^s$, $B_{t \setminus i}^e$, $Y_{t \setminus i}^s$, and $Y_{t \setminus i}^e$ represent all rows except the i -th.

Similarly, for a given label category, -1 and 1 indicate the absence and presence of the label, respectively. Minimizing the Frobenius norm loss between them yields:

$$\min_{B_t^s, B_t^e} \|B_t^s (B_t^e)^T - Y_t^s (Y_t^e)^T\|_F^2$$

where the remaining rows are excluded.

Equation (20) transforms to:

$$\min_{B_t^s} \|B_t^s P - Y_t^s L\|_F^2$$

where L is the label similarity matrix.

Solving Equation (21) gives:

$$B_t^s = Y_t^s L P^T (P P^T + \lambda I)^{-1}$$

To effectively utilize incomplete real label information, when computing the label similarity matrix L , we append the preserved real label information to the corresponding predicted labels. Specifically, we add values from the real label matrix at positions where they equal 1 to the corresponding positions in the predicted label matrix, fully leveraging existing real label information to further improve model accuracy. Since we only need to utilize information about label presence, we replace values of -1 in Y_t^s and Y_t^e with 0 while keeping values of 1 unchanged, denoted as \tilde{Y}_t^s and \tilde{Y}_t^e , respectively. The computation of L is defined as:

$$L = \tilde{Y}_t^s (\tilde{Y}_t^e)^T$$

In summary, we establish the BLPOH model formula:

$$\min_{W_t, V_t, B_t^s, B_t^e, Y_t^s} \|W_t^T X_t^s - B_t^s\|_F^2 + \alpha \|B_t^s Q_t - S_t^s Y_t^s\|_F^2 + \beta \|V_t^T X_t^s - Y_t^s\|_F^2 + \gamma \|B_t^s P - Y_t^s L\|_F^2 + \lambda (\|W_t\|_F^2 + \|V_t\|_F^2) \quad \text{s.t.} \quad B_t^s = Y_t^s L P^T (P P^T + \lambda I)^{-1}$$

1.3 Algorithm Optimization

Due to the binary constraint problem, we adopt an iterative solution approach. When updating one variable, we fix all others and treat them as constants, iteratively solving each row to obtain the final solution.

Update Y_t^s : Fixing all other variables, we learn Y_t^s . The sub-problem is:

$$\min_{Y_t^s} \alpha \|B_t^s Q_t - S_t^s Y_t^s\|_F^2 + \beta \|V_t^T X_t^s - Y_t^s\|_F^2 + \gamma \|B_t^s P - Y_t^s L\|_F^2 \quad \text{s.t.} \quad Y_t^s \in \{-1, 1\}^{u \times n_t}$$

Transforming Equation (23) into a standard linear equation:

$$(\alpha (S_t^s)^T S_t^s + \beta I + \gamma L L^T) \text{vec}(Y_t^s) = \text{vec}(\alpha (S_t^s)^T B_t^s Q_t + \beta V_t^T X_t^s + \gamma B_t^s P L^T)$$

where \otimes denotes the Kronecker product and $I_{un_t \times un_t}$ is a $un_t \times un_t$ identity matrix.

Update Q_t : Fixing all other variables, we update Q_t . The solution process is similar to solving Equation (23). We use the Preconditioned Conjugate Gradient (PCG) method to solve the linear equations in practice.

Algorithm 1 describes BLPOH.

Input: Dataset X , hash code length k , data stream size n , parameters $\alpha, \beta, \gamma, \lambda$, image nearest neighbor count K , and weights η_s, η_d .

Output: W and B , where $B = \text{sgn}(W^T X)$.

- a) Normalize dataset X , split X_t^s from X according to n , randomly initialize W and V , initialize Q as an identity matrix, and compute initial B_t^s and B_t^e .
- b) Compute similarity matrix S_t^s between data in X_t^s based on K nearest neighbors.
- c) Update W_t and V_t using Equations (11) and (13), respectively.
- d) Update B_t^s using Equation (16).
- e) Iteratively update Y_t^s using Equation (22).
- f) Update B_t^e and Q_t using Equations (24) and (25), respectively.
- g) Set $t = t + 1$ and split the next X_t^s from X .
- h) Repeat steps b) through g).

2 Experiments

2.1 Datasets

This paper conducts experiments on two publicly available datasets. To more accurately simulate label loss in real-world scenarios, we construct datasets with deterministic label loss. First, we filter data with a minimum number of specific labels to create the required image datasets and corresponding label sets. We then use the VGG-16 convolutional neural network to extract image features with a dimensionality of 4096. Simultaneously, we encode the corresponding label sets to establish label matrices. For model similarity computation, we mark a label as 1 if an image contains it and -1 otherwise. Finally, following [21], we consider incomplete label rates by randomly discarding labels from training data at rates of $\{0\%, 20\%, 40\%, 60\%, 80\%\}$ to create five sub-datasets for each condition.

The MIR Flickr25k dataset contains 25,000 images and 24 label categories. We filter images with at least 3 labels, obtaining 17,568 images. After feature

extraction and label matrix construction, we randomly discard labels at rates of $\{0\%, 20\%, 40\%, 60\%, 80\%\}$ to simulate label loss. In experiments, except for those analyzing the impact of database size, we randomly select 12,400 images from the filtered 17,568 images, using 10,000 as the database for training and the remaining 2,400 as the query set.

The NUS-WIDE dataset contains 269,648 images and 81 label categories. We select the 21 most frequent categories as labels. Following a similar process as with MIR Flickr25k, we filter 57,073 images and randomly select 12,100 as our experimental dataset, using 10,000 for training and the remaining 2,100 as the query set.

2.2 Baseline Methods and Evaluation Metrics

We compare BLPOH against several state-of-the-art online hashing algorithms: BSODH, MIHASH, OKH, AdaptHash, SketchHash, HMOH, and OHWEU. During experiments, BLPOH and comparison algorithms use identical datasets and label loss rate settings. For each comparison algorithm, parameters are specifically tuned to achieve optimal performance. The experimental environment uses Windows 10, an R5-3600 CPU, and 32GB RAM. For MIR Flickr25k, parameters are set as $\alpha = \beta = 0.8$, $\gamma = 0.9$, $\lambda = 0.6$, stream size = 200, nearest neighbors $K = 10$, $\eta_s = 0.4$, $\eta_d = 1$. For NUS-WIDE, parameters are $\alpha = \beta = \gamma = 0.9$, $\lambda = 0.1$, stream size = 200, nearest neighbors $K = 12$, $\eta_s = 0.2$, $\eta_d = 1$. We use Mean Average Precision (MAP) as the evaluation metric to assess algorithm performance.

2.3.1 MAP Comparison

Tables 2-4 and Figure 2 present MAP results comparing BLPOH with baseline algorithms on MIR Flickr25k and NUS-WIDE datasets under label loss rates of $\{0, 20\%, 40\%, 60\%, 80\%\}$ across different hash code lengths.

As shown in Table 2, on MIR Flickr25k with various label loss rates, BLPOH consistently achieves higher MAP than all comparison algorithms. HMOH and OHWEU perform relatively close to BLPOH, while AdaptHash performs the worst overall. On NUS-WIDE, when the label loss rate is 0, BLPOH's MAP is higher than all algorithms except HMOH. As the label loss rate increases, BLPOH surpasses all comparison algorithms including HMOH, demonstrating its superior performance. Comparing with HMOH on NUS-WIDE reveals that when no labels are missing, HMOH achieves higher MAP, but when the label loss rate reaches 20% and beyond, BLPOH outperforms HMOH, proving BLPOH's effectiveness against label loss. In practical applications where label loss can be severe, BLPOH is more suitable for real-world scenarios.

Figure 2 illustrates how MAP varies with increasing label loss rates for BLPOH and comparison algorithms on MIR Flickr25k and NUS-WIDE across different hash code lengths. The results show that BLPOH's MAP decreases more

gradually and maintains higher levels in most cases, regardless of dataset or hash code length.

2.3.2 Key Parameter Sensitivity Analysis

This subsection analyzes the impact of label category similarity weight parameters η_s and η_d on experimental results. In experiments, we set $\eta_d = 1$ and vary η_s in $\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$ to examine their effects.

As shown in Figure 3(a), on MIR Flickr25k, MAP reaches its maximum when η_s is between (0.3, 0.5), and drops significantly when η_s exceeds 0.8. The lowest MAP occurs when $\eta_s = 1$ (i.e., $\eta_s = \eta_d$), representing the case without label category similarity balancing. We select $\eta_s = 0.4$ as the practical parameter value. Figure 3(b) shows that on NUS-WIDE, MAP is lowest when $\eta_s = 1$, and we ultimately choose $\eta_s = 0.2$ as the practical parameter value. The performance across both datasets demonstrates that MAP without label category similarity balancing is consistently lower than with balancing, proving the effectiveness of this technique for BLPOH.

2.3.3 Impact of Database Sample Size

As an improved online hashing algorithm, we test BLPOH's retrieval performance by analyzing how experimental results vary with increasing database samples. We set query set sizes to 2,400 and 2,100 for MIR Flickr25k and NUS-WIDE, respectively, using the remainder for training. As shown in Figure 4(a), on MIR Flickr25k, BLPOH achieves the worst MAP when the database contains 2,000 samples. MAP gradually increases with sample sizes beyond 4,000 and stabilizes at 10,000 samples. Figure 4(b) shows that on NUS-WIDE, BLPOH's MAP increases progressively from 2,000 samples and stabilizes at 8,000 samples.

2.3.4 Impact of Label Category Similarity Balancing Algorithm

During algorithm optimization, we transform the optimization of predicted label matrix Y_t^s and label category similarity matrix Q_t into solving corresponding linear systems (Equations 23 and 24). In practice, we use Preconditioned Conjugate Gradient (PCG) to solve these equations.

This section analyzes the impact of label category similarity balancing on BLPOH through PCG convergence behavior, using label prediction and similarity matrix solving on NUS-WIDE as examples. As shown in Figure 5(a), during label prediction for each data stream arrival, the algorithm with label category similarity balancing consistently converges faster and more stably than without balancing. Similarly, Figure 5(b) demonstrates that solving the label category similarity matrix is also more stable with balancing.

2.3.5 Image Instance Retrieval Performance

To demonstrate BLPOH's practical retrieval effectiveness, we compare its instance retrieval performance with BSODH on MIR Flickr25k. We select an image from the dataset and return the top 5 most similar images based on models trained under identical conditions. As shown in Figure 6, the BLPOH-trained model returns 5 human portraits, while the BSODH-trained model returns 4 human portraits and one animal image, clearly showing BLPOH's superior retrieval quality.

3 Conclusion

This paper proposes an improved online hashing algorithm that processes data in a streaming fashion, training and updating models online to ensure the algorithm is not constrained by dataset size and maintains good real-time performance. By incorporating information from old data to supervise hash function updates, we enhance compatibility with historical data. Additionally, considering the problem of missing labels in real-world image data, we introduce a label prediction module and innovatively propose a label category similarity balancing algorithm to guide label prediction, making predicted labels more accurate and improving image label similarity accuracy, thereby further enhancing model precision. Our label prediction module not only effectively combats label missingness but also ensures model performance degrades gradually as label loss rates increase, showing promising application prospects for online image retrieval tasks with missing labels. However, our label prediction model is relatively simple, and its robustness is not entirely satisfactory. Future work will consider incorporating ensemble learning to improve model robustness.

References

- [1] Guo Yicun, Chen Huahui. Survey on online hashing algorithm [J]. Journal of Computer Applications, 2021, 41 (04): 1106-1112.
- [2] Lin Guosheng, Shen Chunhua, Shi Qinfeng, et al. Fast supervised hashing with decision trees for high-dimensional data [C]// Proc of the 27th IEEE Conference on Computer Vision and Pattern Recognition. Los Alamitos, CA: IEEE Computer Society, 2014: 1963-1970.
- [3] Gionis A, Indyk P, Motwani R. Similarity search in high dimensions via hashing [C]// Proc of the 25th Very Large Data Base Conference. San Francisco, CA: Morgan Kaufmann Publishers, 1999: 518-529.
- [4] Weiss Y, Fergus R, Torralba A. Multidimensional spectral hashing [C]// Proc of the 12th European Conference on Computer Vision. Berlin: Springer Verlag, 2012: 340-353.

- [5] Wang Jun, Kumar S, Chang S F. Semi-supervised hashing for large-scale search [J]. *IEEE Trans on Pattern Analysis and Machine Intelligence*, 2012, 34 (12): 2393-2406.
- [6] Liu Wei, Mu Cun, Kumar S, et al. Discrete graph hashing [C]// Proc of the 27th International Conference on Neural Information Processing Systems. Cambridge, MA: MIT Press, 2014: 3419-3427.
- [7] Shen Fumin, Shen Chunhua, Liu Wei, et al. Supervised discrete hashing [C]// Proc of the 28th IEEE Conference on Computer Vision and Pattern Recognition. Los Alamitos, CA: IEEE Computer Society, 2015: 37-45.
- [8] Bombara G, Belta C. Offline and online learning of signal temporal logic formulae using decision trees [J]. *ACM Trans on Cyber-Physical Systems*, 2021, 5 (3): 1-23.
- [9] Lu Xu, Zhu Lei, Cheng Zhiyong, et al. Flexible online multi-modal hashing for large-scale multimedia retrieval [C]// Proc of the 27th ACM International Conference on Multimedia. New York: Association for Computing Machinery, 2019: 1129-1137.
- [10] Huang Longkai, Yang Qiang, Zheng Weishi. Online hashing [C]// Proc of the 23rd International Joint Conference on Artificial Intelligence. Menlo Park, CA: AAAI Press, 2013: 1422-1428.
- [11] Cakir F, Bargal S A, Sclaroff S. Online supervised hashing [J]. *Computer Vision and Image Understanding*, 2017, 156: 162-173.
- [12] Cakir F, Sclaroff S. Adaptive hashing for fast similarity search [C]// Proc of the 15th IEEE International Conference on Computer Vision. NW Washington, DC: IEEE Computer Society, 2015: 1044-1052.
- [13] Leng Cong, Wu Jiaxiang, Cheng Jian, et al. Online sketching hashing [C]// Proc of the 28th IEEE Conference on Computer Vision and Pattern Recognition. Los Alamitos, CA: IEEE Computer Society, 2015: 2503-2511.
- [14] Cakir F, He Kun, Adel Bargal S, et al. Mihash: Online hashing with mutual information [C]// Proc of the 16th IEEE International Conference on Computer Vision. NW Washington, DC: IEEE Computer Society, 2017: 437-445.
- [15] Lin Mingbao, Ji Rongrong, Liu Hong, et al. Towards optimal discrete online hashing with balanced similarity [C]// Proc of the 33rd Association for the Advancement of Artificial Intelligence Conference on Artificial Intelligence. Palo Alto, CA: AAAI Press, 2019: 8722-8729.
- [16] Weng Zhenyu, Zhu Yuesheng. Online hashing with efficient updating of binary codes [C]// Proc of the 34th Association for the Advancement of Artificial Intelligence Conference on Artificial Intelligence. Palo Alto, CA: AAAI Press, 2020: 12354-12361.
- [17] Lin Mingbao, Ji Rongrong, Liu Hong, et al. Hadamard matrix guided online hashing [J]. *International Journal of Computer Vision*, 2020, 128 (8): 2279-2306.

- [18] Lin Mingbao, Ji Rongrong, Liu Hong, et al. Supervised online hashing via hadamard codebook learning [C]// Proc of the 26th ACM International Conference on Multimedia. New York: Association for Computing Machinery, 2018: 1635-1643.
- [19] Wang Lin, Zhang Sulan, Yang Haifeng. A nearest neighbor image annotation method based on CNN and weighted bayesian [J]. Computer Technology and Development, 2021, 31 (10): 63-69.
- [20] Jiang Qingyuan, Li Wujun. Deep cross-modal hashing [C]// Proc of the 30th IEEE Conference on Computer Vision and Pattern Recognition. Los Alamitos, CA: IEEE Computer Society, 2017: 3232-3240.
- [21] Dong Haochen, Li Yufeng, Zhou Zhihua. Learning from semi-supervised weak-label data [C]// Proc of the 32nd Association for the Advancement of Artificial Intelligence Conference on Artificial Intelligence. Palo Alto, CA: AAAI Press, 2018: 2926-2933.
- [22] Norouzi M, Punjani A, Fleet D J. Fast search in hamming space with multi-index hashing [C]// Proc of the 25th IEEE Conference on Computer Vision and Pattern Recognition. Los Alamitos, CA: IEEE Computer Society, 2012: 3108-3115.
- [23] Kang Wangcheng, Li Wujun, Zhou Zhihua. Column sampling based discrete supervised hashing [C]// Proc of the 30th Association for the Advancement of Artificial Intelligence Conference on Artificial Intelligence. Palo Alto, CA: AAAI Press, 2016: 1230-1236.

Note: Figure translations are in progress. See original paper for figures.

Source: ChinaXiv –Machine translation. Verify with original.