SAGH a supervised image hashing technique

Guillermo García, Mauricio Villegas and Roberto Paredes
ITI/DSIC, Universitat Politècnica de València
Camí de Vera s/n, 46022 València, Spain

Abstract

Hashing techniques have become very popular to solve the content-based image retrieval problem in gigantic image databases because they allow to represent feature vectors using compact binary codes. Binary codes provide speed and are memory-efficient. Different approaches have been taken by researchers, some of them based on the Spectral Hashing objective function, among these the recently proposed Anchor Graph Hashing. In this paper we propose an extension to the Anchor Graph Hashing technique which deals with supervised/label information. This extension is based on representing the samples in an intermediate semantic space that comes from the definition of an equivalence relation in a intermediate geometric hashing.

1 Introduction

Hashing methods map the high-dimensional representation into a binary representation with a fixed number of bits. Binary codes are very storage-efficient, since relatively few bits are required, millions of images can be stored into computer memory. Moreover, computing the hamming distance for binary codes is very fast, as it can be performed efficiently by using bit XOR operation and counting the number of set bits [1, 2].

The hash function design is crucial, this being mainly the unique difference among all these methods. Generally the different methods learn a hash function that preserves the topology of the samples in the original space, i.e. images that are near in the original high dimensional space share the same (or similar) binary code, while images far in the original space have very different binary codes. These methods work with unsupervised information, thus the preservation of the geometric topology is the unique goal to pursue. However, when there is additional information available, which could be supervised (i.e. labels annotated by a human), better performance can be obtained by methods which try to preserve the semantic topology. Since images visually different could contain similar semantic concepts, in these cases the hash code should be designed to (also) preserve the semantic topology.

In this paper, an extension of the Anchor Graph Hashing technique is proposed, which makes it capable to deal with supervised information and produce a binary embedding that preserves no only the geometrical topology, but also the semantic topology of the data.

2 Notation and background

Let $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \subset \mathbb{R}^d$ be the set of n feature vectors extracted from training images and represented in a d-dimensional space. The goal is to learn a binary embedding function of q bits, $f : \mathbb{R}^d \longrightarrow \{-1, 1\}^q$, where for convenience the binary symbols have been defined as -1 and 1. The training set $\mathcal{X}$ produces the set of binary codes $\mathcal{Y} = \{\mathbf{y}_1, \ldots, \mathbf{y}_n\} \subset \{-1, 1\}^q$, and for an arbitrary input test sample, the same mapping would be used so that the hamming distance can be employed to find its nearest neighbors from the training set. Additionally, in the supervised scenario we assume that each training sample $\mathbf{x}_i$ has an associated label vector $\mathbf{t}_i \in \mathbb{R}^l$ which provides semantic information about the sample. Usually the label vector $\mathbf{t}_i$ is a binary vector indicating the presence or absence of each one of the l terms, $\mathbf{t}_i \in \{-1, 1\}^l$.

The hashing function f should preserve the topology of the data assigning similar codes to near samples and dissimilar codes to far samples. To this end a very effective performance measure is to compute a sum of similarity-weighted squares of differences between codes. More concretely Spectral Hashing [3] proposed the constrained optimization:

$$\min_{\boldsymbol{Y}} \frac{1}{2} \sum_{i,j=1}^{n} \|\mathbf{y}_i - \mathbf{y}_j\|^2 a_{ij} = \mathsf{Tr}(\boldsymbol{Y}^T \boldsymbol{L} \boldsymbol{Y})$$

$$s.t. \;\; \boldsymbol{Y} \in \{1, -1\}^{n \times q}, \;\; \boldsymbol{I}^T \boldsymbol{Y} = \boldsymbol{0}, \;\; \boldsymbol{Y}^T \boldsymbol{Y} = n \boldsymbol{I}_{q \times q} \quad (1)$$

where $\boldsymbol{L} = \mathrm{diag}(\boldsymbol{A}\boldsymbol{1}) - \boldsymbol{A}$, being $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ the

similarity matrix having a_{ij} as its components, and $\boldsymbol{Y} \in \{-1, 1\}^{n \times q}$ is a matrix having in each row the codes of the training samples $\mathbf{y}_i$. However, this integer optimization problem is an NP-hard problem. In order to obtain a tractable optimization the authors proposed the spectral relaxation, by dropping the integer constraint and allow $\boldsymbol{Y} \in \mathbb{R}^{n \times q}$. Therefore an approximate solution given by $\text{sgn}(\boldsymbol{Y})$ yields the final desired hash codes.

In [4] the authors introduced a very effective approach for image hashing based on the approach just mentioned, Anchor Graph Hashing (AGH). This unsupervised technique aims at capturing and preserving the semantic topology assuming that close-by points usually share labels. The solution to this problem as proposed in [4] is to avoid computing the whole similarity matrix $\boldsymbol{A}$ for all the n samples. To this end, a small set of m points being $m \ll n$ called anchors, are selected (e.g. using k-means clusters). With these anchors, the matrix $\boldsymbol{A}$ is approximated as $\hat{\boldsymbol{A}} = \boldsymbol{Z}\boldsymbol{\Lambda}^{-1}\boldsymbol{Z}^T$, where $\boldsymbol{\Lambda} = \text{diag}(\boldsymbol{Z}^T\boldsymbol{1})$, and the matrix $\boldsymbol{Z} \in \mathbb{R}^{n \times m}$ is highly sparse, each column only having s values different from zero, which correspond to similarity values of the s nearest anchors. Because of this sparsity, the solution can be obtained by an eigenvalue decomposition of a much smaller $m \times m$ matrix, instead of $n \times n$ of matrix $\boldsymbol{A}$. For further details, the reader should refer to [4].

3 Supervised AGH

The aim of Anchor Graph Hashing is to preserve the original topology by embedding near images to near hashing codes. The results showed in [4] and the reduced computational complexity make this technique a very interesting hashing method for large-scale scenarios. As mentioned above, the main assumption of AGH is that close-by images share labels. However, we can assume that images far in the original space could also share labels and thus being very close in the semantic space. Taking into account that nowadays the images are represented using a very low-level representation, mainly based on bag-of-visual-words, this second assumption is reasonable and motivates the supervised scenario. We propose an extension to AGH that considers side-information provided by the label vectors $\mathbf{t}$, when such information is available.

Note that the hashing function of AGH depends on the similarity between the input sample and the m anchor vectors, and since the label information is not available for the test samples, the label information cannot be introduced into the similarity matrix $\boldsymbol{A}$ as can be done for other methods. Therefore the label information has to be included in an indirect way.

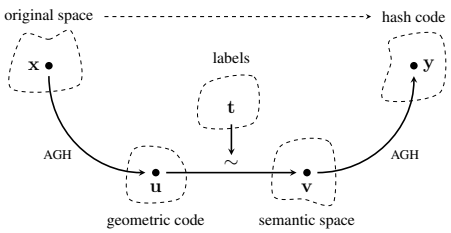

Figure 1. SAGH can be seen as a repeated application of AGH. A first embedding is obtained from the original space to a geometric code. A second embedding is obtained an intermediate semantic space to the final binary hash code.

Our extension, that we call Supervised AGH (SAGH), is based on an two-step repeated application of AGH that uses the label information in an indirect way and the definition of an equivalence relation $\sim$. The resulting procedure can be summarized as follows, first the training samples are embedded into a geometric binary code, then a *semantic* representation is derived from this geometric code and finally a new embedding is performed into the desired binay hash code.

3.1 The proposed SAGH approach

In the proposed Supervised AGH, the intermediate semantic representation of the samples allows that semantically similar samples are coded with similar binary codes despite of being far in the original representation space. From this perspective, SAGH not only can have a better performance because of using the label information, it can also potentially encode the data with shorter binary codes.

The SAGH is performed by first applying the standard AGH to the training set providing an initial hash code of p bits $\mathcal{U} = \{\mathbf{u}_1, \ldots, \mathbf{u}_n\} \subset \{-1, 1\}^p$, usually $p > q$ in order to produce a sparser distribution of the data. We will refer to this first hash code as a the *geometric* code. As mentioned above, the goal of this first hashing is to produce a semantic embedding of the training data. To this end, we define the equivalence relation $\sim$ in the set $\mathcal{X}$. Two samples are equivalent under this relation if these samples have the same geometric code:

$$\mathbf{x}_i \sim \mathbf{x}_j \iff \mathbf{u}_i = \mathbf{u}_j \tag{2}$$

The equivalence class of a particular sample $\mathbf{x} \in \mathcal{X}$ is then defined as:

$$[\mathbf{x}] = \{\mathbf{x}' \in \mathcal{X} \mid \mathbf{u}_{\mathbf{x}} = \mathbf{u}_{\mathbf{x}'}\}$$

With this definition we propose a semantic representation of a particular training sample $\mathbf{x}_i$ as:

$$\mathbf{v}_i = \frac{1}{\mid [\mathbf{x}_i] \mid} \sum_{\mathbf{x} \in [\mathbf{x}_i]} \mathbf{t}_{\mathbf{x}} \tag{3}$$

where $\mid [\mathbf{x}_i] \mid$ is the number of elements in the equivalence class and $\mathbf{t}_{\mathbf{x}}$ is the label vector associated to the sample $\mathbf{x}$. In fact, with this definition all the samples inside an equivalence class share the same semantic representation. Thus, each equivalence class has an associated semantic representation that we denote by $\mathbf{v}_{[\mathbf{x}]}$.

Alternative equivalence relations could be defined in order to group samples depending on different strategies. For instance, several geometric codes could be obtained from the application of AGH with different parameters, e.g. number of nearest anchors s, anchor selection, etc., that yield different geometric codes for the same sample and allowing to define better equivalence relations.

Independently of the equivalence relation definition, equation (3) maps geometric codes $\mathbf{u} \in \{-1, 1\}^p$ into the semantic representations $\mathbf{v} \in \mathbb{R}^l$. As a result we have a representation in a semantic space $\mathcal{V} = \{\mathbf{v}_1, \ldots, \mathbf{v}_n\} \subset \mathbb{R}^l$. This set of semantic representations for the n training samples is considered as a new input to a second AGH that produces an embedding into the final desired binary representation $\mathbf{y} \in \{-1, 1\}^q$ with q bits. This second AGH uses as input the *different* intermediate semantic codes $\mathbf{v}$ generated by the proposed approach. In principle the number of possible different semantic codes can be $\min(n, 2^p)$, but it is much more less in the practical situation. Figure 1 illustrates the SAGH mechanism.

Using this two-step hashing, points that were far in the original space but with similar semantic information should have similar intermediate semantic codes and thus will be mapped to nearby codes in the definitive binary space.

3.2 Hashing query images

The process to obtain a hash code for a query image follows a similar procedure. For a query image $\hat{\mathbf{x}}$ a geometric code $\hat{\mathbf{u}}$ is produced using the first AGH mapping. This geometric code could be the same (i.e. having hamming distance zero) than some geometric code $\mathbf{u}_i$ seen in the training step, thus $\hat{\mathbf{x}} \sim \mathbf{x}_i$ and then we will assign the same intermediate semantic code, $\hat{\mathbf{v}} = \mathbf{v}_i$. But the geometric code $\hat{\mathbf{u}}$ could also be *empty* in the training step, and then there is no semantic code to assign to it. In this case we propose the following procedure. First we have to find the radius R of the minimum hamming ball with non-empty geometric code $\mathbf{u}$ around $\hat{\mathbf{u}}$:

$$R = \min_r \{1, \ldots, p\} \;\; s.t. \, \exists \, \mathbf{x} \in \mathcal{X} : \; d(\mathbf{u}_{\mathbf{x}}, \hat{\mathbf{u}}) = r \tag{4}$$

where $d(\cdot, \cdot)$ is the hamming distance. This radius defines the set $\mathcal{B}_R$ of the different equivalence classes inside this hamming distance. We propose to obtain the semantic embedding of the query point as an average of all the semantic representations associated to the equivalence classes inside $\mathcal{B}_R$:

$$\hat{\mathbf{v}} = \frac{1}{\mid \mathcal{B}_R \mid} \sum_{[\mathbf{x}] \in \mathcal{B}_R} \mathbf{v}_{[\mathbf{x}]} \tag{5}$$

Finally, the semantic code $\hat{\mathbf{v}}$ will be embedded using again AGH to the definitive hash code $\hat{\mathbf{y}}$.

It is important to note that the hashing obtained by SAGH will be affected mainly by the equivalence relation definition (2), the procedure to obtain a semantic representation associated to each equivalence class (3) and the semantic embedding for those query images that fall into empty geometric codes (5).

4 Experiments

In order to assess the performance of the proposed SAGH technique, we have performed experiments using a dataset widely used in the literature. This dataset is a version of the CIFAR[1] dataset [5], which consists 64,185 images selected from the Tiny Images dataset [6]. The original Tiny Images are 32×32 pixels, although they have been represented with grayscale GIST descriptors [7] computed at 8 orientations and 4 different scales, resulting in 320-dimensional feature vectors. These images have been manually grouped into 11 ground-truth classes (airplane, automobile, bird, boat, cat, deer, dog, frog, horse, ship and truck), thus we shall refer to this version of the dataset as CIFAR-11, and it is the same dataset that was used in [8].

For comparison, we also ran the experiments with other hashing techniques found in the literature for which there was code freely available.

4.1 CIFAR-11

To estimate the performance of the different methods for the CIFAR-11 dataset, we have employed a 5-time repeated hold-out procedure. In each of the five rounds,

[1] http://www.cs.toronto.edu/~kriz/cifar.html

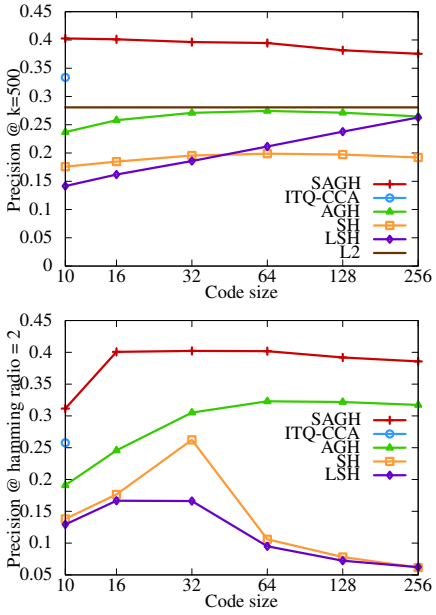

Figure 2. CIFAR-11 dataset. On the left, average precision of the top-500 ranked images. On the right, average precision for a hamming radius of 2.

3,000 images were randomly selected for the test set and the remaining was left as the training set. The final results are the average over the five partitions.

The results are presented in Figure 2. The performance is measured using the class labels as ground truth. In the figure one of the graphs presents the average precision for the first 500 retrieved images when varying the number of bits, this measures the hash ranking performance. For retrieved images having exactly the same hamming distance, a random reordering was applied. The other graph in the figure shows the average precision for a hamming radius of 2 and this measures the hash lookup performance.

As was expected, the two supervised methods, SAGH and ITQ-CCA, perform much better than all of the other unsupervised methods. This is quite understandable since the labels of CIFAR-11 are manually selected and not noisy, thus there is much to gain by using this additional available information. The performance of SAGH is better than ITQ-CCA. Note that in this case, because there are only 11 classes, ITQ-CCA is limited to a maximum of 10 bits, which is a severe limitation. As can be observed, the performance of the proposed SAGH is better than its unsupervised counterpart AGH. This confirms that the proposal effectively is capable of taking advantage the additional information to achieve a better performance. In these results the same behavior as in [4] is observed both for AGH and SAGH. The precision at a hamming radius of 2 does not decrease for large code sizes. Although the performance is not better than for fewer bits.

5 Conclusions

In this paper we propose an extension to the Anchor Graph Hashing technique which is capable of taking advantage of supervised/label information. This extension is based on representing the samples in an intermediate semantic space that comes from the definition of an equivalence relation in an intermediate geometric code. The results show that our approach is a very effective way to incorporate such supervised information to the standard AGH. The standard AGH is clearly outperformed by our SAGH in the CIFAR dataset where the supervised information can be considered very clean. Moreover, SAGH is clearly the best technique on this dataset compared to the state-of-the-art ITQ-CCA.

References

[1] Knuth, D.E.: The Art of Computer Programming, Volume I: Fundamental Algorithms, 3rd Edition. Addison-Wesley (1997)

[2] Wegner, P.: A technique for counting ones in a binary computer. Commun. ACM **3** (1960) 322–

[3] Salakhutdinov, R., Hinton, G.: Semantic hashing. Int. J. Approx. Reasoning **50** (2009) 969–978

[4] Liu, W., Wang, J., Kumar, S., Chang, S.F.: Hashing with graphs. In Getoor, L., Scheffer, T., eds.: Proceedings of the 28th International Conference on Machine Learning (ICML-11). ICML '11, New York, NY, USA, ACM (2011) 1–8

[5] Krizhevsky, A.: Learning multiple layers of features from tiny images. Master's thesis (2009)

[6] Torralba, A., Fergus, R., Freeman, W.T.: 80 million tiny images: A large data set for nonparametric object and scene recognition. IEEE Trans. Pattern Anal. Mach. Intell. **30** (2008) 1958–1970

[7] Oliva, A., Torralba, A.: Modeling the shape of the scene: A holistic representation of the spatial envelope. Int. J. Comput. Vision **42** (2001) 145–175

[8] Gong, Y., Lazebnik, S.: Iterative quantization: A procrustean approach to learning binary codes. In: CVPR. (2011)