Fitted Spectral Hashing

Yu Wang[1,2], Sheng Tang[1], Yalin Zhang[1,2], JinTao Li[1], DanYi Chen[3]
[1]Institute of Computing Technology, Chinese Academy of Sciences,Beijing 100190, China
[2]Graduate University of Chinese Academy of Sciences, Beijing 100190, China
[3]School of Information and Communication Engineering, Beijing University of Posts and Telecommunications,
Beijing 100081, China
{wangyu, ts, zhangyalin, jtli}@ict.ac.cn, rainachen1216@163.com

ABSTRACT

Spectral hashing (SpH) is an efficient and simple binary hashing method, which assumes that data are sampled from a multidimensional uniform distribution. However, this assumption is too restrictive in practice. In this paper we propose an improved method, Fitted Spectral Hashing, to relax this distribution assumption. Our work is based on the fact that one-dimensional data of any distribution could be mapped to a uniform distribution without changing the local neighbor relations among data items. We have found that this mapping on each PCA direction has certain regular pattern, and could fit data well by S-Curve function, Sigmoid function. With more parameters Fourier function also fit data well. Thus with Sigmoid function and Fourier function, we propose two binary hashing methods. Experiments show that our methods are efficient and outperform state-of-the-art methods.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Retrieval models

General Terms

Algorithms, Experimentation

Keywords

Spectral hashing, Sigmoid function, Fourier function

1. INTRODUCTION

Similarity search is an essential problem in the field of machine learning, computer vision and information retrieval. However, with increasing amounts of data, similarity search faces following challenges: efficient storing millions of items in memory and quickly finding similar items to a query item. Recent work [1] shows that binary hashing methods are a powerful way to address those challenges:

- The highly compressed binary codes can be loaded into main memory efficiently;
- Searching similar items can be extremely fast with Hamming distances calculated by bit XOR operation: an ordinary PC today would be able to do millions of hamming distance computation in just a few milliseconds .

The basic idea of binary hashing methods is to formulate projections from items to binary codes, so as to approximately preserve a given similarity function of interest [2]. ”Good” binary codes should meet the entropy maximizing criterion. According to the information theory [3], the maximal entropy of a source alphabet is attained by having a uniform probability distribution. If the entropy of binary codes over data set is small, it means that data are mapped to only a small number of codes, thus rendering the codes inefficient.

However, many state-of-the-art methods do not meet this criterion. One of the most well-known binary hashing methods is locality sensitive hashing method (E2LSH), which calculates binary codes by projecting data on random vectors with random thresholds, and as shown in [4] the hamming distance between binary codes will asymptotically approach the Euclidean distance between data items. The Kernelized version (KLSH) [5] widens the accessibility of E2LSH to generic normalized kernel functions. Rather than using random vectors, the authors have pursued machine learning approaches, e.g. the restricted Boltzmann method (RBM) [6] and Boosting [7], to accelerate the document and image retrieval.

When data are uniformly distributed in a hyper-rectangle, Spectral hashing (SpH) [8], derived from the spectral graph partitioning problem, meets the entropy maximizing criterion. Bits can be calculated efficiently by the eigenfunctions of the weighted Laplacian defined on R^1. This simple method outperforms above methods. However, the assumption of SpH is too restrictive in practice. Like SpH, Self-Taught Hashing (STH) [1] is also related to the spectral graph partitioning, but uses ration-cut to address the entropy maximizing criterion and applies support vector machine (SVM) to yield hash codes for out-of-sample objects. STH can work with any data distribution, while suffering with high computational cost. The binarized dimensionality reduction technique Latent Semantic Indexing (LSI) [9] and its improved version Laplacian Co-Hashing (LCH) [10] are efficient to get binary codes of documents. Via setting the threshold to the median value of left singular vectors of

MM'13, October 21–25, 2013, Barcelona, Spain.
Copyright 2013 ACM 978-1-4503-2404-5/13/10 ...$15.00.
http://dx.doi.org/10.1145/2502081.2502169.

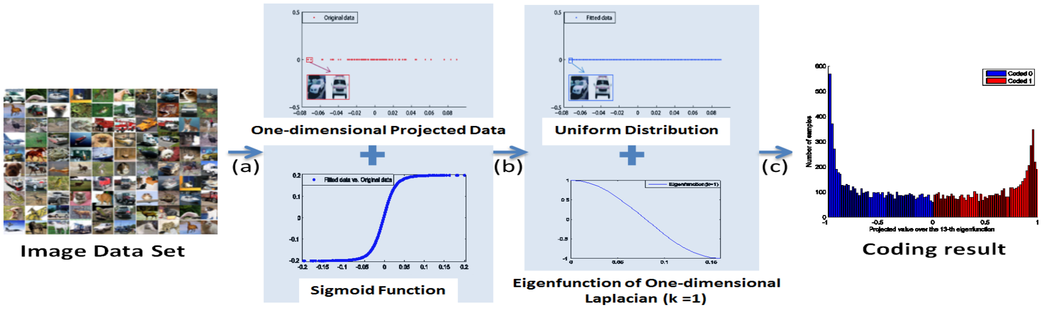


Image Data Set
(a)
One-dimensional Projected Data
Original data
Sigmoid Function
Fitted data vs. Original data
(b)
Uniform Distribution
Fitted data
Eigenfunction of One-dimensional Laplacian (k =1)
Eigenfunction (k=1)
(c)
Coding result
Coded 0
Coded 1
Number of samples
Projected value over the 13-th eigenfunction

Figure 1: Illustration of using our method to generate one bit binary code for image data set, and the steps are: (a) Project data on one PCA direction; (b) Fit the projected data with Sigmoid function, in order to map one-dimensional data to a uniform distribution; (c) Use eigenfuction $(k = 1)$ of one-dimensional Laplacian on uniform distribution to generate binary code.

the whole data matrix, they also meet the entropy maximizing criterion.

In order to relax the restrictive assumption of SpH on data distribution, we propose an improved version Fitted Spectral Hashing as shown in Figure 1. Our work is based on this obvious fact that any distribution of one-dimensional data could be mapped to a uniform distribution without changing the local neighbor relations among them. The main contributions are as follows:

1 We have found that this mapping has certain regular pattern on each PCA direction. It could be well fitted by S-Curve function, Sigmoid function. With more parameters, the Fourier function also fits data well.

2 We integrate Spectral hashing with fitting functions to approximately meet the entropy maximization criterion, and propose two binary hashing methods, Sigmoid Fitting Spectral Hashing (SFSpH) and Fourier Fitting Spectral Hashing (FFSpH).

2. SPECTRAL HASHING

In this section we briefly introduce the related binary hashing method, SpH. The SpH method is derived from the graph partitioning problem. Let $\{y_i\}_{i=1}^n$ be the list of binary codes for n data points $\{x_i\}_{i=1}^n$, and $W_{n\times n}$ be the similar matrix, where $W(i,j) = exp(- \parallel x_i - x_j \parallel^2 / \epsilon^2)$. The average Hamming distance between similar neighbors can be written: $\sum\limits_{i,j} W(i,j) \parallel y_i - y_j \parallel^2$. With the assumption that data point x is sampled from a probability distribution $p(x)$, the SpH problem can be written as:

$$\min \int \parallel y(x_1) - y(x_2) \parallel^2 W(x_1,x_2)p(x_1)p(x_2)dx_1dx_2,$$
$$s.t. \int y(x)p(x)dx = 0, \int y(x)y(x)^T P(x)dx = I$$
$$y_i \in \{-1,1\}^k. \quad (1)$$

Relaxing the constraint that $y_i \in \{-1,1\}^k$ gives a spectral problem, whose solutions are eigenfunctions of the weighted Laplace-Beltrami operators defined on manifold [11]. When $p(x)$ is a separable and multidimensional uniform distribution $Pr(x) = \prod\limits_i u_i(x_i)$, where $u_i(x_i)$ is a one-dimensional uniform distribution on $[a,b]$, the solutions are

$$\phi_k(x) = sin(\frac{\pi}{2} + \frac{k\pi}{b-a}x). \quad (2)$$

$$\lambda_k = 1 - e^{-\frac{\epsilon^2}{2}|\frac{k\pi}{b-a}|^2}. \quad (3)$$

The Eq.2 can be used to code data points $\{x_i\}_{i=1}^n$ directly. However, this simple algorithm has an obvious limitation: it assumes data points are generated from a multidimensional uniform distribution. When dealing with the actual data, we have found that the SpH algorithm can hardly meet the entropy maximizing criterion as shown in Figure 2(b).

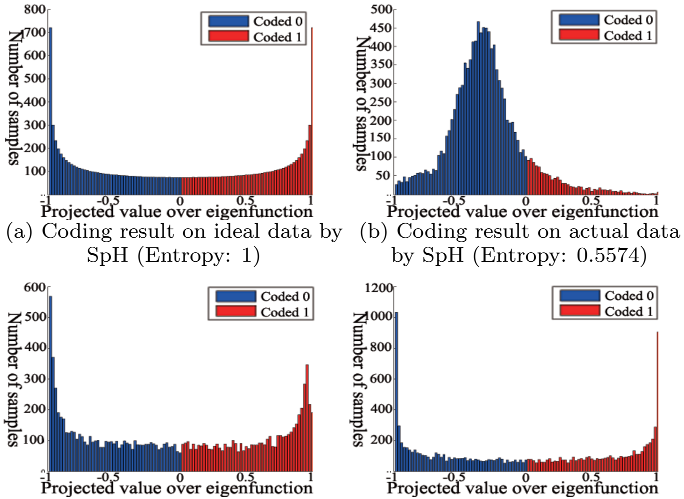
3. PROPOSED APPROACH

The fitted SpH uses PCA to align the axes like SpH, but doesn't need the distribution assumption. It is based on this simple fact that any distribution of one-dimensional data could be mapped to a uniform distribution. Therefore, with a uniform distribution the fitted SpH algorithm can approximately meet the entropy maximizing criterion, and corresponding entropies[1] are shown in Figure 2(c) and Figure 2(d). However, there are two questions: (1) Does this mapping change local neighbor relations among data items? (2) Could this fitting model be efficiently computed for out-of-sample objects?

Proposition 3.1. *Let $\{p_i\}_{i=1}^n$ be the projected values of $\{x_i\}_{i=1}^n$ on arbitrary PCA axis, $\{q_i\}_{i=1}^n$ be data, which obey uniform distribution, mapped from $\{p_i\}_{i=1}^n$. After this mapping, we claim that: (1) the entropy of binary coding $\{q_i\}_{i=1}^n$ can be maximized. (2) If the mapped function f is monotonically increasing function, the sequence of $\{p_i\}_{i=1}^n$ could be preserved by $\{q_i\}_{i=1}^n$. (3) The adjacency relationships among data items can be preserved after this mapping.*

Proof. (1) q obey the uniform distribution on $[a,b]$, and are translated to $[0, b-a]$ without loss. Since $q \in (0, \frac{b-a}{2k})$, $\phi_k(q) > 0$; $q \in (\frac{b-a}{2k}, \frac{b-a}{k})$, $\phi_k(q) < 0$, half of q on uniform distribution could be coded as 1 or 0 by Eq.2 in the range $(0, \frac{b-a}{k})$. Because Eq.2 is a periodic function, the entropy of binary coding $\{q_i\}_{i=1}^n$ in every range $(0+(i-1)\times\frac{b-a}{k}, \frac{b-a}{k} + (i-1)\times\frac{b-a}{k}))$, $i = 1, ..., k$, can be maximized. (2) $f(p_i) = q_i$, when $\{p_i > p_z > \cdots > p_j\}$, because f is monotonically increasing function, then $\{q_i > q_z > \cdots > q_j\}$, the sequence of

[1] Let pr be the probability of one bit be 0, the entropy of pr is: $H(pr) = -prlog_2(pr) - (1-pr)log_2(1-pr)$



(a) Coding result on ideal data by SpH (Entropy: 1)

(b) Coding result on actual data by SpH (Entropy: 0.5574)

(c) Coding result on actual data by FFSpH (Entropy: 0.9935)

(d) Coding result on actual data by SFSpH (Entropy: 0.9909)

Figure 2: With ideal data(a), which obey uniform distribution, half of the training samples are coded as one or zero by SpH, while (b) is not balanced. (c) and (d) are approximately balanced.

$\{p_i\}_{i=1}^n$ is preserved by $\{q_i\}_{i=1}^n$ (3) PCA axes are independent from each other, therefore the mapping on one PCA axis doesn't affect others. Because the adjacency relationships on each axis are preserved, the adjacency relationships among data items are preserved. □

For out-of-sample objects, we have found that this mapping could be well fitted by Sigmoid function:

$$f(x) = a_0 \times (1 + e^{(\frac{-(x-a_1)}{a_2})})^{-1} + a_3, \tag{4}$$

where a_0, a_1, a_2 and a_3 are 4 parameters of Sigmoid function. Obviously Eq.4 is monotonically increasing function. It has been observed in the literature [12] that the density projections of large high-dimensional data sets onto a random line generally follows a normal distribution, thus the Cumulative Distribution Function (CDF) function could be used as fitting function. Because Sigmoid function is the commonly used CDF of normal function, thus it can fit data well. For comparison, we also use the Fourier function which can approximate any function:

$$\begin{aligned} f^*(x) = a_0 + a_1 \times cos(x \times w) + b_1 \times sin(x \times w) + \dots \\ + a_8 \times cos(8 \times x \times w) + b_8 \times sin(8 \times x \times w), \end{aligned} \tag{5}$$

where $a_0, w, a_1, \dots, a_8, b_1, \dots, b_8$ are 18 parameters and more parameters ensure the low Sum of Squares due to Error (SSE) of Fourier function. Though with a little higher SSE, the computational cost of Sigmoid function is lower as showed in Experiments. There are many tools available to solve the fitting functions, and CFtool in Matlab is used in this paper.

In Algorithm 1, the cost of eigenvalue decomposition of matrix $X^* \in R^{m \times m}$ is lower than state-of-art methods dealing with $R^{n \times n}$ [1] [9] [10], where m is the dimensionality and n is the number of samples. This Algorithm first learns the fitting function on each selected PCA direction, and then uses Eq.2 to generate the binary codes for one-dimensional uniform data.

Algorithm 1 Fitted Spectral Hashing

```
Input: Matrix $X \in R^{n \times m}$, the number of bits $k$
Output: Binary codes $Y \in R^{n \times k}$, Mode $M$
Compute eigenvectors and eigenvalues of $X^*$, $X^* = X' \times X$;
According to the order of eigenvalues, select k eigenvectors $\{V_1, V_2, \dots, V_k\}$ as the PCA axes and save them in $M$;
for i=1 to k do
    Compute projected value $p$ with $X$ and $V_i$
    Compute $f_i$ using Eq.4 or Eq.5 with $p$
    Evaluate the k smallest eigenvalues $\{\lambda_1, \lambda_2, \dots, \lambda_k\}$ by Eq.3 with $p$
    Save $f_i$, $\{\lambda_1, \lambda_2, \dots, \lambda_k\}$ in $M$
end for
Select $k$ eigenfunctions $\{\phi_1, \phi_2, \dots, \phi_k\}$ by Eq.2 according to the order of $\lambda$ in $M$ and save them in $M$
for j=1 to k do
    Find corresponding $V_i$ according to $\lambda_j$
    Compute projected value $p$ with $X$ and $V_i$
    $q = f_i(p)$
    Threshold eigenfunctions $\phi_j(a)$ at zero to obtain binary codes $Y(:, j)$
end for
```

4. EXPERIMENTS

In this section, we evaluate the proposed methods and discuss their computational costs. We compare our methods with LSI [9], LCH [10] and STH [1], which are well known binary hash methods and meet the entropy maximization criterion. SpH is used as baseline.

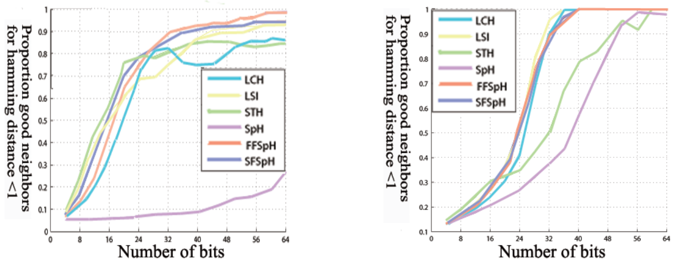
4.1 Data Sets

We choose the well-known tiny image dataset CIFAR-10(60K) and real-world text dataset 20Newsgroups as our experiment data sets. CIFAR-10 contains 60K 32×32 color images of 10 classes and 6K images in each class. We extract 512-D GIST features for each image and use 80% images for training, 20% left for testing. The 20Newsgroups corpus contains 18846 documents distributed across 20 categories. The document dataset has been pre-processed by stopword removal, Poster stemming, and TF-IDF weighting. The time-based split leads to 11314 (60%) documents for training and 7532 (40%) documents for testing.

4.2 Results and Discussions

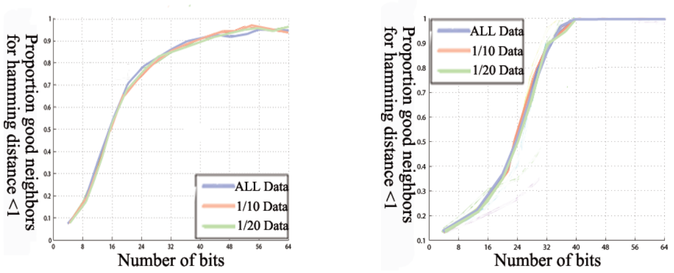
Retrieval performance: As shown in the Figure 3 the proposed methods give the best results on 20Newsgroup, and top-3 results on CIFAR-10. Our methods use fitting functions to approximately meet entropy maximization criterion and achieve average 60% improvement on 20Newsgroups and average 19% improvement on CIAFR-10 compared with SpH. Compared with other methods [9] [10] [1], the PCA directions, which represent the directions of maximum variance, ensure better retrieval performace.

Our methods, SFSpH and FFSpH, work almost the same on 20Newsgroups. With more parameters, the performance of FFSpH on CIFAR-10 is slightly better than SFSpH's.



(a) 20Newsgroups (b) CIAFR-10

Figure 3: Comparisons with different methods:(a) Our methods give the best results on 20Newsgroups. (b) Our method give the top-three results on CIAFR-10.



(a) 20Newsgroups (b) CIAFR-10

Figure 4: The performances of SFSpH under different sampling rates are almost as good as using all samples.

Thus we claim that Sigmoid function could fit data well with much less parameters.

Computational cost: As shown in Figure 5 the computational cost of SFSpH is much lower than other methods', only 60% of FFSpH's, 25.3% of LCH's, 25.2% of LSI's and 3% of STH's, while a litter higher than SpH's. Because SpH only needs to calculate the PCA directions, while our methods need extra fitting function computation.

The computational cost of fitting function is relevant with the number of parameters and the scale of samples. The Sigmoid function needs 4 parameters, while Fourier function needs 18 parameters. Therefore the cost of FFSpH is higher than that of SFSpH. Due to the simplicity of Sigmoid function, low sampling rate gives almost as good result as full sampling. The performance comparisons of SFSpH under different sampling rates are shown in Figure 4.

The LCH and LSI need the SVD (singular vectors decomposition) of whole data matrix $R^{n\times n}$ to compute projecting directions, thus the computational complexity is higher than proposed methods with matrix $R^{m\times m}$. The STH method has to construct the spectral graph with complexity of $O(mn^2)$, and compute the eigenvectors of similar matrix $R^{n\times n}$, then train k SVM models to generate binary codes, therefore its cost is the highest.

5. CONCLUSIONS

In this paper, we propose two fitted spectral hashing methods and prove their rationality. The hashing codes of our methods approximately meet the entropy maximizing criterion. Experiments show that these two fitted spectral hashing methods out-perform stat-of-the-art methods. Furthermore, due to the efficiency, we claim that our methods can be efficient to train large data sets with short code lengths.

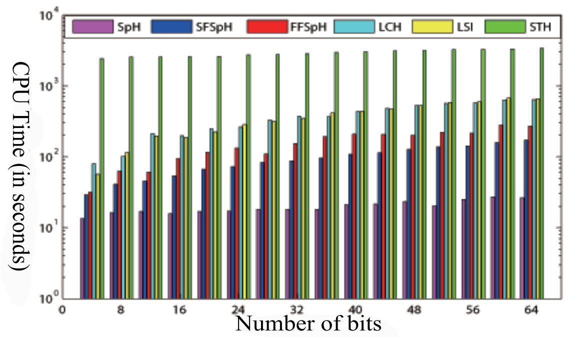


Figure 5: The CPU time comparisons of different methods. Compared with LCH, LSI and STH, which all meet the entropy maximization,the cost of our method is much lower. Though SpH has the lowest cost, it doesn't meet this criterion.

6. ACKNOWLEDGMENTS

This work was supported in part by the National Nature Science Foundation of China (61173054), and the National Key Technology Research and De-velopment Program of China(2012BAH39B02).

7. REFERENCES

[1] D. Zhang, J. Wang, D. Cai, J. Lu, Self-taught hashing for fast similarity search, In ACM SIGIR, 2010.

[2] R. Cipolla, S. Battiato, G. M. Farinella, Machine Learning for Computer Vision, Springer Berlin Heidelberg, 2013.

[3] C. E. Shannon, A mathematical theory of communication, Bell System Technical Journal, 1948.

[4] A. Andoni, P. Indyk, Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions, In FOCS, 2006.

[5] B. Kulis, K. Grauman, Kernelized locality-sensitive hashing, In IEEE PAMI, 2012.

[6] R. Salakhutdinov, G. Hinton, Semantic hashing, In ACM SIGIR, 2007.

[7] A. Torralba, R. Fergus, Y. Weiss, Small codes and large image databases for recognition, In IEEE CVPR, 2008.

[8] Y. Weiss, A. Torralba, R. Fergus, Spectral hashing, Advances in Neural Information Processing Systems, 2009.

[9] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, R. Harshman, Indexing by latent semantic analysis, Journal of the American Society of Information Science, 1990.

[10] D. Zhang, J. Wang, D. Cai, J. Lu, Laplacian co-hashing of terms and documents, In ECIR, 2010.

[11] B. Nadler, S. Lafon, R. R. Coifman, I. G. Kevrekidis, Diffusion maps, spectral clustering and reaction coordinates of dynamical systems, Arxiv, 2008.