Incremental Hashing with Sample Selection Using Dominant Sets

Wing W. Y. Ng[a], Xiaoxia Jiang[a], Xing Tian[b,]*, Marcello Pelillo[c], Hui Wang[d], Sam Kwong[b]

[a]Guangdong Provincial Key Laboratory of Computational Intelligence and Cyberspace Information, School of Computer Science and Engineering, South China University of Technology, Guangzhou, Guangdong, 510006, China

[b]Department of Computer Science, City University of Hong Kong, Hong Kong, China

[c]Department of Environmental Sciences, Informatics and Statistics, University of Venice, 30172 Venice, Italy, and also with the European Centre for Living Technology, University of Venice, 30172 Venice, Italy

[d] School of Computing, Ulster University, Jordanstown, United Kingdom

Abstract

In the world of big data, large amounts of images are available in social media, corporate and even personal collections. A collection may grow quickly as new images are generated at high rates. The new images may cause changes in the distribution of existing classes or the emergence of new classes, resulting in the collection being dynamic and having concept drift.

For efficient image retrieval from an image collection using a query, a hash table consisting of a set of hash functions is needed to transform images into binary *hash codes* which are used as the basis to find similar images to the query. If the image collection is dynamic, the hash table built at one time step may not work well at the next due to changes in the collection as a result of new images being added. Therefore, the hash table needs to be rebuilt or updated at successive time steps.

Incremental hashing (ICH) is the first effective method to deal with the concept drift problem in image retrieval from dynamic collections. In ICH, a new hash table is learned based on newly emerging images only which represent data distribution of the current data environment. The new hash table is used to generate hash codes for all images including old and new ones. Due to the dynamic nature, new images of one class may not be similar to old images of the same class. In order to learn new hash table that preserves within-class similarity in both old and new images, *incremental hashing with sample selection using dominant sets* (ICHDS) is proposed in this paper, which selects representative samples from each class for training the new hash table. Experimental results show that ICHDS yields better retrieval performance than existing dynamic and static hashing methods.

Keywords

Image Retrieval, Incremental Hashing, Semi-Supervised Hashing, Concept Drift, Dominant Sets.

1. Introduction

With the rapid development of digital technologies, multimedia data such as videos, images and audios are generated in large quantities and at high rates. Similarity search is thus becoming more and more important. How to quickly find the most relevant data from large collections of multimedia data is a challenge. The hashing method has been successfully applied for approximate nearest neighbor search from large-scale image collections [1]. Hashing methods learn hash functions to map images from high-dimensional content feature space to low-dimensional Hamming space. By training the hash function, the compact hash code representation of the massive data is learned so that the hash code can maintain the similarity relationship of the data in the original space. Hamming distance between binary hash codes of images are computed to evaluate the similarities between images. Since hash codes are binary with much smaller dimension comparing to the original feature of images, hashing methods retrieve relevant images of the query efficiently with lower storage space cost.

In the literature, a lot of hashing methods have been proposed for image retrieval, but most of which learn hash functions in static environments. The real-life data environment will change over time and the newly generated data may cause change to the distribution of the current collections, resulting in the so-called *concept drift*. Therefore, the hash model trained in a static environment usually loses the ability to learn new samples and cannot adapt well to changes in the data distribution of the dynamic environment.

There are some online or incremental hashing methods being proposed to address data changes in a dynamic environment, such as OKH [23], OSH [26], ICH [30], etc. However, most of them only assume the data appearing in an online manner, without considering the distribution changes and emerging of new classes, i.e. concept drift problem. Incremental hashing (ICH) is the first image retrieval method to deal with the concept drift problem in dynamic environment. However, the new hash table learned at each time step in ICH is based on the newest data batch only, which just contains the similarity information of the current data environment. Since hash codes of all images including old and new ones are generated based on this new hash table, similarity information in both old and new images should be considered holistically for the training of new learned hash table.

Therefore, a semi-supervised *incremental hashing with sample selection using dominant sets* (ICHDS) is proposed in this paper, which selects representative samples for each class from old and new image collections. Dominant sets clustering method is employed to evaluate the semantic concept differences between samples for sample selection when new data appears. Selected representative samples are combined as the semantic pool for training. ICHDS utilizes all the images in the semantic pool to train a new hash table and updates the weights of hash tables at successive time steps, thereby dynamically learning the changes of the data distribution due to the new images appearing.

The contributions of this paper are summarized as follows:

1) A sample selection method based on dominant sets is proposed in this paper to select representative samples from each class for hash learning. Dominant sets sample selection method is robust to noise and does not require prior knowledge about the number of classes and distribution of samples in each class. These selected representative samples provide comprehensive data distribution information of each class for the training of a new hash table.

2) A semi-supervised *incremental hashing with sample selection using dominant sets* (ICHDS) is proposed, which employs multiple hash tables with corresponding weights to generate a multi-hashing system. At each time step, a new hash table is trained based on the similarity information of images in the semantic pool. Hash tables and corresponding weights in the multi-hashing system of ICHDS are updated dynamically with new data appearing.

3) ICHDS is compared with state-of-the-art hashing methods in 11 dynamic data scenarios. Experimental results show that ICHDS achieves significant performance improvement over the original ICH method and outperforms other comparative hashing methods as well.

This paper is organized as follows. Section 2 describes related works of static and dynamic hashing methods. We propose ICHDS in Section 3. Section 4 shows experimental results in 11 dynamic data scenarios in three real world image datasets. Finally, Section 5 concludes this work.

2. Related Works

Most of hashing methods for image retrieval are presented in a static data environment. As time goes by, the scale of new data continues to increase. Static hashing methods do not adapt well to changes in dynamic data environments. Some hashing methods for image retrieval problems in dynamic data environments have been studied. Sections 2.1 and 2.2 describe the existing static hashing and dynamic hashing methods, respectively.

2.1 Static Hashing Methods

According to whether the semantic similarity information is used for the training of hash functions, existing static hashing methods can be summarized into three broad categories: unsupervised hashing, supervised hashing, and semi-supervised hashing, respectively.

Unsupervised hashing does not require labeled information of data. The unsupervised locality-sensitive hashing (LSH) [2] which randomly constructs hash function is considered as the most primitive image retrieval method. KLSH [3] is an improvement and extension based on LSH. KSLH employs kernel functions and sparse sets in the image collections to construct random maps. A subset of the threshold feature vectors of the Laplacian matrix of the graph is used in SH [4] to obtain a hash function by the relaxation condition. Iterative quantization (ITQ) [5] employs PCA to reduce the dimensions of data and projects the data to the vertices of a binary hypercube. MIPS [6] maximizes the correlation between the raw data and the inner product of the binary code based on the inner product fitting framework. The objective functions including binary quantization loss, shared subspace contribution, and spectral embedding loss are introduced in [7] to guide the learning of the hashing function. In [8], the objective function is optimized by augmenting the Lagrangian method. QRank [9] trains multiple hash tables and employs query-adaptive bitwise weighting for each hash table.

Due to the existence of the semantic gap, the unsupervised hashing method does not preserve the similarity relationship of semantic images. Therefore, supervised hashing utilizing semantic information is proposed to achieve higher retrieval accuracy. Supervised hashing with kernels (KSH) [10] replaces the Hamming distance with the coded inner product and correlates well with the similarity matrix, which greatly reduces the computation time and improves the efficiency. Top-RSBC [11] aims to optimize the accuracy of the top position in the Hamming distance ranking list to protect semantic information. The sorting information of the sample is carried in the loss function. The key of SDH [12] is to minimize the loss function to obtain hash code and regress the obtained hash code to its corresponding label. Different from the traditional SDH method, in FSDH [13], the class label of the training example is regressed to the corresponding hash code to speed up the training of hash model. Another supervised discrete hashing COSDISH [14] directly trains discrete hash codes and selects the hash code in each iteration. COSDISH decomposes the selected hash code into two parts and optimizes them alternately. The hashing method of deep learning is also a hot topic today. Deep discrete supervised hashing (DDSH) [15] directly guides hash codes and depth features with paired semantic information to enhance feedback between each other.

Supervised hashing method requires the identification of all data in dataset, but it is difficult to achieve in real life. Therefore, semi-supervised hashing (SSH) [16] is proposed. SSH not only uses semantic information, but also minimizes empirical errors on labeled data. At the same time, SSH also uses the theoretical regularization of unlabeled data and labeled data. Classic unsupervised hashing methods are SPLH [17] and BSPLH [18]. SPLH iteratively updates the pairwise label matrix, and the training of each hash function is corrected according to the error caused by the training of the previous hash function. BSPLH effectively corrects errors based on training of samples that have not been learned by all previous hash functions. Unlike SSH and BSPLH, SCEM-SSH [33] sequentially maximizes the joint entropy between the hash codes of each bit to train the hash function. SCEM-SSH corrects all previous hash bits that have not been learned well. DCH [19] is multiple hash tables learning method, and each hash table is trained using SPLH. DCH gradually learns the hash function and the hash table by performing error correction based on the previous hash function and the hash table. SSMDH [20] learns the relationship between multiple features of different view samples, and combines discrete learning hash functions. BSPLH is also used in BBSHR [21] to train multiple hash tables. BBSHR uses query adaptive weighting for re-ranking to improve retrieval accuracy. SIF [34] uses bucket and location sensitivity measurement to score the retrieved image. RBFNN training is used in SIF to select out dissimilar images. In addition, in the field of deep learning, the hash code obtained by semi-supervised deep hashing (SSDH) [22] can maintain semantic similarity information and distribution of underlying data.

2.2 Dynamic Hashing Methods

Most of existing hashing models are offline, however the real-life data environment is not always static. The offline hashing and static hashing methods can't adapt well to the dynamic data environment. Existing static hashing methods usually lose the learning ability against new data samples when the difference of distribution between the previous and new data is large. In order to solve this problem, several online hashing and dynamic hashing methods have been proposed to address data changes in a dynamic environment. Online kernel-based hashing (OKH) [23] constantly updates the hash function according to the similarity of the newly added pairwise samples. Online hashing (OH) [24] is an expanded version of OKH. OH minimizes the loss function of the similarity of the pairwise samples in Hamming space to update the hash function. A multi-model MMOH proposed in OH aims to train multiple complementary hashes based on the similarity loss function. Another online hashing MIHash [25] optimizes objective functions online based on mutual information using gradient descent method, thereby reducing unnecessary hash table updates. Unsupervised online sketching hashing (OSH) [26] is proposed to update the hash function online based on the concept of data sketching. OSH trains the new hash function based on the small size sketch of dataset to get main features needed. Unlike unsupervised OSH, supervised OSSH [35] combines supervised semantic information with streaming data to construct data sketch to guide the training of hash functions. Based on error correcting output codes (ECOCs), the hash function of online supervised hashing [27] learns new labeled data, and updates the hash function in a discerning manner. The stochastic gradient descent (SGD) method is employed in [28] to iteratively update the hash function based on the newly emerging samples.

These online hashing methods only consider the constant emergence of data streams, regardless of the new data that may result in data distribution changes. Incremental hash-bit learning (IBL) [29] and incremental hashing (ICH) [30] are currently proposed for image retrieval to solve the problem of concept drift in dynamic environment. IBL trains a new hash table with the new data batch, and picks up the hash functions that best fit the current environment to update the hash table. In ICH, the latest data batch is used to train a new hash table and calculate the weights of all hash tables.

Semantic image retrieval is queried from the entire collections, including the latest data batch and data emerging in previous time steps. However, ICH only uses all the images in the latest data batch to train a new hash table. Using the representative data that more fully expresses the relationship between the images of each concept, the hash code obtained by the newly learned hash function training can divide the concept as well as possible. Therefore, this paper proposes a semi-supervised incremental hashing method with sample selection using dominant sets (ICHDS). Different from ICH, ICHDS not only considers the distribution of current new samples, but also considers the semantic information in the previous collections to adapt to the dynamic data environment, thereby improving retrieval performance.

In [36], a preliminary version of this work is proposed. Different from this preliminary method, the process of sample selection using dominant sets in this paper is based on the nature characteristics of the images, i.e. the similarity measure and the relationship matrix. It is not necessary to randomly initialize the cluster centroid position, which reduces the error of sample selection. The selected representative samples describe the comprehensive distribution of each class, not just filtering noisy or anomalous samples.

3. Incremental Hashing with Sample Selection Using Dominant Sets

There may be differences in the distribution between the newly emerged data batch and the previous data batches. The purpose of ICHDS is to dynamically update the multi-hashing system Ω^T at time *T* to adapt to changes of data distribution with new images emerging sequentially. It is assumed that that a new data batch $D^T \in R^{n \times d}$ emerges in each time step, where *n* and *d* represent the number of images and the feature dimension of each image, respectively. The amount of images in data batch at each time step is consistent. The data batch D^T can be divided into labeled dataset D_l^T and unlabeled dataset D_{ul}^T. All representative samples used for training generate a semantic pool, i.e. X_l^T. In ICHDS, representative samples are firstly selected from the new labeled dataset D_l^T and the data in old semantic pool X_l^{T-1} using dominant sets clustering at time *T*. The selected representative data is used to update the semantic pool X_l^T. The semantic pool which provides supervised information is combined with the latest unlabeled dataset D_{ul}^T as training set X^T for training the new hash table.

ICHDS generates a multi-hashing system Ω^T at time *T*. The new system at this time step consists of two parts, *M* hash tables and corresponding weights. In ICHDS, each hash table contains *K* hash functions. *M* hash tables in this system are trained by employing data from different time steps. At each time step, ICHDS employs semi-supervised BSPLH and training set X^T to train a new hash table. Then ICHDS computes the weights of existing *M* + 1 hash tables based on the retrieval performance of each hash table. Only *M* hash tables with the highest weight are kept. The final *M* hash tables and their corresponding weights v_i^T generate a multi-hashing system Ω^T as follows:

$$\Omega^T = \left\{ \left(v_i^T, W_i^T \right) \right\}, i \in \{1, 2, ..., M\} \qquad (1)$$

where W_i^T, v_i^T denote the i^{th} hash table and the weight of the hash table at time *T*, respectively.

3.1 Sample Selection for Training Using Dominant Sets

For the scenario in which samples of new classes emerge, the semantic information and distribution information of the samples are different from the old samples. Samples of new classes provide useful similarity information to train a new hash table. Therefore, samples of new classes are added to the semantic pool directly. For the scenario in which the distribution drift occurs for existing classes, the distribution between samples of the same class may also have large differences. For example, whales and dolphins are aquatic mammals, but their visual characteristics are significantly different.

The distribution difference between samples is judged by the idea of dominant sets clustering in [31, 32]. Samples that are not in a dominant set are considered to have large differences with other samples. These samples located distinctively provide representative semantic information and are added to the semantic pool for training. In this way, new samples in a dominant set are viewed as redundant data and not used for training.

3.1.1 Dominant Sets

Among existing clustering methods, dominant sets [32] achieve excellent clustering performance and competitiveness in terms of stability, segmentation accuracy, robustness, and computation time comparing to K-means and other spectral methods, such as the Normalized Cut (Ncut) method. Therefore, the idea of dominant sets clustering is employed for sample selection in ICHDS.

Images being clustered are defined as an undirected graph $G = (V, E, w)$, where $V = \{1, 2, ..., n\}$, $E \subseteq V \times V$, w denote vertex set, edge set, and the weights of all edges, respectively. Relationship matrix $A = a_{ij}$ represents the relationship between images, where a_{ij} represents the similarity between the i^{th} image and the j^{th} image. Images in a dominant set are equivalent to the fact that the images belong to a cluster. The problem of finding a dominant set can be transformed into a problem of solving the quadratic maximum of a standard simplex, i.e.

$$\begin{array}{ll} \max mize & f(p) = p'Ap \quad , \\ \text{subject to} & p \in \Delta \end{array} \qquad (2)$$

where $\Delta = \left\{ p \in R^n : p_i \geq 0 \;\; and \;\; \sum_{i \in V} p_i = 1, \;\; \forall i \in V \right\}$.

The support set of vector p can be defined as a subscript set of non-zero elements in vector p, then its support is a dominant set, as follows:

$$\sigma(p) = \{ i \in V : p_i \neq 0 \} \qquad (3)$$

Due to the correspondence between dominant sets and quadratic form (2), solving quadratic maximum on this simplex can be solved by the equation derived from decision theory [31] as follows:

$$p_i(\tau + 1) = p_i(\tau) \frac{(Ap(\tau))_i}{p(\tau)' Ap(\tau)} \qquad (4)$$

where $p_i(\tau)$ denotes the element in vector $p(\tau)$ and τ is the number of iteration steps. The element value in p indicates the possibility of the corresponding sample belonging to the cluster. The support set of the vector p is the vertex corresponding to the dominant set, as the segmentation criterion [31]. Each element $p_i(\tau)$ corresponds to an image in the original image set V. When the value of this element is larger, the possibility of the corresponding image belonging to the cluster is higher.

ICHDS draws on the selection of the relationship matrix in [32], and defines the relationship matrix $A = a_{ij}$ as follows:

$$a_{ij} = \begin{cases} \exp(-\frac{w(i,j)^2}{r^2}), & if\,(i,j) \in E \\ 0, & otherwise \end{cases} \qquad (5)$$

where $w(i, j)$ represents the similarity measure function of the i^{th} image and the j^{th} image in the image feature library. The parameter $r > 0$ is a scale factor, which plays an important role in regulating cluster sensitivity. The similarity relationship between images can be measured by Euclidean distance as follows:

$$\eta(i,j) = sqrt\left(\sum_{k=1}^{d} \left(i_k - j_k \right)^2 \right) \qquad (6)$$

where *d* denotes the feature dimension of each data. The Euclidean distance $\eta(i, j)$ is employed in this paper which can be described as $w(i, j) = \eta(i, j)$ according to Eq.(6).

In the clustering process, images with low weights of edges are not forced into a certain cluster. The number of clusters generated by this method is not set by the user in advance. Rather, it is determined by the nature of the image, that is, the similarity measure and the relationship matrix. Therefore, the size and structure of the class is determined by the characteristics of the image itself. Algorithm 1 describes the pseudo code of dominant sets clustering.

Algorithm 1：Dominant sets clustering.

Input: Relationship matrix A.
Output: Support set $\sigma(p)$.

1. Initialize vector $p(\tau)=\left[\frac{1}{n},\frac{1}{n},\ldots,\frac{1}{n}\right]_{n\times 1}$.

2. Compute $p_i(\tau+1)=p_i(\tau)\frac{(Ap(\tau))_i}{p(\tau)'Ap(\tau)}$ in Eq.(4).

3. Output the clustering result $\sigma(p)$ of the original images according to Eq.(3).

4. The corresponding value of the image in the previous cluster is removed from the relation matrix A.

5. Determine whether most of the images are clustered. If yes, the algorithm is terminated. If not, the remaining image sets perform Step 1 to Step 4.

3.1.2 Sample Selection Using Dominant Sets

The first data batch D^0 that emerges at time T = 0 serves as the most primitive data for training. The labeled data in D^0 is used to build the original semantic pool X_l^0. For the scenario in which new labeled data $x_l^t \in D_l^t$ of existing classes emerges at time T = t, there are a total of $C_{old} \in \{c_{old1},\ldots,c_{olds}\}$ classes in the database. For a certain class $C \in C_{old}$, the latest labeled data in this class and the data of the same class in the semantic pool establish a two-point-one-line edge relationship. These labeled samples are clustered using the concept of dominant sets based on the similarity weights of the edges. The data in a dominant set is considered to be the most similar data with redundant information, and is not utilized for training. The remaining labeled data is stored in the semantic pool X_l^t for training of new hash table.

For the scenario in which labeled data $x_l^t \in D_l^t$ of new classes $C_{new} \in \{c_{new1},\ldots,c_{news}\}$ emerges at time T = t. The previous hash tables could not preserve the relevant semantic information well. Therefore, the data of a new emerging class $C \in C_{new}$ is informative and valuable for the training of new hash table. These data of new emerging classes is added to the semantic pool X_l^t directly for the training of new hash table. ICHDS evaluates each labeled data in semantic pool with function $\Psi(\bullet)$ to determine whether new labeled data of existing classes is used for training as follows:

$$\Psi(x_l^t)=\begin{cases}1, & if\ x_l^t \notin X_{\sigma(p)}\\ 0, & if\ C \in C_{new}\\ -1, & otherwise\end{cases} \tag{7}$$

where $x_l^t \notin X_{\sigma(p)}$ denotes the labeled data is not in a dominant set. According to Eq.(7), new labeled sample is added to semantic pool when its $\Psi(\bullet)$ function value equals to 0 or 1. The semantic pool at each time step is updated as follows:

$$X_l^T = X_l^{T-1} \cup \{x_l^T \mid \Psi(x_l^T)=0\} \cup \{x_l^T \mid \Psi(x_l^T)=1\} \tag{8}$$

The semantic pool X_l^T combines with the latest unlabeled dataset D_{ul}^T as training set X^T for each time step to train the new hash table, as follows:

$$X^T = X_l^T \cup D_{ul}^T \tag{9}$$

Figure 1 shows the steps of sample selection in IHCDS at time $T = t$.

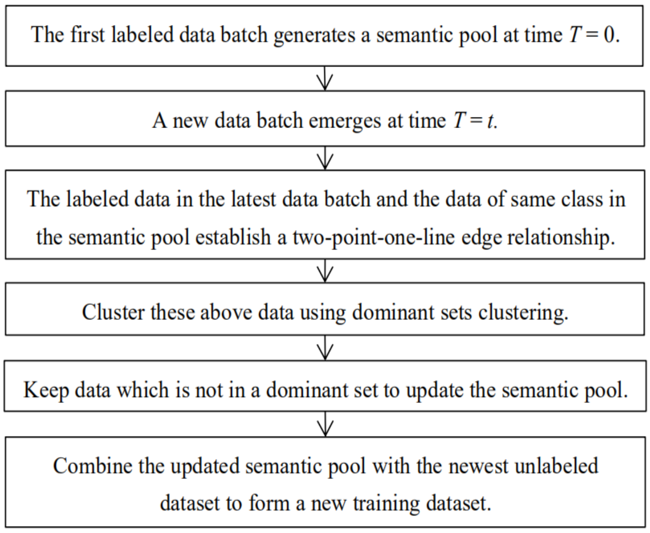

The first labeled data batch generates a semantic pool at time T = 0.
A new data batch emerges at time T = t.
The labeled data in the latest data batch and the data of same class in the semantic pool establish a two-point-one-line edge relationship.
Cluster these above data using dominant sets clustering.
Keep data which is not in a dominant set to update the semantic pool.
Combine the updated semantic pool with the newest unlabeled dataset to form a new training dataset.

Figure 1. The step of sample selection using dominant sets clustering in ICHDS.

3.2 Hash Table Training and Calculation of Weights for Ranking

In each time step, ICHDS employs semi-supervised BSPLH to train a new hash table. The objective function of the semi-supervised hashing is described in the form of a compact matrix. It tries to maximize the objective functions as follows:

$$\max_W \frac{1}{2} tr\{W'X_lSX_l'W\} + \frac{\lambda}{2} tr\{W'XX'W\} \tag{10}$$

where S and λ denote the similarity matrix of pairwise labeled data X_l and the parameter of regularization, respectively. The training of the objective function consists of two aspects. The former $tr\{W'X_lSX_l'W\}$ is the empirical accuracy using the labeled information X_l to learn the hash function. The latter $tr\{W'XX'W\}$ employs the regularization item to prevent over-fitting problems caused by the number of labeled data being too small in the entire dataset. To improve generalization capabilities, regularization employs all labeled and unlabeled data which is used for training, rather than relying solely on labeled data. Hash functions in one hash table is trained sequentially in ICHDS. Each hash function is trained by correcting the errors caused by its previous ones.

The weight of hash table is evaluated based on its performance on the latest training images. Firstly, the Hamming distance based on the K-bit hash codes of images should be consistent with the real semantic similarity between images based on the label, i.e. semantic consistency. Moreover, hash functions are expected to partition the dataset in balance which leads maximum entropy and variance for each hash bit. Therefore, hash code variance of each hash table is used as the second part for weighting. The weight of the i^{th} hash table at time T consists of two parts, i.e. the semantic consistency γ_{i} and the corresponding hash code variance δ_i, which is computed as follows:

$$v_i^T = \gamma_i \delta_i \tag{11}$$

Based on the weights of hash tables in the multi-hashing system of ICHDS, weighted Hamming distance between images are computed to evaluate their similarities. Figure 2 shows the overview of ICHDS. The pseudo code of ICHDS at time T = t is described in Algorithm 2.

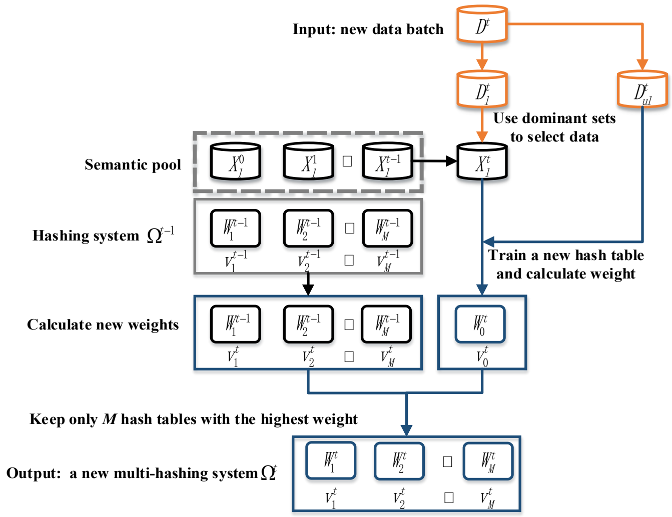


Figure 2. The overview of ICHDS.

Algorithm 2: ICHDS algorithm at time $T = t$.

```
Input: $M$ hash tables, new data batch $D^t$ containing labeled
       dataset $D_l^t$ and unlabeled dataset $D_{ul}^t$, the semantic pool
       $X_l^{t-1}$.
Output: Generate a new multi-hashing system $\Omega^t$.
1. Create the relationship of the edges between the data in the semantic pool $X_l^{t-1}$ and the labeled data $D_l^t$ in new data batch $D^t$.
2. Employ dominant sets clustering according to the similarity of edges.
3. Select representative data using Eq.(7).
4. Update the semantic pool $X_l^t$ using Eq.(8).
5. The latest training set $X^t$ used to train the hash tables is composed of the latest unlabeled dataset $D_{ul}^t$ and the updated semantic pool $X_l^t$ using Eq.(9).
6. Employ semi-supervised BSPLH and latest training set $X^t$ to train a new $K$-bit hash table $W_0^t$.
7. Calculate the weight $v_i^t$ of all $M+1$ hash tables $(W_{\text{i}}^t (i \in \{0,1,2,...,M\}))$ using Eq.(11).
8. Rank the weights and the $M$ hash tables with the highest weight are kept.
9. Generate a new multi-hashing system $\Omega^t$ using the final $M$ hash tables and their associated weights.
```

4. Experiments

The experimental results of the emergence of new classes and distribution drift are shown in Sections 4.1 and 4.2, respectively. Section 4.3 is the parameter selection section about the clustering sensitivity scale factor r in Eq.(5). Comparative experiments are performed based on CIFAR-10, MNIST, and CIFAR-100 image databases to compare the existing hashing methods with ICHDS.

CIFAR-10 dataset consists of 60000 32x32 color images belonging to 10 classes. Each image belongs to one class and is described by a 512-dimensional GIST feature vector. The dataset is originally divided into training set with 50000 images and the test set with 10000 images.

MNIST dataset is 28x28 handwritten digital grayscale images database of 10 classes. Each image is described by a 784-dimensional GIST feature vector. The training set of the MNIST database has 60000 images, and the test set has 10000 images. Each image belongs to a unique label from 0 to 9.

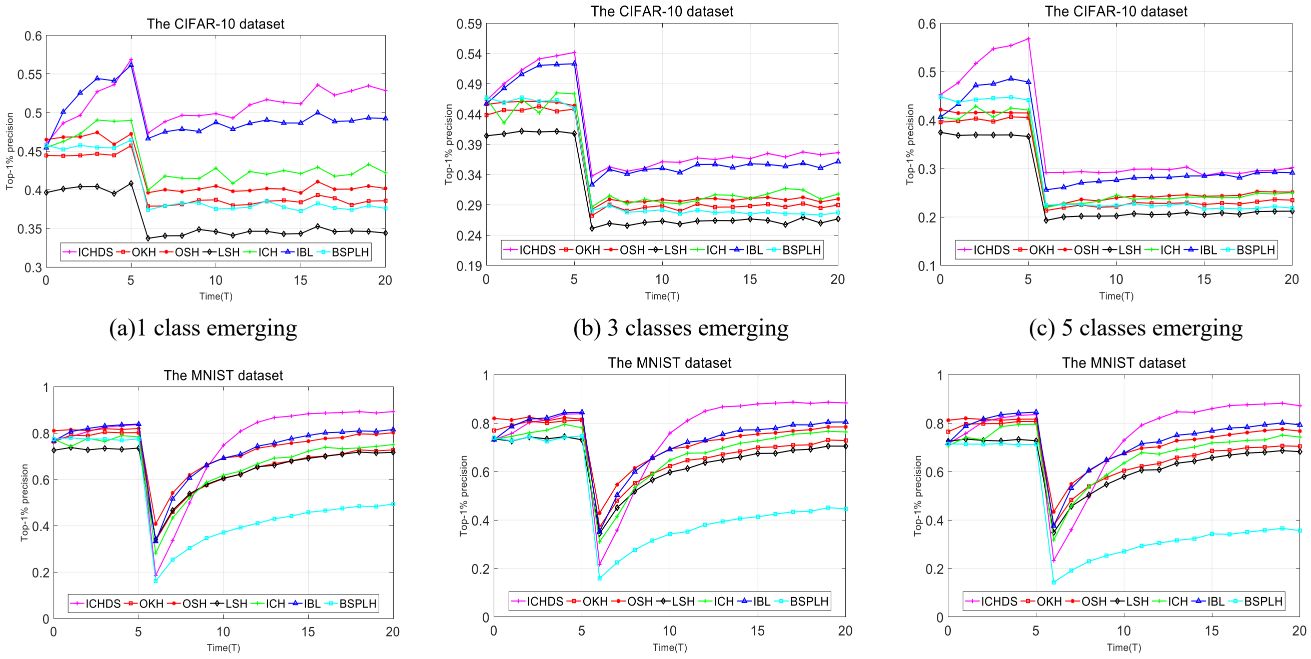
CIFAR-100 dataset is similar to CIFAR-10 dataset, except that it has 100 classes and 600 images per class. Each class has 500 training images and 100 test images. 100 classes with a “fine” label in CIFAR-100 are subordinate to 20 superclasses with a “coarse” label. Each superclass contains 5 subclasses.

The experimental part of ICHDS mainly includes two aspects: the emergence of new classes and the distribution drift of existing classes. The datasets used to simulate data scenarios of the emergence of new classes are CFIAR-10 dataset and MNIST dataset, respectively. These two datasets have in common that they all have different classes from 0 to 9, with a total of 10 classes. The CFIAR-100 dataset is used to simulate data scenarios of distribution drift. According to [30], in the experimental setting aspect, a total of 0 to 20 time steps of data simulations are set. Each data batch randomly selects n images from the dataset. In our experiments, n = 1000. That is to say, 1000 images are selected in each time step as new data batch for training. The training set is used to learn the hash function and construct a hash lookup table. In the setting of the training set, 100 images are randomly selected as the labeled information for training hash function. The remaining images are utilized as unlabeled information for regularization along with the labeled images. Further, the value of the multiple hash tables M is set as 5, and the value of bits K of each hash table is set as 64. In the experiments of ICHDS, the performance of the hashing methods are evaluated using the Top 100 precision and Top 1% precision. The Top 1% precision refers to the accuracy of the top 1% image returned by the ranked image retrieval results. The Top 100 precision is the retrieval result of the top 100 images. In order to reduce the random effects, experiments of concept drift are repeated 5 times. The average of 5 times experiments is calculated as the final experimental result.

4.1 Experiments Involving the Emergence of New Classes

The datasets used to simulate the data scenarios of new classes emerging are the CIFAR-10 and MNIST datasets. The simulation of data scenarios from 0 to 20 time steps is arranged as follows. At time T = 0 to 5, 5 classes are randomly selected from the dataset as the initial dataset for image training. At time T > 5, the remaining 5 classes of these two datasets are randomly selected to simulate the emergence of 1 new class, 3 new classes and 5 new classes, respectively. Figure 3 and Figure 4 show the accuracy of ICHDS and the compared hashing method in Top1% precision and Top100 precision, respectively. Among them, (a) to (c) and (d) to (f) are the results of ICHDS and comparative methods on simulated data scenarios of the CIFAR-10 dataset and the MNIST dataset, respectively.

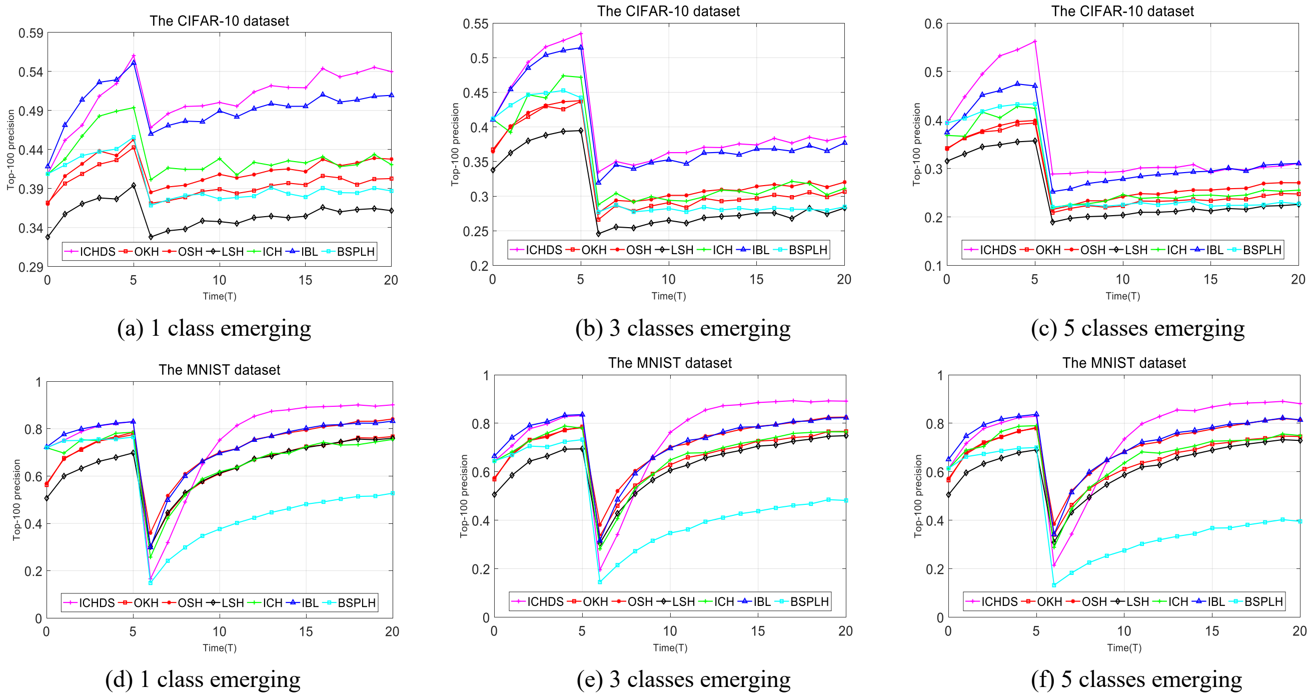
The experimental results show that the classes of new images emerging at time T = 6 reduce the retrieval performance of all hashing methods. This is because all methods have not trained samples of these new classes before. Compared with the CIFAR-10 dataset, the proportion and characteristics of objects in MNIST are different. The noise of the MNIST dataset is small and the image is easy to recognize. Therefore, after the emergence of the new classes, the retrieval performance of all hashing methods on the MNIST dataset increases over time. ICHDS proposed in this paper can adapt to dynamic data changes well and achieve the best retrieval results. Existing dynamic hashing methods IBL, ICH, OKH, and OSH basically have better retrieval effects than the static hashing methods BSPLH and LSH. IBL and ICH, designed to solve the problem of concept drift in a dynamic environment, yield better performance than other hashing methods. Online hashing OKH and OSH dynamically update the hash function. OKH is a supervised hashing method that trains with all label information. Unsupervised OSH uses all unlabeled information for training. Unsupervised static LSH trains the hash function with all unlabeled information. However, the hash function of LSH is only trained at the beginning, without updating over time. The hash function of semi-supervised static BSPLH is trained at the beginning only, using both supervised and unsupervised information of the training set. Hash functions of BSPLH are also not updated over time.



(a)1 class emerging (b) 3 classes emerging (c) 5 classes emerging

(d) 1 class emerging (e) 3 classes emerging (f) 5 classes emerging

Figure 3. Top 1% precision of ICHDS and the comparative hashing methods in experiments involving the emergence of new classes.



(a) 1 class emerging

(b) 3 classes emerging

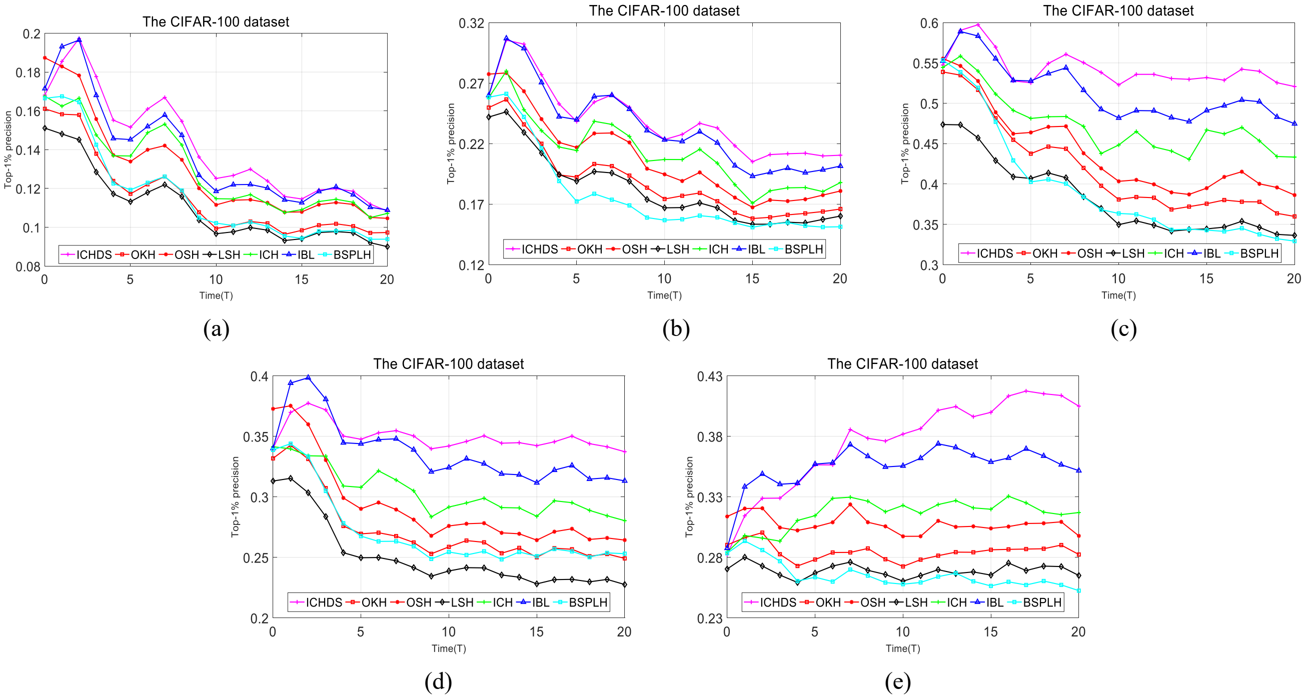
(c) 5 classes emerging

(d) 1 class emerging

(e) 3 classes emerging

(f) 5 classes emerging

Figure 4. Top 100 precision of ICHDS and the comparative hashing methods in experiments involving the emergence of new classes.



(a)

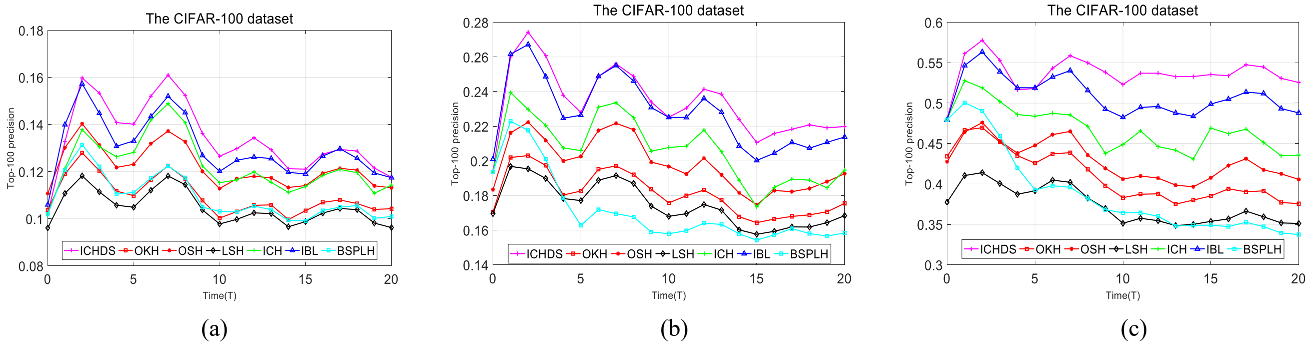
(b)

(c)

(d)

(e)

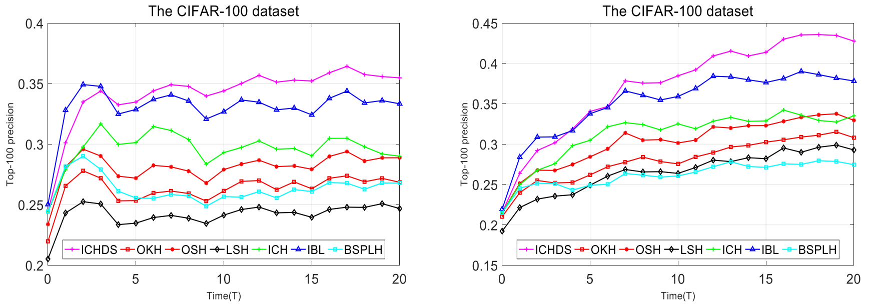
Figure 5. Top 1% precision of ICHDS and the comparative hashing methods in experiments involving distribution drift.



(a)

(b)

(c)



(d) (e)

Figure 6. Top 100 precision of ICHDS and the comparative hashing methods in experiments involving distribution drift.

Table 1 Experimental setting involving distribution drift.

$T = 0$	$T > 0$	Figure
Images randomly selected from 20 superclasses construct the scenario of distribution drift.	Each superclass has distribution drift.	(a)
Images randomly selected from 10 superclasses construct the scenario of distribution drift.	Same as the setting of Figure (a).	(b)
Images randomly selected from 5 superclasses construct the scenario of distribution drift.	Same as the setting of Figure (a).	(c)
Same as the setting of Figure (b).	Only 6 superclasses have distribution drift.	(d)
Same as the setting of Figure (b).	Only 3 superclasses have distribution drift.	(e)

Table 2 Average Top 1% precision of ICHDS and the comparative hashing methods in 11 dynamic data scenarios.

Figure name	**LSH**	**BSPLH**	**OKH**	**OSH**	**ICH**	**IBL**	**ICHDS**
Figure 3 (a)	0.3608±0.0267	0.4006±0.0366	0.4021±0.0294	0.4201±0.0313	0.4358±0.0283	0.4953±0.0266	**0.5105±0.0250**
Figure 3 (b)	0.3038±0.0683	0.3306±0.0847	0.3320±0.0741	0.3435±0.0746	0.3463±0.0732	0.3939±0.0717	**0.4058±0.0715**
Figure 3 (c)	0.2523±0.0762	0.2855±0.1026	0.2764±0.0810	0.2913±0.0814	0.2887±0.0822	0.3309±0.0846	**0.3589±0.1066**
Figure 3 (d)	0.6568±0.1025	0.5059±0.1919	0.6749±0.1210	0.7368±0.1052	0.6757±0.1286	0.7393±0.1242	**0.7623±0.1927**
Figure 3 (e)	0.6475±0.1041	0.4706±0.1877	0.6831±0.1135	0.7326±0.1008	0.6839±0.1257	0.7321±0.1203	**0.7620±0.1826**
Figure 3 (f)	0.6360±0.1018	0.4153±0.2011	0.6722±0.1128	0.7253±0.0997	0.6775±0.1170	0.7309±0.1138	**0.7512±0.1785**
Figure 5 (a)	0.1103±0.0192	0.1159±0.0248	0.1157±0.0215	0.1298±0.0263	0.1292±0.0214	0.1376±0.0270	**0.1413±0.0268**
Figure 5 (b)	0.1824±0.0294	0.1772±0.0358	0.1889±0.0298	0.2077±0.0349	0.2124±0.0288	0.2333±0.0336	**0.2396±0.0293**
Figure 5 (c)	0.3822±0.0461	0.3941±0.0702	0.4188±0.0578	0.4395±0.0537	0.4739±0.0375	0.5143±0.0346	**0.5428±0.0211**
Figure 5 (d)	0.2506±0.0280	0.2707±0.0311	0.2726±0.0291	0.2922±0.0358	0.3041±0.0195	0.3371±0.0255	**0.3496±0.0108**
Figure 5 (e)	0.2690±0.0051	0.2652±0.0108	0.2845±0.0067	0.3077±0.0071	0.3159±0.0128	0.3548±0.0184	**0.3754±0.0378**

4.2 Experiments Involving Distribution Drift

In the dynamic data environment, the image distribution of existing classes may change. The data scenarios that simulate changes in the distribution of existing classes use the CIFAR-100 dataset. CIFAR-100 is a large dataset with 20 superclasses which divided into 100 subclasses. Each of these 5 subclasses is classified as a superclass. The distribution and characteristic between the 5 subclasses in a superclass are somewhat different. In actual situations, not all superclasses of images may have a distribution drift. Therefore, 5 kinds of data scenarios of distribution drift are simulated. The distribution of superclass data in the experiment varies from T = 0 to 20 over time. The images of the subclasses in the superclasses change according to the ratio in [30]. Table 1 describes the experimental setting of Figures 5 and 6. Figures 5 and 6 show the experimental results of the Top 1% precision and the Top 100 precision of the distribution drift, respectively.

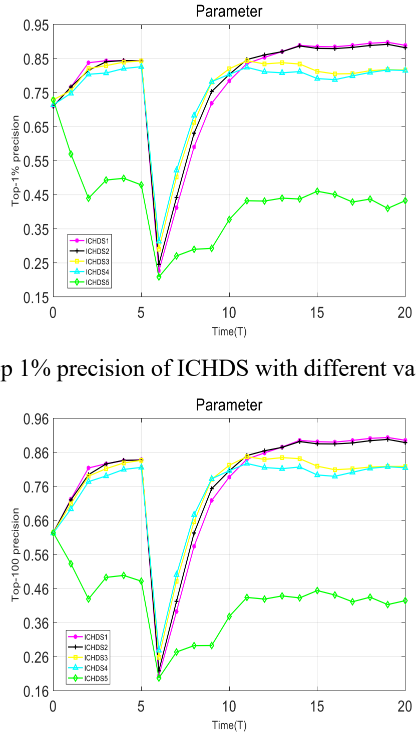
In a dynamic data environment, there are scenarios in which only the data distribution of existing classes changes and data of new classes does not emerge. Therefore, the semantic information learned in the early stage can still adapt to the new data environment changes. When the distribution drift occurs, the image retrieval performances of all hashing methods fluctuate significantly. In the presence of distribution drift, ICHDS is superior to all other methods. IBL and ICH methods, which continuously update the hash function with new data batch, take good results. Dynamic OKH and OSH that can update the hash function also take better image retrieval performance. Static unsupervised LSH and semi-supervised BSPLH using single-table training have the worst retrieval performance. Therefore, static hashing methods cannot adapt to the current changing data environment.

When new data emerges, ICHDS not only considers the semantic information of data in the current batch, but also judges the semantic difference between data in the current batch and data in the old batch by clustering. Redundant data is no longer reused for training, which greatly reduces the difficulty of classifying new samples. The current data distribution is described by selecting representative data to improve image retrieval efficiency. Therefore, the retrieval performance of ICHDS is better than ICH.

In addition, we calculated the average Top 1% precision in 11 dynamic scenarios as shown in Table 2, in which the mean and standard deviation of the Top 1% precision value over all time steps are calculated. According to the average results of the Top 1% precision, ICHDS is superior to all state-of-the-art hashing methods.

4.3 Parameter Selection Experiment

In Eq.(5), $r > 0$ is a cluster-sensitive scale factor. Therefore, the experiment of parameter selection is mainly to find a suitable value of r. In the Euclidean space, when the the i^{th} image is very similar to the j^{th} image, the Euclidean distance of the two images is smaller. The larger r is, the smaller the difference in the relationship matrix a_{ij} will be. However, the size of the weights of all edges in dominant sets represents that the edge of the more similar images has a larger weight. This paper clusters samples in the same class to select representative samples. The Euclidean distances of the samples in the same class are not much different. Therefore, the difference of the relationship matrix a_{ij} is small when $r > 2$. In the experimental parameter adjustment, the verification of the 5 values from 0 to 2, which are 0.2, 0.4, 0.8, 1, and 2, respectively. In Figure 7, the ICHDS1 to ICHDS5 in the experimental results correspond to the above five parameters, respectively. The experimental results show that r has a good effect between 0 and 1, and $r = 0.4$ is set in this paper.



(a) Top 1% precision of ICHDS with different values of *r*.

(b) Top 100 precision of ICHDS with different values of *r*.

Figure 7. The experimental results of ICHDS with different values of *r* on MNIST dataset involving the emergence of 1 new class.

5. Conclusion

In this paper, we propose a semi-supervised incremental image hashing method, i.e. ICHDS, to improve the accuracy and efficiency of image retrieval in large-scale dynamic environments. ICHDS can effectively solve the problem of concept drift that occurs in dynamic data environments, including the emergence of new classes and distribution drift of existing classes. In ICHDS, representative samples are selected by dominant sets clustering which can provide more comprehensive data distribution information of each class in image collections for the training of hash tables. Experimental results in a variety of data scenarios show that our method is better than existing dynamic hashing and static hashing methods.

The selected representative samples are employed to train the new hash table in ICHDS. Moreover, weights of hash tables in multi-hashing system are calculated by their performance to the current data environment. It is important that the selected representative samples provide comprehensive distribution information for each class. Therefore, in future work more efficient algorithms for image selection will be studied in order to reduce training time and improve retrieval performance. It would make sense to design a more efficient weighting scheme to improve the stability of the ICHDS method. The proposed method can be further extended to various scenarios with dynamic environments, and the possibility of optimization during binary code learning is investigated.

Acknowledgement

This work was supported in part by the National Natural Science Foundation of China under Grant 61876066, Grant 61572201, Grant 61772344, and Grant 61672443, in part by the Guangzhou Science and Technology Plan Project under Grant 201804010245, and EU Horizon 2020 Programme (700381, ASGARD), and in part by the Hong Kong RGC General Research Funds under Grant 9042489 (CityU 11206317), Grant 9042816 (CityU 11209819) and Grant 9042322 (CityU 11200116).

Reference

[1] L. Hong, R. R. Ji, J. D. Wang, C. H. Shen. 'Ordinal Constraint Binary Coding for Approximate Nearest Neighbor Search'. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 41/no. 4, (2019), pp. 941-955.

[2] D. Mayur, P. Indyk, N. Immorlica, V. S. Mirrokni. 'Locality-Sensitive Hashing Scheme Based on P-Stable Distributions'. Proceedings of the Annual Symposium on Computational Geometry, (2004), pp. 253-262.

[3] K. Brian, K. Grauman. 'Kernelized Locality-Sensitive Hashing'. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 34/no. 6, (2012), pp. 1092-1104.

[4] W. Yair, A. Torralba, R. Fergus. 'Spectral Hashing'. International Conference on Neural Information Processing Systems, (2009), pp. 1753-1760.

[5] Y. C. Gong, L. Svetlana, G. Albert, P. Florent. 'Iterative Quantization: A Procrustean Approach to Learning Binary Codes for Large-Scale Image Retrieval'. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 35/no. 12, (2013), pp. 2916-2929.

[6] F. M. Shen, W. Liu, S. T. Zhang, Y. Yang, H. T. Shen. 'Learning Binary Codes for Maximum Inner Product Search'. IEEE International Conference on Computer Vision, (2015), pp. 4148-4156.

[7] X. L. Liu, Y. D. Mu, D. C. Zhang, B. Lang, X. L. Li. 'Large-Scale Unsupervised Hashing with Shared Structure Learning'. IEEE Transactions on Cybernetics, vol. 45/no. 9, (2015), pp. 1811-1822.

[8] M. Lopamudra, R. Sathya N, I. Vamsi K, H. Tyler, S. Vikas. 'An NMF Perspective on Binary Hashing'. IEEE International Conference on Computer Vision, (2015), pp. 4184-4192.

[9] X. L. Liu, L. Huang, C. Deng, B. Lang, D. C. Tao. 'Query-Adaptive Hash Code Ranking for Large-Scale Multi-View Visual Search'. IEEE Transactions on Image Processing, vol. 25/no. 10, (2016), pp. 4514-4524.

[10] W. Liu, J. Wang, R. R. Ji, Y. G. Jiang, S. F. Chang. 'Supervised Hashing with Kernels'. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, (2012), pp. 2074-2081.

[11] D. J. Song, W. Liu, R. R. Ji, M. David A, S. John R. 'Top Rank Supervised Binary Coding for Visual Search'. IEEE International Conference on Computer Vision, (2015), pp. 1922-1930.

[12] F. M. Shen, C. H. Shen, W. Liu, H. T. Shen. 'Supervised Discrete Hashing'. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, (2015), pp. 37-45.

[13] J. Gui, T. L. Liu, Z. N. Sun, D. C. Tao, T. N. Tan. 'Fast Supervised Discrete Hashing'. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 40/no. 2, (2018), pp. 490-496.

[14] K. W. Cheng, W. J. Li, Z. H. Zhou. 'Column Sampling Based Discrete Supervised Hashing'. Thirtieth AAAI Conference on Artificial Intelligence, (2016), pp. 1230-1236.

[15] Q. Y. Jiang, X. Cui, W. J. Li. 'Deep Discrete Supervised Hashing'. IEEE Transactions on Image Processing, vol. 27/no. 12, (2018), pp. 5996-6009.

[16] J. Wang, S. Kumar, S. F. Chang. 'Semi-Supervised Hashing for Large-Scale Search'. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 34/no. 12, (2012), pp. 2393-2406.

[17] J. Wang, S. Kumar, S. F. Chang. 'Sequential Projection Learning for Hashing with Compact Codes'. Proceedings of the International Conference on Machine Learning, (2010), pp. 1127-1134.

[18] C. X. Wu, J. K. Zhu, D. Cai, C. Chen, J. J. Bu. 'Semi-Supervised Nonlinear Hashing using Bootstrap Sequential Projection Learning'. IEEE Transactions on Knowledge and Data Engineering, vol. 25/no. 6, (2013), pp. 1380-1393.

[19] P. Li, J. Cheng, H. Q. Lu. 'Hashing with Dual Complementary Projection Learning for Fast Image Retrieval'. Neurocomputing, vol. 120, (2013), pp. 83-89.

[20] C. H. Zhang, W. S. Zheng. 'Semi-Supervised Multi-View Discrete Hashing for Fast Image Search'. IEEE Transactions on Image Processing, vol. 26/no. 6, (2017), pp. 2604-2617.

[21] W. W. Y. Ng, X. C. Zhou, X. Tian, X. Z. Wang, D. S. Yeung. 'Bagging-Boosting-Based Semi-Supervised Multi-Hashing with Query-Adaptive Re-Ranking'. Neurocomputing, vol. 27, (2018), pp. 916-923.

[22] J. Zhang, Y. X. Peng. 'SSDH: Semi-Supervised Deep Hashing for Large Scale Image Retrieval'. IEEE Transactions on Circuits and Systems for Video Technology, vol. 29/no. 1, (2019), pp. 212-225.

[23] L. K. Huang, Q. Yang, W. S. Zheng. 'Online Hashing'. International Joint Conference on Artificial Intelligence, (2013), pp. 1422-1428.

[24] L. K. Huang, Q. Yang, W. S. Zheng. 'Online Hashing'. IEEE Transactions on Neural Networks and Learning Systems, vol. 29/no. 6, (2018), pp. 2309-2322.

[25] F. Cakir K. He, S. A. Bargal, S. Sclaroff. 'MIHash: Online Hashing with Mutual Information'. IEEE International Conference on Computer Vision, (2017), pp. 437-445.

[26] C. Leng, J. X. Wu, J. Cheng, X. Bai, H. Q. Lu. 'Online Sketching Hashing'. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 7-12, (2015), pp. 2503-2511.

[27] F. Cakir, S. Sclaroff. 'Online Supervised Hashing'. IEEE International Conference on Image Processing, vol. 2015, (2015), pp. 2606-2610.

[28] F. Cakir, S. Sclaroff. 'Adaptive Hashing for Fast Similarity Search'. IEEE International Conference on Computer Vision, (2015), pp. 1044-1052.

[29] W. W. Y. Ng, X. Tian, W. Pedrycz, X. Z. Wang, D. S. Yeung. 'Incremental Hash-Bit Learning for Semantic Image Retrieval in Nonstationary Environments'. IEEE Transactions on Cybernetics, vol. 49/no. 11, (2019), pp. 3844-3858.

[30] W. W. Y. Ng, X. Tian, Y. M. Lv, D. S. Yeung, W. Pedrycz. 'Incremental Hashing for Semantic Image Retrieval in Nonstationary Environments'. IEEE Transactions on Cybernetics, vol. 47/no. 11, (2017), pp. 3814-3826.

[31] M. Pavan, M. Pelillo. 'A New Graph-Theoretic Approach to Clustering and Segmentation'. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, (2003), pp. I-I.

[32] M. Pavan, M. Pelillo. 'Dominant Sets and Pairwise Clustering'. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 29/no. 1, (2007), pp. 167-172.

[33] W. W. Y. Ng, Y. M. Lv, Z. Q. Zeng, D. S. Yeung, P. P. K. Chan. 'Sequential Conditional Entropy Maximization Semi-Supervised Hashing for Semantic Image Retrieval'. International Journal of Machine Learning and Cybernetics, vol. 8/no. 2, (2017), pp. 571-586.

[34] W. W. Y. Ng, J. C. Li, S. Y. Feng, D. S. Yeung, P. P. K. Chan. 'Sensitivity Based Image Filtering for Multi-Hashing in Large Scale Image Retrieval Problems'. International Journal of Machine Learning and Cybernetics, vol. 6/no. 5, (2015), pp. 777-794.

[35] Z. Y. Weng, Y. S. Zhu. 'Online Supervised Sketching Hashing for Large-Scale Image Retrieval'. IEEE Access, vol. 7, (2019), pp. 88369-88379.

[36] X. X. Jiang, W. W. Y. Ng, X. Tian, S. Kwong, H. Wang. 'Incremental Hashing with Undersampling'. IEEE International Conference on Systems, Man, and Cybernetics, (2019), early access.