Learning Supervised Binary Hashing without Binary Code Optimization

Miguel Á. Carreira-Perpiñán
EECS, University of California, Merced
mcarreira-perpinan@ucmerced.edu

Ramin Raziperchikolaei
EECS, University of California, Merced
rraziperchikolaei@ucmerced.edu

Introduction. Searching a large database of high-dimensional images for the most similar images to a query image is a nearest-neighbor search whose exact solution takes too long to be practical. One way to approximate this is to map the query image to a short binary vector and search for this instead, possibly using an inverted index [2]. This is much faster because the binarized database takes far less space, so it may fit in fast memory, and Hamming distances are fast with hardware support [2]. The success of this approach crucially relies on designing a binary hash function $\mathbf{h}$: $\mathbb{R}^D \rightarrow \{-1,+1\}^b$ (mapping input $\mathbf{x} \in \mathbb{R}^D$ to a b-bit code $\mathbf{z} = \mathbf{h}(\mathbf{x}) \in \{-1,+1\}^b$) such that the ground-truth similarity between any two given images correlates well with the Hamming distance between their corresponding binary codes. We focus on the supervised hashing where the semantic similarity between the images defines the ground-truth. So two images that are far in Euclidean distance may in fact be similar (e.g. an object seen from different viewpoints). The only way to define good hash functions is to learn them from similarity information provided for the training data.

Learning the hash function: optimization-based approach vs diversity-based approach. To learn the hash function, the leading approach has so far been *optimization-based* [4, 5, 7, 10]. The approach works by first defining an objective function over the hash function and then minimizing it. The objective formalizes the notion that similar images should have lower Hamming distance than dissimilar images. However, the optimization is difficult, usually NP-complete, and the existing optimization algorithms are approximate and slow, and most do not scale to large training sets.

A recent work, *Independent Laplacian Hashing (ILH)* [1], has proposed a very different, *diversity-based approach*. Rather than optimizing over the hash function of every code bit jointly, it trains the b single-bit hash functions $\mathbf{h}(\cdot) = (h_1(\cdot), \dots, h_b(\cdot))$ independently from each other while ensuring they differ via diversity-inducing mechanisms from the ensemble learning literature. For example, optimizing each single-bit code on a different, random data subset, and then fitting a hash function (binary classifier) to each bit.

To learn each single-bit hash function, ILH samples a training set of N points and assigns a single-bit code to each of them by optimizing the following objective function:

$$E(\mathbf{z}) = \sum_{n,m=1}^{N} y_{nm}(z_n - z_m)^2, \quad \mathbf{z} \in \{-1,+1\}^N \tag{1}$$

where $y_{nm} = +1$ or $y_{nm} = -1$ indicates that the images n and m are similar or dissimilar. Then, ILH learns the hash function by training a classifier given the original points as the input and the single-bit codes as the output. This performs surprisingly well compared to approaches based purely on optimization, while being simpler, more scalable and embarrassingly parallel.

Our proposed method: independent supervised binary hashing (ISH). The motivation for our approach stems from trying to push the frontier of independent single-bit hash function learning. While in the b-bit case the binary code space can have 2^b different codes, with $b = 1$ there are just two possible codes, $+1$ and -1, and every training point must be assigned to one of them. This partitions the training set into two classes. What do these two codes, or classes, represent?

Recall the purpose of defining an objective function over binary codes: if for image $\mathbf{x}_n$ we know that $\mathbf{x}_m$ is a similar point and $\mathbf{x}_q$ is a dissimilar point, then ideally $\mathbf{x}_n$ and $\mathbf{x}_m$ should have the same code (say, $+1$) and $\mathbf{x}_q$ a different code (necessarily -1). This will assign a Hamming distance of 0 to $(\mathbf{x}_n, \mathbf{x}_m)$ and of 1 to $(\mathbf{x}_n, \mathbf{x}_q)$. In fact, the objective function was a mathematical device precisely designed to be able to translate the available similarity information into codes whose Hamming distances preserve such similarity. Hence, all we have to do to learn a single-bit hash function is to pick a point $\mathbf{x}_n$ (the "seed") and find a sample $\mathcal{S}_+$ of points that are similar to $\mathbf{x}_n$ ($y_{nm} > 0$) and a sample $\mathcal{S}_-$ of points that are dissimilar to $\mathbf{x}_n$ ($y_{nm} < 0$). This defines a two-class problem on the training set $\mathcal{S}_+ \cup \mathcal{S}_-$, on which we can train a classifier to use as single-bit hash function.

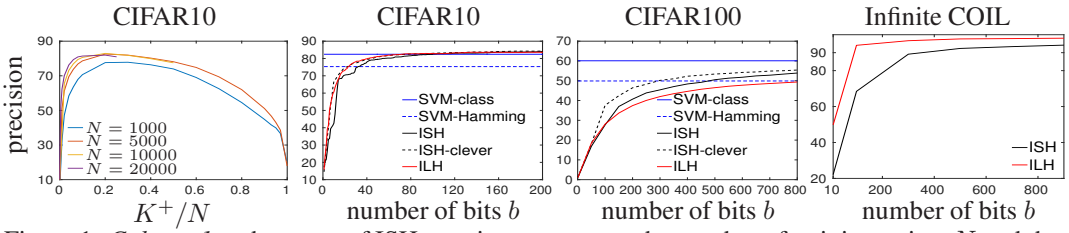

Figure 1: *Column 1:* robustness of ISH over its parameters, the number of training points N and the number of similar points K^+. *Columns 2–4:* precision as a function of the number of bits b.

We call this *Independent Supervised Hashing (ISH)*. It has the following advantages. It is very simple, requiring only the available similarity values and training binary classifiers. We need not encode the supervision information into some matrix $\mathbf{Y}$ for use in a single-bit objective function (1). Like ILH, ISH is embarrassingly parallel over the b bits, and suitable for implementation in a distributed-data setting. It is faster than ILH, because it eliminates the NP-complete optimization of the single-bit objective, and much faster than approaches that optimize over the b bits for the entire dataset jointly. It scales to bigger datasets, essentially as big as long as we are able to train a binary classifier on them, while ILH is still somewhat limited because of the NP-complete optimization. One can keep adding single-bit hash functions until a desired precision and/or maximum number of bits is reached, effectively selecting the value of b. As we show later, ISH learns hash functions comparable to the state-of-the-art. The precision of ISH (and ILH) consistently increases as more bits are added over a range of b values. Finally, we can *prune* the ensemble of b single-bit hash functions produced by ISH as proposed in [8] and achieve a smaller ensemble with similar precision performance.

When class labels are available for the training set, supervised binary hashing works define the ground-truth for a query as all the points in its class. This is problematic because perfect retrieval can be achieved by training C one-vs-all perfect binary classifiers (assuming C classes), and returning the entire class predicted for a query at test time. So, in this case, binary hashing experiments should report the retrieval performance for a C-class classifier as a baseline. Although ISH works with arbitrary similarity values, it is instructive to see how it behaves in this case. The training set for a given single-bit hash function consists of a sample $\mathcal{S}_+$ of points from class k (the class of the seed) and a sample $\mathcal{S}_-$ from all other classes. *This is (a sampled version of) a one-vs-all classifier.*

Experimental setup. We report the results on CIFAR10 [3], CIFAR100 [3] and Infinite COIL datasets. Infinite COIL is created by adding 100 images uniformly along the straight line segment between every pair of consecutive images of COIL20 [6]. We use $D = 4\,096$ VGG network features (the last fully connected layer of VGG) [9]. The ground-truth is defined based on the class labels in CIFAR0 and CIFAR100 and based on both the class labels and the angles in Infinite COIL. We compare the proposed method ISH with ILH [1], which outperforms the state-of-the art methods.

Experiments: parameter robustness. The parameters of ISH are the size K^+ of the seed class and the training set size N for a single-bit hash function. Intuitively it might seem that using $K^+/N = \frac{1}{2}$ would work well, and our experiments show this is true. The first column of fig. 1 shows the ISH precision as a function of the ratio K^+/N for $b = 100$ bits and for $N \in [1\,000, 20\,000]$, in CIFAR10. ISH is very robust, performing reasonably well over a wide range of K^+/N values. The same experiment on the parameters of ILH shows that ILH is not as robust as ISH.

Experiments: precision over the number of bits. The last three columns of fig. 1 compare the following methods. (1) ISH, selecting the seeds randomly to create the training set. (2) ISH-clever: selects seeds by cycling over the C classes in the labeled datasets. (3) SVM-class: for the labeled datasets with C labels, we train C one-vs-all classifiers and report the classification accuracy. (4) SVM-Hamming: We use the C one-vs-all classifiers of SVM-class as the hash functions. (5) ILH. When the ground-truth is given by the class label, SVM-class gives better precision than the hashing methods, but is inapplicable to the Infinite COIL, which has no image labels. ILH and ISH outperform SVM-Hamming and are generally comparable in different datasets, sometimes a bit better, sometimes a bit worse. The precision of ISH and ILH keeps growing throughout the range of b.

Conclusion. ISH is a drastic innovation in the field of binary hashing: *we have essentially redefined the problem of supervised binary hashing as a collection of independent binary classification problems*. We have demonstrated that it is not necessary to optimize an objective function of binary codes in order to learn good hash functions for information retrieval. We assign binary codes to the points based on the similarity to a set of seeds. The proposed algorithm is simple, fast, embarrassingly parallel, robust, and achieves state-of-the-art performance in precision and recall.

References

[1] M. Á. Carreira-Perpiñán and R. Raziperchikolaei. An ensemble diversity approach to supervised binary hashing. In D. D. Lee, M. Sugiyama, U. von Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems (NIPS)*, volume 29, pages 757–765. MIT Press, Cambridge, MA, 2016.

[2] K. Grauman and R. Fergus. Learning binary hash codes for large-scale image search. In R. Cipolla, S. Battiato, and G. Farinella, editors, *Machine Learning for Computer Vision*, pages 49–87. Springer-Verlag, 2013.

[3] A. Krizhevsky. Learning multiple layers of features from tiny images. Master's thesis, Dept. of Computer Science, University of Toronto, Apr. 8 2009.

[4] G. Lin, C. Shen, D. Suter, and A. van den Hengel. A general two-step approach to learning-based hashing. In *Proc. 14th Int. Conf. Computer Vision (ICCV'13)*, pages 2552–2559, Sydney, Australia, Dec. 1–8 2013.

[5] G. Lin, C. Shen, Q. Shi, A. van den Hengel, and D. Suter. Fast supervised hashing with decision trees for high-dimensional data. In *Proc. of the 2014 IEEE Computer Society Conf. Computer Vision and Pattern Recognition (CVPR'14)*, pages 1971–1978, Columbus, OH, June 23–28 2014.

[6] S. A. Nene, S. K. Nayar, and H. Murase. Columbia object image library (COIL-20). Technical Report CUCS–005–96, Dept. of Computer Science, Columbia University, Feb. 1996.

[7] R. Raziperchikolaei and M. Á. Carreira-Perpiñán. Optimizing affinity-based binary hashing using auxiliary coordinates. In D. D. Lee, M. Sugiyama, U. von Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems (NIPS)*, volume 29, pages 640–648. MIT Press, Cambridge, MA, 2016.

[8] R. Raziperchikolaei and M. Á. Carreira-Perpiñán. Learning independent, diverse binary hash functions: Pruning and locality. In *Proc. of the 17th IEEE Int. Conf. Data Mining (ICDM 2016)*, pages 1173–1178, Barcelona, Spain, Dec. 12–15 2016.

[9] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proc. of the 3rd Int. Conf. Learning Representations (ICLR 2015)*, San Diego, CA, May 7–9 2015.

[10] Y. Weiss, A. Torralba, and R. Fergus. Spectral hashing. In D. Koller, Y. Bengio, D. Schuurmans, L. Bottou, and A. Culotta, editors, *Advances in Neural Information Processing Systems (NIPS)*, volume 21, pages 1753–1760. MIT Press, Cambridge, MA, 2009.