A Framework for Image Dark Data Assessment

Yu Liu[1], Yangtao Wang[1], Ke Zhou[1(✉)], Yujuan Yang[1], Yifei Liu[1], Jingkuan Song[2], and Zhili Xiao[3]

[1] Huazhong University of Science and Technology, Wuhan, China
{liu_yu,ytwbruce,k.zhou,gracee,yifeiliu}@hust.edu.cn
[2] University of Electronic Science and Technology of China, Chengdu, China
jingkuan.song@gmail.com
[3] Tencent Inc., Shenzhen, China
tomxiao@tencent.com

Abstract. Blindly applying data mining techniques on image dark data whose content and value are not clear, is highly likely to bring undesired result. Therefore, we propose an assessment framework which includes offline and online stages for image dark data. In offline stage, we first transform images into hash codes by Deep Self-taught Hashing (DSTH) algorithm, then construct a semantic graph, and finally use our designed Semantic Hash Ranking (SHR) algorithm to calculate the importance score. During online stage, we first translate the user's query into hash codes, then match the suitable data contained in the dark data, and finally return the weighted average value of these matched data to help the user cognize the dark data. The results on real-world dataset show our framework can apply to large-scale datasets, help the user conduct subsequent data mining work.

Keywords: Image dark data · Deep self-taught hashing · Semantic hash ranking · Assessment

1 Introduction

Dark data is defined as the information assets that can be easily collected and stored, but generally fail to use for data analytics and mining[1]. Most of these data are unstructured data represented by images. Many social platforms store image data (i.e., albums and chat images) as an independent resource separated from other businesses. These massive image data quickly turn into dark data, which contain lots of historical records and thus are not allowed to be removed. However, they consistently occupy the storage space but can not produce great value.

[1] https://www.gartner.com/it-glossary/dark-data/.

The original version of this chapter was revised: The abstract section and the keywords of this chapter have been exchanged. This have been now corrected. The correction to this chapter is available at https://doi.org/10.1007/978-3-030-26072-9_31

J. Shao et al. (Eds.): APWeb-WAIM 2019, LNCS 11641, pp. 3–18, 2019.
https://doi.org/10.1007/978-3-030-26072-9_1

Therefore, developers are eager to mine image dark data in order to improve the cost performance of storage. However, owing that the image dark data lack labels and associations, owners have no idea how to apply these data. For a given target, blindly conducting data mining techniques on the dark data is highly to get little feedback. Faced with image dark data whose content and value are not clear, the primary issue is to judge whether this dataset are worth mining or not. Therefore, it is of great significance to evaluate the value of image dark data according to the user's requirement. Given this, faced with the user's requirement, which way shall be taken to make the user aware of the dark data? There exist following challenges when executing association analysis on dark data.

(1) How to extract semantic information with generalization ability? Reasonable semantic extraction method is the key to correctly understand the user's requirement and analyze semantic distribution of dark dataset. An excellent deep model will obtain desired classification effects. However, deep model suffers from a poor generalization ability when extracting semantic information for unknown samples. Our goal is to express semantic distance for different images including unknown samples. For example, when training the semantic model, we make the semantic feature of a cat similar to that of a dog but different from that of an airplane.

(2) How to evaluate relevance? There are two-level evaluation for the relevance: (1) the amount of data that meets the user's requirements; (2) the matching degree of these relevant data. Traditional clustering methods need many iterations and will take a long time to complete the evaluation. In addition, graph-based computing [4] is another way to achieve global evaluation. The most well-known one is PageRank algorithm [14], which determines the importance of web pages according to the links. However, PageRank can only express directional attributes on a directed graph, so it fails to measure the mutual extent of relevance between objects. Events detecting [2] can find the hot events though connections on an undirected graph, but the representative data are very limited. Once the query data can not match any hot event, no assessment result will be returned, so it does not apply to our task.

(3) How to reduce the query cost? Even if the above problems have been solved, we still need to find the corresponding related data in the whole dataset to measure the feedback of the query. Online computing millions of high-dimensional floating-point vectors means a huge resource consumption. Besides, the assessment task will receive frequent query requests for different mining tasks. Thus, the assessment work is supposed to be built on more efficient distance measurement for practical feasibility.

In this paper, we propose an assessment framework for image dark data. The framework consists of four parts. First, we use deep self-taught hashing (DSTH) algorithm to transform unlabeled images into deep semantic hash codes with generalization ability. Second, we build the semantic undirected graph using restricted Hamming distance. According to what DCH [3] describes, we cut off a lot of unreasonable connections and improve the efficiency of construction and

subsequent calculation on graph. Third, on the built graph, we design semantic hash ranking (SHR) algorithm to calculate the importance score for each node by random walk and obtain the rank for each image. It is worth mentioning that we improve the PageRank algorithm and extend it to undirected weighted graph, which takes both the number of connected links and the weight on edges into consideration. At last, according to the user's requirements, we match the corresponding data contained in the dataset which are restricted within a given Hamming distance range, calculate the weighted semantic importance score of these data, and return the ranking. The user can decide whether conducting data mining on this dark dataset based on the returned result. The major contributions of this paper are summarized as follows:

- We design a deep self-taught hashing (DSTH) algorithm, which can extract semantic features without labels and solve the out-of-sample problem.
- Based on the built semantic graph, we propose a semantic hash ranking (SHR) algorithm to calculate the overall importance score for each node (image) according to random walk, which takes both the number of connected links and the weight on edges into consideration.
- We propose an analysis-query-assessment framework including offline calculation and online assessment, which applies to assessing large-scale datasets.
- Our framework can help the user to detect the potential value of the dark data, avoid unnecessary mining cost. To the best of our knowledge, this is the first attempt that assesses image dark data.

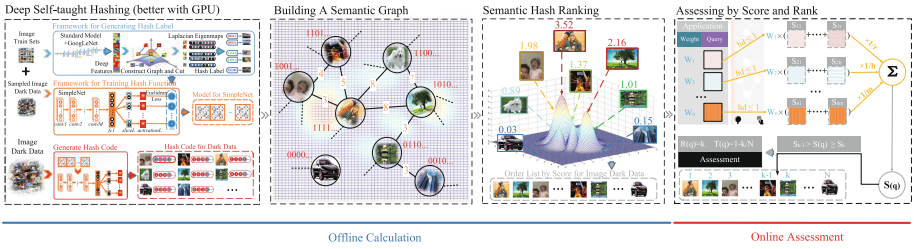


Fig. 1. The framework for image dark data assessment.

2 Design Overview

For a large-scale image dark dataset, in order to make our assessment framework effective for real-time analysis, we perform offline analysis on the dataset to get the score (rank) of each image. Then given an online matched request, we evaluate whether the dataset can be used for the query on-the-fly. As shown in Fig. 1, the framework consists of four steps. The first three steps give an offline evaluation on the dark dataset, which calculates the importance score (rank) for each image. The last step provides an online matching and weighted computing for the user, which returns the ranking score of the user's requirement based on the computed score.

Offline Evaluation. We design three steps to effectively calculate the semantic importance score and provide each image with a rank. Formally, we first train a Deep Self-taught Hashing (DSTH) model and transform all images into hash codes, then build a semantic undirected graph with restricted Hamming distance, and finally calculate the overall importance score (rank) for each image by our designed Semantic Hash Ranking (SHR) algorithm.

(1) Step 1: Deep Self-taught Hashing. As shown in the first frame of Fig. 1, we adopt the DSTH algorithm to encode each image of the dark dataset. DSTH contains three stages: hash label acquiring stage, hash function training stage and hash code generating stage. First, it is important to acquire hash labels, because the premise of feature extraction using deep learning is based on semantic labels. We choose ImageNet and the same amount of sampled image dark data as the training data and GoogLeNet trained on ImageNet to extract semantic features of these data. Next, we use the features to construct a graph via K-NN ($K = 12$), then map data to predefined l-dimensional space by means of Laplacian Eigenvalue decomposition (LE), and finally binarize all data to generate hash labels. We conduct clustering on extracted semantic features, which not only preserves original semantic classification information but also makes these semantics automatically closer or estranged according to the similarity between themselves (challenge (1) in Sect. 1). Those labels have the semantics with generalization ability, which directly affects the next hash function learning. Note that the hash function is specially trained on above sampled dark data. At last, according to the obtained deep hash functions, we transform each image of the dark dataset into a hash code which represents the semantic feature of the data. Our method (DSTH) converts high-dimensional dark data into low-dimensional hash vectors that can be easily but fast measured. The mathematical expression of DSTH and the advantages are described in Sect. 3.

(2) Step 2: Semantic Graph Construction. As shown in the second frame of Fig. 1, we model the images as a graph G where each node is an image and edges are relationships between images. In order to speed up the graph construction, we cut off those edges on which the weight exceeds half of the length of hash code, according to the conclusion of Long [3]. Let N_* denote the $*$-th node of G, $H(N_*)$ denote hash code of N_* and l denote length of hash codes. We define XOR operation as $\oplus$. Therefore, the Hamming distance weight on the undirected link between N_i and N_j can be defined as

$$d_{ij} = \begin{cases} H(N_i) \oplus H(N_j) & i \neq j, H(N_i) \oplus H(N_j) \leq \Omega, \\ NULL & otherwise. \end{cases} \tag{1}$$

where $\Omega = \lceil l/s \rceil$ and $s \in [1, l]$. In practice, the determination of Ω is based on efficiency of building a graph with tolerable loss. Formally, we define the precision of i-th node as C_i/L_i, where L_i represents the number of all nodes connected to i-th node and there exist C_i nodes of the L_i nodes that have the same label as the i-th node. Therefore, the precision of graph $P(G|\Omega)$ is defined as

$$P(G|\Omega) = \frac{1}{N}\sum_{i=1}^{N}\frac{C_i}{L_i} \tag{2}$$

(3) Step 3: Semantic Hash Ranking. As shown in the third frame of Fig. 1, after building the graph with restricted Hamming distance in *Step 2*, we calculate the importance score for each node by random walk in order to obtain the overall objective evaluation value. We extend the PageRank algorithm and propose the SHR algorithm which takes both the number of connected links and the weight on edges into consideration (challenge (2) in Sect. 1). Note that we specially design how to reasonably calculate the extent of relevance between nodes, aiming at making full use of the Hamming distance of similarity hash with generalization ability. On the built semantic graph, we use SHR to calculate the importance score for each node. At the same time, according to the physical meaning of Hamming distance, we redesign the iteration matrix elements for obtaining reasonable importance scores. SHR makes the dominant semantics more prominent, thus reinforcing the user's cognition of the dark dataset. We introduce the detailed calculation process of SHR in Sect. 4.

Online Query Assessment. As shown in the last frame of Fig. 1, for the image dark data consisting of N images, the query will be mapped to hash codes by hash function calculated in Sect. 3 and match images contained in the dark data (challenge (3) in Sect. 1). The matching range is defined as hd and we set $hd = 1$ to conduct matching shown in the last frame of Fig. 1. Mathematically, let q denote a query with n images, img_i denote the i-th image where $i \in [1, n]$, m_i denote the number of matched images for the i-th image of the query q. Meanwhile, let $S_j(img_i)$ denote the score of the j-th image where $j \in [1, m_i]$. Therefore, the score of q is defined as follows:

$$S(q) = \sum_{i=1}^{n}\frac{1}{m_i}\sum_{j=1}^{m_i}\beta_i S_j(img_i) \quad s.t. \sum_{i=1}^{n}\beta_i = 1 \tag{3}$$

where $\beta_i \in [0, 1]$ represents the importance weight of the i-th image.

Compared with the ranked scores denoted as $\{S_1, S_2, ..., S_N\}$ of image dark data calculated by SHR, we can acquire the rank of $S(q)$ denoted as k, where $S_{k-1} > S(q) \geq S_k$. Further, $T(q) = 1 - \frac{k}{N}$ represents importance degree of image dark data for the requirement. As a result, the user can decide whether the image dark data are worth fine-mining.

3 Deep Self-taught Hashing (DSTH)

In this section, we detailedly describe DSTH algorithm including how to integrate clustering information into semantic learning under deep learning framework, how to generate hash labels, and how to conduct the training process. And then, we elaborate on the advantages of DSTH.

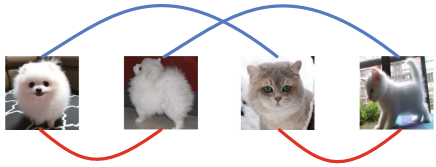


Fig. 2. Red line represents the deep classification result based on classification labels, while blue line represents the classification result by LE. (Color figure online)

3.1 DSTH Algorithm

The algorithm mainly contains hash label generating stage and hash function training stage.

(1) Hash Label Generating Stage. The prime task of DSTH is to acquire semantic labels, because labels determine which semantic informations should be extracted from data and directly affect the subsequent function learning. Semantic labels acquiring aims at extracting semantic information and semantic feature with generalization ability. Supervised deep learning algorithm is able to better extract semantic information owing to the accurate hand-crafted labels denoted as red line in Fig. 2. LE algorithm extracts semantics according to the similarity between data themselves, which applies to those scenes without hand-crafted labels denoted as blue line in Fig. 2. Our method combines these two advantages, which can not only obtain semantic information without labels but also reach the balance between human semantic cognition and data semantic cognition, so as to acquire our expected semantic features (labels).

We apply the deep and shallow mixed learning method, which integrates clustering information and improves the generalization ability of feature extraction. In the absence of labels, we use the trained deep model to get features. After that, we use LE method and binarization to transform the extracted deep features to hash codes which serve as hash labels for next stage. Mathematically, we use n m-dimensional vectors $\{x_i\}_{i=1}^n \in \mathbb{R}^m$ to denote the image features and use $ED(i,j) = ||x_i - x_j||_2^2$ to denote the Euclidean distance between i-th and j-th image. For θ_t ($t \in [1, n-1]$), we denote $\{ED(i,\theta_1), ED(i,\theta_2), ..., ED(i,\theta_{n-1})\} = \{ED(i,1), ED(i,2), ..., ED(i,i-1), ED(i,i+1), ..., ED(i,n)\}$, where $ED(i,\theta_t) \leq ED(i,\theta_{t+1})$. We define that $TK_{ED}(i,j) = t$, if $ED(i,j) = ED(i,\theta_t)$. Further, we use $N_K(i,j)$ to denote neighbor relationship between i-th and j-th data, which is defined as

$$N_K(i,j) = \begin{cases} True & TK_{ED}(i,j) \leq K, \\ False & TK_{ED}(i,j) > K. \end{cases} \tag{4}$$

Next, we use x_i and y_i to represent the i-th sample and its hash codes where $y_i \in \{0,1\}^\gamma$ and γ denotes the length of hash codes. We set $y_i^\rho \in \{0,1\}$ as the ρ-th element of y_i. The hash code set for n samples can be represented as $[y_1, \ldots, y_n]^T$. Our $n \times n$ local similarity matrix W is

$$W_{ij} = \begin{cases} 0 & \text{if } N_K(i,j) \text{ is False,} \\ \dfrac{x_i^T x_j}{\|x_i\| \cdot \|x_j\|} & \text{otherwise.} \end{cases} \tag{5}$$

Furthermore, we apply diagonal matrix

$$D_{ii} = \sum_{j=1}^{n} W_{ij} \tag{6}$$

Meanwhile, we use the number of different bits for calculating Hamming distance between y_i and y_j as

$$H_{ij} = \|y_i - y_j\|^2/4 \tag{7}$$

We define an object function ζ to minimize the weighted average Hamming distance.

$$\zeta = \sum_{i=1}^{n}\sum_{j=1}^{n} W_{ij}H_{ij} \tag{8}$$

To calculate ζ, we transform it to $\xi = tr(Y^T LY)/4$, where $L = D - W$ is Laplacian matrix and $tr(\cdot)$ means trace of matrix. At last, we transform ξ to LapEig problem ψ with slacking constraint $y_i \in \{0,1\}^t$, and obtain the optimal t-dimensional real-valued vector $\tilde{y}$ to represent each sample. ψ is the following:

$$\psi = arg \min_{\tilde{Y}} \quad Tr(\tilde{Y}^T L\tilde{Y}) \quad s.t. \begin{cases} \tilde{Y}^T D\tilde{Y} = I \\ \tilde{Y}^T D1 \ = 0 \end{cases} \tag{9}$$

where $Tr(\tilde{Y}^T L\tilde{Y})$ gives the real relaxation of the weighted average Hamming distance $Tr(T^T LY)$. The solution of this optimization problem is given by $\tilde{Y} = [v_1, \ldots, v_t]$ whose columns are the t eigenvectors corresponding to the smallest eigenvalues of following generalized eigenvalue problem. The solution of ψ can be transformed to $Lv = \lambda Dv$ where vector v is the t eigenvectors which are corresponding to the t smallest eigenvalues (nonzero).

Then, we convert the t-dimensional real-valued vectors $\tilde{y_1}, \ldots, \tilde{y_n}$ into binary codes according to the threshold. We set δ^p to present threshold and $\tilde{y_i^p}$ equivalent to p-th element of $\tilde{y_i}$. The hash label as final result value of y_i^p is

$$y_i^p = \begin{cases} 1 & \tilde{y_i^p} \geqslant \delta^p, \\ 0 & otherwise. \end{cases} \quad where \quad \delta^p = \frac{1}{n}\sum_{i=1}^{n} \tilde{y_i^p} \tag{10}$$

(2) Hash Model Training Stage. We implement an end-to-end hashing deep learning module. Firstly, we employ CNNs again to receive fine-grained features. After that, we adopt encoding module which is Divide and Encode Module [9] associated with activation function of BatchNorm [7] to approximate hash labels

generated in previous stage. The learning framework is the artificial neural network on the multi-output condition. Formally, we set a function $f : \mathbb{R}^I \rightarrow \mathbb{R}^O$, where I is the input set, O is the output set and x is the input vector. The formulation is

$$f^{(k)} = \begin{cases} \varphi(W^{(k)}x + b^{(k)}) & k = 1, \\ \varphi(W^{(k)}f^{(k-1)} + b^{(k)}) & k = 2, ..., K. \end{cases} \tag{11}$$

where b is the bias vector, W is the weight matrix of convolution and $\varphi(*)$ is ReLU and BatchNorm function. When the core of $\varphi(x)$ is BatchNorm, the function is calculated as follows:

$$\tilde{x}^{(k)} = \frac{x^{(k)} - E\left(x^{(k)}\right)}{\sqrt{Var\left(x^{(k)}\right)}} \tag{12}$$

where

$$E\left(x\right) = \frac{1}{m}\sum_{i=1}^{m} x_i \quad Var\left(x\right) = \frac{1}{m}\sum_{i=1}^{m}\left(x_i - E\left(x\right)\right)^2 \tag{13}$$

In the last layer of CNN, we split a 1024-dimensional vector into 16 groups, and each group is mapped to z elements. The output number $16 \times z$ is the hash code length. Denote the output as one $m \times d$ matrix (m is the number of samples in batch and d is the number of output in last full connection layer), x is the output vector, y is the corresponding label. We define the loss function as follows:

$$F\left(x\right) = min\sum_{i=1}^{m}\sum_{j=1}^{d}\left\|x_i^{(j)} - y_i^{(j)}\right\|_2^2 \tag{14}$$

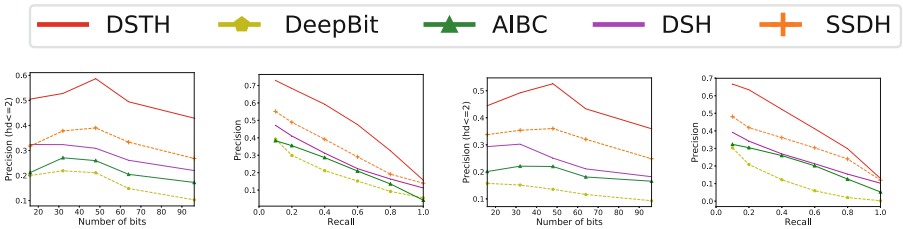
At last, we define the threshold as the same as Eq. (10). Usually, we apply the threshold values of each bit calculated in the hash label generating stage.

4 Semantic Hash Ranking (SHR)

In this section, we introduce SHR algorithm in detail, which considers both the number of connected links and the weight on edges into consideration, designs impact factor between different nodes according to hash code, and calculate the importance score for each node by random walk. Let L_* denote number of links to N_*. Draw rank factor $R(N_*)$ for N_* and impact factor $I(N_{ij})$ for N_j to N_i, where $I(N_{ij})$ is defined as

$$I(N_{ij}) = \begin{cases} \dfrac{l - d_{ij}}{\sum\limits_{t \in T_j} l - d_{tj}} R(N_j) & \exists d_{ij}, \\ 0 & otherwise. \end{cases} \tag{15}$$

where T_j is the set including orders of all nodes associated with N_j. Specially, we design the formulation according to two principals. Firstly, the less d_{ij} is,



(a) mAP@5% CLA without Cat (b) 48-bit codes PR Training without Cat (c) mAP@5% CLA Training without Cat and Automobile (d) 48-bit codes PR Training without Cat and Automobile

Fig. 3. mAP@5% Code Length Analysis (CLA) and Precision-Recall (PR) curve on CIFAR-10.

the greater influence N_j contributes to N_i is. Meanwhile, the longer hash code is, the more compact the similarity presented by d_{ij} is. Secondly, PageRank considers the weights on each edge as the same, but we extend it to be applied to different weights on edges. As a result, when weights on different edges are the same, Eq. (15) should be the same as the impact factor formulation of PageRank. Consequently, $R(N_i)$ should be equal to the sum of the impact factors of all nodes linked to N_i

$$R(N_i) = \sum_{j=1, j\neq i}^{n} I(N_{ij}) \tag{16}$$

Let f_{ij} denote the coefficient of $R(N_j)$ in $I(N_{ij})$. We draw iteration formula as

$$\begin{bmatrix} R^{c+1}(N_1) \\ R^{c+1}(N_2) \\ \cdots \\ R^{c+1}(N_n) \end{bmatrix} = \begin{bmatrix} 0 & f_{12} & \cdots & f_{1n} \\ f_{21} & 0 & \cdots & f_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ f_{n1} & f_{n2} & \cdots & 0 \end{bmatrix} \begin{bmatrix} R^{c}(N_1) \\ R^{c}(N_2) \\ \cdots \\ R^{c}(N_n) \end{bmatrix} \tag{17}$$

where c is the number of iteration rounds. We set the termination condition as

$$R^{c+1}(N_m) - R^{c}(N_m) \leq \varepsilon \tag{18}$$

where $m \in [1, n]$ and $\forall N_m$ should satisfy Eq. (18). Meanwhile, ε is constant. Let $SHR(N_*)$ denote semantic rank of N_*. The last results are

$$SHR(N_m) = R^{\eta}(N_m) \tag{19}$$

where η is the round on termination.

5 Evaluation

In this section, we evaluate our framework and conduct extensive experiments as follows:

(1) Using the feature extraction method with generalization ability, DSTH can solve the out-of-sample problem (see Sect. 5.1).

(2) The efficiency of graph building using hash codes generated by DSTH can be greatly improved with allowed accuracy loss (see Sect. 5.2).
(3) SHR can help highlight and prepose those data whose semantic information account for higher proportion in original dataset (see Sect. 5.3).
(4) Our framework can deal with large-scale datasets and return a concise score (assessment result) based on the user's query, which assists the user to make a correct decision on subsequent operations with this dataset (see Sect. 5.4).

We implement the first three experiments on the public CIFAR-10 dataset, and adopt large-scale Tencent datasets to conduct the last one experiment. Our evaluation is executed using Python tools including TensorFlow and Scikit-Learn library. Our experiments are run on two 10-core Intel Xeon E5-2640 machines with 128 GB of DDR4 memory. At last, we conduct the experiment on Tencent dataset using 12 machines.

5.1 Generalization Ability

In this section, we verify the effectiveness of DSTH by executing code length analysis (CLA) and precision-recall (PR) on CIFAR-10. We compare with the state-of-the-art methods on reorganized CIFAR-10 dataset, our DSTH shows a stronger generalization ability, which solves the out-of-sample problem.

In practice, we execute code length analysis (CLA) and precision-recall (PR) on CIFAR-10, compared with the state-of-the-art of single target unlabeled deep hashing algorithms including DeepBit [10], AIBC [16], DSH [11] and SSDH [20]. CIFAR-10 is a labeled data set, which consists of 60,000 32×32 color images in 10 classes, with 6000 images per class. There are 5000 training images and 1000 test images in each class. Particularly, we select the average value of the top 5% nodes in terms of precision in each class as the precision of CLA. In the experiment, we select GoogLeNet and classification model trained on ImageNet to extract deep features. Meanwhile, the CNN structure for generating hash model is similar to [12,23,24].

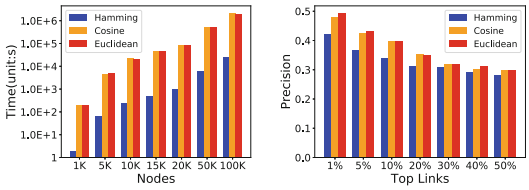
In order to validate the advantage in solving the out-of-samples problem mentioned in Sect. 6, we adjust the distribution of CIFAR-10 by taking the image of cat or automobile off from the training set. Besides, we stipulate that it is correct to classify a cat as a dog and an automobile as a truck. Figure 3(a) and (b) show the mAP@5% CLA and 48-bit codes PR performance without training set of cat. Figure 3(c) and (d) shows the mAP@5% CLA and 48-bit codes PR performance without training set of cat and automobile. As shown in Fig. 3, our results of mAP@5% CLA and PR also yield a significant dominance. As the same as the code length of previous, 48-bit is best for redefined CIFAR-10 data sets at precision of 58.66% and 52.56% respectively. Specially, as shown in Fig. 3(a) and (c), although the gap is reduced, performances of DSTH with 48-bit codes are higher than that of SSDH by 0.196 and 0.165 respectively in precision, illustrating the superiority of ours for solving the problem of out-of-samples.

5.2 Graph Building Efficiency

In this section, for verifying that building a graph by Hamming distance is more efficient than Cosine and Euclidean distance, we exhibit the time of graph building using three metrics with 48-bit vectors (48-bit hash codes and 48 float numbers). In order to ensure the fairness, we set $\Omega = 48$, making all nodes fully connected. As shown in Fig. 4(a), the horizontal coordinate represents the number of nodes while the ordinate represents the graph building time. With the same scale of nodes, the graph building time of Hamming distance is nearly 100 times less than that of Cosine and Euclidean, which shows that Hamming distance has overwhelming predominance over other two metrics in building a graph. Especially, with the scale of nodes increasing, the graph building time of Cosine and Euclidean grows exponentially which is unacceptable, making that Hamming distance becomes the better choice.

In order to compare precision of graph in three metrics, we choose more accurate links from top 1% to top 50% according to the weight of edges with 100,000 nodes. For example, we choose those edges on which the Hamming distance is smaller, while selecting the edges whose Cosine and Euclidean distance is larger. As shown in Fig. 4(b), Hamming distance is 0.070 lower than Euclidean at top 1% links in the worst case and 0.010 lower than Cosine at top 30% links in the best case in term of precision of graph. Averagely, Hamming distance is 0.038 lower than other two metrics in seven cases.

On the whole, there is not a marked difference of precision between three metrics, although hashing will bring certain loss to precision. However, Hamming distance has overwhelming predominance in building a graph in term of time cost. We use hashing and Hamming distance in the follow-up work with comprehensive consideration of tradeoff between efficiency and precision, since an acceptable margin of error is allowed.

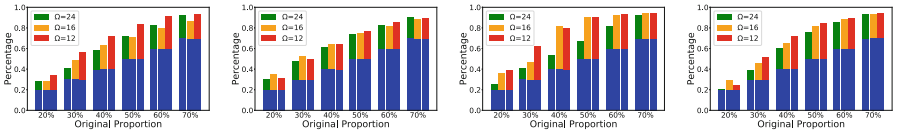


(a) Graph Buliding Time (b) Precision of Graph

Fig. 4. Graph building time with different scale of nodes and precision of graph with 100,000 nodes using Hamming, Cosine and Euclidean distance.

5.3 Predominant Semantics

In this section, we verify SHR can highlight and prepose those data whose semantic information account for higher proportion in this section, which shows our



(a) Percentage of Cat in Top 5% (b) Percentage of Cat in Top 20% (c) Percentage of Automobile in Top 5% (d) Percentage of Automobile in Top 20%

Fig. 5. Trend for percentage of cat and automobile in ranked result using different Ω with 48-bit codes. (Color figure online)

algorithm has practical significance for assessment tasks. In the next experiment, if the ranked results are correct, those images whose semantic distribution account for higher proportion in original data set will obtain larger scores and higher ranks. Thus, based on CIFAR-10 test set, under the premise that the amount of images of other classes remains unchanged, we choose one class as a study object to be added to the sample, making the amount of this class reach 20%, 30%, 40%, 50%, 60% and 70% respectively on the whole data set. We collect proportion of this class in the top 5% and top 20% of ranked results in six cases mentioned above. We set $\varepsilon = 1.0\text{E}{-7}$ and choose $\Omega = 24$, 16 and 12 to conduct the experiments.

Figure 5(a) and (b) show the percentage with different Ω in the top 5% and top 20% of ranked results respectively when choosing cat as the study object. As shown in Fig. 5(a) and (b), in all cases, SHR magnifies original proportion of cat (the part that goes beyond the blue column), indicating the efficiency of this algorithm. Detailedly, compared with $\Omega = 16$ or 24, setting $\Omega = 12$ yields better performance on the magnification of the cat percentage in the top 5% of ranked results, where the cat percentage is averagely 27.7% higher than original proportion in six cases. Among them, the best result exceeds the original proportion by 33.3% in the case of 50%. In the top 20% of ranked results, choosing $\Omega = 12$ yields better performance in most of the cases, where the cat percentage is averagely 21.2% higher than original proportion in six cases. Among them, the best result exceeds the original proportion by 26.3% in the case of 50%. Similarly, Fig. 5(c) and (d) show the results choosing automobile as the study object. As is shown in Fig. 5(c) and (d), SHR achieves the same effect. Detailedly, in the top 5% of ranked results, compared with other setting of Ω, the percentage of automobile shows the superiority in most of the cases while choosing $\Omega = 12$, which is averagely 31.7% higher than original proportion in six cases. Among them, the best result exceeds the original proportion by 40% in the case of 50%. Besides, in the top 20% of ranked results, the percentage of automobile also shows great superiority with $\Omega = 12$, which is averagely 31.0% higher than original proportion in six cases. Among them, the best result exceeds the original proportion by 34.7% in the case of 50%.

It should be explained that better precision and shorter time cost can be captured theoretically when $\Omega < 12$. However, the reduction of links causes too

many isolated nodes all of whom get the same score, which may bring certain loss to the ranked results. Usually, with a larger scale of nodes, hash codes are more widely distributed, thus setting a smaller Ω will not result in too many isolated nodes. In our follow-up research, we intend to study this issue in depth.

As above experimental results show, both in the top 5% and 20% of ranked results, SHR effectively highlights and preposes the data whose semantic information account for higher proportion in original data set after ranking. Thus, our SHR is correct and effective in practical applications.

5.4 Assessment of Query

In this section, we verify that our framework can efficiently complete online assessment work according to the user's query task on large-scale real dark dataset. Also, it can guide and assist the user to conduct subsequent data mining work in order to show our framework is effective to complete the dark dataset assessment. We apply our framework to real-world data set of Tencent which express support for QQ albums, QQ chat and WeChat in a certain period. The size of data is around 5TB consisting of 1,000,000 images. Specially, according to the results shown in Sect. 5.3 that a smaller Ω is proved to be feasible at a million scale, we select $\Omega = 2$ to conduct this experiment.

We use the general method mentioned in Sect. 2 for assessment by analyzing the value of Tencent data set for three tasks which include human intimacy as task-A, lovers traveling in the outskirts as task-B, and driving on road as task-C respectively. The query images are collected from Baidu search and their respective weights are given below. Figure 6 displays the assessing process and results for above tasks. As shown in Fig. 6, the intimacy image representing the first task (query) matches three images whose ranks are high, so it is worth carrying out data mining on this data set for the task-A. For task-B which matches two images contained in Tencent data set, the images of lovers have high scores and images about landscape own medium ranks. However, the weighted score of this task is relatively high, which shows this data set can help analyze images about lovers traveling in the outskirts. Although there are three images

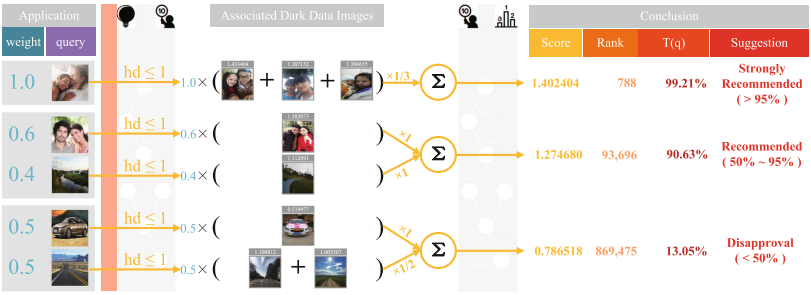


Fig. 6. The process of assessment on Tencent data set for three real applications.

that match the task-C, neither automobiles nor highways obtain high scores, so this data set are not suitable for the task-C.

For exposing semantic information of the dark data, we use deep learning to coarsely explore the content of the whole data set. According to the results, except that the amount of images including people is obviously dominant (66.13%), no more information can be captured for more specific assessment. Because of the better expressive ability, our framework can achieve better assessment results for different queries than classification algorithms.

6 Related Works

Dark Data. [5] has demonstrated the value of dark data by the long tail theory in economics and given the concept of dark data lightening which means constructing relationship according to a new task. [17] proposes using *File WinOver* System to complete the dark data judgment and risk assessment through fingerprint. [1] mentioned that the value of dark data depends on both the requirements of the task and the ability of value extraction. [6] presents the implementation of astronomical dark data management using unified databases. *GeoDeepDive* [21] and *DeepDive* [22] proposed a pragmatic scheme of dark data mining system by correcting annotations and associations of data according to feedback from users.

Content-Based Hashing for Image. Content-based Hashing is a technique that generates compact hash codes from the original data to represent the main content which preserves the data semantic relationship. With the success of Convolution Neutral Network (CNN) [8] in feature extraction, deep hashing [18, 19] becomes the mainstream for image hashing. For unlabeled images, DSTH has better ability to solve the problem of out-of-samples, because it is able to regard the instances beyond scope of cognition as the samples which have been learnt in the model as close as possible. Therefore, DSTH is a better hashing method to reduce the sensitivity of non-cognitive objects which are widely distributed in large-scale data set.

Graph-Based Ranking. Calculating the importance score of each node is a special quantization method without clustering. It is more effective to get evaluation standards by ranking for each node globally. PageRank [14] considers out-degree of related nodes as impact factor for data ranking. [15] applies random walking to ranking community images for searching, which has achieved good results. TextRank [13] and SentenceRank [4] take the weights on edges into consideration, both of which mentioned applying PageRank to improve their algorithms.

7 Conclusions

In this paper, we proposed a framework for image dark data assessment. We first transformed unlabeled images into hash codes by our developed DSTH algorithm, then constructed a semantic graph using restricted Hamming distance,

and finally used our designed SHR algorithm to calculate the overall importance score for each image. During online assessment, we first translated the user's query into hash codes using DSTH model, then matched the suitable data contained in the dark data, and finally returned the weighted average value of these matched data to help the user cognize the dark data. Experimental results showed DSTH could extract semantic features with generalization ability, and SHR could correctly calculate the importance scores according to the similarity between data, and our framework could apply to large-scale datasets and had an overwhelming advantage over deep model.

Acknowledegments. This work is supported by the Innovation Group Project of the National Natural Science Foundation of China No. 61821003 and the National Key Research and Development Program of China under grant No. 2016YFB0800402 and the National Natural Science Foundation of China No. 61672254.

References

1. Cafarella, M.J., Ilyas, I.F., Kornacker, M., Kraska, T., Ré, C.: Dark data: are we solving the right problems? In: ICDE, pp. 1444–1445 (2016)
2. Cai, H., Huang, Z., Srivastava, D., Zhang, Q.: Indexing evolving events from tweet streams. In: ICDE, pp. 1538–1539 (2016)
3. Cao, Y., Long, M., Liu, B., Wang, J.: Deep cauchy hashing for hamming space retrieval. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2018
4. Ge, S.S., Zhang, Z., He, H.: Weighted graph model based sentence clustering and ranking for document summarization. In: ICIS, pp. 90–95 (2011)
5. Heidorn, P.B.: Shedding light on the dark data in the long tail of science. Libr. Trends **57**(2), 280–299 (2008)
6. Heidorn, P.B., Stahlman, G.R., Steffen, J.: Astrolabe: curating, linking and computing astronomy's dark data. CoRR abs/1802.03629 (2018)
7. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. In: ICML, pp. 448–456 (2015)
8. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: NIPS, pp. 1106–1114 (2012)
9. Lai, H., Pan, Y., Liu, Y., Yan, S.: Simultaneous feature learning and hash coding with deep neural networks. In: CVPR, pp. 3270–3278 (2015)
10. Lin, K., Lu, J., Chen, C., Zhou, J.: Learning compact binary descriptors with unsupervised deep neural networks. In: CVPR, pp. 1183–1192 (2016)
11. Liu, H., Wang, R., Shan, S., Chen, X.: Deep supervised hashing for fast image retrieval. In: CVPR, pp. 2064–2072 (2016)
12. Liu, Y., et al.: Deep self-taught hashing for image retrieval. IEEE Trans. Cybern. **49**(6), 2229–2241 (2019)
13. Mihalcea, R.: Graph-based ranking algorithms for sentence extraction, applied to text summarization. Unt Sch. Works **170–173**, 20 (2004)
14. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: bringing order to the web. Technical report, Stanford InfoLab (1999)
15. Richter, F., Romberg, S., Hörster, E., Lienhart, R.: Multimodal ranking for image search on community databases. In: MIR, pp. 63–72 (2010)

16. Shen, F., Liu, W., Zhang, S., Yang, Y., Shen, H.T.: Learning binary codes for maximum inner product search. In: ICCV, pp. 4148–4156 (2015)
17. Shukla, M., Manjunath, S., Saxena, R., Mondal, S., Lodha, S.: POSTER: winover enterprise dark data. In: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, CO, USA, 12–16 October 2015, pp. 1674–1676 (2015)
18. Song, J., Gao, L., Liu, L., Zhu, X., Sebe, N.: Quantization-based hashing: a general framework for scalable image and video retrieval. Pattern Recogn. **75**, 175–187 (2018)
19. Song, J., He, T., Gao, L., Xu, X., Shen, H.T.: Deep region hashing for efficient large-scale instance search from images (2017)
20. Yang, H., Lin, K., Chen, C.: Supervised learning of semantics-preserving hash via deep convolutional neural networks. TPAMI **40**, 437–451 (2017)
21. Zhang, C., Govindaraju, V., Borchardt, J., Foltz, T., Ré, C., Peters, S.: Geodeepdive: statistical inference using familiar data-processing languages. In: SIGMOD, pp. 993–996 (2013)
22. Zhang, C., Shin, J., Ré, C., Cafarella, M.J., Niu, F.: Extracting databases from dark data with deepdive. In: SIGMOD, pp. 847–859 (2016)
23. Zhou, K., Liu, Y., Song, J., Yan, L., Zou, F., Shen, F.: Deep self-taught hashing for image retrieval. In: MM, pp. 1215–1218 (2015)
24. Zhou, K., Zeng, J., Liu, Y., Zou, F.: Deep sentiment hashing for text retrieval in social ciot. Future Gener. Comput. Syst. **86**, 362–371 (2018)