Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)

Binary Embedding with Additive Homogeneous Kernels

Saehoon Kim, Seungjin Choi
Department of Computer Science and Engineering
Pohang University of Science and Technology
77 Cheongam-ro, Nam-gu, Pohang 37673, Korea
{kshkawa,seungjin}@postech.ac.kr

Abstract

Binary embedding transforms vectors in Euclidean space into the vertices of Hamming space such that Hamming distance between binary codes reflects a particular distance metric. In machine learning, the similarity metrics induced by Mercer kernels are frequently used, leading to the development of binary embedding with Mercer kernels (BE-MK) where the approximate nearest neighbor search is performed in a reproducing kernel Hilbert space (RKHS). Kernelized locality-sensitive hashing (KLSH), which is one of the representative BE-MK, uses kernel PCA to embed data points into a Euclidean space, followed by the random hyperplane binary embedding. In general, it works well when the query and data points in the database follow the same probability distribution. The streaming data environment, however, continuously requires KLSH to update the leading eigenvectors of the Gram matrix, which can be costly or hard to carry out in practice. In this paper we present a completely randomized binary embedding to work with a family of additive homogeneous kernels, referred to as BE-AHK. The proposed algorithm is easy to implement, built on Vedaldi and Zisserman's work on explicit feature maps for additive homogeneous kernels. We show that our BE-AHK is able to preserve kernel values by developing an upper- and lower-bound on its Hamming distance, which guarantees to solve approximate nearest neighbor search efficiently. Numerical experiments demonstrate that BE-AHK actually yields similarity-preserving binary codes in terms of additive homogeneous kernels and is superior to existing methods in case that training data and queries are generated from different distributions. Moreover, in cases where a large code size is allowed, the performance of BE-AHK is comparable to that of KLSH in general cases.

Introduction

Binary embedding (BE) refers to the methods that transform examples in $\mathbb{R}^d$ into the vertices of Hamming space, i.e., $\{0,1\}^k$, in which the normalized Hamming distance between binary codes preserves a particular distance measure, including angular distance (Charikar 2002) and kernel-induced distance (Kulis and Grauman 2009) (Li, Samorodnitsky, and Hopcroft 2012) (Raginsky and Lazebnik 2009). Most notably, random hyperplane binary embedding (Charikar 2002) involves random projection followed by binary quantization, which aims to preserve angular distance between two vectors.

Randomized binary embedding (RBE) seeks to develop an embedding function without requiring any training data points. Contrary to RBE, *data-dependent binary embedding* (DBE) makes use of a training set to construct compact binary codes (Weiss, Torralba, and Fergus 2008), (Gong and Lazebnik 2011), (Li et al. 2014). We observe that DBE performs poorly in the case that the query and training data points are generated from different distributions. Recently, online DBE (Huang, Yang, and Zhang 2013) (Leng et al. 2015) sequentially learns an embedding function for large-scale or streaming data. However, it still incurs overhead to re-compute all binary codes as a new point arrives. Therefore, it is necessary to develop RBE with different types of streaming data. For example, χ^2 and intersection kernels have been frequently used as a distance metric for histograms, which makes it necessary to develop RBE with such kernels.

Binary embedding with Mercer kernels (BE-MK) (Kulis and Grauman 2009) (Raginsky and Lazebnik 2009) (Mu et al. 2014) (Jiang, Que, and Kulis 2015) employs feature maps (kernel PCA or Nyström approximation) followed by RBE, such that the normalized Hamming distance between codes preserves Mercer kernels. Since it requires training examples to build the feature map, we observe that it might not be adequate for streaming environment. For example, kernelized locality-sensitive hashing (KLSH) (Jiang, Que, and Kulis 2015), which is one of the representative example of BE-MK, employs KPCA for the feature map, which requires a set of training data points to compute the leading eigenvector of Gram matrix. If the data distribution changes over time, the performance of KLSH is steadily degraded over time.

In this paper, we propose a completely randomized binary embedding with additive homogeneous kernels, referred to as RBE-AHK, where data points are embedded onto $\mathbb{R}^m$ by the explicit feature map for additive homogeneous kernels (Vedaldi and Zisserman 2012) and then are transformed into the vertices of Hamming space by the random hyperplane binary embedding. The contribution of this paper is summarized below.

- We propose a RBE algorithm for additive homogeneous kernels and conduct the numerical experiments to show

that the proposed algorithm is superior to existing BE-MK methods in case that training data and queries are generated from different distributions.

- We present the lower and upper bounds on Hamming distance between binary codes generated by the proposed algorithm, which guarantees to solve approximate nearest neighbor search problem and large-scale machine learning efficiently.

Background

In this section, we briefly review some prerequisites to describe the proposed algorithm.

Random Hyperplane Binary Embedding

Random hyperplane binary embedding (Charikar 2002), referred to as RHBE, involves a random projection followed by binary quantization, whose an embedding function is formally defined as $h(\boldsymbol{x}) \triangleq \text{sgn}\left(\boldsymbol{w}^\top \boldsymbol{x}\right)$, where $\boldsymbol{w} \in \mathbb{R}^d$ is a random vector sampled on a unit d-sphere and $\text{sgn}(\cdot)$ is the sign function which returns 1 whenever the input is non-negative and -1 otherwise. It was shown in (Charikar 2002) that RHBE naturally gives an unbiased estimator of angular distance such that the expectation of Hamming distance is the angle between two vectors, i.e.,

$$\mathbb{E}\Big[\mathcal{I}\left[h(\boldsymbol{x}) \neq h(\boldsymbol{y})\right]\Big] \quad = \quad \frac{\theta_{\boldsymbol{x},\boldsymbol{y}}}{\pi}, \tag{1}$$

where $\mathcal{I}[\cdot]$ is an indicator function which returns 1 whenever the input argument is true and 0 otherwise, and $\theta_{\boldsymbol{x},\boldsymbol{y}}$ denotes the angle between two vectors. It is easy to verify that RHBE is $(r, r(1+\epsilon), 1-\frac{r}{\pi}, 1-\frac{r(1+\epsilon)}{\pi})$-sensitive locality-sensitive hashing family (Indyk and Motwani 1998), (Gionis, Indyk, and Motawani 1999), leading to a $O(n^\rho)$ time complexity algorithm to solve an approximate nearest neighbor search problem, where $\rho = \frac{\log(1-\frac{r}{\pi})}{\log(1-\frac{r(1+\epsilon)}{\pi})}$.

Binary Embedding with Mercer Kernels

Binary embedding with Mercer kernels (BE-MK) (Kulis and Grauman 2009) (Mu et al. 2014) (Jiang, Que, and Kulis 2015) employs feature maps followed by RBE, which is defined as follows:

$$h(\boldsymbol{x}) = \text{sgn}\left(\boldsymbol{w}^\top \boldsymbol{\phi}(\boldsymbol{x})\right), \tag{2}$$

where $\boldsymbol{\phi}(\cdot)$ is a feature map (kernel PCA or Nyström approximation) for a particular Mercer kernel and $\boldsymbol{w}$ is a random vector in $\mathbb{R}^m$. For example, kernelized locality-sensitive hashing (KLSH) (Kulis and Grauman 2009) (Jiang, Que, and Kulis 2015) for a kernel $k(\cdot,\cdot)$ involves the following feature mapping:

$$\boldsymbol{\phi}(\boldsymbol{x}) \triangleq \boldsymbol{U}_k^\top \left[k(\boldsymbol{x}_1^*, \boldsymbol{x}); \cdots; k(\boldsymbol{x}_m^*, \boldsymbol{x})\right], \tag{3}$$

where $\boldsymbol{U}_k$ is the k leading eigenvectors of the Gram matrix. Instead of such *data-dependent* embedding, a fully randomized binary embedding is developed to preserve Mercer kernels, which includes χ^2 kernel (Li, Samorodnitsky, and Hopcroft 2012) and shift-invariant kernels (Raginsky and Lazebnik 2009).

Table 1: Additive homogeneous kernels with closed-form feature maps.

kernel	$k(x,y)$	$\Phi_w(x)$
Hellinger's	$\sqrt{xy}$	$\sqrt{x}$
χ^2	$2\frac{xy}{x+y}$	$e^{iw\log x}\sqrt{x\text{sech}(\pi w)}$
intersection	min{x, y}	$e^{iw\log x}\sqrt{\frac{2x}{\pi}\frac{1}{1+4w^2}}$

For existing algorithms, we observe the following limitations:

- It requires a training dataset for data-dependent binary embedding to construct feature maps, resulting in the poor performance when the query and training data are generated from very different distributions.
- Up to our knowledge, there does not exist a completely randomized algorithm to work with a large family of kernels. For example, additive homogeneous kernels, which includes very common kernels (e.x. χ^2 kernels, intersection kernels, etc.), are not be considered in RBE.

Explicit Feature Maps for Additive Homogeneous Kernels

Additive homogeneous kernels are said to be a family of Mercer kernels $K : \mathbb{R}_+^d \times \mathbb{R}_+^d \to \mathbb{R}$, which is defined as follows:

$$K(\boldsymbol{x},\boldsymbol{y}) \quad = \quad \sum_{i=1}^{d} k(x_i, y_i), \quad \forall(\boldsymbol{x},\boldsymbol{y}) \in \mathbb{R}_+^d \times \mathbb{R}_+^d,$$

where $k(\cdot,\cdot)$ is a homogeneous kernel and x_i is the ith element of $\boldsymbol{x}$. There exist many popular kernels in the family, including Hellinger's, χ^2, and intersection kernels, which have been frequently used for distance measures between histograms. For the sake of simplicity, kernels are always meant to be additive homogeneous kernels in this paper.

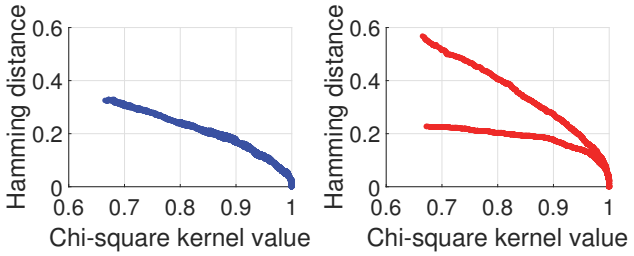
According to (Vedaldi and Zisserman 2012), homogeneous kernels can be represented by the explicit feature map $\Phi_w(\cdot)$ such that

$$k(x,y) = \int_{-\infty}^{\infty} \Phi_w(x)^* \Phi_w(y) dw, \tag{4}$$

where $k(x,y)$ is a homogeneous kernel and the feature maps associated with homogeneous kernels are described in Table 1. To approximate it in a finite-dimensional vector, (Vedaldi and Zisserman 2012) constructs m-dimensional feature maps denoted as $\widehat{\boldsymbol{\Phi}}_m(\cdot)$ by proposing an efficient technique to sample w in Eq. 4. For example, if m samples are used, the kernel is approximated by $2m+1$-dimensional feature maps. Therefore, in case of additive homogeneous kernel, the kernel is approximated by $d(2m+1)$-dimensional features, where d is the data dimension.

Proposed Method

In this section, we propose a randomized binary embedding with additive homogeneous kernels, which is referred to as



(a) Biased selection (b) Random selection

Figure 1: Scatter plots of the normalized Hamming distance versus χ^2 kernel values on the two-dimensional synthetic dataset. The left panel is the case where the landmarks are selected from a limited domain, in which the one of dimension is limited to [0.8, 1]. The right panel is the case where the landmarks are randomly selected from the entire domain.

BE-AHK. Before describing the details of the proposed algorithm, we describe the main motivation of this algorithm.

Motivation

Kernelized LSH (KLSH), along with other data-dependent binary embedding with Mercer kernels, requires a set of training points, which results in performance degradation in the case that queries are generated from the different distribution of the training points. The degradation is caused by the inaccurate estimation of the kernel values between the query and data points if landmark points are not sufficiently selected to cover the data distribution.

Figure 1 represents the scatter plot of the normalized Hamming distance versus χ^2 kernel values on the two-dimensional synthetic dataset, in which all data points are the absolute values of random samples from a two-dimensional normal distribution with zero mean and unit variance. The points are transformed into 1K bits by KLSH. As observed in Figure 1, if the landmark points are biased (i.e. not sufficient to cover the whole dataset), KLSH fails to preserve kernel values. Since the data distribution steadily changes in a streaming environment, it is necessary to develop a completely randomized binary embedding algorithm for additive homogeneous kernels.

Algorithm

The proposed algorithm, BE-AHK, is composed of the following steps: (1) data points are embedded onto a m-dimensional space by explicit feature maps (Vedaldi and Zisserman 2012), and (2) the embedded points are transformed into binary codes by random hyperplane binary embedding. Specifically, given an example $\boldsymbol{x} \in \mathbb{R}^d$, an embedding function is defined by

$$h(\boldsymbol{x}) \triangleq \text{sgn}\left(\boldsymbol{w}^\top \widehat{\boldsymbol{\Phi}}_m(\boldsymbol{x})\right), \tag{5}$$

where $\widehat{\boldsymbol{\Phi}}_m(\boldsymbol{x})$ is a m-dimensional feature map described in Eq. 4 and $\boldsymbol{w} \in \mathbb{R}^m$ drawn from $\mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$. It is worth noting that the length of feature map $\widehat{\boldsymbol{\Phi}}_m(\boldsymbol{x})$ cannot be larger than 1 and always the same regardless of input vectors, which is formally described in the next section.

Even though BE-AHK is easy to implement, it yields the similarity-preserving binary codes in terms of additive homogeneous kernels, which can be used for approximate nearest neighbor search by addressing the following questions:

- Can BE-AHK approximate the kernel values up to small distortion? According to (Vedaldi and Zisserman 2012), the explicit feature map leads to small approximation error between $K(\boldsymbol{x}, \boldsymbol{y})$ and $\widehat{\boldsymbol{\Phi}}_m(\boldsymbol{x})^\top \widehat{\boldsymbol{\Phi}}_m(\boldsymbol{y})$. However, due to the presence of sign function, it is not clear how such error is related to binary embedding.
- Can BE-AHK provide an efficient solution to approximate nearest neighbor search problem or large-scale machine learning? For example, if BE-AHK yields similarity-preserving binary codes, it naturally leads to a sub-linear running time algorithm for retrieving nearest neighbors.

Theoretical Analysis

In this subsection, we analyze BE-AHK to prove that it preserves additive homogeneous kernels. First, we present Corollary 1 to show that the length of explicit feature maps cannot be larger than 1.

Corollary 1. *Given d-dimensional histograms $\boldsymbol{x}$ and $\boldsymbol{y}$, let $\widehat{\boldsymbol{\Phi}}_m(\boldsymbol{x})$ be the explicit feature map to approximate additive homogeneous kernels. Suppose that the feature map is obtained by the uniform window and a period Λ larger than 2π. Then, for any $\boldsymbol{x} \in \mathbb{R}^d_+$, $\widehat{\boldsymbol{\Phi}}_m(\boldsymbol{x})^\top \widehat{\boldsymbol{\Phi}}_m(\boldsymbol{x}) \leq 1$.*

Proof. According to Eq. 20 (Vedaldi and Zisserman 2012),

$$\begin{aligned} \widehat{\boldsymbol{\Phi}}_m(\boldsymbol{x})^\top \widehat{\boldsymbol{\Phi}}_m(\boldsymbol{x}) &\leq \sum_{j=-\infty}^{+\infty} \widehat{k}_j \\ &= \sum_{j=-\infty}^{+\infty} \frac{2\pi}{\Lambda} k(j\frac{2\pi}{\Lambda}), \end{aligned}$$

where $\widehat{k}_j \triangleq \frac{2\pi}{\Lambda} k(j\frac{2\pi}{\Lambda})$ and $k(\cdot)$ is the spectrum of an additive homogeneous kernel. Since $\int_{-\infty}^{+\infty} k(x)dx = 1$, it is trivial to show that $\widehat{\boldsymbol{\Phi}}_m(\boldsymbol{x})^\top \widehat{\boldsymbol{\Phi}}_m(\boldsymbol{x}) \leq 1$. □

Moreover, it is easy to show that the length of feature map should be always the same regardless of input vectors, because the spectrum of kernel is invariant under the inputs. Therefore, Corollary 1 says that all data points transformed by feature maps lie on a sphere, which makes it reasonable to apply RHBE into the transformed points.

We present Corollary 2 to show how many samples for the feature map are needed to approximate kernels up to δ-distortion.

Corollary 2. *Let $K(\boldsymbol{x}, \boldsymbol{y})$ be a smooth additive homogeneous kernel (i.e. χ^2 or JS kernel) for any d-dimensional histograms $\boldsymbol{x}$ and $\boldsymbol{y}$. Given $\delta \in (0, 1)$, there exists*

$O\left(d \log^2 \frac{1}{\delta}\right)$-dimensional feature map $\widehat{\mathbf{\Phi}}_m(\cdot)$ to approximate the kernels up to δ-distortion such that

$$|K(\boldsymbol{x}, \boldsymbol{y}) - \widehat{\mathbf{\Phi}}_m(\boldsymbol{x})^\top \widehat{\mathbf{\Phi}}_m(\boldsymbol{y})| < \delta.$$

Proof. The proof is a simple consequence of Lemma 3 in (Vedaldi and Zisserman 2012). □

Corollary 2 shows that the dimension of feature map grows logarithmically to approximate smooth kernels [1] up to δ-distortion, which makes it possible to drive the tight upper and lower bounds on Hamming distance.

Theorem 1. *Given $\delta \in (0, 1)$, suppose that the feature map $\widehat{\mathbf{\Phi}}_m(\cdot)$ approximates the kernel $K(\cdot, \cdot)$ up to δ-distortion. Then, the lower and upper bounds on Hamming distance induced by Eq. 5 are summarized as*

$$f\left(K(\boldsymbol{x}, \boldsymbol{y})\right) \leq \mathbb{E}\Big[\mathcal{I}\left[h(\boldsymbol{x}) \neq h(\boldsymbol{y})\right]\Big] \leq g\left(K(\boldsymbol{x}, \boldsymbol{y})\right), \quad (6)$$

where $f(\cdot)$ and $g(\cdot)$ are defined as

$$f(x) = -\frac{3}{5\pi}\left(\frac{x+\delta}{1-\delta}\right)^3 - \frac{x+\delta}{\pi(1-\delta)} + \frac{1}{2} \quad (7)$$

$$g(x) = -\frac{(x-\delta)^3}{6\pi} - \frac{x-\delta}{\pi} + \frac{1}{2}. \quad (8)$$

Proof. According to Eq. 1,

$$\begin{aligned}\mathbb{E}\Big[\mathcal{I}\left[h(\boldsymbol{x}) \neq h(\boldsymbol{y})\right]\Big] &= \frac{\theta_{\widehat{\mathbf{\Phi}}_m(\boldsymbol{x}), \widehat{\mathbf{\Phi}}_m(\boldsymbol{y})}}{\pi} \\ &= \frac{1}{\pi}\cos^{-1}\left(\frac{\widehat{\mathbf{\Phi}}_m(\boldsymbol{x})^\top \widehat{\mathbf{\Phi}}_m(\boldsymbol{y})}{||\widehat{\mathbf{\Phi}}_m(\boldsymbol{x})||||\widehat{\mathbf{\Phi}}_m(\boldsymbol{y})||}\right).\end{aligned}$$

According to Corollary 1 and 2, it is trivial to show that

$$K(\boldsymbol{x}, \boldsymbol{y}) - \delta \leq \frac{\widehat{\mathbf{\Phi}}_m(\boldsymbol{x})^\top \widehat{\mathbf{\Phi}}_m(\boldsymbol{y})}{||\widehat{\mathbf{\Phi}}_m(\boldsymbol{x})||||\widehat{\mathbf{\Phi}}_m(\boldsymbol{y})||} \leq \frac{K(\boldsymbol{x}, \boldsymbol{y}) + \delta}{1-\delta}.$$

Moreover, Corollary 1 also says that $\frac{\widehat{\mathbf{\Phi}}_m(\boldsymbol{x})^\top \widehat{\mathbf{\Phi}}_m(\boldsymbol{y})}{||\widehat{\mathbf{\Phi}}_m(\boldsymbol{x})||||\widehat{\mathbf{\Phi}}_m(\boldsymbol{y})||}$ is bounded from $[0, 1]$. Then, we obtain the following inequalities:

$$\begin{aligned}\cos^{-1}\left(\frac{K(\boldsymbol{x}, \boldsymbol{y}) + \delta}{1-\delta}\right) &\leq \cos^{-1}\left(\frac{\widehat{\mathbf{\Phi}}_m(\boldsymbol{x})^\top \widehat{\mathbf{\Phi}}_m(\boldsymbol{y})}{||\widehat{\mathbf{\Phi}}_m(\boldsymbol{x})||||\widehat{\mathbf{\Phi}}_m(\boldsymbol{y})||}\right) \\ \cos^{-1}\left(K(\boldsymbol{x}, \boldsymbol{y}) - \delta\right) &\geq \cos^{-1}\left(\frac{\widehat{\mathbf{\Phi}}_m(\boldsymbol{x})^\top \widehat{\mathbf{\Phi}}_m(\boldsymbol{y})}{||\widehat{\mathbf{\Phi}}_m(\boldsymbol{x})||||\widehat{\mathbf{\Phi}}_m(\boldsymbol{y})||}\right).\end{aligned}$$

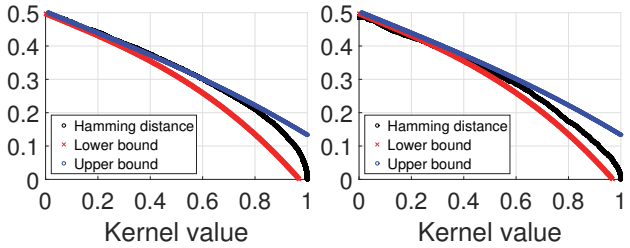
First, we derive the lower bound on Hamming distance. Considering the lower bound on the inverse cosine function,

$$\cos^{-1}(x) \geq -\frac{3}{5}x^3 - x + \frac{\pi}{2}, \quad \text{for } x \in [0, 1],$$

we obtain

$$\begin{aligned}&-\frac{3}{5\pi}\left(\frac{K(\boldsymbol{x}, \boldsymbol{y}) + \delta}{1-\delta}\right)^3 - \frac{K(\boldsymbol{x}, \boldsymbol{y}) + \delta}{\pi(1-\delta)} + \frac{1}{2} \\ \leq\ & \mathbb{E}\Big[\mathcal{I}\left[h(\boldsymbol{x}) \neq h(\boldsymbol{y})\right]\Big].\end{aligned}$$

[1] For non-smooth kernels including an intersection kernel, $O(d\delta^{-\frac{1}{c-1}})$-dimensional feature maps should be used to approximate the kernel up to δ-distortion, where $c > 1$ is a constant.



(a) χ^2 kernel (b) Intersection kernel

Figure 2: Plots of the upper and lower bounds on Hamming distance in case of $\delta = 0.01$. The blue (red) dotted line shows the upper (lower) bound and the black circles mean the normalized Hamming distance on the synthetic dataset, in which two-dimensional 10K data points sampled from χ^2 distribution (with one degree of freedom) are transformed into 10K bits.

Second, we derive the upper bound on Hamming distance. According to Taylor expansion of $\cos^{-1}(\cdot)$,

$$\cos^{-1}(x) = \frac{\pi}{2} - \sum_{n=0}^{\infty} C(n)\,(x)^{2n+1},$$

where $C(n) = \binom{2n}{n}/(4^n(2n+1))$. Then,

$$\cos^{-1}(x) \leq -\frac{1}{6}x^3 - x + \frac{\pi}{2}.$$

Therefore, we can derive the lower bound on the collision probability:

$$\begin{aligned}&\mathbb{E}\Big[\mathcal{I}\left[h(\boldsymbol{x}) \neq h(\boldsymbol{y})\right]\Big] \\ \leq\ & -\frac{(K(\boldsymbol{x}, \boldsymbol{y}) - \delta)^3}{6\pi} - \frac{K(\boldsymbol{x}, \boldsymbol{y}) - \delta}{\pi} + \frac{1}{2}.\end{aligned}$$

□

As shown in Figure 2, Theorem 1 roughly says that the normalized Hamming distance between binary codes is related to their kernel value in case that an explicit feature map approximates a kernel up to small distortion. According to (Indyk and Motwani 1998), (Gionis, Indyk, and Motawani 1999), it is trivial to show that BE-AHK can be used as LSH to retrieve approximate nearest neighbors in sub-linear time complexity [2], because it yields similarity-preserving binary codes. Due to the page limit, we omit the details of LSH construction and its analysis.

Numerical Experiments

In this section, we conducted numerical experiments to validate the benefits of BE-AHK compared to existing methods, which includes of KLSH (Jiang, Que, and Kulis 2015),

[2] A better version of LSH in terms of query time (Andoni et al. 2014) (Andoni and Razenshteyn 2015) has been recently proposed, which works well when all data points lie on a sphere. However, they do not consider a kernel-induce distance metric.

RHBE(Charikar 2002), and sign Cauchy random projection (Li, Samorodnitsky, and Hopcroft 2012). Specifically, BE-AHK is at least comparable to existing methods in general cases and superior to them when queries and landmark points are from different distributions. For additive homogeneous kernels, we selected the following kernels:

$$K_{\chi^2}(\boldsymbol{x}, \boldsymbol{y}) \triangleq \sum_{i=1}^{d} 2\frac{x_i y_i}{x_i + y_i},$$

$$K_{inters}(\boldsymbol{x}, \boldsymbol{y}) \triangleq \sum_{i=1}^{d} \min(x_i, y_i),$$

where $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$ and x_i is the i-th component of $\boldsymbol{x}$.

For experiments, we used two popular datasets:

- MNIST is composed of 784-dimensional 50,000 training data and 10,000 testing data with 10 classes.
- GIST1M (Jégou, Douze, and Schmid 2011) is composed of 920-dimensional 1 million GIST descriptors with additional 1,000 queries.

For both datasets, all data points are L_1 normalized.

For a fair comparison, we used the following configuration of compared methods:

- BE-AHK (the proposed method) has one hyperparameter to set the number of samples per dimension to construct feature maps. We used three (ten) samples per dimension for χ^2 (intersection) kernel.
- KLSH (Jiang, Que, and Kulis 2015) has two hyperparameters: the number of landmark points and the rank of KPCA. We fixed the number of landmark points to be 1,000. We tested different ranks for KPCA, because it it sensitive to the performance of KLSH.
- RHBE(Charikar 2002) and SCR (Li, Samorodnitsky, and Hopcroft 2012) do not have any tuning parameters.

To avoid any bias, we repeated all experiments five times to produce the mean and standard deviation.

Kernel Preservation Evaluation

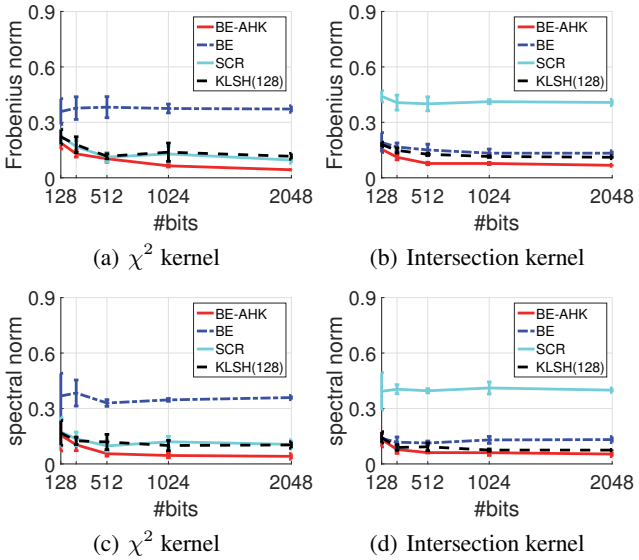
We measured the difference between the normalized Hamming distance and "acos-kernel", which is defined as follow:

$$\frac{\cos^{-1}\left(K(\boldsymbol{x}, \boldsymbol{y})\right)}{\pi},$$

where $K(\cdot, \cdot)$ is an additive kernel. According to (Li, Samorodnitsky, and Hopcroft 2012), the performance of non-linear classifiers with "acos-kernel" is similar to the classifier with original kernel. It motivates us to measure how much the normalized Hamming distance can preserve the "acos-kernel" by the following metric used in (Yang et al. 2014):

$$\frac{||\boldsymbol{H} - \boldsymbol{K}||_F}{||\boldsymbol{K}||_F} \quad \text{and} \quad \frac{||\boldsymbol{H} - \boldsymbol{K}||_2}{||\boldsymbol{K}||_2},$$

where $[\boldsymbol{H}]_{ij}$ is the normalized Hamming distance between two vectors and $[\boldsymbol{K}]_{ij}$ is the "acos-kernel" of two points. $||\cdot||_F$ and $||\cdot||_2$ are Frobenius and spectral norms.



(a) χ^2 kernel (b) Intersection kernel

(c) χ^2 kernel (d) Intersection kernel

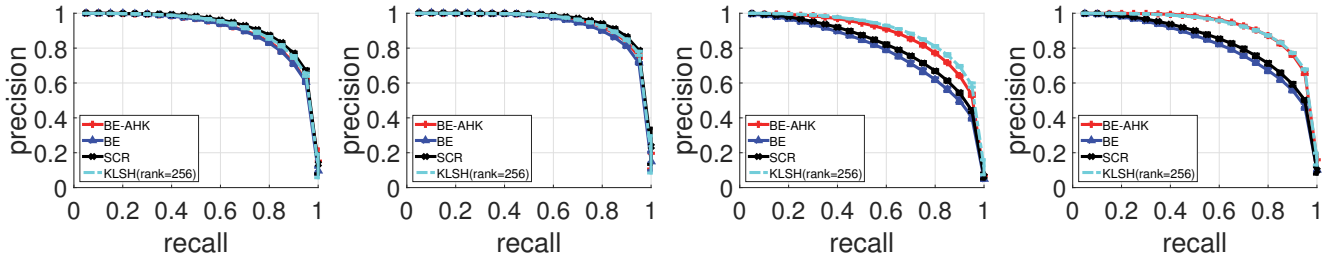
Figure 3: Plots for the difference between the normalized Hamming distance and "acos-kernel" values on the subset of GIST1M dataset with respect to the number of bits, in which KLSH(32) means that the rank of KPCA is 32. The first row is obtained by Frobenius norm and the second one by spectral norm.

Figure 3 shows the difference between Hamming distance and "acos-kernel" with respect to the number of bits on a subset of GIST1M dataset, where 5,000 points are randomly chosen. Clearly, BE-AHK almost acts as an unbiased estimator of "acos-kernel", because the difference becomes zero when the number of bits increases. It is worth noting that SCR does not work with an intersection kernel, which makes it inappropriate to use SCR with other kernels instead of χ^2 kernel.

Hamming Ranking Evaluation

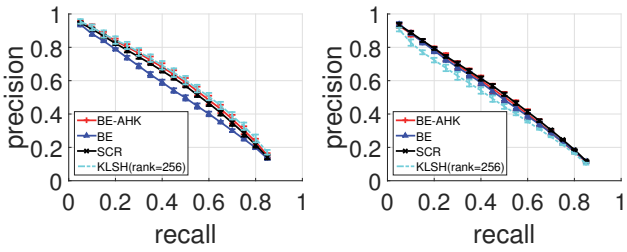
We computed precision-recall curves on MNIST and GIST1M datasets with respect to the number of bits to compare BE-AHK with existing methods in terms of approximate nearest neighbor search. For both datasets, we randomly selected 100 queries from test sets and computed 100 nearest neighbors for ground-truths, in which χ_2 or intersection kernels are used for similarity measures. We followed Hamming ranking to compute precision and recall, which is one of the standard measures (Wang, Kumar, and Chang 2010a) (Wang, Kumar, and Chang 2010b) to evaluate binary embedding or LSH, in which distance between query and data points is computed by Hamming distance and is sorted in ascending order to find the nearest neighbors of query.

Figure 4 and 5 represent precision-recall curves on MNIST and GIST1M datasets. In any cases, we observed that BE-AHK is comparable to existing methods in case of a large code size. Specifically, BE-AHK and KLSH work well for both χ^2 and intersection kernels. However, as shown in the second row in Figure 4, BE and SCR do not perform well



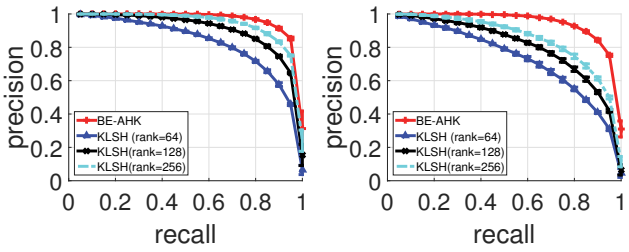
(a) χ_2, #bit=2,048 (b) χ_2, #bit=4,096 (c) inters, #bit=2,048 (d) inters, #bit=4,096

Figure 4: Precision-recall curves on MNIST with respect to the number of bits, in which KLSH(rank=256) means that KLSH uses KPCA with rank 256. The first two figures represent the results for χ_2 and the rest ones for the intersection kernel.



(a) χ^2 kernel (b) Intersection kernel

Figure 5: Precision-recall curves on GIST1M with 8,192bits, in which the left figure shows the results for χ_2 and the right one for the intersection kernel.



(a) χ^2 kernel (b) Intersection kernel

Figure 6: Precision-recall curves on MNIST with 8,192bits when landmark points and queries are generated from different distribution. The left figure shows the results for χ_2 and the right one for the intersection kernel.

with an intersection kernel on MNIST, because the nearest neighbors computed by angular and χ^2 distances are very different from the ones by an intersection kernel. As observed in Figure 3, BE-AHK more accurately estimates kernel values as the number of bit increases, which makes it reasonable to observe that BE-AHK works well for both kernels when a large code size is given.

Finally, we compared BE-AHK with KLSH in the case that the training data points and queries are generated from different distributions, which is designed to mimic streaming data environment. In order to do that, the data points are grouped into ten clusters and landmark points are chosen from one of the cluster, which naturally induces different distributions for training data and queries. As shown in Figure 6, it is very clear that the performance of KLSH is significantly dropped compared to Figure 4 while the performance of BE-AHK is the same. Since BE-AHK is a data-independent algorithm, we observe that it is superior to KLSH in streaming data environment.

Conclusion

We proposed a completely randomized binary embedding to work with a family of additive homogeneous kernels, referred to as BE-AHK. The proposed algorithm is built on Vedaldi and Zisserman's work on explicit feature maps for additive homogeneous kernels, which consists of two steps: (1) data points are embedded onto a m-dimensional space by explicit feature maps, and (2) the embedded points are transformed into binary codes by random hyperplane binary embedding. We theoretically and empirically confirmed that BE-AHK is able to generate similarity-preserving binary codes, which guarantees to retrieve nearest neighbors efficiently. For future work, we will extend BE-AHK to work with other families of kernels, such as shift-invariant kernels or polynomial kernels.

Acknowledgements

This work was supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIP) (No.B0101-16-0307, Basic Software Research in Human-level Lifelong Machine Learning (Machine Learning Center)). S. Kim was partially supported by POSTECH-Qualcomm Ph.D. fellowship award (2015).

References

Andoni, A., and Razenshteyn, I. 2015. Optimal data-dependet hashing for approximate near neighbors. In *Proceedings of the Annual ACM Symposium on Theory of Computing (STOC)*.

Andoni, A.; Indyk, P.; guyen, H. L.; and Razenshteyn, I.

2014. Beyond locality-sensitive hashing. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA)*.

Charikar, M. S. 2002. Similarity estimation techniques from rounding algorithms. In *Proceedings of the Annual ACM Symposium on Theory of Computing (STOC)*.

Gionis, A.; Indyk, P.; and Motawani, R. 1999. Similarity search in high dimensions via hashing. In *Proceedings of the International Conference on Very Large Data Bases (VLDB)*.

Gong, Y., and Lazebnik, S. 2011. Iterative quantization: A procrustean approach to learning binary codes. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*.

Huang, L.-K.; Yang, Q.; and Zhang, W.-S. 2013. Online hashing. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*.

Indyk, P., and Motwani, R. 1998. Approximate nearest neighbor towards removing the curse of dimensionality. In *Proceedings of the Annual ACM Symposium on Theory of Computing (STOC)*, 604–613.

Jégou, H.; Douze, M.; and Schmid, C. 2011. Product quantization for nearest neighbor search. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33.

Jiang, K.; Que, Q.; and Kulis, B. 2015. Revisiting kernelized locality-sensitive hashing for improved large-scale image retrieval. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*.

Kulis, B., and Grauman, K. 2009. Kernelized locality-sensitive hashing for scalable image seaerch. In *Proceedings of the International Conference on Computer Vision (ICCV)*.

Leng, C.; Wu, J.; Cheng, J.; Bai, X.; and Lu, H. 2015. Online sketching hashing. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*.

Li, W.; Mu, C.; Kumar, S.; and Chang, S.-F. 2014. Discrete graph hashing. In *Advances in Neural Information Processing Systems (NIPS)*.

Li, P.; Samorodnitsky, G.; and Hopcroft, J. 2012. Sign cauchy projections and chi-square kernel. In *Advances in Neural Information Processing Systems (NIPS)*.

Mu, Y.; Hua, G.; Fan, W.; and Chang, S.-F. 2014. Hash-svm: scalable kernel machines for large-scale visual classification. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*.

Raginsky, M., and Lazebnik, S. 2009. Locality-sensitive binary codes from shift-invariant kernels. In *Advances in Neural Information Processing Systems (NIPS)*, volume 22. MIT Press.

Vedaldi, A., and Zisserman, A. 2012. Efficient additive kernels via explicit feature maps. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34.

Wang, J.; Kumar, S.; and Chang, S. F. 2010a. Semi-supervised hashing for scalable image retrieval. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*.

Wang, J.; Kumar, S.; and Chang, S. F. 2010b. Sequential projection learning for hashing with compact codes. In *Proceedings of the International Conference on Machine Learning (ICML)*.

Weiss, Y.; Torralba, A.; and Fergus, R. 2008. Spectral hashing. In *Advances in Neural Information Processing Systems (NIPS)*, volume 20. MIT Press.

Yang, J.; Sindhwani, V.; Avron, H.; and Mahoney, M. W. 2014. Quasi-monte carlo feature maps for shift-invariant kernels. In *Proceedings of the International Conference on Machine Learning (ICML)*.