Kernelized Locality Sensitive Hashing

Daniel Speyer and Geelon So
{dls2192,geelon}@columbia.edu

December 14, 2017

Abstract

1 LSH

Locality sensitive hashing is a general solution for searches in which the desired key is near or nearest to the query, rather than equal. It is not as efficient as tree-based solutions for $\mathbb{R}^n$ with small n, but unlike those solutions, it maintains its performance in high dimensions or in arbitrary metric spaces.

Most near search problems can be reduced to cr-near search. Suppose we have a database of points $x_1, x_2, ..., x_n$ drawn from some arbitrary metric space $\mathcal{X}$ and some query q also from $\mathcal{X}$, plus constants $c, r \in \mathbb{R}$. A cr-near search seeks either an x within distance cr of q or a statement that there are none within r. It is generally best to think of r as part of the query and c as the acceptable level of error.

To use LSH for problems like this we will need a family of hashing functions $h_1, h_2, ...$ that map $\mathcal{X}$ into some small finite space, all following two key inequalities:

$$\begin{aligned} ||x_1 - x_2|| < r &\rightarrow Pr[h(x_1) = h(x_2)] \geq p_1 \\ ||x_1 - x_2|| > cr &\rightarrow Pr[h(x_1) = h(x_2)] \leq p_2 \end{aligned}$$

Note that we may require an arbitrary number of hashing functions, which must be generated randomly. It is this randomness that allows the use of probabilities in the preceding inequalities. (The distribution of hashing functions is often written $\mathcal{H}$, but we'll be reserving that symbol for Hilbert spaces later.)

In the simplest example, $\mathcal{X}$ is a high dimensional Hamming space and the hash functions sample random columns. This may provide useful intuition for thinking about hash functions.

Once we have a source of functions and a gap between p_1 and p_2, we can widen the gap by taking n functions and concatenating their outputs (getting p_1^n and p_2^n). We can then deal with too low a p_1^n by taking m repetitions and searching all of them (getting mp_1^n and mp_2^n). This solves the cr-near problem with arbitrarily high probability, which can in turn solve other, similar problems.

2 Kernels

In this section, we will motivate and describe the theory of kernel methods. We often develop mathematics and algorithms for objects that explicitly encode a lot of structure. For example, LSH has been developed for Hamming spaces, finite-dimensional vector spaces, n-spheres, and so on. These objects are particularly well-suited for analysis, as the very way we *represent* them directly showcases their *properties*. It is often the harmony between representation and structure that enables elegance and empowers analysis.

But we can't expect harmony always. Nonetheless, might we be able to extend these methods to objects that have implicit structure?

Kernel methods are one way of approaching such situations. Let $\mathcal{X}$ be our object—just an abstract set. But perhaps it has implicit structure. How might we access this structure? Very naturally, we might encode this structure into a mapping from that abstract set to a structure-rich mathematical object.

We often do this without thinking, for example, assigning numbers to objects in order to compare them—a mapping of the form $\mathcal{X} \to \mathbb{R}$. Indeed, at a high level, we can consider different mathematical objects as the 'canonical' idealization of certain types of structure; then, functions from $\mathcal{X}$ to these objects allow us to 'pull back' or instantiate that structure within our specific $\mathcal{X}$.

Of course, the structure we care about with respect to LSH is *similarity*, one idealization of which is the *(real) inner product*. In fact, with kernel methods, we will be able to tightly bind any abstract object with the right notion of similarity to an inner product space, and this will give us both new insight and new algorithms. Let us take a moment to reexamine the inner product on finite real vector spaces as a notion of similarity, before generalizing that notion using *kernels* to infinite dimensional vector spaces and arbitrary sets.

2.1 Finite-Dimensional Inner Product Spaces

Definition 1. Let V be a (possibly infinite) vector space over $\mathbb{R}$. An *inner product* on V is a bilinear map $K : V \times V \to \mathbb{R}$ that is *symmetric* and *positive-definite*. We call the pair (V, K) an *inner product space*.

Symmetry and positive-definiteness are two of the most obvious requirements for any self-respecting function that calls itself a measure of similarity:

1. the similarity between two points x and y better not depend on the order we specify them, so we require $K(x, y) = K(y, x)$,

2. a nonzero object should be similar to itself, so if $x \neq 0$, then $K(x, x) > 0$.

These two simple intuitions of a 'similarity measure' then precisely define the conditions of an inner product on V.[1] Allow us to be (perhaps overly) rigorous in the following discussion, for it is better to bore in the finite case than to confuse in the general case.

[1] For now, we can just view bilinearity as a necessary condition to ensure that K respects the real numbers as an algebraic object. So, in some sense, bilinearity is not intrinsic to K; rather, it is a condition *induced* by the field $\mathbb{R}$. Later on, when we generalize, we won't have a notion of (bi)linearity on arbitrary sets $\mathcal{X}$; however, there is a natural 'completion' of $\mathcal{X}$ as a vector space $\mathcal{H}$. Here, $\mathbb{R}$ induces the linear structure.

For concreteness, let $V = \mathbb{R}^n$ with a fixed basis. With any $x, y \in V$, denote by $\langle x, y\rangle$ the *formal dot product* of x and y as an algebraic construction.[2] Then, as K is a bilinear form, there exists a matrix $\mathbf{K}$ where

$$K(x, y) = \langle x, \mathbf{K}y\rangle. \tag{1}$$

The symmetry condition on the inner product K forces the matrix $\mathbf{K}$ to be symmetric. The spectral theorem on symmetric matrices tell us that V has an eigenbasis with respect to $\mathbf{K}$. Consider an eigenvalue $\mathbf{K}v = \lambda v$. As we require K to be positive-definite, we know that

$$K(v, v) = \langle v, \mathbf{K}v\rangle = \lambda\langle v, v\rangle > 0.$$

Since the dot product of a vector with itself in any basis is nonnegative, this implies that the eigenvalue λ is positive; so in its eigenbasis, $\mathbf{K}$ is diagonal with positive terms on its diagonal—it follows that $\mathbf{K}^{1/2}$ exists, and this allows us to write:

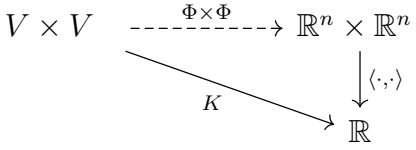
$$K(x, y) = \langle \mathbf{K}^{1/2}x, \mathbf{K}^{1/2}y\rangle.$$

We deduce that there exists a linear map $\Phi : V \to \mathbb{R}^n$ (spoiler: we call Φ a *feature map*) where the inner product on V directly corresponds to the usual dot product on $\mathbb{R}^n$. That is,

$$K(x, y) = \langle \Phi(x), \Phi(y)\rangle. \tag{2}$$

In other (ridiculously abstract) words, Equation 2 precisely says:

Proposition 2. *Let V be an n-dimensional real vector space. If $K : V \times V \to \mathbb{R}$ is an inner product, then there exists a map $\Phi : V \to \mathbb{R}^n$ such that the following diagram commutes:*



where $\langle\cdot,\cdot\rangle$ is the standard dot product on $\mathbb{R}^n$.

This is the main payoff to all this formality: we may view the pair $(\mathbb{R}^n, \langle\cdot,\cdot\rangle)$ as the canonical n-dimensional real inner product space. So, no matter which inner product space (V, K) we want to study, we're in fact guaranteed the existence of an isomorphism Φ to the familiar inner product space $\mathbb{R}^n$, and we may freely interchange the two objects in our minds.

Furthermore, we are now justified in dropping the distinction between the formal dot product and inner product; let us denote both by $\langle\cdot,\cdot\rangle$.

[2] One point of confusion may be that $\langle\cdot,\cdot\rangle$ often denotes an inner product—usually unproblematic as the dot product is in fact the inner product *induced by the choice of basis* on V. However, an abstract vector space has no canonical basis; thus, it has no canonical inner product. In particular, here, the inner product corresponding to the dot product is in general unrelated to the inner product K. We will reconcile these two views at Equation 2, which shows that every inner product corresponds to the dot product in an appropriate basis. Conversely, as every dot product is an inner product, if V is an abstract finite-dimensional real vector space, *specifying an inner product on V is equivalent to specifying a basis on V.*

2.2 Generalization: Kernels and Hilbert Spaces

First, we will need to extend the 'similarity measure' from vector spaces V to abstract sets $\mathcal{X}$. In the former, notice that given a basis $\{v_1, \ldots, v_n\}$, the inner product K is fully determined by the collection of values $K(v_i, v_j)$, where i, j range between 1 and n. This follows from bilinearity of K and is equivalent to saying that we can represent the bilinear form K by a matrix of $n \times n$ values $\mathbf{K}$. In some sense, this means that there are only 'n ways' or 'n directions' in which objects of V may be (dis)similar.

This suggests that we can generalize Definition 1 because we didn't need all of V to start with! Suppose someone secretly had an n-dimensional inner product space (V, K), but just gave us n linearly independent vectors, say $[n] := \{v_1, \ldots, v_n\}$, along with the corresponding restriction of the similarity measure $K : [n] \times [n] \to \mathbb{R}$.

Our view of $[n]$ would just be a collection of n abstract objects, and K just an $n \times n$ matrix over $\mathbb{R}$ (this is often called the *Gram matrix*). Still, we would have produced the same feature map $\Phi_{[n]} : [n] \hookrightarrow \mathbb{R}^n$, albeit restricted to $[n] \subset V$. But in the previous analysis, since V and $\mathbb{R}^n$ are isomorphic via Φ, this implies that $\Phi_{[n]}$ actually *recovers* V by its identification with $\mathbb{R}^n$; it is as though we've discovered that $[n]$ actually lived inside the space V.

Very naturally, the generalization of K from $[n]$ to $\mathcal{X}$ takes the view that K is a similarity 'matrix' on $\mathcal{X}$ (though we call it a *kernel*):

Definition 3. Let $\mathcal{X}$ be a nonempty set. A *kernel*[3] is a map $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. We say that K is *positive-definite* if for any finite subsets A of $\mathcal{X}$, the corresponding Gram matrix of A is symmetric and positive-definite.

In the finite case, we produced an embedding $\Phi_{[n]} : [n] \to \mathbb{R}^n$ of $[n]$ satisfying Equation 2 (K corresponds to inner product). As we had only n objects, it is clear that we needed an inner product space with at most n dimensions, hence $\mathbb{R}^n$.

However, in the general case, where $\mathcal{X}$ may be infinite, we may need infinite dimensions. Consider the real vector space $\mathbb{R}^{\mathcal{X}}$ (vector addition and scalar multiplication are defined pointwise), and the map $\Phi : \mathcal{X} \to \mathbb{R}^{\mathcal{X}}$ defined by:

$$\Phi(x) := K(\cdot, x).$$

[3]Unfortunately, the terminology *kernel* can be terribly confusing. Within functional analysis, kernels are often interchangably called *kernel maps*, *kernel functions*, and *positive-definite kernels*. Yet, each of these also take on other meanings depending on author and context. 'Kernels' and 'kernel maps' are almost always synonymous. Sometimes, 'kernel functions' denote kernels of the form $K(x, y) = k(x-y)$, in situations where subtraction is defined. And in many instances, when context is clear, the author cares only for positive-definite kernels, so omitting the specifier *positive-definite*.

Furthermore, *kernels* used in this sense are unrelated to the algebraic kernel of a linear map. The terminology comes from the theory of integral operators. In particular, infinite-dimensional function spaces often have inner products of the form:

$$\langle f, g \rangle = \int_{X \times X} K(x, y) f(x) g(y) \, \mathrm{d}x \mathrm{d}y,$$

where the K here performs the analogous role as $\langle x, \mathbf{K}y \rangle$ in Equation 1. Actually, it performs precisely the role we desire in the general case. These maps K were historically called kernels.

For short, let $k_x := \Phi(x)$. Then, Φ maps $\mathcal{X}$ into a subset of $\mathbb{R}^{\mathcal{X}}$; let $V = \text{span}(\Phi(\mathcal{X}))$ be a linear subspace of $\mathbb{R}^{\mathcal{X}}$, so that elements f of V are of the form:

$$f = \alpha_1 k_{x_1} + \cdots + \alpha_n k_{x_n},$$

with $\alpha_i \in \mathbb{R}$ and n ranging over $\mathbb{N}$. We claim that because K is a positive-definite kernel, there exists a well-defined bilinear map on V, corresponding to our desired inner product. Of course, we want the inner product to satisfy:

$$\langle k_x, k_y \rangle = K(x, y).$$

But in fact, once we specify this, linearity determines the inner product on general elements $f = \sum_{i=1}^{n_1} \alpha_i k_{x_i}$ and $g = \sum_{j=1}^{n_2} \beta_j k_{y_j}$:

$$\langle f, g \rangle = \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \alpha_i \beta_j K(x_i, y_j).$$

This should feel quite familiar, recalling how the Gram matrix of n linearly independent vectors fully determined the inner product in the finite case. However, we must be a little more careful here, as we are not guaranteed that the collection of k_x's are linearly independent. In particular, suppose that f, as follows, is identically zero:

$$0 \equiv f = \sum_{i=1}^{n} \alpha_i k_{x_i}.$$

That is, $f(x) = 0$ for all $x \in \mathcal{X}$. Then, $\langle f, g \rangle$ better equal 0 for all $g \in V$. Notice that it is sufficient to show that $\langle f, k_x \rangle = \langle k_x, f \rangle = 0$ for each $x \in \mathcal{X}$. And indeed, by assumption

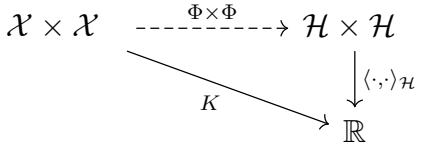
$$\langle k_x, f \rangle = \sum_{i=1}^{n} k_{x_i}(x) = f(x) = 0.$$

This proves that V admits an inner product that is compatible with K in the sense of Equation 2. We will in fact go beyond producing an inner product space V; we can complete the space with respect to the norm induced by the inner product, producing a *Hilbert space*.

That is, let $\mathcal{H}$ be the completion of V by taking equivalence classes of Cauchy sequences on V. We claim that $\mathcal{H} \subset \mathbb{R}^{\mathcal{X}}$ is a collection of functions in the sense that $f(x) < \infty$ for all $f \in \mathcal{H}$ and $x \in \mathcal{X}$.[4] But let us relegate the proof to references, say [P2009, Thm 3.16], for it is not particularly enlightening. However, the consequences are quite important.

First of all, we immediately attain the analog to Proposition 2:

Proposition 4 (Moore)**.** *Let $\mathcal{X}$ be a set, and $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ a positive-definite kernel. Then, there exists a map $\Phi : \mathcal{X} \to \mathcal{H}$ into a Hilbert space such that the following diagram commutes:*



where $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ is the inner product on $\mathcal{H}$.

[4]Note that in general, it is not the case that the completion of a function space consists of functions. Elements of $L^2(\mathbb{R})$, for example, are equivalence classes of functions.

This is a powerful result because we may now view $\mathcal{X}$ not as an abstract set, but as vectors within a Hilbert space, which carries with it both algebraic and geometric properties which $\mathcal{X}$ now inherits. It is particularly amazing because the only thing we needed required was a positive-definite kernel K to obtain $\mathcal{H}$. And in fact:

Fact 5. *The Hilbert space $\mathcal{H}$ induced by K is unique. We call $\mathcal{H}$ the* reproducing kernel Hilbert space (RKHS) *on $\mathcal{X}$ with respect to K. (See [P2009, Prop 3.3]).*

Later, we'll see a characterization of RKHS's: if $\mathcal{H}$ is one, then $\mathcal{H}$ induces a unique kernel K on $\mathcal{X}$. So, analogous to the finite-dimensional case, we can think of $\Phi(\mathcal{X})$ in $\mathcal{H}$ as the 'canonical form' of $\mathcal{X}$ with respect to the similary measure K.

Despite this, the way we've presented the embedding of $\mathcal{X}$ into $\mathcal{H}$ is somewhat unnatural. While it makes sense to consider the collection of functions k_x, the identification of x with k_x may seem *ad hoc*. But if we venture a bit further into functional analysis, we may obtain a clearer understanding of RKHS's. Along the way, hopefully we can clarify why we might call this identification a feature map.

2.3 Feature Maps

Let us once again consider an abstract set $\mathcal{X}$, and a collection of functions over $\mathcal{X}$, say $\mathcal{H} \subset \mathbb{R}^{\mathcal{X}}$. We may imagine each $f \in \mathcal{H}$ as an observable or a feature, where $f(x)$ gives the *measured value* of the object x by feature f.

As a result, every $x \in \mathcal{X}$ is associated to the Cartesian product of measurements,

$$x \overset{\Psi}{\mapsto} \prod_{f \in \mathcal{H}} f(x). \tag{3}$$

The high-level punchline here will be that $\Psi(x)$ is actually a point in the *dual* of $\mathcal{H}$, following the canonical mapping of $\mathcal{X}$ into the double dual $\mathcal{X}^{**}$. But in the case where $\mathcal{H}$ is a Hilbert space and all the $\Psi(x) \in \mathcal{H}^*$ are bounded (i.e. continuous), then $\mathcal{H}$ is isomorphic to $\mathcal{H}^*$. It turns out that $\Psi(x)$ is naturally identified with k_x, justifying the initial definition $\Phi : x \mapsto k_x$.

Before elaborating on this, let us provide a visual example; perhaps a bit of concreteness will convince the reader why we should even care about the association in Equation 3 in the first place.

Imagine $\mathcal{X}$ is a set of abstract cats, while $\mathcal{H} = \{f_1, \ldots, f_n\}$ represents the pixels within the lens of a camera. Given a cat x, the value $f_i(x)$ gives the color the ith pixel on the camera. Then, the product $\Psi(x)$ in Equation 3 represents the *image* of the cat—the collection of colors all the camera pixels see as the camera takes the picture of cat x.

Here, it makes sense to call the mapping in Equation 3 the *feature map*, for it maps the cat to a collection of features, specifically its colors at different locations. In general, given any function $f : \mathcal{X} \to \mathbb{R}$, we may dually view points of x as functionals, $\mathrm{ev}_x : \mathcal{H} \to \mathbb{R}$, where

$$\mathrm{ev}_x(f) := f(x).$$

In functional analysis, we say that ev_x is the *evaluation at x*, and if it is continuous, then we view it as an element of a larger set of continuous functionals on $\mathcal{H}$, the *continuous dual*,

$\mathcal{H}^*$.[5] Let us return to kernels. But whereas in the previous section, the feature map fell out of the construction almost incidentally, here, we begin with it: $\Psi(x) = \text{ev}_x$.

In particular, we equivalently define an RKHS (equivalency will become clear):

Definition 6. Let $\mathcal{H} \subset \mathbb{R}^{\mathcal{X}}$ be a Hilbert space. It is an RKHS if for all $x \in \mathcal{X}$, the evaluation functional ev_x is continuous.

That is, $\mathcal{H}$ is an RKHS if $\Psi(\mathcal{X}) \subset \mathcal{H}^*$. As made intuitive by the cat images example, $\Psi(\mathcal{X})$ is the representation of $\mathcal{X}$ seen through the lens of features $\mathcal{H}$. And if those features are quite discerning, for any distinct objects $x, y \in \mathcal{X}$, we might expect that their differences will be measurable; there exists some $f \in \mathcal{H}$ such that $f(x) \neq f(y)$. When this is the case, we say that $\mathcal{H}$ *separates points*, and it follows that:

Proposition 7. *The mapping $\Psi : \mathcal{X} \to \mathcal{H}^*$ is injective if and only if $\mathcal{H}$ separates points.*

So in many cases, we may think of Ψ as an embedding of $\mathcal{X}$ into $\mathcal{H}^*$. But then, Riesz representation theorem states that $\mathcal{H} \cong \mathcal{H}^*$. This is a powerful isomorphism, for it implies that every x, which corresponds to a unique $\Psi(x) \in \mathcal{H}^*$, then corresponds to some unique point k_x in $\mathcal{H}$ such that:

$$\Psi(x) = \langle \cdot, k_x \rangle_{\mathcal{H}}.$$

We call k_x the *reproducing kernel for the point x*, and for consistency, we can let $\Phi : x \mapsto k_x$. But via the isomorphism, we may afford to drop the distinction between Ψ and Φ.

Now, as vectors in a Hilbert space, the points k_x inherits the similarity measure from the inner product on $\mathcal{H}$, obtaining a positive-definite kernel:

$$K(x, y) = \langle k_y, k_x \rangle_{\mathcal{H}}.$$

We call K the *reproducing kernel of $\mathcal{H}$*, for $\mathcal{H}$ uniquely determines K. Recalling from before—every kernel K uniquely determines an $\mathcal{H}$—we now see that these two views of RKHS's we've provided are equivalent. And whenever we are given a set $\mathcal{X}$ with a positive-definite kernel, we can reason as though we know that they belong to a Hilbert space $\mathcal{H}$.

2.4 In Practice

In many cases, one important limitation is the infeasibility of construction Φ. And especially in a computational setting, if $\mathcal{H}$ is infinite dimensions, representation of its vectors becomes even more problematic.

Still, we can still take advantage of the existence of Φ; sometimes we can reason about properties of $\mathcal{X}$ as points in $\mathcal{H}$, but pull the analysis back through K.

In the rest of the paper, we'll look at how this may be done for LSH.

[5] The earlier point that $\mathcal{H}$ is a collection of functions over $\mathcal{X}$ is equivalent to saying that the evaluation functionals are continuous.

3 Similarity Metrics, Unusual and Usual

4 KLSH

Now we are ready to put the pieces together into KLSH. Suppose we have a set of objects $\mathcal{X}$ on which the only defined function is a simularity metric $K : \mathcal{X} \times \mathcal{X} \to [0, 1]$ (with $K(x, x) = 1$). In the most common example, $\mathcal{X}$ is a collection of images and K is whatever kernel the computer vision community recommends. And let us suppose we know K to be positive-definite.

Note that $\mathcal{X}$ is *not* generally a vector space. We cannot add two images, nor take their inner product. We cannot generate a random image drawn from a Gaussian distribution. These things are not defined. Furthermore, we might be unable to find a function Φ that would map the image into a vector space. We know one exists, but we do not have it, nor do we know any interesting properties of the Hilbert space it maps into.

Nevertheless, we would like to apply LSH techniques that are defined in the kernel space to search problems in the object space.

4.1 LSH on the Sphere

4.2 Attempt at KLSH

Let's try to kernelize the ball-carving LSH method. Given $\mathcal{X}$ and a normalized positive-definite kernel K,[6] one very natural way is to consider the embedding $\Phi(\mathcal{X}) \subset \mathcal{H}$ inside the RKHS.

In particular, because K is normalized, all the points $\Phi(x)$ fall onto the unit sphere in $\mathcal{H}$, so it seems that we just need to ball-carve the sphere in $\mathcal{H}$. In the usual LSH situation in $\mathbb{R}^n$, we drew a Gaussian $g \sim \mathcal{N}(0, 1)^n$ and hashed according the function:

$$h(x) = \text{sign } \langle x, g \rangle.$$

So it seems that we just need to replicate this within $\mathcal{H}$ instead of $\mathbb{R}^n$:

$$h(x) = \text{sign } \langle \Phi(x), g \rangle_{\mathcal{H}},$$

where g is drawn according to a Gaussian in $\mathcal{H}$. But here, we run into two problems:

1. if $\mathcal{H}$ is infinite-dimensional, it does not have a canonical Gaussian distribution,
2. if we don't have Φ, how can we compute the inner product of $\Phi(x)$ with g?

Let's take a closer look at why there is no canonical Gaussian distribution—in fact, this will provide us with some heuristic to think about how Φ might embed $\mathcal{X}$ into $\mathcal{H}$. From that insight, we will be able to reduce the infinite-dimensional problem down to a finite-dimensional one, which we essentially already know how to solve.

[6] One can normalize any positive-definite kernel $K(x, y)$ by:

$$K'(x, y) = \frac{K(x, y)}{\sqrt{K(x, x)K(y, y)}}$$

.

4.3 Projection to Finite Dimensions

Proposition 8 ([L2012, Example 2.2])**.** *Let $\mathcal{H}$ be a separable Hilbert space. A Gaussian distribution with covariance Σ exists iff, in an appropriate base, Σ has a diagonal form with non-negative eigenvalues, and the sums of these eigenvalues is finite.*

In particular, this implies that no analog of the normal distribution $\mathcal{N}(0, I)$ exists in $\mathcal{H}$ as the trace of I is infinite. Although we won't prove this,[7] we can give intuition: the normal distribution in $\mathbb{R}^n$ is the product of n independent normal Gaussians along n orthogonal directions. As the number of dimensions go to infinity, we would find that the probability distribution would converges to 0 on $\mathcal{H}$. And so any Gaussian that exists on $\mathcal{H}$ must have 'most' of its variance concentrated within a finite number of directions.

But this leads to some possible intuition utilized by papers such as [K2012,J2015], ultimately resulting in the KLSH method. We might ask how is $\Phi(\mathcal{X})$ 'distributed' within $\mathcal{H}$?[8] The proposition above therefore suggests that most of the variance of $\Phi(\mathcal{X})$ lie on a finite number of dimensions.

Slightly more precisely we could look at how a large but finite subset $S \subset \mathcal{X}$ is distributed within $\mathcal{H}$ (so as to avoid unreliable intuition if $\mathcal{X}$ is infinite). Let Σ be the covariance of $\Phi(S) = \{\Phi(x_1), \ldots, \Phi(x_m)\}$. For simplicity, let's consider Σ in its eigenbasis, so that

$$\Sigma = \sum_{i=1}^{d_\Phi} \lambda_i v_i \otimes v_i,$$

where $\lambda_1 \geq \lambda_2 \geq \ldots \geq 0$ are the eigenvalues and $v_1, v_2, \ldots$ are the (orthonormal) eigenbasis (by d_Φ, we mean the dimension of $\mathcal{H}$, which may be infinite). By Proposition 8, the sum of the eigenvalues is bounded, so for any $\epsilon > 0$, there is some $N \in \mathbb{N}$ such that:

$$\sum_{i=N}^{d_\Phi} \lambda_i < \epsilon.$$

Thus, if we project $\Phi(x)$ onto the first N eigenbasis (so, truncate $\Phi(x)$ at N coordinates in the covariance eigenbasis):

$$\pi : \Phi(x) \mapsto \sum_{i=1}^{N} \langle \Phi(x), v_i \rangle \; v_i,$$

then the expected truncation error will be less than ϵ. This essentially describes *principal component analysis* from within the RKHS (this is called KPCA).

If we project $\mathcal{H}$ into a finite-dimensional vector space by this truncation, then we can follow up with the usual LSH algorithm. At its core, this is how KLSH works, as described in [K2012]. Interestingly enough, while they did not at first view KLSH as a combination of KPCA and standard LSH at first, their later paper [J2015] proved the equivalence of their

[7] Additionally, the reader might notice that we have not actually defined what a Gaussian measure on $\mathcal{H}$ is; we refer the reader to [L2012] for details.

[8] This question is not very precise: do we have a measure on $\mathcal{X}$? Does that induce a measure in $\mathcal{H}$? Or, is there already a measure on $\mathcal{H}$, and we assume that $\Phi(\mathcal{X})$ is measurable? It might be worth pursing these questions more in the future.

original method to KPCA+LSH. While the intuition is useful, we still need to resolve one more matter: truncated or not, how is $\langle \Phi(x), g \rangle$ computed? We still only have access to $\mathcal{H}$ through the kernel map K. It turns out that this will enough.

4.4 Using the Kernel

[NEED TO WRITE]

That is, suppose we had some random variable $X \in \mathcal{X}$, inducing a random variable $\Phi(X) \in \mathcal{H}$, with some mean μ and covariance Σ. Then, if we drew t samples, $\Phi(X_1), \ldots, \Phi(X_t)\}$, CLT implies that the random variable

$$\Sigma^{-1/2} Z := \Sigma^{-1/2} \left(\sqrt{t} \sum_{i=1}^{t} \Phi(X_i) - \mu \right)$$

will converge to a standard Gaussian as t approaches ∞.

In general, $n = 30$ will be sufficient in KLSH to make a good enough approximation, though the theoretical bound is poorly studied, and extremely multimodal distributions might require larger values of n.

Let z be a realization of Z. Then, it follows that we could ball-carve using the hash:

$$h(x) = \text{sign } \left\langle \Phi(x), \Sigma^{-1/2} z \right\rangle .$$

Let's examine the term $\left\langle \Phi(x), \Sigma^{-1/2} z \right\rangle$ in the eigenbasis of Σ, where Σ has eigenvalues $\lambda_1 \geq \lambda_2 \geq \ldots \geq 0$ and eigenvalues $v_1, v_2, \ldots$. Then,

If we select a random point x from our database (equal chance of each point) and consider its $\Phi(x)$, this is a distribution in $\mathcal{H}$, albeit a strange and discrete one. That's good enough for the central limit theorem to apply. This lets us "draw" a point in $\mathcal{H}$ from $\mathcal{N}(\mu, \Sigma)$. Furthermore, we can convert this into a point from $\mathcal{N}(0, I)$ by subtracting μ and multiplying by $\Sigma^{-1/2}$. That is to say:

$$g = \Sigma^{-1/2} \left(\frac{1}{n} \sum_{i}^{n} \Phi(x_i) \right) - \mu \sim \mathcal{N}(0, I)$$

Granted, we still cannot compute this. By approximating Σ, however, we can compute per-point scalars w_i such that $g = \sum_i w_i \Phi(x_i)$. We can then use this form to compute inner products:

$$\begin{aligned} \langle \Phi(q), g \rangle &= \langle \Phi(q), \sum_i w_i \Phi(x_i) \rangle \\ &= \sum_i w_i \langle \Phi(q), \Phi(x_i) \rangle \\ &= \sum_i w_i K(q, x_i) \end{aligned}$$

4.5 Analysis of Algorithm

5 Data-Dependent KLSH

5.1 Data-Dependent LSH

Random hyperplanes are not the most efficient form of LSH for a unit sphere. They achieve only [TODO: look this up] while the best-known technique achieves $\rho = 1/(2c^2 - 1)$. This is better than the theoretical limit and is made possible by basing the hashing functions on the data.

5.1.1 Approximate Evenness

5.2 Smaller Caps

5.3 Making the Calculations

6 References

[A1950] Aronszajn, Nachman. *Theory of reproducing kernels.* Transactions of the American mathematical society 68.3 (1950): 337-404.

[E2016] Eldredge, Nathaniel *Analysis and probability on infinite-dimensional spaces.* arXiv preprint arXiv:1607.03591 (2016).

[G2013] Gretton, Arthur. *Introduction to RKHS, and some simple kernel algorithms.* Adv. Top. Mach. Learn. Lecture Conducted from University College London (2013).

[H2008] Hofmann, Thomas, Bernhard Schlkopf, and Alexander J. Smola. *Kernel methods in machine learning.* The annals of statistics (2008): 1171-1220.

[J2015] Jiang, Ke, Qichao Que, and Brian Kulis. *Revisiting kernelized locality-sensitive hashing for improved large-scale image retrieval.* Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015.

[K2012] Kulis, Brian, and Kristen Grauman. *Kernelized locality-sensitive hashing.* IEEE Transactions on Pattern Analysis and Machine Intelligence 34.6 (2012): 1092-1104.

[L2012] Lifshits, Mikhail. *Lectures on Gaussian processes.* Lectures on Gaussian Processes. Springer Berlin Heidelberg, 2012. 1-117.

[P2009] Paulsen, Vern I., and Mrinal Raghupathi. *An introduction to the theory of reproducing kernel Hilbert spaces.* Vol. 152. Cambridge University Press, 2016.