Improving Locality Sensitive Hashing Based Similarity Search and Estimation for Kernels

Aniket Chakrabarti (✉), Bortik Bandyopadhyay, and Srinivasan Parthasarathy

Department of Computer Science and Engineering
The Ohio State University
{chakrabarti.14,bandyopadhyay.14}@osu.edu,srini@cse.ohio-state.edu

Abstract. We present a novel data embedding that significantly reduces the estimation error of locality sensitive hashing (LSH) technique when used in reproducing kernel Hilbert space (RKHS). Efficient and accurate kernel approximation techniques either involve the kernel principal component analysis (KPCA) approach or the Nyström approximation method. In this work we show that extant LSH methods in this space suffer from a bias problem, that moreover is difficult to estimate apriori. Consequently, the LSH estimate of a kernel is different from that of the KPCA/Nyström approximation. We provide theoretical rationale for this bias, which is also confirmed empirically. We propose an LSH algorithm that can reduce this bias and consequently our approach can match the KPCA or the Nyström methods' estimation accuracy while retaining the traditional benefits of LSH. We evaluate our algorithm on a wide range of realworld image datasets (for which kernels are known to perform well) and show the efficacy of our algorithm using a variety of principled evaluations including mean estimation error, KL divergence and the Kolmogorov-Smirnov test.

Keywords: Locality Sensitive Hashing, Kernel Similarity Measure, Similarity Estimation, Nyström Method

1 Introduction

In recent past, Locality Sensitive Hashing (LSH) [1] has gained widespread importance in the area of large scale machine learning. Given a high dimensional dataset and a distance/similarity metric, LSH can create a small sketch (low dimensional embedding) of the data points such that the distance/similarity is preserved. LSH is known to provide approximate and efficient solution for estimating the pairwise similarity among data points, which is critical in solving applications for many domains ranging from image retrieval to text analytics and from protein sequence clustering to pharmacogenomics. Recently kernel-based similarity measures [22] have found increased use in such scenarios in part because the data becomes easily separable in the kernel induced feature space. The challenges of working with kernels are two fold – (1) explicit embedding of data points in the kernel induced feature space (RKHS) may be unknown

or infinite dimensional and (2) generally the kernel function is computationally expensive. The first problem prohibits building of a smart index structure such as kdtrees [3] that can allow efficient querying, while the second problem makes constructing the full kernel matrix infeasible.

LSH has been used in the context of kernels to address both of the aforementioned problems. Existing LSH methods for kernels [13, 12] leverage the KPCA or Nyström techniques to estimate the kernel. The two methods differ only in the form of covariance operator that they use in the eigenvector computation step to approximately embed the data in RKHS. While KPCA uses the centered covariance operator, Nyström method uses the uncentered one (second moment operator). Without loss of generality, for the rest of the paper, we will use the Nyström method and hence by covariance operator we will mean the uncentered one. The LSH estimates for kernel differ significantly from the Nyström approximation. This is due to the fact that the projection onto the subspace (spanned by the eigenvectors of covariance operator) results in reduction of norms of the data points. This reduction depends on the eigenvalue decay rate of the covariance operator. Therefore, this norm reduction is difficult to estimate apriori. Assume that the original kernel was normalized with norm of the data points (self inner product) equaling 1. As a consequence of this norm reduction, in the resulting subspace the Nyström approximated kernel is not normalized (self inner product less than 1). Now, it is shown in [6] that LSH can only estimate normalized kernels. Thus in the current setting, instead of the Nyström approximated kernel, it estimates the *re-normalized version of it*. The bias arising out of this re-normalization depends on the eigenvalue decay rate of the covariance operator, and is unknown to the user apriori. This is particularly problematic, since for the LSH applications (index building and estimation) in the context of similarity (not distance), accurate estimation is paramount. For instance, the *All Pairs Similarity Search* (APSS) [2, 5, 4] problem finds all pairs of data points whose similarity is above a user defined threshold. Therefore, APSS quality will degrade in case of high estimation error. In APSS using LSH [5], it is clearly noticeable that the quality for non-kernel similarity measures is better than their kernel counterparts.

We propose a novel embedding of data points that is amenable to LSH sketch generation, while still estimating the Nyström approximated kernel matrix instead of the re-normalized version (which is the shortcoming of existing work). Specifically the contributions of this paper are as follows:

1. We show that Nyström embedding based LSH generates the LSH embedding for a slightly different kernel rather than the Nyström approximated one. This bias becomes particularly important during the LSH index construction where similarity threshold (or distance radius) is a mandatory parameter. Since this radius parameter is given in terms of the original similarity (kernel) measure, if the LSH embedding results in a bias (estimating a slightly different kernel), then the resulting index generated will be incorrect.

2. We propose an LSH scheme to estimate the Nyström approximation of the original input kernel and develop an algorithm for efficiently generating the LSH embedding.
3. Finally we empirically evaluate our methods against state-of-the-art KLSH [12, 13] and show that our method is substantially better in estimating the original kernel values. We additionally run statistical tests to prove that the statistical distribution of pairwise similarity in the dataset is better preserved by our method. Preserving the similarity distribution correctly is particularly important in applications such as clustering.

Our results indicate upto 9.7x improvement in the kernel estimation error and the KL divergence and Kolmogorov-Smirnov tests [15] show that the estimates from our method fit the pairwise similarity distribution of the ground truth substantially better than the state-of-the-art KLSH method.

2 Background and Related Works

Table 1: Key Symbols

n, d	Number of data points, Dimensionality of data
p, c	Parameters: Number of eigenvectors to use, Number of extra dimensions to use
$\kappa(x, y)$	Kernel function over data points x, y $(=< \Phi(x), \Phi(y) >)$
$\Phi(x)$	Kernel induced feature map for data point x
X_i	i^{th} data point (i^{th} row of matrix $X_{n\times d}$)
$K_{i,j}$	$(i,j)^{th}$ value of true kernel matrix $K_{n\times n}$ $(= \kappa(X_i, X_j))$
Y_i	p dimensional Nystroḿ embedding of $\Phi(X_i)$ (i^{th} row of $Y_{n\times p}$ matrix)
$\widehat{K_{i,j}}$	Approximation of $K_{i,j}$ due to Nystroöm embedding $(=< Y_i, Y_j >)$
Z_i	Our $(p+n)$ dimensional Augmented Nystroḿ embedding of $\Phi(X_i)$
$\widehat{K_{(Z)i,j}}$	Approximation of $K_{i,j}$ due to Augmented Nystroḿ embedding $(=< Z_i, Z_j >)$
Z_i'	Our $(p+c)$ dimensional Remapped Augmented Nystroḿ embedding of $\Phi(X_i)$

2.1 LSH for Cosine Similarity

A family of hash functions F is said to be locality sensitive with respect to some similarity measure, if it satisfies the following property [6]:

$$P_{h\in F}(h(x) = h(y)) = sim(x, y) \tag{1}$$

Here x, y is a pair of data points, h is a hash function and sim is a similarity measure of interest. LSH for similarity measures can be used in two ways:

1. **Similarity Estimation:** If we have k i.i.d. hash functions $\{h_i\}_{i=1}^{k}$, then a maximum likelihood estimator (MLE) for the similarity is:

$$\widehat{sim(x,y)} = \frac{1}{k}\sum_{i=1}^{k} I(h_i(x) = h_i(y)) \tag{2}$$

2. **LSH Index Search:** The concatenation of the aforementioned k hash functions form a signature and suppose l such signatures are generated for each data point. Then for a query data point q, to find the nearest neighbor, only those points that have at least one signature in common with q need to be searched. This leads to an index construction algorithm that results in a sublinear time search. It is worth noting that a similarity threshold is a mandatory parameter for an LSH index construction. Consequently, a bias in its estimation may lead to a different index than the one intended based on input similarity measure.

Charikar [6] introduced a hash family based on the rounding hyperplane algorithm that can very closely approximate the cosine similarity. Let $h_i(x) = sign(r_i x^T)$, where $r_i, x \in R^d$ and each element of r_i is drawn from i.i.d. $N(0,1)$. Essentially the hash functions are signed random projections (SRP). It can be shown that in this case,

$$\begin{aligned} P(h_i(x) = h_i(y)) &= 1 - \frac{\theta(x,y)}{\pi} = sim(x,y) \\ \implies cos(\theta(x,y)) &= cos(\pi(1 - sim(x,y))) \end{aligned}$$

where $\theta(x,y)$ is the angle between x, y. The goal of this work is to find a locality sensitive hash family for the Nyström approximation $\hat{\kappa}$ of any arbitrary kernel κ that will satisfy the following property:

$$P(h_i(x) = h_i(y)) = 1 - \frac{cos^{-1}(\widehat{\kappa(x,y)})}{\pi} \tag{3}$$

2.2 Existence of LSH for Arbitrary Kernels

Kernel similarity measures are essentially the inner product in some transformed feature space. The transformation of the original data into the kernel induced feature space is usually non-linear and often explicit embedding in the kernel induced space are unknown, only the kernel function can be computed. Shrivastava *et. al.* [23] recently proved the non-existence of LSH functions for general inner product measures. In spite of the non-existence of LSH for kernels in the general case, LSH can still exist for a special case, where the kernel is normalized – in other words the inner product is equal to the cosine similarity measure. As mentioned in previous section, Charikar [6] showed that using signed random projections, cosine similarity can be well approximated using LSH. To summarize, LSH in kernel context is meaningful in the following two cases:

1. The case where the kernel is normalized with each data object in the kernel induced feature space having unit norm.

$$||\Phi(x)||^2 = \kappa(x, x) = 1 \tag{4}$$

   Here $\kappa(.,.)$ is the kernel function and $\Phi(.)$ is the (possibly unknown) kernel induced feature map in RKHS.
2. In the case equation 4 does not hold, LSH does not exist for $\kappa(.,.)$. But it exists for a normalized version of κ, say $\kappa_N(.,.)$, where:

$$\kappa_N(x, y) = \frac{\kappa(x, y)}{\sqrt{\kappa(x, x)}\sqrt{\kappa(y, y)}} \tag{5}$$

2.3 Kernelized Locality Sensitive Hashing

KLSH [13] is an early attempt to build an LSH index for any arbitrary kernel similarity measure. Later work by Xia *et. al.* [26] tries to provide bounds on kernel estimation error using Nyström approximation [25]. This work also provides an evaluation of applying LSH directly on explicit embedding generated by KPCA [21]. A follow up [12] to KLSH provided further theoretical insights into KLSH retrieval performance and proved equivalence of KLSH and KPCA+LSH.

KLSH computes the dot product of a data point and a random Gaussian in the approximate RKHS spanned by the first p principal components of the empirical centered covariance operator. It uses an approach similar to KPCA to find out a data point's projection onto the eigenvectors in the kernel induced feature space and it approximates the random Gaussian in the same space by virtue of the central limit theorem (CLT) of Hilbert spaces by using a sample of columns of the input kernel matrix. Let $X_{n\times d}$ denote the dataset of n points, each having d dimensions. We denote the i^{th} row/data point by X_i and i, j^{th} element of X by $X_{i,j}$. Let $K_{n\times n}$ be the full kernel matrix ($K_{i,j} = \kappa(X_i, X_j)$). KLSH takes as input p randomly selected columns from kernel matrix - $K_{n\times p}$. The algorithm to compute the hash bits is as follows:

1. *Extract $K_{p\times p}$ from input $K_{n\times p}$. $K_{p\times p}$ is a submatrix of $K_{n\times n}$ created by sampling the same p rows and columns.*
2. *Center the matrix $K_{p\times p}$.*
3. *Compute a hash function h by forming a binary vector e by selecting t indices at random from $1, ..., p$, then form $w = K_{p\times p}^{-1/2} e$ and assign bits according to the hash function*

$$h(\Phi(X_a)) = sign(\sum_i w(i)\kappa(X_i, X_a))$$

One thing worth noting here is, unlike vanilla LSH, where an LSH estimator tries to estimate the similarity measure of interest directly, in case of KLSH, the estimator tries to estimate the kernel similarity that is approximated by the KPCA embedding. The idea is that the KPCA embedding should lead to good

approximations of the original kernel and hence KLSH should be able to approximate the original kernel as well. Alternatively, instead of directly computing the dot product in RKHS, one may first explicitly compute the KPCA/Nyström $p-$dimensional embedding of the input data and generate a $p-$dimensional multivariate Gaussian, and then compute the dot product. The two methods are equivalent [12]. Next, we discuss why approximation error due to applying LSH on kernels may be significant.

3 Estimation Error of LSH for Kernels

According to Mercer's theorem [16], the kernel induced feature map $\Phi(x)$ can be written as $\Phi(x) = [\phi_i(x)]_{i=1}^{\infty}$ where $\phi_i(x) = \sqrt{\sigma_i}\psi_i(x)$ and σ_i and ψ_i are the eigenvalues and eigenfunctions of the covariance operator whose kernel is κ. The aforementioned infinite dimensional kernel induced feature map can be approximated explicitly in finite dimensions by using Nyström style projection [25] as described next. This can be written as $\widehat{\Phi(x)} = [\widehat{\phi_i(x)}]_{i=1}^{p}$ where $\widehat{\phi_i(x)} = \frac{1}{\sqrt{\lambda_i}} < K(x,.), u_i >$. Here $K(x,.)$ is a vector containing the kernel values of data point x to the p chosen points, λ_i and u_i are the i^{th} eigenvalue and eigenvector of the sampled $p \times p$ kernel matrix $K_{p\times p}$. Note that, both the KPCA and Nyström projections are equivalent other than the fact that in case of KPCA, $K_{p\times p}$ is centered, whereas in case of Nyström, it is uncentered. Essentially, $\widehat{\Phi(x)} = P_{\hat{S}}\Phi(x)$, where $P_{\hat{S}}$ is the projection operator that projects $\Phi(x)$ onto the subspace spanned by first p eigenvectors of the empirical covariance operator. Let $Y_{n\times p}$ represent this explicit embedding of the data points.

In the next lemma, we show that the above approach results in a bias for kernel similarity approximation from LSH.

Lemma 1. *If $\widehat{K_{(LSH)i,j}}$ is the quantity estimated by using LSH on Nyström embedding, then $\widehat{K_{(LSH)i,j}} \geq \widehat{K_{i,j}}$.*

Proof. Since $\widehat{K_{(LSH)}}$ is the quantity estimated by the LSH estimator for cosine similarity on embedding $Y_{n\times p}$, then by equation 5

$$\widehat{K_{(LSH)i,j}} = \frac{Y_i Y_j^T}{||Y_i||||Y_j||} = \frac{\widehat{K_{i,j}}}{\sqrt{\widehat{K_{i,i}}}\sqrt{\widehat{K_{j,j}}}} \tag{6}$$

where Y_i is the i^{th} row of Y.

By assumption, $||\Phi(X_i)|| = 1,\ \forall i$. Hence

$$\widehat{K_{i,i}} = < P_{\hat{S}}\Phi(X_i), P_{\hat{S}}\Phi(X_i) > = ||P_{\hat{S}}\Phi(X_i)||^2 \leq 1,\ \forall i$$

(since $P_{\hat{S}}$ is a projection operator onto a subspace). Specifically if $i \in p$, then $\widehat{K_{i,i}} = K_{i,i}$. Putting $\widehat{K_{i,i}} \leq 1$ in equation 6, we get the following.

$$\widehat{K_{(LSH)i,j}} \geq \widehat{K_{i,j}}$$

Thus, applying LSH to the Nyström embedding results in an overestimation of the kernel similarity when compared to the Nyström approximation to the kernel similarity. In terms of our goal, equation 3 will have $\widehat{K_{(LSH)}}$ instead of $\hat{K}$ (Nyström approximated kernel). Unlike $\hat{K}$, $\widehat{K_{(LSH)}}$ does not approximate K (true kernel) well, unless p is extremely large. This is not feasible since eigendecomposition is $O(p^3)$. Interestingly, the above bias $||\Phi(x) - P_{\hat{S}}\Phi(x)||$ depends on the eigenvalue decay rate [28], that in turn depends on the data distribution and the kernel function. Hence this error in estimation is hard to predict beforehand.

Additionally, another cause of estimation error, specifically for KLSH is due to the fact that KLSH relies on the CLT in Hilbert space to generate the random Gaussians in the kernel induced feature space. Unlike the single dimensional CLT, Hilbert space's CLT's convergence rate could be much worse [20], implying that the sample size requirement may be quite high. However, the number of available samples is limited by p (number of sampled columns). Typically p is set very small for performance consideration (in fact we found that p=128 performs extremely well for dataset size upto one million).

We next propose a transformation over the Nyström embedding on which the SRP technique can be effectively used to create LSH that approximates the input kernel $\kappa(.,.)$ (K) well. Our methods apply to centered KPCA case as well.

4 Augmented Nyström LSH Method (ANyLSH)

In this section we propose a data embedding that along with the SRP technique forms an LSH family for the RKHS. Given n data points and p columns of the kernel matrix, we first propose a $p+n$ dimensional embedding for which the bias is 0 (LSH estimator is an unbiased one for the Nyström approximated kernel). Since $p + n$ dimensional embedding is infeasible in practice due to large n, we propose a $p + c$ dimensional embedding, where c is a constant much smaller than n. In this case the estimator is biased, but that bias can be bounded by setting c and this bound hence is independent of the eigenvalue decay rate of the covariance operator. We provide theoretical analysis regarding the preservation of the LSH property and we also give the runtime and memory cost analysis.

4.1 Locality Sensitive Hash Family

We identify that the major problem with using Nyström embedding for LSH is the underestimation bias of the norms ($\widehat{K_{i,i}}$) of these embedding. Hence, though the estimates of the numerator of equation 6 are very good, the denominator causes estimation bias. We propose a new embedding of the data points such that the numerator will remain the same, but the norms of the embedding will become 1.

Definition 1. *We define the augmented Nyström embedding as the feature map $Z_{n\times(p+n)}$ such that $Z_{n\times(p+n)} = [Y_{n\times p}\ V_{n\times n}]$, where $V_{n\times n}$ is an $n \times n$ diagonal matrix with the diagonal elements as $\left\{\sqrt{1 - \sum_{j=1}^{p} Y_{i,j}^2}\right\}_{i=1}^{n}$.*

Lemma 2. *For $Z_{n\times(p+n)}$, if $\widehat{K_{(Z)n\times n}}$ is the inner product matrix, then for (i) $i=j$, $\widehat{K_{(Z)i,j}}=1$ and (ii) for $i\neq j$, $\widehat{K_{(Z)i,j}}=\widehat{K_{i,j}}$*

Proof. Case (i):

$$\begin{aligned}\widehat{K_{(Z)i,j}} &= Z_i Z_j^T \\ &= \sum_{k=1}^{p} Y_{i,k}^2 + \sum_{l=1}^{n} V_{i,l}^2 \\ &= \sum_{k=1}^{p} Y_{i,k}^2 + \left(\sqrt{1-\sum_{j=1}^{p} Y_{i,j}^2}\right)^2 \\ &= 1\end{aligned}$$

Case (ii):

$$\begin{aligned}\widehat{K_{(Z)i,j}} &= Z_i Z_j^T \\ &= \sum_{k=1}^{p} Y_{i,k}Y_{j,k} + \sum_{l=1}^{n} V_{i,l}V_{j,l} \\ &= \sum_{k=1}^{p} Y_{i,k}Y_{j,k} + 0 \ (V \textit{ is a diagonal matrix}) \\ &= Y_i Y_j^T \\ &= \widehat{K_{i,j}}\end{aligned}$$

Hence Z_i gives us a $p+n$ dimensional embedding of the data point X_i where Z_i approximates $\Phi(X_i)$. The inner product between two data points using this embedding gives the cosine similarity as the embedding are unit norm and the inner products are exactly same as that of Nyström approximation. Hence we can use SRP hash family on $Z_{n\times(p+n)}$ to compute the LSH embedding related to cosine similarity. Essentially we have:

$$P(h(Z_i)=h(Z_j)) = 1 - \frac{cos^{-1}(\widehat{K_{i,j}})}{\pi} \tag{7}$$

Hence we are able to achieve the LSH property of the goal equation 3.

4.2 Quality Implications

The quality of an LSH estimator depends on (i) similarity and (ii) number of hash functions. It is independent of the original data dimensionality. From equation 1, it is easy to see that each hash match is a i.i.d. Bernoulli trial with success probability $sim(x,y)$ (s). For k such hashes, the number of matches follow a binomial distribution. Hence the LSH estimator $\hat{s}$ of equation 2 is an MLE for the binomial proportion parameter. The variance of this estimator is known to be $\frac{s(1-s)}{k}$. Therefore, even with the increased dimensionality of $p+n$, the estimator variance remains the same.

4.3 Performance Implications

The dot product required for a single signed random projection for Z_i can be computed as follows:

$$\begin{aligned} Z_i r_j^T &= \sum_{l=1}^{p+n} Z_{i,l} R_{j,l} \\ &= \sum_{l=1}^{p} Y_{i,l} R_{j,l} + \sum_{k=1} nV_{i,k} R_{j,p+k} \\ &= \sum_{l=1}^{p} Y_{i,l} R_{j,l} + V_{i,i} R_{j,p+i} \end{aligned}$$

Hence there are (p+1) sum operations ($O(p)$). Though $Z_i \in R^{p+n}$, the dot product for SRP ($Z_i r_j^T$) can be computed in $O(p)$ (which is the case for vanilla LSH). Since $V_{n\times n}$ is a diagonal matrix, the embedding storage requirement is increased only by n (still $O(np)$). However, the number of $N(0,1)$ Gaussian samples required is $O(k(p+n))$, where as in case of vanilla LSH it was only $O(kp)$ (k is the number of hash functions). In the next section, we develop an algorithm with probabilistic guaranty that can substantially reduce the number of hashes required for the augmented Nyström embedding.

4.4 Two Layered Hashing Scheme

Next we define a $p+c$ dimensional embedding of a point X_i to approximate $\Phi(X_i)$. The first p dimensions contain projections onto p eigenvectors (same as first p dimensions of Z_i). In the second step, the norm residual (to make the norm of this embedding 1.0) will be randomly projected to 1 of c remaining dimensions, other remaining dimensions will be set zero.

Definition 2. *Remapped augmented Nyström embedding is an embedding* $Z'_{n\times(p+c)}$ *($\forall i, Z'_i \in R^{p+c}$) obtained from $Z_{n\times(p+n)}$ ($\forall i, Z_i \in R^{p+n}$) such that, (i) $\forall j \leq p$, $Z'_{i,j} = Z_{i,j}$ and (ii) $Z'_{i,p+a_i} = Z_{i,p+i}$, where $a_i \sim unif\{1,c\}$.*

Definition 3. *$C(i,j)$ is a random event of collision that is said to occur when for two vectors $Z'_i,\ Z'_j \in Z$, $a_i = a_j$.*

Since this embedding is in R^{p+c} rather than R^{p+n}, the number of $N(0,1)$ samples required will be $O(k(p+c))$, rather than $O(k(p+n))$. Next we show that using SRP on $Z'_{n\times(p+c)}$ yields LSH embedding, where the estimator converges to $\widehat{K_{n\times n}}$ with $c \to n$.

Lemma 3. *For $Z'_{n\times(p+c)}$, the LSH property that will be satisfied is*

$$P(h(Z'_i) = h(Z'_j)) =$$

$$\frac{1}{c}\Big[1 - \frac{cos^{-1}(\widehat{K_{i,j}} + \sqrt{1-\sum_{l=1}^{p} Y_{i,l}^2}\sqrt{1-\sum_{l=1}^{p} Y_{j,l}^2})}{\pi}\Big] + \frac{c-1}{c}\Big[1 - \frac{cos^{-1}(\widehat{K_{i,j}})}{\pi}\Big]$$

Proof. For the remap we used, collision probability is given by,

$$P(C(i,j)) = \frac{1}{c} \tag{8}$$

If there is a collision, then the norm correcting components will increase the dot product value.

$$P(h(Z'_i) = h(Z'_j)|C(i,j)) = \\ 1 - \frac{cos^{-1}(\widehat{K_{i,j}} + \sqrt{1 - \sum_{l=1}^{p} Y_{i,l}^2}\sqrt{1 - \sum_{l=1}^{p} Y_{j,l}^2})}{\pi} \tag{9}$$

If there is no collision, LSH will be able to approximate the Nyström method.

$$P(h(Z'_i) = h(Z'_j)|\neg\ C(i,j)) = 1 - \frac{cos^{-1}(\widehat{K_{i,j}})}{\pi} \tag{10}$$

We can compute the marginal distribution as follows:

$$P(h(Z'_i) = h(Z'_j)) = P(h(Z'_i) = h(Z'_j)|C(i,j))P(C(i,j)) \\ + P(h(Z'_i) = h(Z'_j)|\neg\ C(i,j))P(\neg C(i,j))$$

Applying equations 8,9 and 10 above, we get the result.

There are two aspects to note about the aforementioned lemma:

1. According to Nyström approximation [25], as we increase p (higher rank approx.), the quantity $\sqrt{1 - \sum_{l=1}^{p} Y_{i,l}^2}$ tends to 0 and the lemma leads to the desired goal of equation 3, but at a computational cost of $O(p^3)$ for the eigendecomposition operation. Of course increasing p improves the overall quality of Nyström approximation itself, however in practice small values of p suffice.
2. Interestingly, instead of p, if we increase c, then also we converge to the goal of equation 3 as the first term of the lemma converges to 0. The computational cost is $O(k(p+c))$ which usually is much less than $O(p^3)$. This is the strategy we adopt and as we will show shortly, small values of c are sufficient even for large scale datasets. Hence c can be used to bound the bias (difference from the probability of equation 3).

5 Evaluation

5.1 Datasets and Kernels

We evaluate our methodologies on five real world image datasets varying from 3030 data points to 1 million and three popular kernels known to work well on them. Summary of the datasets can be found in Table 2.

Caltech101: This is a popular image categorization dataset [9]. We use 3030 image from this data. Following KLSH [12, 13] we use this dataset with the CORR kernel [27].
PASCAL VOC: This is also an image categorization dataset [8]. We use 5011 images from this data. Following [7] we use the additive χ^2 kernel for this data.
Notre Dame image patches: This dataset contains 468159 small image patches of Notre Dame [10] and the image patch descriptors used are as per [24]. We use the Gaussian RBF kernel on this data.
INRIA holidays: To test at large scale, we use 1 million SIFT as well as 1 million GIST descriptors from the INRIA holidays dataset [11]. Following KLSH [12, 13] we use the additive χ^2 kernel with this data.

Table 2: dataset and kernel details

Dataset	Size	Kernel
Caltech101	3030	CORR
PASCAL VOC 2007	5011	Additive χ^2
Notre Dame image patches	468159	Gaussian RBF
INRIA holidays SIFT-1M	1000000	Additive χ^2
INRIA holidays GIST-1M	1000000	Additive χ^2

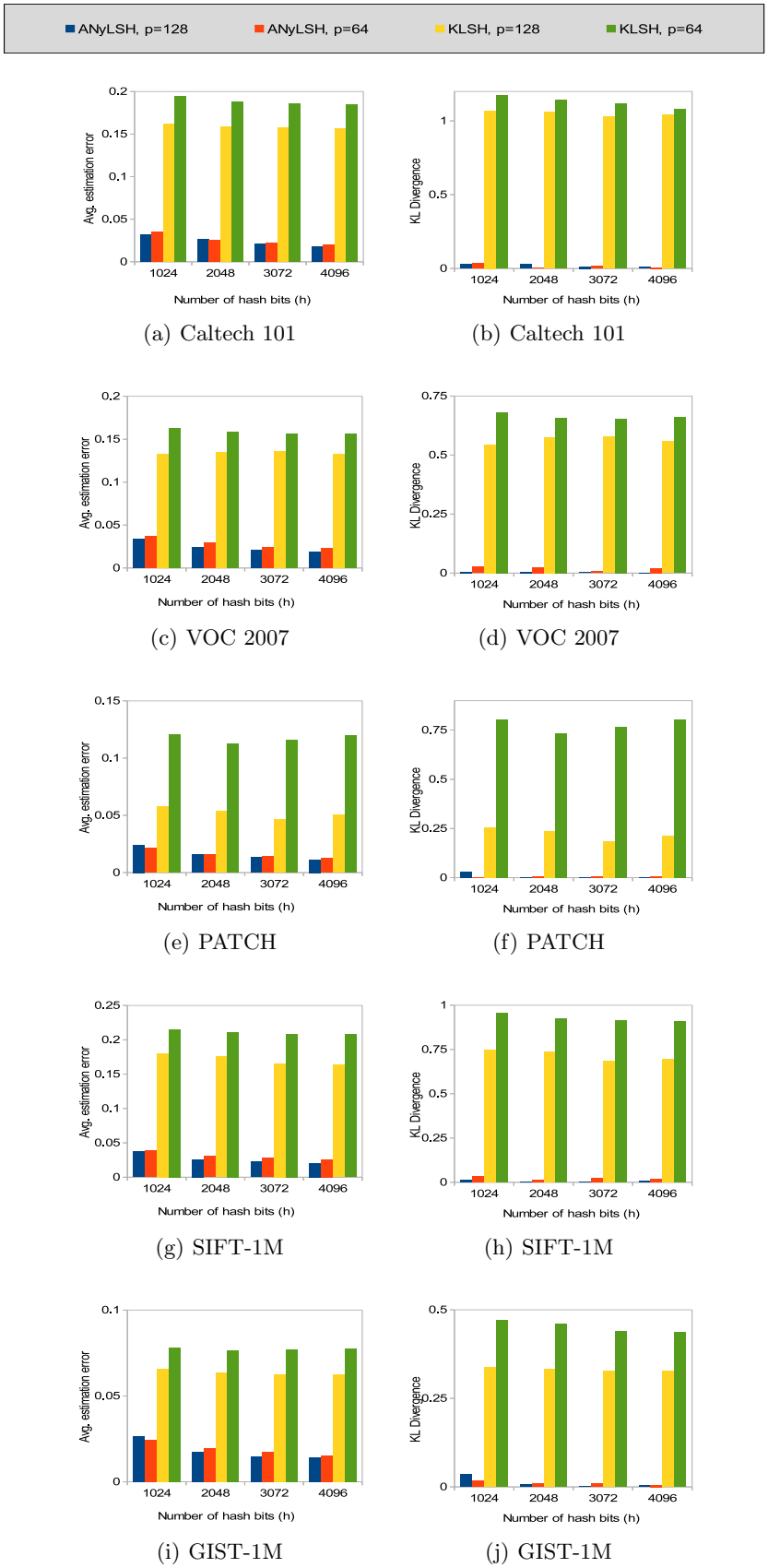
5.2 Evaluation Methodology

The focus of this work is accurate estimation of the input kernel similarity measure through LSH. For evaluating the quality of similarity estimation, we use two approaches - (i) we take a sample of pairs from each dataset, and compute the average estimation error directly and (ii) we use a sample of pairs from each dataset, compute the similarity of the pairs, both accurately (ground truth) and approximately (ANyLSH) and then compare the statistical distribution of the pairwise similarity of ground truth with ANyLSH. The former gives a direct measure of estimation accuracy, while the latter gives us insights on how well the pairwise similarity distribution is preserved. In terms of execution times, our algorithm performs the same as the baseline we compare against.

We use state-of-the-art KLSH as our baseline. We randomly sample 1000 pairs of data points from each dataset for our experiments. We use the values 64 and 128 for p, and vary h from 1024 to 4096 in steps of 1024. For ANyLSH, we set $c = 1000$. In our evaluation, we see that c generalizes well to varying data sizes. For KLSH, we set $q = 16, 32$ for $p = 64, 128$ respectively as per the guideline in the source code [13].

5.3 Results

Similarity Estimation Comparisons Figures 1(a),1(c),1(e),1(g) and 1(i) report the results on estimation error. We clearly see that our ANyLSH method



(a) Caltech 101 (b) Caltech 101

(c) VOC 2007 (d) VOC 2007

(e) PATCH (f) PATCH

(g) SIFT-1M (h) SIFT-1M

(i) GIST-1M (j) GIST-1M

Fig. 1. Estimation error and KL Divergence are reported in the first and second columns respectively for all datasets.

outperforms KLSH in every single case by a large margin. The improvement of estimation error varies from a minimum of 2.4x (Figure 1(e), $p = 128, h = 1024$) to a maximum of 9.7x (Figure 1(e), $p = 64, h = 4096$), with average reduction in error of 5.9x across all datasets. With fixed p, the estimation error of our method decreases consistently across all datasets with the increase of hashes, as should be the case per equation 2. Interestingly, for KLSH, there are multiple cases when with the increase in hashes, the estimation error also increased. For instance, in Figure 1(i), at $p = 64$, by increasing h from 2048 to 4096, KLSH's error increased from 0.076 to 0.078. This provides empirical evidence as well that not only the estimates are off, but in case of KLSH, they are converging towards a biased value as described in Lemma 1. Additionally note that our average absolute error varies between $0.011 - 0.038$ across all datasets and there is no trend that the error increases with larger datasets. This provides strong empirical evidence to the theoretical insight that at fixed c (1000 in our case), the average estimation error generalizes extremely well to different datasets of varied sizes and different kernels. Though the error is a function of the eigenvalue decay rate, it is upper bounded by ANyLSH.

Similarity Distribution Comparisons As second part of our qualitative evaluation, in this section, we investigate how well the pairwise similarity distribution of the data is preserved. This is particularly important in applications that rely heavily on similarity distribution such as clustering. Our goal is to compare the two distributions in a non-parametric fashion as we do not have any prior knowledge of these distribution. Our first approach is to compare normalized histograms (probabilities). We choose the popular KL divergence measure to compare probability distributions represented by histograms. We discretized both our data and the ground truth by splitting the similarity range $0 - 1$ into fixed length bins of length 0.1. Figures 1(b),1(d),1(f),1(h) and 1(j) report the KL divergence numbers. The improvement in terms of KL divergence is even better, with upto two orders of magnitude improvement over KLSH. This improvement can be partly attributed to the discretization process – since we used length 0.1 bins and our estimation errors are significantly less than 0.1, most of our errors get absorbed in the discretization process. With KLSH's error being substantially higher than 0.1, it's KL divergence becomes very high.

To account for the binning issue, we additionally run the non-parametric Kolmogorov-Smirnov two sample test that is more suitable for comparing empirical distributions of continuous data. This test is particularly challenging in our setting as the test statistic is the supremum of absolute differences across all values in the empirical CDFs. Thus, error in even a single region may result in the failure of this test. Moreover, our proposed method being an approximate one, will always have some estimation error. The null hypothesis is that the two samples come from the same underlying distribution and the alternative hypothesis is that they are from different distributions. The results for $p = 128$ and $h = 4096$ are are reported in Table 3. All of the datasets (only exception being Caltech) did not reject the null hypothesis, providing strong evidence that

they are indeed from same underlying distribution. Note that, even the Caltech result was very close to the threshold. For KLSH, in every single dataset the null hypothesis was rejected and the $p - values$ were extremely far away from the threshold. This conclusively proves that applying KLSH to a dataset significantly changes the pairwise similarity distribution.

Table 3: Results of Kolmogorov Smirnov tests on ANyLSH method. Critical Value at 1% significance level was 0.073.

Dataset	p-value	Test statistic
Caltech101	0.006	0.076
PASCAL VOC 2007	0.716	0.031
Patch	0.565	0.035
INRIA SIFT-1M	0.603	0.034
INRIA GIST-1M	0.011	0.072

6 Future Works

There has been a wide range of works that build on the KLSH foundations - improve quality through supervised learning [14, 17]; develop LSH for non-metric measures [18]; We believe that these methods can be used in conjunction with our hashing scheme as well to improve performance, and in future, we propose to investigate them. Additionally, we plan to explore the case of non-normalized kernel measures. Though LSH is known not to exist in the general case for maximum inner product search, but augmented data embedding along with modified LSH functions [23, 19] are known to work well for maximum inner product search. We believe these ideas can be leveraged by our data embedding framework to handle kernel similarities for the general case.

7 Conclusion

In this paper we proposed a locality sensitive hash family for arbitrary normalized kernel similarity measures. We analytically showed that the existing methods of LSH for kernel similarity measures based on KPCA/Nyström projections suffer from an estimation bias, specific to the LSH estimation technique. In other words, these LSH estimates differ from the KPCA/Nyström estimates of the kernel. This bias depends on the eigenvalue decay rate of the covariance operator and as such unknown apriori. Our method, ANyLSH, can directly estimate the KPCA/Nyström approximated input kernel efficiently and accurately in a principled manner. Key to our method are novel data embedding strategies. We showed that, given p columns of the input kernel matrix, the bias can be completely removed by using a $p + n$-dimensional embedding. Since n can

be rather large and also not fixed, we additionally propose a $p + c$-dimensional embedding where c is fixed and much smaller than n. In our analysis we showed that in this case the worst case bias can be controlled by the user by setting c. Consequently, we overcame the short coming that resulted from the bias term being unknown to the user apriori. Our methods, when compared to the state-of-the-art KLSH improves the kernel similarity estimation error by upto 9.7x. Further evaluations based on the KL divergence and Kolmogorov-Smirnov tests provide strong evidence that pairwise similarity distribution is well preserved by ANyLSH.

Acknowledgments: We thank the anonymous reviewers for their feedback. This work is supported in part by NSF grants CCF-1217353 and DMS-1418265.

References

1. Andoni, A., Indyk, P.: Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. Communications of the ACM 51, 117–122 (2008)
2. Bayardo, R., Ma, Y., Srikant, R.: Scaling up all pairs similarity search. In: WWW (2007)
3. Bentley, J.L.: Multidimensional binary search trees used for associative searching. Communications of the ACM 18(9), 509–517 (1975)
4. Chakrabarti, A., Parthasarathy, S.: Sequential hypothesis tests for adaptive locality sensitive hashing. In: Proceedings of the 24th International Conference on World Wide Web. pp. 162–172. ACM (2015)
5. Chakrabarti, A., Satuluri, V., Srivathsan, A., Parthasarathy, S.: A bayesian perspective on locality sensitive hashing with extensions for kernel methods. ACM Transactions on Knowledge Discovery from Data (TKDD) 10(2), 19 (2015)
6. Charikar, M.S.: Similarity estimation techniques from rounding algorithms. In: STOC '02 (2002), http://doi.acm.org/10.1145/509907.509965
7. Chatfield, K., Lempitsky, V., Vedaldi, A., Zisserman, A.: The devil is in the details: an evaluation of recent feature encoding methods. In: British Machine Vision Conference (2011)
8. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html
9. Fei-Fei, L., Fergus, R., Perona, P.: Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories. In: Computer Vision and Pattern Recognition Workshop, 2004. CVPRW'04. Conference on. pp. 178–178. IEEE (2004)
10. Goesele, M., Snavely, N., Curless, B., Hoppe, H., Seitz, S.M.: Multi-view stereo for community photo collections. In: Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on. pp. 1–8. IEEE (2007)
11. Jegou, H., Douze, M., Schmid, C.: Hamming embedding and weak geometric consistency for large scale image search. In: Computer Vision–ECCV 2008, pp. 304–317. Springer (2008)
12. Jiang, K., Que, Q., Kulis, B.: Revisiting kernelized locality-sensitive hashing for improved large-scale image retrieval. arXiv preprint arXiv:1411.4199 (2014)

13. Kulis, B., Grauman, K.: Kernelized locality-sensitive hashing. Pattern Analysis and Machine Intelligence, IEEE Transactions on 34(6), 1092–1104 (2012)
14. Liu, W., Wang, J., Ji, R., Jiang, Y.G., Chang, S.F.: Supervised hashing with kernels. In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. pp. 2074–2081. IEEE (2012)
15. Massey Jr, F.J.: The kolmogorov-smirnov test for goodness of fit. Journal of the American statistical Association 46(253), 68–78 (1951)
16. Mercer, J.: Functions of positive and negative type, and their connection with the theory of integral equations. Philosophical transactions of the royal society of London. Series A, containing papers of a mathematical or physical character 209, 415–446 (1909)
17. Mu, Y., Shen, J., Yan, S.: Weakly-supervised hashing in kernel space. In: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on. pp. 3344–3351. IEEE (2010)
18. Mu, Y., Yan, S.: Non-metric locality-sensitive hashing. In: AAAI (2010)
19. Neyshabur, B., Srebro, N.: On symmetric and asymmetric lshs for inner product search. In: Proceedings of The 32nd International Conference on Machine Learning. pp. 1926–1934 (2015)
20. Paulauskas, V.: On the rate of convergence in the central limit theorem in certain banach spaces. Theory of Probability & Its Applications 21(4), 754–769 (1977)
21. Schölkopf, B., Smola, A., Müller, K.R.: Nonlinear component analysis as a kernel eigenvalue problem. Neural computation 10(5), 1299–1319 (1998)
22. Shawe-Taylor, J., Cristianini, N.: Kernel methods for pattern analysis. Cambridge university press (2004)
23. Shrivastava, A., Li, P.: Asymmetric lsh (alsh) for sublinear time maximum inner product search (mips). In: Advances in Neural Information Processing Systems. pp. 2321–2329 (2014)
24. Simonyan, K., Vedaldi, A., Zisserman, A.: Descriptor learning using convex optimisation. In: Computer Vision–ECCV 2012, pp. 243–256. Springer (2012)
25. Williams, C., Seeger, M.: Using the nyström method to speed up kernel machines. In: Proceedings of the 14th Annual Conference on Neural Information Processing Systems. pp. 682–688. No. EPFL-CONF-161322 (2001)
26. Xia, H., Wu, P., Hoi, S.C., Jin, R.: Boosting multi-kernel locality-sensitive hashing for scalable image retrieval. In: Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval. pp. 55–64. ACM (2012)
27. Zhang, H., Berg, A.C., Maire, M., Malik, J.: Svm-knn: Discriminative nearest neighbor classification for visual category recognition. In: Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on. vol. 2, pp. 2126–2136. IEEE (2006)
28. Zwald, L., Blanchard, G.: On the convergence of eigenspaces in kernel principal component analysis. In: NIPS (2005)