Theoretical Foundations of Hyperdimensional Computing

Anthony Thomas  
Department of Computer Science  
University of California, San Diego  
San Diego, CA 92093, USA

Sanjoy Dasgupta  
Department of Computer Science  
University of California, San Diego  
San Diego, CA 92093, USA

Tajana Rosing  
Department of Computer Science  
University of California, San Diego  
San Diego, CA 92093, USA

Abstract

Hyperdimensional (HD) computing is a set of neurally inspired methods for obtaining high-dimensional, low-precision, distributed representations of data. These representations can be combined with simple, neurally plausible algorithms to effect a variety of information processing tasks. HD computing has recently garnered significant interest from the computer hardware community as an energy-efficient, low-latency, and noise robust tool for solving learning problems. In this work, we present a unified treatment of the theoretical foundations of HD computing with a focus on the suitability of representations for learning. In addition to providing a formal structure in which to study HD computing, we provide useful guidance for practitioners and lay out important open questions warranting further study.

1. Introduction

Hyperdimensional (HD) computing is an emerging area at the intersection of computer architecture and theoretical neuroscience (Kanerva, 2009). It is based on the observation that brains are able to perform complex tasks using circuitry that: (1) uses low power, (2) requires low precision, and (3) is highly robust to data corruption. HD computing aims to carry over similar design principles to a new generation of digital devices that are highly energy-efficient, fault tolerant, and well-suited to natural information processing (Rahimi et al., 2018).

The wealth of recent work on neural networks also draws its inspiration from the brain, but modern instantiations of these methods have diverged from the desiderata above. The success of these networks has rested upon choices that are not neurally plausible: significant depth and training via backpropagation. Moreover, from a practical perspective, training these models often requires high-precision and substantial amounts of energy. While a large body of literature has sought to ameliorate these issues with neural networks, these fixes have largely amounted to ad-hoc patches designed to address a specific performance limitation. By contrast, the desiderata listed above emerge naturally from the basic architecture of HD computing.
Hyperdimensional computing focuses on the very simplest neural architectures. Typically, there is a single, static, mapping from inputs $x$ to much higher-dimensional “neural” representations $\phi(x)$ living in some space $\mathcal{H}$. All computational tasks are performed in $\mathcal{H}$-space, using simple, element-wise operations like additions and dot products. The mapping $\phi$ is often taken to be random, and the embeddings have coordinates that have low precision; for instance, they might take values $-1$ and $+1$. The entire setup is elementary and lends itself to fast, low-power hardware realizations.

Indeed, a cottage industry has emerged around developing optimized implementations of HD computing based algorithms on hardware accelerators (Imani et al., 2017b; Rahimi et al., 2018; Gupta et al., 2018; Manuel et al., 2019; Salamat et al., 2019; Imani et al., 2019d). Broadly speaking, this line of work touts HD computing as an energy efficient, low-latency, and noise-resilient alternative to conventional realizations of general purpose ML algorithms like support vector machines, multilayer-perceptrons, and nearest-neighbor classifiers. While this work has reported impressive performance benefits, there has been relatively little formal treatment of HD computing as a tool for general purpose learning. Most existing analysis has focused on the ability of HD computing to store and recall specific patterns, which is a distinct problem from learning (Kanerva, 2009; Gallant and Okaywe, 2013; Frady et al., 2018).

This review has two broad aims. The first, more modest, goal is to introduce the area of hyperdimensional computing to a machine learning audience. The second is to develop a mathematical framework for understanding and analyzing these models. The recent literature has suggested a variety of different HD architectures that conform to the overall blueprint given above, but differ in many important details. We present a unified treatment of such architectures that enables their properties to be compared. The most basic types of questions we wish to answer are:

1. How can individual items, sets of items, or sequences of items, be represented and stored in $\mathcal{H}$-space, in a manner that permits reliable decoding?
2. What kinds of noise can be tolerated in $\mathcal{H}$-space?
3. What kinds of structure in the input $x$-space are preserved by the mapping to $\mathcal{H}$-space?
4. What is the power of linear separators on the $\phi$-representation?

We answer many of these questions in generality. We then give an overview of some of the most exciting open problems in the theory of HD computing.

2. Introduction to HD Computing

In the following section we provide an introduction to the fundamentals of HD computing and provide some brief discussion of its antecedents in the neuroscience literature.

2.1 High-Dimensional Representations in Neuroscience

Neuroscience has proven to be a rich source of inspiration for the machine learning community: from the perceptron (Rosenblatt, 1958), which introduced a simple and general-purpose learning algorithm for linear classifiers, to neural networks (Rumelhart et al., 1986),
Theoretical Foundations of Hyperdimensional Computing

to convolutional architectures inspired by visual cortex (Fukushima, 1980), to sparse coding (Olshausen and Field, 1996), and independent component analysis (Bell and Sejnowski, 1995). One of the most consequential discoveries from the neuroscience community, underlining much research at the intersection of neuroscience and machine learning, has been the notion of high-dimensional distributed representations as the fundamental data structure for diverse types of information. In the neuroscience context, these representations are also typically sparse.

To give a concrete example, the sensory systems of many organisms have a critical component consisting of a transformation from relatively low dimensional sensory inputs to much higher-dimensional sparse representations. These latter representations are then used for subsequent tasks such as recall and learning. In the olfactory system of the fruit fly (Masse et al., 2009; Turner et al., 2008; Wilson, 2013; Caron et al., 2013), the mapping consists of two steps that can be roughly captured as follows:

1. An input \( x \in \mathbb{R}^n \) is collected via a sensory organ and mapped under an random linear transformation to a point \( \phi(x) \in \mathbb{R}^d \) \((d \gg n)\) in a high-dimensional space.

2. The coordinates of \( \phi(x) \) are “sparsified” by a thresholding operation which just retains the locations of the largest \( k \) coordinates.

In the fly, the olfactory input is a roughly 50-dimensional vector \((n = 50)\) while the sparse representation is roughly 2,000-dimensional \((d = 2000)\). A similar “expand-and-sparsify” template is also found in other species, suggesting that this process somehow exposes the information present in the input signal in a way that is amenable to learning by the brain (Stettler and Axel, 2009; Olshausen and Field, 2004; Chacron et al., 2011). The precise mechanisms by which this occurs are still not fully understood, but may have close connections to some of the literature on the theory of neural networks and kernel methods (Cybenko, 1989; Barron, 1993; Rahimi and Recht, 2008).

2.2 HD Computing

The notion of high-dimensional, distributed, data representations has engendered a number of computational models that have collectively come to be known as vector symbolic architectures (VSA) (Levy and Gayler, 2008). In general, VSAs provide a systematic way to generate and manipulate high-dimensional representations of symbols so as to implement cognitive operations like association between related concepts. Notable examples of VSAs include “holographic reduced representations” (Plate, 1995), “binary spatter codes” (Kanerva, 1994, 1995), and “sparse distributed memory” (Kanerva, 1988). HD computing can be seen as a successor to these early VSA models, with a strong additional slant towards hardware efficiency. While our treatment focuses primarily on recent work on HD computing, many of our results apply to these earlier VSA models as well.

An overview of data-flow in HD computing is given in Figure 1. The first step in HD computing is encoding, which maps a piece of input data to its high-dimensional representation under some function \( \phi : \mathcal{X} \rightarrow \mathcal{H} \). The nature of \( \phi \) depends on the type of input and the choice of \( \mathcal{H} \). In this review, we consider inputs consisting of sets, sequences, and structures composed from a finite alphabet as well as vectors in a Euclidean space. The space \( \mathcal{H} \) is some \( d \)-dimensional inner-product space defined over the real numbers or a subset thereof.
For computational reasons, it is common to restrict $\mathcal{H}$ to be defined over integers in the range $[-b, b]$.

The HD representations of data are manipulated using two key operators: “bundling” and “binding” (Kanerva, 2009). The bundling operator is used to compile a set of elements in $\mathcal{H}$ and takes the form of a function $\oplus : \mathcal{H} \times \mathcal{H} \to \mathcal{H}$. The function takes two points in $\mathcal{H}$ and returns a third point similar to both operands. The binding operator is used to create ordered tuples of points in $\mathcal{H}$ and is similarly a function $\otimes : \mathcal{H} \times \mathcal{H} \to \mathcal{H}$. The function takes a pair of points in $\mathcal{H}$ as input, and produces a third point dissimilar both operands. We make these notions more precise in our subsequent discussion of encoding.

Given the HD representation $\phi(S)$ of a set of items $S \subset \mathcal{X}$, we may be interested to query the representation to determine if it contains the encoding of some $x \in \mathcal{X}$. To do so, we compute a metric of similarity $\rho(\phi(x), \phi(S))$ and declare that the item is present in $S$ if the similarity is greater than some critical value $\alpha$. This process can be used to decode the HD representation so as to recover the original points in $\mathcal{X}$. We may additionally wish to assert that we can decode reliably even if $\phi(S)$ has been corrupted by some noise process. One of our chief aims is to characterize sufficient conditions for robust decoding and analyze this question under different noise models and input data types throughout this work.

Beyond simply storing and recalling specific patterns, HD representations may also be used for learning. HD computing is most naturally applicable to classification problems. Suppose we are given some collection of labeled examples $\mathcal{S} = \{(x_i, y_i)\}_{i=1}^N$, where $x_i \in \mathcal{X}$ and $y_i \in \{c_i\}_{i=1}^K$ is a categorical variable indicating the class label of a particular $x_i$. In its simplest form, HD classification simply bundles together the data corresponding to a particular class to generate a “prototypical” example for the class (Kanerva, 2009;
Kleyko et al., 2018; Rahimi et al., 2018):

$$\phi(c_k) = \bigoplus_{i: y_i = c_k} \phi(x_i)$$  \hspace{1cm} (1)

The resulting $$\phi(c_k)$$ are sometimes quantized to lower precision or sparsified via a thresholding operation. A nice feature of this scheme is that it is extremely simple to implement in an on-line fashion—that is on streaming data arriving continuously over time (Rahimi et al., 2018). It is common to fine-tune the class prototypes using a few rounds of perceptron training (Imani et al., 2017a, 2019b). Given some piece of query data $$x_q \in X$$ for which we do not know the correct label we simply return the label of the most similar prototype:

$$k^* = \arg \max_{k \in 1, \ldots, K} \rho(\phi(x_q), \phi(c_k)).$$

The similarity metric $$\rho$$ is typically taken to be the dot-product—where the operands are normalized if necessary. Thus, on the whole, the scheme is quite similar to classical simple statistical classifiers like naive-Bayes and Fisher’s linear discriminant.

3. Encoding and Decoding Discrete Data

The central object in HD computing is the mapping from inputs to their high-dimensional representations. The design of this mapping, typically referred to as “encoding” in the literature on HD computing, has been the subject of considerable research. There is a wide range of possible encoding methods. Some of these have been introduced in the HD computing literature and studied in isolation (Plate, 1995; Gallant and Okaywe, 2013; Kleyko et al., 2018). In this review, we present a novel unifying framework in which to study these mappings and to characterize their key properties. We first discuss the encoding and decoding of sets in some detail. Most HD encoding procedures for more complex data types such as sequences and multivariate data essentially amount to transforming the data into a set and then applying the standard set-encoding method.

3.1 Finite Sets

Let $$\mathcal{A} = \{a_i\}_{i=1}^m$$ be some finite alphabet of $$m$$ symbols. A symbol $$a \in \mathcal{A}$$ is mapped to $$\mathcal{H}$$ under an encoding function $$\phi: \mathcal{A} \rightarrow \mathcal{H}$$. Our goal in this section is to consider the encoding of sets $$\mathcal{S}$$ whose elements are drawn from $$\mathcal{A}$$. The HD representation of $$\mathcal{S}$$ is constructed by superimposing the embeddings of the constituent elements using the bundling operator $$\oplus: \mathcal{H} \times \mathcal{H} \rightarrow \mathcal{H}$$. The encoding of $$\mathcal{S}$$ is defined to be $$\phi(\mathcal{S}) = \oplus_{a \in \mathcal{S}} \phi(a)$$. We first focus on the intuitive setting in which $$\oplus$$ is the element-wise sum and then address other forms of bundling.

To determine if some $$a \in \mathcal{A}$$ is contained in $$\mathcal{S}$$, we check if the dot product $$\langle \phi(a), \phi(\mathcal{S}) \rangle$$ exceeds some fixed threshold. If the codewords are orthogonal, then we have $$\langle \phi(a), \phi(\mathcal{S}) \rangle = L1(a \in \mathcal{S})$$, where $$1$$ is the indicator function which evaluates to one if its argument is true and zero otherwise. However, when the codewords are not perfectly orthogonal, we have $$\langle \phi(a), \phi(\mathcal{S}) \rangle = L1(a \in \mathcal{S}) + \Delta$$, where $$\Delta$$ is a noise term caused by interference—commonly referred to as “cross-talk”—between the codewords. In order to decode reliably, we must...
ensure the contribution of the cross-talk is small and bounded. We formalize this using the notion of incoherence popularized in the sparse coding literature. We define incoherence formally as (Donoho et al., 2005):

**Definition 1. Incoherence.** For \( \mu \geq 0 \), we say \( \phi : \mathcal{A} \rightarrow \mathcal{H} \) is \( \mu \)-incoherent if for all distinct \( a, a' \in \mathcal{A} \), we have

\[
|\langle \phi(a), \phi(a') \rangle| \leq \mu L^2
\]

where \( L = \min_{a \in \mathcal{A}} \| \phi(a) \| \).

The ideal incoherence is 0, but this is only achievable if the dimension of \( \mathcal{H} \) is at least \( |\mathcal{A}| \).

### 3.1.1 Exact Decoding of Sets

In the following section, we show how the cross-talk can be bounded in terms of the incoherence of \( \phi \), and use this to derive a simple threshold rule for exact decoding.

**Theorem 2.** Let \( L = \min_{a \in \mathcal{A}} \| \phi(a) \| \) and let the bundling operator be the element wise sum. To decode whether an element \( a \) lies in set \( S \), we use the rule

\[
\langle \phi(a), \phi(S) \rangle \geq \frac{1}{2} L^2.
\]

This gives perfect decoding for sets of size \( \leq s \) if \( \phi \) is \( 1/(2s) \)-incoherent.

**Proof** Consider some symbol \( a \). Then:

\[
\langle \phi(a), \phi(S) \rangle = \mathbb{1}(a \in S) \langle \phi(a), \phi(a) \rangle + \sum_{a' \in S \setminus \{a\}} \langle \phi(a), \phi(a') \rangle
\]

If \( a \in S \), then the above is lower bounded by \( L^2 - sL^2 \mu \). Otherwise, it is upper bounded by \( sL^2 \mu \). So we decode perfectly if \( sL^2 \mu < L^2/2 \), or \( \mu < 1/(2s) \). \( \square \)

### 3.1.2 Random Codebooks

In practice, each \( \phi(a) \) is typically generated by sampling from some distribution over \( \mathcal{H} \), or a subset thereof (Kanerva, 2009; Klevko et al., 2018; Rahimi et al., 2018). We typically require that this distribution is factorized so that coordinates of \( \phi(a) \) are i.i.d.. Intuitively, the incoherence condition stipulated in Theorem 2 will hold if dot products between two different codewords are concentrated around zero. Furthermore, we would like it to be the case that this concentration occurs quickly as the encoding dimension is increased. It turns out that a broad family of simple distributions satisfies this property.

As an example, suppose \( \phi(a) \) is sampled from the uniform distribution over \( \{\pm1\}^d \), which we denote \( \phi(a) \sim \{\pm1\}^d \). In this case, \( L = \sqrt{d} \) exactly, and a direct application of Hoeffding’s inequality and the union bound yields:

\[
P(\exists a, a' \text{ s.t. } |\langle \phi(a), \phi(a') \rangle| \geq \mu d) \leq m^2 \exp \left\{ -\frac{\mu^2 d}{2} \right\}.
\]

6
Stated another way, with high probability, \( \mu = O(\sqrt{(\log m)/d}) \), meaning that we can make \( \mu \) as small as desired by increasing \( d \).

In fact, the same basic approach holds for a much broader class of distributions known as the sub-Gaussians, which are characterized as follows (Wainwright, 2019):

**Definition 3** Sub-Gaussian Random Variable. A random variable \( X \sim P_X \) is said to be sub-Gaussian if there exists \( \sigma \in \mathbb{R}^+ \), referred to as the sub-Gaussian parameter, such that:

\[
E[\exp \{ \lambda (X - E[X]) \}] \leq \exp \left\{ \frac{\sigma^2 \lambda^2}{2} \right\} \text{ for all } \lambda \in \mathbb{R}.
\]

Intuitively, the tails of a sub-Gaussian random variable decay at least as fast those of a Gaussian. Like the conventional Gaussian distribution, sub-Gaussianity is preserved under linear transformations. That is, if \( x = \{x_i\}_{i=1}^n \) is a sequence of i.i.d. sub-Gaussian random variables and \( a \) is an arbitrary vector in \( \mathbb{R}^n \), then \( \langle a, x \rangle \) is sub-Gaussian with parameter \( \sigma \|a\|_2 \) (Wainwright, 2019). We can obtain a more general version of the previous result which applies to \( \phi(a) \) sampled from any sub-Gaussian distribution.

**Theorem 4** Let \( \phi \) be randomly sampled from a mean-zero distribution with sub-Gaussian parameter \( \sigma^2 \). Then, for \( \mu > 0 \),

\[
P(\exists \, a, a' \text{ s.t. } |\langle \phi(a), \phi(a') \rangle| \geq \mu L^2) \leq m^2 \exp \left( -\frac{\mu^2 \kappa L^2}{2\sigma^2} \right)
\]

for some constant \( \kappa \leq 1 \).

**Proof** Fix some \( a \) and \( a' \). Treating \( \phi(a) \) as a fixed vector in \( \mathbb{R}^d \) and using the fact that sub-Gaussianity is preserved under linear transformations, we may apply Hoeffding’s inequality to obtain:

\[
P(|\langle \phi(a), \phi(a') \rangle| \geq \mu L^2) \leq 2 \exp \left( -\frac{\mu^2 L^4}{2\sigma^2 \|\phi(a)\|_2^2} \right) \leq 2 \exp \left( -\frac{\mu^2 L^4}{2\sigma^2 L_{max}^2} \right)
\]

where \( L_{max} = \max_{a \in A} \|\phi(a)\|_2 \). Therefore, taking \( \kappa = L^2 / L_{max}^2 \), we have:

\[
P(|\langle \phi(a), \phi(a') \rangle| \geq \mu L^2) \leq 2 \exp \left( -\frac{\mu^2 \kappa L^2}{2\sigma^2} \right)
\]

and the claim follows by applying the union bound over all \( \binom{m}{2} < m^2/2 \) pairs of codewords. We can further note that the variance of a sub-Gaussian distribution is upper bounded by its sub-Gaussian parameter, and so \( E[L_{max}^2] = E[L^2] \leq \sigma^2 d \), meaning \( \kappa \to 1 \) as \( d \) becomes large.

The assumption that the codewords are sampled from a mean-zero distribution is made for simplicity of notation and is in no way restrictive. To be concrete and provide useful practical guidance, we here introduce three running examples of codeword distributions.

**Dense Binary Codewords.** In our first example, the most common in practice in our impression, \( \phi(a) \) is sampled from the uniform distribution over the \( d \)-dimensional unit
cube. This approach is advantageous because it leads to efficient hardware implementa-
tions (Imani et al., 2017a; Rahimi et al., 2018) and is simple to analyze.

**Gaussian Codewords.** Our second example consists of codewords sampled from the
dimensional Gaussian distribution (Plate, 1995). That is, \( \phi(a) \sim \mathcal{N}(0_d, \sigma^2 I_d) \), where \( 0_d \) is the
dimensional zero vector. Here, the codewords will not be of exactly the same length.
However, we can note that \( \|\phi(a)\|_2^2 = \sigma^2 d (1 \pm \epsilon) \) where \( \epsilon \to 0 \) as \( d \to \infty \). We consider
this case largely for historical reasons and to understand theoretical implications of using
codewords from a continuous distribution. We see little advantage to using continuous
codewords in practice.

In both cases, we can see that to obtain a \( \mu \)-incoherent codebook with probability \( 1 - \delta \),
is it sufficient to choose:

\[
d = O \left( \frac{2}{\mu^2} \log \frac{m}{\delta} \right)
\]

Or, stated another way, we have \( \mu = O(\sqrt{(\log m)/d}) \) with high probability. The key point
in the two examples above is that the encoding dimension is inversely proportional to \( \mu^2 \).
Per Theorem 2, to decode correctly it is sufficient to have \( \mu = 1/2s \), meaning that the
encoding dimension exhibits quadratic scaling with the number of items to encode.

We will also consider a third example in which the codewords are sparse and binary.
However, we defer this for the time being as slightly different encoding methods and analysis
techniques are appropriate.

3.1.3 Decoding with Small Probability of Error

The analysis above gives strong uniform bounds showing that, with probability at least
1 − \( \delta \) over random choice of the codebook, every subset of size at most \( s \) will be correctly
decoded. However, this guarantee requires us to impose the unappealing restriction that
\( s \ll \sqrt{d} \) which is a significant practical limitation. We here show that we can obtain
\( s = O(d) \) but with a weaker pointwise guarantee: with probability 1 − \( \delta \), again over random
samplings of the codebook, we can decode any arbitrary set of size at most \( s \). Rather than
insist on a hard upper bound on the incoherence of the codebook, we can instead require
the milder condition that random sums over dot-products between \( \leq s \) codewords are small
with high-probability. We define this property more formally as follows:

**Definition 5 Subset Incoherence.** For \( \tau > 0 \), we say a random mapping \( \phi : A \to H \)
satisfies \( (s, \tau, \delta) \)-subset incoherence if, for any \( S \subset A \) of size at most \( s \), with probability at
least 1 − \( \delta \) over the choice of \( \phi \):

\[
\max_{a \in S} \left| \sum_{a' \in S} \langle \phi(a), \phi(a') \rangle \right| \leq \tau L^2
\]

where \( L = \min_{a \in A} ||\phi(a)|| \).

Per the analysis of Theorem 4, for some fixed \( a \neq a' \), the distribution over \( \langle \phi(a), \phi(a') \rangle \)
is sub-Gaussian with parameter \( \sigma^2 ||\phi(a)|| \). Then, again using the fact that sub-Gaussianity is
preserved under sums, we can see that codewords sampled from a sub-Gaussian distribution
will satisfy a desired subset-incoherence condition when the bundling operator is the sum. Fixing a set $S$, again by Hoeffding’s inequality and the union bound, we have:

$$\Pr(\exists a \notin S \text{ s.t. } \left| \sum_{a' \in S} \langle \phi(a), \phi(a') \rangle \right| \geq \tau L^2) \leq 2m \exp \left\{ -\frac{\kappa \tau^2 L^2}{2s\sigma^2} \right\}$$

As a concrete example, in the practically relevant case that $\phi \sim \{\pm 1\}^d$ the above boils down to:

$$\Pr(\exists a \notin S \text{ s.t. } \left| \sum_{a' \in S} \langle \phi(a), \phi(a') \rangle \right| \geq \tau d) \leq 2m \exp \left\{ -\frac{\tau^2 d}{2s} \right\}$$

Stated another way, we have: $\tau = O(\sqrt{(s \log m)/d})$. Following Theorem 2, in order to ensure correct decoding with high probability, we must simply argue that the codebook satisfies the subset-incoherence property with $\tau = 1/2$, meaning we should choose the encoding dimension to be $d = O(s \log m)$.

This method of analysis is similar to that of (Plate, 1995; Gallant and Okaywe, 2013), who reach the same conclusion vis-à-vis linear scaling using the central limit theorem. However, our formalism is non-asymptotic and more general.

### 3.1.4 Comparing Set Representations

We can estimate the size of a set by computing the norm of its encoding, where the precision of the estimate can be bounded in terms of the incoherence of $\phi$. In the following discussion, we make the simplifying assumption that the codewords are all of a constant length $L$. We again note that this assumption is not onerous in practice since, for codewords sampled i.i.d. from a sub-Gaussian distribution, the codeword lengths converge fairly rapidly to the same value as $d$ becomes large.

**Theorem 6** Let $S$ be a set of size $s$. Then:

$$s(1 - s\mu) \leq \frac{1}{L^2} \|\phi(S)\|_2^2 \leq s(1 + s\mu)$$

**Proof** The proof is by direct manipulation:

$$\frac{1}{L^2} \|\phi(S)\|_2^2 = \frac{1}{L^2} \langle \phi(S), \phi(S) \rangle = \frac{1}{L^2} \sum_{a \in S} \langle \phi(a), \phi(a) \rangle + \frac{1}{L^2} \sum_{a, a' \neq a \in S} \langle \phi(a), \phi(a') \rangle$$

$$\leq \frac{1}{L^2} (sL^2 + s^2 L^2) \mu$$

The other direction is analogous. $\blacksquare$

Given a pair of sets $S, S'$ over the same alphabet, we can estimate the size of their intersection and union directly from their encoded representation.

**Theorem 7** Let $S$ and $S'$ be sets of size $s$ and $s'$, drawn from a common alphabet of $M$ symbols and denote their encodings by $\phi(S)$ and $\phi(S')$ respectively.

$$|S \cap S'| - ss'\mu \leq \frac{1}{L^2} \langle \phi(S), \phi(S') \rangle \leq |S \cap S'| + ss'\mu$$
Proof Expanding the dot product between the two representations:

\[
\frac{1}{L^2} \langle \phi(S), \phi(S') \rangle = \frac{1}{L^2} \sum_{a \in S \cap S'} \langle \phi(a), \phi(a) \rangle + \frac{1}{L^2} \sum_{a \in S \setminus S'} \sum_{a' \in S' \setminus S} \langle \phi(a), \phi(a') \rangle
\]

\[\leq |S \cap S'| + ss'\mu.\]

The other direction is analogous. ■

Noting as well that \(|S \cup S'| = |S| + |S'| - |S \cap S'|\), we see that we can estimate the size of the union using the previous theorem. In practice, it may be unnecessary to compute these quantities with a high degree of precision. For instance, it may only be necessary to identify sets with a large intersection-over-union. Provided the definition of “large” is somewhat loose, we can accept a larger incoherence among the codewords in exchange for reducing the encoding dimension.

3.1.5 Sparse and Low-Precision Encodings

In the previous discussion, we assumed the bundling operator was the element-wise sum. This is a natural choice when the codewords are dense or non-binary. However, the resulting encodings are of unconstrained precision which may be undesirable from a computational perspective. For the purposes of representing sets of size \(\leq s\), we may truncate \(\phi(S)\) to lie in the range \([-c, c]\), with negligible loss in accuracy provided \(c = O(\sqrt{s})\). In practice, it is common to quantize the encodings more aggressively to binary precision by thresholding (Rahimi et al., 2017; Kim et al., 2018; Burrello et al., 2018; Imani et al., 2019). In other words, we encode as \(\phi(S) = g_t(\phi(S))\), where \(g_t\) is a function applied coordinate wise and has the effect of thresholding at \(t\). That is, \(g_t(x) = 1\) if \(x \geq t\) and 0 otherwise.

As a notable special case of the thresholding rule described above, we here consider encoding with sparse codewords. In this case, we assume that a coordinate in a codeword is non-zero with some small probability. In other words, \(\phi(a)_i \sim \text{Bernoulli}(p)\), where \(p \ll 1/2\). We may then choose the threshold \(t = 1\), which is equivalent to taking the element wise maximum over the codewords. That is, \(\phi(S) = \max_{a \in S} \phi(a)\), where the max operator is applied coordinate-wise. Noting that the max is upper bounded by the sum, the analysis of Theorem 2 continues to apply and we can decode by thresholding \(\langle \phi(a), \phi(S) \rangle\) at \(L^2/2\), where \(L = \min_{a \in A} \|\phi(a)\|_2\).

This encoding procedure is essentially a standard implementation of the popular “Bloom filter” data structure for representing sets (Bloom, 1970). The conventional Bloom filter differs slightly in that the typical decoding rule is to threshold \(\langle \phi(a), \phi(S) \rangle\) at \(\|\phi(a)\|_1\).

There is a vast literature on Bloom filters with applications ranging from networking and database systems to neural coding, and several schemes for generating good codewords have been proposed (Broder and Mitzenmacher, 2004; Pagh et al., 2005; Dasgupta et al., 2018). Using the random coding scheme described here, the optimal value of \(p\) can be seen to be \((\ln 2)/s\) and, to ensure the probability of a false positive is at most \(\delta\), the encoding dimension should be chosen on the order of \(s \log(1/\delta)\) (Broder and Mitzenmacher, 2004).

A disadvantage of this encoding is that the density of \(\phi(S)\) increases as more elements are encoded into the set, necessitating a larger encoding dimension. In the HD literature, a method known as “context dependent thinning” (CDT) is used to control the density of the
representation (Rachkovskii, 2001; Kleyko et al., 2018). CDT takes the logical “and” of \( \phi(S) \) and some permutation \( \sigma(\phi(S)) \) to obtain the thinned representation \( \phi(S)' = \phi(S) \land \sigma(\phi(S)) \). This process can be repeated until the desired density of \( \phi(S) \) is achieved. A capacity analysis of CDT representations can be found in (Kleyko et al., 2018).

3.2 Robustness to Noise

In this section we explore the noise robustness properties of the encoding methods discussed above. Noise robustness is an oft cited strength of HD computing; however, we are unaware of any work providing a broad, systematic analysis of the effects of noise. We consider a noise process which corrupts the encoding of a set \( S \subseteq A \) of size at most \( s \) according to \( \tilde{\phi}(S) = \phi(S) + \Delta S \). We say \( \Delta S \) is \( \rho \)-bounded if:

\[
\max_{a \in A} |\langle \phi(a), \Delta S \rangle| \leq \rho.
\]

We are interested in understanding the conditions under which we can still decode reliably.

**Theorem 8** Suppose \( S \) has size \( \leq s \) and \( \Delta S \) is \( \rho \)-bounded. Then we can correctly decode \( S \) using the thresholding rule from Theorem 2 if:

\[
\frac{\rho}{L^2} + s\mu < \frac{1}{2}
\]

where \( L = \min_{a \in A} \|\phi(a)\|_2 \)

**Proof** Consider some symbol \( a \in A \). In the event \( a \in S \):

\[
\langle \phi(a), \phi(S) + \Delta S \rangle = \langle \phi(a), \phi(S) \rangle + \langle \phi(a), \Delta S \rangle \geq L^2 - sL^2\mu - \rho
\]

and when \( a \notin S \):

\[
\langle \phi(a), \phi(S) + \Delta S \rangle \leq sL^2\mu + \rho
\]

Therefore we can decode correctly if:

\[
\frac{\rho}{L^2} + s\mu < \frac{1}{2}
\]

We can analyze several practically relevant noise models by placing additional restrictions on \( \Delta S \) and by considering worst or typical case bounds on \( \rho \). We here consider different forms of noise under constraints on \( \mathcal{H} \). Our goal is to understand how the magnitude of noise that can be tolerated scales with the encoding dimension, size \( s \) of the encoded set, and size \( m \) of the alphabet. In the interest of clarity, we here make the simplifying assumption that all codewords are of the same length and approximate the incoherence as \( \mu \approx \sqrt{(\log m)/d} \).

### 3.2.1 Gaussian Codewords

We here consider the case that codewords are drawn from a standard Gaussian distribution, and that \( \mathcal{H} = \mathbb{R}^d \). This accords with the practically relevant setting discussed above in which the bundling operator is chosen to be the element wise sum. We first consider a simple additive white Gaussian noise model.
**Lemma 9** Let $\Delta_S \sim \mathcal{N}(0_d, \sigma^2_d I_d)$. Then we can decode reliably if

$$\sigma_\Delta = O(\sqrt{d/\log m})$$

**Proof** Fixing some $\phi$, and again recalling that Gaussianity is preserved under linear combinations, we have:

$$\langle \phi(a), \Delta_S \rangle \sim \mathcal{N}(0, \sigma^2_\Delta \| \phi(a) \|^2_2)$$

Then, using a standard Gaussian tail bound (Wainwright, 2019), and the union bound:

$$P(\exists a \text{ s.t. } |\langle \phi(a), \Delta_S \rangle| \geq \rho) \leq m^2 \exp\left\{-\frac{\rho^2}{2\sigma^2_\Delta L^2}\right\}$$

meaning $\rho = O(\sigma_\Delta L \sqrt{\log m})$. Using Theorem 8 yields:

$$\frac{\sigma_\Delta \sqrt{\log m}}{L} + s\mu < \frac{1}{2}$$

Noting that, with high-probability, $L = O(\sqrt{d})$, we have:

$$\frac{\sqrt{\log m}}{\sqrt{d}}(\sigma_\Delta + s) < \frac{1}{2}$$

from which the result follows.

We can additionally consider an “adversarial” setting in which a malicious actor gains access to some set of coordinates and tries to corrupt them so as to cause a decoding error. We suppose that the adversary can access any set of coordinates, but that the total magnitude of noise introduced is bounded.

**Lemma 10** Suppose $\|\Delta_S\|_2 \leq \omega L$. Then we can decode reliably if:

$$\omega < \frac{1}{2} - s\sqrt{(\log m)/d}$$

**Proof** The result is obtain by using the Cauchy-Schwarz inequality: $\rho = |\langle \phi(a), \Delta_S \rangle| \leq \omega L^2$, and Theorem 8.

**3.2.2 Dense Binary Codewords**

We here suppose that $\phi \sim \{\pm 1\}^d$, and that $H$ is restricted to be integers in the range $\{-c, ..., c\}$. As an analogue to the white Gaussian noise model considered above, we suppose each coordinate in $\phi(S)$ is corrupted by a value chosen uniformly at random from $\{-c, ..., c\}$.

**Lemma 11** Let $\Delta_S \sim \{-c, ..., c\}^d$ for some $c \in \mathbb{Z}^+$. Then we can decode reliably if:

$$c = O(\sqrt{d/\log m})$$
Proof Noting that \( \langle \phi(a), \Delta_S \rangle \) is the sum of \( d \) random variables bounded in the range \([-c, c]\), another application of Hoeffding’s inequality yields that, with high probability \( \rho = O(c \sqrt{d \log m}) \). Applying Theorem 8 and noting that \( L = \sqrt{d} \) exactly:

\[
\frac{c \sqrt{\log m}}{\sqrt{d}} + s \frac{\sqrt{\log m}}{\sqrt{d}} < \frac{1}{2}
\]

from which the result follows.

We can also consider the effect of adversarial noise in this context. We assume that the adversary gains access to some fraction \( \omega \cdot d \) of bits in the representation and tries to corrupt them so as to cause a decoding error.

**Lemma 12** Suppose \( \| \Delta_S \|_1 \leq \omega d \). Then we can decode reliably if:

\[
\omega < \frac{1}{c^2} \left( \frac{1}{2} - \frac{s}{p} \sqrt{(\log m)/d} \right)
\]

**Proof** The result is obtained by noting that \( |\langle \phi(a), \Delta_S \rangle| \leq \| \phi(a) \|_\infty \| \Delta_S \|_1 \leq \omega c^2 d \) and applying Theorem 8.

Noting that \( \mu \geq 0 \), there is a hard limit of \( \omega < 1/2c^2 \) that cannot be improved upon by simply choosing a larger encoding dimension.

### 3.2.3 Sparse Binary Codewords

Finally, we consider the case that \( \mathcal{H} = \{0, 1\}^d \) and codewords are sparse and binary as in Section 3.1.5. We here again consider the case that an adversary gains access to some arbitrary set of \( \omega \cdot d \) bits in the representation.

**Lemma 13** Suppose \( \| \Delta_S \|_1 \leq \omega \cdot d \). Then we can decode reliably if:

\[
\omega < \frac{p}{2} - sp \sqrt{(\log m)/d}
\]

where \( p \) is the probability a coordinate in \( \phi \) is non-zero.

**Proof** We first observe that \( \rho = |\langle \phi(a), \Delta_S \rangle| \leq \| \phi(a) \|_\infty \| \Delta_S \|_1 \leq \omega d \) for any \( a \in \mathcal{A} \). Furthermore, we have \( L^2 = p \cdot d(1 \pm \epsilon) \), where \( \epsilon \to 0 \) in \( d \). Then, by Theorem 8 we want:

\[
\frac{\omega}{p} + s \mu < \frac{1}{2}
\]

The result follows by some brief algebra.

The lemma above illustrates an important tradeoff with sparse codewords in the presence of adversarial noise. When the codewords are sparse, only a few bits matter for decoding a particular codeword, and so the representation is more susceptible to adversarial noise in a worst case sense.
4. Encoding Structures

We are often interested in representing more complex data types, such as objects with multiple attributes. In general, we suppose that we observe a set of attributes $\mathcal{F}$ whose values are assumed to lie in some set $\mathcal{A}$. Let $\psi : \mathcal{F} \rightarrow \mathcal{H}$ be an embedding of features, and $\phi : \mathcal{A} \rightarrow \mathcal{H}$ be an embedding of values. We associate a feature with its value through use of the binding operator $\otimes : \mathcal{H} \times \mathcal{H} \rightarrow \mathcal{H}$ that creates an embedding for a (feature,value) pair. For a feature $f \in \mathcal{F}$ taking on a value $a \in \mathcal{A}$, its embedding is constructed as $\psi(f) \otimes \phi(a)$.

A data point $x = \{(f_i \in \mathcal{F}, x_i \in \mathcal{A})\}_{i=1}^n$ consists of $n$ such pairs. The entire embedding for $x$ is constructed as (Kanerva, 2009):

$$\phi(x) = \bigoplus_{i=1}^n \psi(f_i) \otimes \phi(x_i) \quad (2)$$

As with sets we would like $\phi(x)$ to be decodable in the sense that we can recover the value associated with a particular feature, and comparable in the sense that $\langle \phi(x), \phi(x') \rangle$ is reflective of a reasonable notion of similarity between $x$ and $x'$.

From a formal perspective, we require that $\mathcal{H}$ be a group under the binding operator. That is to say, the binding operator should be associative, there must exist an identity element with respect to $\otimes$, and any point in $\mathcal{H}$ must have an inverse with respect to $\otimes$. Additionally, binding should not change the length of an embedding, that is to say: $\|\psi(f) \otimes \phi(a)\|_2 = \|\phi(a)\|_2$. Finally, the bound pairs should satisfy an incoherence property.

We say the binding is $\mu$-incoherent if:

$$\max_{a \in \mathcal{A}} \max_{a' \in \mathcal{A}, f \in \mathcal{F}} \langle \phi(a), \psi(f) \otimes \phi(a') \rangle \leq \mu L^2$$

where $L = \min_{a \in \mathcal{A}} \|\phi(a)\|_2$. A natural choice of embedding satisfying these properties is to sample $\phi(f)$ randomly from $\{\pm 1\}^d$ and choose $\otimes$ to be the element-wise product. In this case $\psi(f)$ is its own inverse and an extension of Theorem 4 will show that incoherence is satisfied with high probability:

**Theorem 14** Fix $d, n, m \in \mathbb{Z}^+$ and $\mu \in \mathbb{R}^+$. Let $\phi(a)$ be sampled from a centered sub-Gaussian distribution with parameter $\sigma^2$, $\otimes$ be the element-wise product, and $\psi(f) \sim \{\pm 1\}^d$. Then:

$$\mathbb{P}(\exists a, a', f \text{ s.t. } \|\phi(a) \otimes \phi(a') \otimes \psi(f)\| \geq \mu L^2) \leq nm^2 \exp \left\{ -\frac{\kappa \mu^2 L^2}{2\sigma^2} \right\}$$

where $L = \min_{a \in \mathcal{A}} \|\phi(a)\|_2$ and $\kappa$ as defined in Theorem 4.

**Proof** Note first that $\|\phi(a) \otimes \psi(f)\|_2 = \|\phi(a)\|_2$. Then, fixing $a, a'$ and $f$, by Hoeffding’s inequality:

$$\mathbb{P}(\|\phi(a) \otimes \phi(a') \otimes \psi(f)\| \geq \mu L^2) \leq 2 \exp \left\{ -\frac{L^4 \mu^2}{2\sigma^2 \|\phi(a)\|_2^2} \right\} \leq 2 \exp \left\{ -\frac{k \mu^2 L^2}{2\sigma^2} \right\}$$

The result follows by the union bound over all $< nm^2$ combinations of $a, a', f$. ■

We can more directly relate the above to the encoding dimension $d$, by again noting that
the lengths of the codewords are concentrated around their mean of $\sigma \sqrt{d}$, meaning that, to obtain a $\mu$-incoherent binding, we should choose $d = O((\log nm)/\mu^2)$. As a corollary to the previous theorem, we also obtain the following incoherence property:

$$\mathbb{P}(\exists a, a', f \neq f' \text{ s.t. } |(\phi(a) \otimes \psi(f), \phi(a') \otimes \psi(f'))| \geq \mu L^2) \leq m^2 n^2 \exp\left\{ -\frac{\mu^2 L^2}{2\sigma^2} \right\}$$ (3)

We note that the above refers to symbols in different positions and thus does not require any particular incoherence assumption on the $\phi(a)$.

### 4.1 Decoding Structures

This representation can be decoded to recover the value associated with a particular feature. To recover the value of the $i$-th feature, we use the following rule:

$$\hat{x}_i = \arg\max_{a \in A} \langle \phi(a), \phi(x) \otimes \psi^{-1}(f_i) \rangle$$ (4)

Since the binding operator is assumed to distribute over bundling, the dot-product above expands to:

$$\langle \phi(a), \phi(x_i) \rangle + \sum_{j \neq i} \langle \phi(a), (\phi(x_j) \otimes \psi(f_j)) \otimes \psi^{-1}(f_i) \rangle$$

$$\begin{cases}
\geq L^2(1 - n\mu) & \text{if } x_i = a \\
\leq nL^2\mu & \text{otherwise}
\end{cases}$$

where $\mu$ can be bounded in terms of $L$ as described in Theorem 14. Following Theorem 2, we can decode if $\mu < 1/2n$.

### 4.2 Comparing Structures

As with sets, we may wish to compare two structures without decoding them. As one would expect given Theorem 7, this is can be achieved by computing the dot-product between their encodings:

**Theorem 15** Let $x$ and $x'$ be two structures drawn from a common alphabet $F \times A$. Denote their encodings using Equation 2 by $\phi(x)$ and $\phi(x')$. Then:

$$|x \cap x'| - n^2 \mu \leq \frac{1}{L^2} \langle \phi(x), \phi(x') \rangle \leq |x \cap x'| + n^2 \mu$$

where $x \cap x'$ is defined to be the set $\{i : x_i = x'_i\}_{i=1}^N$.

**Proof** Expanding:

$$\langle \phi(x), \phi(x') \rangle = \sum_{i=1}^{n} \phi(x_i) \otimes \psi(f_i), \sum_{j=1}^{n} \phi(x'_j) \otimes \psi(f_j)$$

$$= \sum_{i=1}^{n} \langle \phi(x_i) \otimes \psi(f_i), \phi(x'_i) \otimes \psi(f_i) \rangle + \sum_{i \neq j} \langle \phi(x_i) \otimes \psi(f_i), \phi(x'_j) \otimes \psi(f_j) \rangle$$
A term in the first sum is $L^2$ if $x_i = x_i'$ and $\in \pm L^2 \mu$ otherwise. So the expression above is bounded as:

$$\leq L^2 |x \cap x'| + L^2 n^2 \mu$$

and the other direction of the inequality is analogous.

As a practical example, in bioinformatics it is common to search for regions of high similarity between a “reference” and “query” genome. Work in Imani et al. (2018); Kim et al. (2020) explored the use HD computing to accelerate this process by encoding short segments of DNA and estimating similarity on the HD representations.

4.3 Encoding Sequences

Sequences are an important form of structured data. In this case, the feature set is simply the list of positions $\{1, 2, 3, \ldots\}$ in the sequence. In practical applications, we are often interested in streams of data which arrive continuously over time. Typically, real-world processes do not exhibit infinite memory and we only need to store the $n \geq 1$ most recent observations at any time. In the streaming setting, we would like to avoid needing to fully re-encode all $n$ data points each time we receive a new sample, as would be the case using the method described above. This motivates the use of shift based encoding schemes (Kanerva, 2009; Rahimi et al., 2017b; Kim et al., 2018). Let $\rho^{(i)}(z)$ denote a cyclic left-shift of the elements of $z$ by $i$ coordinates, and $\rho^{(-i)}(z)$ denotes a cyclic right-shift by $i$ coordinates.

In other words:

$$\rho^{(1)}((z_1, z_2, \ldots, z_{d-1}, z_d)) = (z_2, z_3, \ldots, z_d, z_1).$$

In shift-based encoding a sequence $x = (x_1, \ldots, x_n)$ is represented as:

$$\phi(x) = \bigoplus_{i=1}^{n} \rho^{(i-1)}(\phi(x_i)),$$

where we take $\oplus$ to be the element wise sum. Now suppose we receive symbol $n+1$ and wish to append it to $\phi(x)$ while removing $\phi(x_1)$. Then using the fact that $\rho^{-1}(\rho^2(\phi(x))) = \rho(\phi(s))$ we apply the rule:

$$\rho^{-1}(\phi(x) - \phi(x_1)) \oplus \rho^n(\phi(x_{n+1})) = \phi(x_2) \oplus \bigoplus_{i=1}^{n} \rho^{(i)}\phi(x_{i+2})$$

where we can additionally note that $\rho$ is a special type of permutation and that permutations distribute over sums. However, in order to decode correctly, each $\phi(a)$ must satisfy an incoherence condition with the $\rho^{(j)}(\phi(a'))$. We can again use the randomly generated nature of the codewords to argue this is the case, however, we must impose the additional restriction that the $\phi(a)$ be bounded, and accordingly restrict attention to the case $\phi(a) \sim \{\pm 1\}^d$.

Theorem 16 Fix $d, m, n < d \in \mathbb{Z}^+$ and $\mu \in \mathbb{R}^+$ and let $\phi(a) \sim \{\pm 1\}^d$. Then:

$$\mathbb{P}(\exists a, a', i \neq 0 \text{ s.t. } |\langle \phi(a), \rho^{(i)}(\phi(a')) \rangle| \geq \mu d) \leq nm^2 \exp\left\{ -\frac{\mu^2 d}{4} \right\}$$

16
Proof Fix some \(a, a'\) and \(i\). First consider the case that \(a \neq a'\). Then \(\phi(a)\) and \(\rho(i)(\phi(a'))\) are mutually independent and their dot product can be bounded using Hoeffding’s inequality as above:

\[
P(\|\langle \phi(a), \rho(i)(\phi(a')) \rangle \| \geq \mu d) \leq 2 \exp \left\{ -\frac{\mu^2 d}{2} \right\}
\]

When \(a = a'\), \(\phi(a)\) and \(\rho(i)(\phi(a))\) are pairwise, but not mutually, independent. Let \(f(\phi(a)) = \langle \phi(a), \rho(i)(\phi(a)) \rangle\), and denote by \(\phi(a)\) the vector formed by replacing the \(k\)-th coordinate in \(\phi(a)\) with an arbitrary value \(\in \{\pm 1\}\). Then \(|f(\phi(a)) - f(\phi(a)^k)| \leq 4\) and we can recover the same bound as above using McDiarmid’s inequality (Wainwright, 2019):

\[
P(\|\langle \phi(a), \rho(i)(\phi(a')) \rangle \| \geq \mu d) \leq 2 \exp \left\{ -\frac{\mu^2 d}{4} \right\}
\]

5. Encoding Euclidean Data

One option for encoding Euclidean vectors is to treat them as a special case of the “structured data” considered in the preceding section. As before, we think of our data as a collection of (feature,value) pairs \(x = \{f_i, x_i\}_{i=1}^n\) with the important caveat that \(x_i \in \mathbb{R}^n\). This case is more complex because the feature values may now be continuous, and because the data possesses geometric structure which is typically relevant for downstream tasks and must be preserved by encoding. We here analyze two of the most widely used methods for encoding Euclidean data and discuss general properties of structure preserving embeddings in the context of HD computing.

5.1 Position-ID Encoding

Perhaps the most widely-used method in practice relies on quantizing the raw signal to a suitably low precision and then simply applies the structure encoding method discussed in the previous section (Rachkovskiy et al., 2005a,b; Klevko et al., 2018).

In this approach, we first quantize the support of each feature \(f \in \mathcal{F}\) into some set of \(m\) bins with centroids \(a_1, \ldots, a_m\) and assign each bin a codeword \(\phi(a) \in \mathcal{H}\). However, instead of requiring the codewords to be incoherent, we now require the correlation between codewords to reflect the distance between corresponding quantizer bins. In other words \(\langle \phi(a), \phi(a') \rangle\) should be monotonically decreasing in \(|a - a'|\).

A simple method can be used to generate monotonic codebooks when the codewords are randomly sampled from \(\{\pm 1\}^d\) (Imani et al., 2017a; Rahimi et al., 2018). Fixing some feature \(f\), the codeword for the minimal quantizer bin–\(\phi(a_1)\)–is generated by sampling randomly from \(\{\pm 1\}^d\). To generate the codeword for the second bin, we simply flip some set of \([b]\) bits in \(\phi(a_1)\), where:

\[
b = \frac{a_2 - a_1}{a_m - a_1} \cdot \frac{d}{2}
\]

The codeword for the third bin is generated analogously from the second, where we assume the bits to be flipped are sampled such that a bit is flipped at most once. Thus the codewords
for the minimal and maximal bins are orthogonal and the correlation between codewords for intermediate bins is proportional to their distance.

In practice, it seems to be typical to use a single codebook for all features and for the quantizer to be a set of evenly spaced bins over the support of the data. While simple, this approach is likely to have sub-optimal rate when the features are on different scales or are far from the uniform distribution. Encoding then proceeds as follows:

\[
\phi(x) = \sum_{i=1}^{n} \phi(x_i) \otimes \psi(f_i)
\]

where, as before \(\psi \in \{\pm 1\}^{d}\) is a vector which encodes the index \(i\) of a feature value \(x_i\) as in the previous section on encoding sequences—hence the name “position-ID” encoding. There are several minor variations on this theme which are compared empirically in (Kleyko et al., 2018).

This general encoding method was analyzed in (Rachkovskiy et al., 2005b), in the specific case of sparse and binary codewords, who show it preserves the \(L_1\) distance between points in expectation but do not provide distortion bounds. We here provide such bounds using our formalism of matrix incoherence. We assume that the underlying quantization of the points is sufficiently fine that it is a low-order term that can be ignored.

**Theorem 17** Let \(x\) and \(x'\) be points in \([0,1]^n\) with encodings \(\phi(x)\) and \(\phi(x')\) generated using the rule described above. Assume that \(\phi\) satisfies \(\langle \phi(a), \phi(a')\rangle = d(1 - |a - a'|)\) for all \(a, a' \in A\), and let \(\psi \sim \{\pm 1\}^{d}\). Then:

\[
2d(\|x - x'\|_1 - 2n^2 \mu) \leq \|\phi(x) - \phi(x')\|^2 \leq 2d(\|x - x'\|_1 + 2n^2 \mu)
\]

**Proof** Expanding:

\[
\|\phi(x) - \phi(x')\|^2 = \|\phi(x)\|_2^2 + \|\phi(x')\|_2^2 - 2\langle \phi(x), \phi(x')\rangle
\]

Noting first that \(\|\phi(x)\|_2^2 = nd + \Delta\), where \(\Delta\) is a mean-zero noise term due to cross-talk between the codewords. The dot-product expands to, neglecting minor errors from the ceiling function:

\[
\langle \phi(x), \phi(x')\rangle = \sum_{i=1}^{n} \langle \phi(x_i) \otimes \psi(f_i), \phi(x'_i) \otimes \psi(f_i) \rangle + \sum_{i \neq j} \langle \phi(x_i) \otimes \psi(f_i), \phi(x'_j) \otimes \psi(f_j) \rangle
\]

\[
= \sum_{i=1}^{n} \langle \phi(x_i), \phi(x'_i) \rangle + \Delta' = \sum_{i=1}^{n} d(1 - |a(x_i) - a(x'_i)|) + \Delta'
\]

where \(a(x_i)\) is taken to be the centroid corresponding to \(x_i\). Putting both together and noting that \(\Delta, \Delta' \leq n^2 d \mu\) we have:

\[
2d(\|x - x'\|_1 - 2n^2 \mu) \leq \|\phi(x) - \phi(x')\|^2 \leq 2d(\|x - x'\|_1 + 2n^2 \mu)
\]

where the incoherence can be bounded as in Corollary 3.
A pointwise version of the result above can also be obtained using the notion of subset incoherence. The practical implication of the previous theorem is that the position-ID encoding method preserves the L1 distance between points up to an additive distortion which can be bounded by the incoherence of the codebook. Noting that \( ||\phi(x)||^2_2 \leq nd \pm n^2d\mu \), we can see that the encodings of each point are roughly of equal norm and lie in a ball of radius at most \( n\sqrt{d\mu} \), where the exact position depends on the instantiation of the codebook. Thus, we can loosely interpret the encoding procedure as mapping the data into a thin shell around the surface of a high dimensional sphere.

5.2 Random Projection Encoding

Work in (Imani et al., 2019c) uses an encoding method based on random embeddings of data to avoid the need for quantizing the low-dimensional representation of the signal. Suppose we wish to encode a vector \( x \in \mathbb{R}^n \). We can randomly embed the data into \( \mathcal{H} \) under a linear map \( \Phi \) followed by quantization:

\[
\phi(s) = g(\Phi x)
\]

where \( \Phi \in \mathbb{R}^{d \times n} \) is a matrix whose rows are sampled randomly from the surface of the \( n \)-dimensional unit sphere, and \( g \) is a quantizer – typically the sign function (Imani et al., 2019c) – restricting the embedding to \( \mathcal{H} \). This encoding method has also been studied in the context of kernel approximation where it is used to approximate the angular kernel (Choromanski et al., 2017), and to construct low-distortion binary embeddings (Jacques et al., 2013; Plan and Vershynin, 2014; Huynh and Saab, 2020). While the following result is known, we here show this encoding method preserves angular distance up to an additive distortion as this fact is important for subsequent analysis.

**Theorem 18** Let \( S^{n-1} \subset \mathbb{R}^n \) denote the \( n \)-dimensional unit sphere. Let \( \Phi \in \mathbb{R}^{d \times n} \) be a matrix whose rows are directions sampled uniformly at random from \( S^{n-1} \). Let \( X \) be a set of points supported on \( S^{n-1} \). Denote the embedding of a point by \( \phi(x) = \text{sign}(\Phi x) \). Then, for any \( x, x' \in X \), with high probability:

\[
d\theta - O(\sqrt{d}) \leq d_{\text{ham}}(\phi(x), \phi(x')) \leq d\theta + O(\sqrt{d})
\]

where \( d_{\text{ham}}(a, b) \) is the Hamming distance between \( a \) and \( b \), defined to be the number of coordinates on which \( a \) and \( b \) differ, and \( \theta = \frac{1}{\pi} \cos^{-1}(\langle x, x' \rangle) \in [0, 1] \) is proportional to the angle between \( x \) and \( x' \).

**Proof** Let \( \phi_i \) denote the \( i \)-th row of the matrix \( \Phi \). Then, the \( i \)-th coordinate in the embedding of \( x \) can be written as \( \text{sign}(\langle \phi_i, x \rangle) \). The probability that the embeddings differ on their \( i \)-th coordinate, that is \( \langle \phi_i, x \rangle \langle \phi_i, x' \rangle < 0 \), is exactly \( \theta/\pi \), where \( \theta \) is the angle between \( x \) and \( x' \).

Therefore, the number of coordinates on which \( \phi(x) \) and \( \phi(x') \) disagree is, concentrated in the range, \( d(\theta \pm \epsilon) \). By Chernoff/Hoeffding, we have that with probability \( 1 - \delta \):

\[
d\epsilon \leq \sqrt{2d \log \frac{2}{\delta}}.
\]

Noting that \( \langle \phi(x), \phi(x') \rangle = d - 2d_{\text{ham}}(\phi(x), \phi(x')) \), we obtain the following simple corollary:
Corollary 19 Let $\phi$ and $\theta$ be as defined in Theorem 18. Then, with high probability:

$$d(1 - 2\theta) - O(\sqrt{d}) \leq \langle \phi(x), \phi(x') \rangle \leq d(1 - 2\theta) + O(\sqrt{d})$$

To obtain a more explicit relationship with the dot product, we can use the first-order approximation $\cos^{-1}(x) \approx \frac{1}{2} - \frac{1}{\pi^2} \langle x, x' \rangle$, to obtain $\theta \approx \frac{1}{2} - \frac{4}{\pi^2} \langle x, x' \rangle$, from which we obtain the approximation:

$$d(1 - 2\theta) \approx \frac{2d}{\pi} \langle x, x' \rangle.$$

5.2.1 Connection with Kernel Approximation

This method of encoding is closely related to the notion of “random Fourier features” which has been widely used for kernel approximation (Rahimi and Recht, 2008). The basic idea of random Fourier features is to construct a randomized feature map $\phi : \mathbb{R}^n \rightarrow \mathbb{R}^d$, such that $\langle \phi(x), \phi(x') \rangle \approx k(x, x')$, where $k$ is a shift-invariant kernel. The construction exploits the fact that the Fourier transform of a properly scaled, shift-invariant kernel $k$ is a probability measure—a well known result from functional analysis known as Bochner’s Theorem (Rudin, 1962). The feature map itself is given by $\phi(x) = \frac{1}{\sqrt{d}} \cos(\Phi x + b)$, where the rows of $\Phi$ are sampled from the distribution induced by $k$ and the coordinates of $b$ are sampled uniformly at random from $[0, 2\pi]$.

Subsequent work in (Raginsky and Lazebnik, 2009) gave a simple scheme for quantizing the embeddings produced from random Fourier features to binary precision. Their construction yields an embedding $\psi : \mathbb{R}^n \rightarrow \{0, 1\}^d$ such that:

$$f_1(k(x, x')) - \Delta \leq \frac{1}{d} d_{ham}(\psi(x), \psi(x')) \leq f_2(k(x, x')) + \Delta$$

where $f_1, f_2 : \mathbb{R} \rightarrow \mathbb{R}$ are independent of the choice of kernel, and $\Delta$ is a distortion term. The embedding itself is constructed by applying a quantizer $Q_t(x) = \text{sign}(x + t)$ coordinate wise over the embeddings constructed from random Fourier features. In other words $\psi(x)_i = \frac{1}{2}(1 + Q_{t_i}(\phi(x)_i))$, where $t_i \sim \text{Unif}[-1, 1]$, and $\phi(x)$ is a random Fourier feature. An appealing property of this construction is that the distortion term scales with the intrinsic dimension of the data. When the data is a finite set $\mathcal{X} \subset \mathbb{R}^n$ lying on an $m < n$ dimensional Riemannian sub-manifold, $\Delta$ scales with $m$ rather than $n$. This implies it is actually possible to obtain low-distortion dimensionality reducing embeddings when the intrinsic dimension of the data is small.

This connection is highly appealing for HD computing. The quantized random Fourier feature scheme presents a simple, neurally plausible, recipe for constructing encoding methods meeting the desiderata of HD computing and preserving a rich variety of structure in data. For instance, shift-invariant kernels preserving the L1 and L2 distance—among many others—can be approximated using the method discussed above. Furthermore, this observation provides a natural point of contact between HD computing and the vast literature on kernel methods which has produced a wealth of algorithmic and theoretical insights. The authors are of the opinion that further exploring these connections would be of benefit for those interested in HD computing.
5.3 Consequences of Distance Preservation

The encoding methods discussed above are both appealing because they preserve reasonable notions of distance between points in the original data. Distance preservation is a sufficient condition to establish other desirable properties of encodings, namely preservation of neighborhood/cluster structure, robustness to various forms of noise, and in some cases preservation of linear separability. We address the first two items here and defer the latter for our discussion of learning on HD representations. We formalize our notion of distance preservation as follows:

**Definition 20 Metric-Preserving Embedding:** Let \( \delta \) be a metric on \( \mathbb{R}^n \) and \( \delta_H \) be a metric on \( \mathcal{H} \). We say \( \phi \) preserves the \( \delta \) metric under \( \delta_H \) if there exist functions \( \alpha, \beta : \mathbb{Z}^+ \rightarrow \mathbb{R} \) such that, for any \( d \in \mathbb{Z}^+ \):

\[
\alpha(d) \delta(x, x') - \beta(d) \leq \delta_H(\phi(x), \phi(x')) \leq \alpha(d) \delta(x, x') + \beta(d) \tag{5}
\]

We typically wish the metric \( \delta_H \) on \( \mathcal{H} \) to be simple to compute. In practice, it is often taken to be the Euclidean, Hamming, or angular distance. The position-ID method preserves the L1 distance with \( \delta_H \) the Euclidean distance, \( \alpha(d) = 2d \), and \( \beta(d) \leq n^2 \mu d \). The signed random-projection method preserves the angular distance with \( \alpha(d) = O(d) \), \( \beta(d) = O(\sqrt{d}) \), and \( \delta_H \) the Hamming, angular, or Euclidean distance. To be concrete, unless otherwise stated, we will take \( \delta_H \) to be the Euclidean distance as it accords naturally with the two encoding methods we consider in detail.

5.3.1 Preservation of Cluster Structure

In general, there is no universally applicable definition of cluster structure. Indeed, numerous algorithms have been proposed in the literature to target various reasonable notions of what constitutes a “cluster” in the data. Preservation of a metric accords naturally with K-means like algorithms which compute a set of centroids \( C = \{c_i\}_{i=1}^k \), and define associated clusters as the Voronoi cells associated with each centroid. We here adopt this notion and state that cluster structure is preserved if, for any \( x \):

\[
\arg\min_{c \in C} \delta(x, c) = \arg\min_{c \in C} ||\phi(x) - \phi(c)||_2^2
\]

In other words, that the set of points bound to a particular cluster centroid does not change under the encoding. We can restate the above as requiring that, for some point \( x \) bound to a cluster centroid \( c \), it is the case that:

\[
||\phi(x) - \phi(c)||_2^2 < ||\phi(x) - \phi(c')||_2^2
\]

for any \( c' \in C \setminus \{c\} \). Using the definition of distance preservation above, we can see this property will be satisfied if:

\[
\frac{\beta(d)}{\alpha(d)} < \min_{x \in X} \min_{c, c' \in C} \frac{1}{2} \delta(x, c') - \delta(x, c)
\]

A sufficient condition for the existence of some \( d \) satisfying this property is that \( \alpha(d) \) is monotone increasing and that \( \alpha(d) \) is faster growing than \( \beta(d) \). This condition is satisfied for both the random projection and position-ID encoding methods.
5.3.2 Noise Robustness

It is also of interest to consider robustness to noise in the context of encoding Euclidean data. Suppose we have a set of points, \( \mathcal{X} \), in \( \mathbb{R}^n \), and a metric of interest \( \delta(\cdot, \cdot) \) which is preserved à la Equation 5. Given an arbitrary point \( x \in \mathcal{X} \) we consider a noise model which corrupts \( \phi(x) \) to \( \tilde{\phi}(x) = \phi(x) + \Delta \), where \( \Delta \) is some unspecified noise process. Along the lines of Section 3.2, we say \( \Delta \) is \( \rho \)-bounded if:

\[
\max_{x \in \mathcal{X}} \langle \phi(x), \Delta \rangle \leq \rho
\]

We want to ensure the encodings can distinguish between all points at a distance \( \leq \epsilon_1 \) from \( x \) and all points at a distance \( \geq \epsilon_2 \). That is:

\[
\|\tilde{\phi}(x) - \tilde{\phi}(x')\|^2_2 < \|\tilde{\phi}(x) - \tilde{\phi}(x'')\|^2_2
\]

for all \( x' \in \mathcal{X} \) such that \( \delta(x, x') \leq \epsilon_1 \) and all \( x'' \in \mathcal{X} \) such that \( \delta(x, x'') \leq \epsilon_2 \). We say such an encoding is \((\epsilon_1, \epsilon_2)\)-robust.

**Theorem 21** Let \( \delta \) be a metric on \( \mathbb{R}^n \) and suppose \( \phi \) is an embedding preserving \( \delta \) under the Euclidean distance on \( \mathcal{H} \) as described in Definition 20. Let \( \mathcal{X} \) be a set of points in \( \mathbb{R}^n \) and suppose \( \Delta \) is \( \rho \)-bounded noise. Then \( \phi \) is \((\epsilon_1, \epsilon_2)\)-robust if:

\[
\rho \leq \frac{\alpha(d)}{8}(\epsilon_2 - \epsilon_1) - \frac{\beta(d)}{4}
\]

**Proof** Fix a point \( x \). Then for all \( x' \in \mathcal{X} \) s.t. \( \delta(x, x') \leq \epsilon_1 \) and for all \( x'' \in \mathcal{X} \) s.t. \( \delta(x, x'') \geq \epsilon_2 \), we wish to be the case that:

\[
\|\phi(x) + \Delta - \phi(x')\|^2_2 < \|\phi(x) + \Delta - \phi(x'')\|^2_2
\]

\[
\|\phi(x) - \phi(x')\|^2_2 + \|\Delta\|^2_2 + 4\rho < \|\phi(x) - \phi(x'')\|^2_2 + \|\Delta\|^2_2 - 4\rho
\]

\[
8\rho < \alpha(d)(\delta(x, x'') - \delta(x, x')) - 2\beta(d)
\]

The result follows by noting that, by construction, \( \delta(x, x'') \geq \epsilon_2 \) and \( \delta(x, x') \leq \epsilon_1 \).

**Additive White Noise.** First consider the case that \( \mathcal{H} = \mathbb{R}^d \) and \( \Delta \) is additive white noise drawn from a mean-zero sub-Gaussian distribution with variance \( \sigma_\Delta^2 \). Then, as before, we can note that \( \langle \phi(x), \Delta \rangle \sim \mathcal{N}(0, \sigma_\Delta^2 \|\phi(x)\|^2_2) \). Then, with very high probability \( \rho < 4L\sigma \), where \( L = \max_{x \in \mathcal{X}} \|\phi(x)\| \). So then, we have the desired robustness property if:

\[
\sigma < \frac{\alpha(d)}{32L}(\epsilon_2 - \epsilon_1) - \frac{\beta(d)}{16L}
\]

Assuming that \( \alpha(d) \) is faster growing in \( d \) than \( L \) and \( \beta(d) \) there will exist some encoding dimension for which we can tolerate an arbitrary level of noise. In the case of the random projection encoding scheme described above \( \alpha(d) = O(d), \beta(d) = O(\sqrt{d}) \) and \( L = \sqrt{d} \) exactly. And so we can tolerate noise on the order of:

\[
\sigma \approx \sqrt{d}(\epsilon_2 - \epsilon_1)
\]
For the position-ID encoding method, \( \alpha(d) = O(d) \), \( L = O(\sqrt{nd}) \) and \( \beta(d) = O(n^2d\mu) \), and so we can tolerate noise:
\[
\sigma \approx \sqrt{\frac{d}{\sqrt{n}}((\epsilon_2 - \epsilon_1) - n^2 \mu)}
\]

**Adversarial Noise.** We now consider the case that \( H = \{ \pm 1 \} \)—as in the random-projection encoding method for instance—and \( \Delta \) is noise in which some fraction \( \omega \cdot d \) of coordinates in \( \phi(x) \) are maliciously corrupted by an adversary. Since \( \|\Delta\|_1 \leq \omega d \), we have, for any \( x \in X \):
\[
|\langle \phi(x), \Delta \rangle| \leq \|\phi(x)\|_\infty \|\Delta\|_1 \leq \omega d
\]

So then we can tolerate \( \omega \) on the order of:
\[
\rho < \frac{\alpha(d)}{8d}(\epsilon_2 - \epsilon_1) - \frac{\beta(d)}{4d}
\]

In the case of the random-projection encoding method this boils down to:
\[
\rho \approx (\epsilon_2 - \epsilon_1) - \frac{1}{\sqrt{d}},
\]

meaning the total number of coordinates that can be corrupted is \( O(d(\epsilon_2 - \epsilon_1)) \).

### 6. Learning on HD Data Representations

We now turn to the question of using HD representations in learning algorithms. Our goal is to clarify in what precise sense the HD encoding process can make learning easier. We study two ways in which this can happen: the encoding process can increase the separation between classes and/or can induce sparsity. Both of these characteristics can be exploited by neurally plausible algorithms to simplify learning. Throughout this discussion, we assume access to a set of \( N \) labelled examples \( S = \{(x_i, y_i)\}_{i=1}^N \), where \( x_i \) lies in \([0, 1]^n\) and \( y_i \in \mathcal{C} \) is a categorical variable indicating the class label. In general, we are interested in the case that training examples arrive in a streaming, or “online,” fashion, although our conclusions apply to fixed and finite data as well.

#### 6.1 Learning by Bundling

The simplest approach to learning with HD representations is to bundle together the training examples corresponding to each class into a set of exemplars—often referred to as “prototypes”—which are then used for classification (Kleyko et al., 2018; Rahimi et al., 2018; Burrello et al., 2018). More formally, as described in Section 2, we construct the prototype \( c_k \) for the \( k \)-th class as:
\[
c_k = \bigoplus_{i \text{ s.t. } y_i=k} \phi(x_i)
\]

and then assign a class label for some “query” point \( x_q \) as:
\[
\hat{y} = \text{argmax}_{k \in \mathcal{C}} \frac{\langle c_k, \phi(x) \rangle}{\|c_k\|} \quad (6)
\]
This approach bears a strong resemblance to naive Bayes and Fisher’s linear discriminant, which are both classic simple statistical procedures for classification (Bishop, 2006). Like these methods, the bundling approach is appealing due to its simplicity. However, it also shares their weaknesses in that it is less general than arbitrary linear separators, and one can easily construct simple data sets that cannot be correctly classified using this approach.

6.2 Learning Arbitrary Linear Separators

Linear separability is one of the most basic notions of simplifying structure that can aid learning. The theory of linear models is well developed and several simple, neurally plausible, algorithms for learning linear separators are known, for instance, the Perceptron and Winnow (Rosenblatt, 1958; Littlestone, 1988). Thus, if our data is linearly separable in low-dimensional space we would like it to remain so after encoding so that these methods can be applied. Preservation of distance is sufficient, under some conditions, to preserve linear separability.

**Theorem 22** Let \( X \) and \( X' \) be two disjoint, closed, and convex sets of points in \( \mathbb{R}^n \). Let \( p \) and \( q \) be the closest pair of points in either set. Suppose \( \phi \) preserves a metric \( \delta \) under the L2 distance on \( H \) in the sense of Definition 20. Then, if \( \delta \) is induced by an inner product, the function \( f(x) = \langle \phi(x), \phi(p) - \phi(q) \rangle - \frac{1}{2} (||\phi(p)||^2 - ||\phi(q)||^2) \) is positive for all \( x \in X \) and negative for all \( x' \in X' \) provided:

\[
\frac{4\beta(d)}{\alpha(d)} < \min_{x \in X} \langle x, p - q \rangle
\]

**Proof** By assumption, L2 distances in \( H \) preserve a metric \( \delta \) on \( \mathbb{R}^n \) induced by an inner product. Therefore, we may write Equation 5 as:

\[
\alpha(d) \langle x, x' \rangle - \frac{3}{2} \beta(d) \leq \langle \phi(x), \phi(x') \rangle \leq \alpha(d) \langle x, x' \rangle + \frac{3}{2} \beta(d)
\]

Consider some point \( x \in X \). Expanding and using the inner-product version of the distance preserving property given above:

\[
\langle \phi(x), \phi(p) - \phi(q) \rangle - \frac{1}{2} (||\phi(p)||^2 - ||\phi(q)||^2)
\]

\[
= \langle \phi(x), \phi(p) \rangle - \langle \phi(x), \phi(q) \rangle - \frac{1}{2} (||\phi(p)||^2 - ||\phi(q)||^2)
\]

\[
\geq \alpha(d)(\langle x, p - q \rangle - \frac{1}{2} (||p||^2 - ||q||^2)) - 4\beta(d)
\]

By a standard proof of the hyperplane separation theorem (Boyd and Vandenberghe, 2004), \( \langle x, p - q \rangle - \frac{1}{2} (||p||^2 - ||q||^2) > 0 \) for any \( x \in X \). Therefore, the expression above is positive if:

\[
\frac{4\beta(d)}{\alpha(d)} < \min_{x \in X} \langle x, p - q \rangle - \frac{1}{2} (||p||^2 - ||q||^2)
\]

Since we may assume w.l.o.g. that the data has been normalized, this can be simplified to:

\[
\frac{4\beta(d)}{\alpha(d)} < \min_{x \in X} \langle x, p - q \rangle
\]
the proof for \( x \in X' \) is analogous.

### 6.2.1 Learning Sparse Classifiers on Random Projection Encodings

The random projection encoding method can be seen to lead to representations that are **sparse** in the sense that only a subset of \( k \ll d \) coordinates are relevant for determining the class label. This setting accords naturally with the Winnow algorithm which is known to make on the order of \( k \log d \) mistakes when the target function class is a monotone disjunction (Littlestone, 1988). This can offer substantially faster convergence than the Perceptron when the margin is small. Curiously, while the Perceptron algorithm is commonly used on HD encodings, we are unaware of any work using Winnow for learning and suggest this to practitioners.

**Theorem 23** Let \( \mathcal{X}^{(+)} \) and \( \mathcal{X}^{(-)} \) be two sets of points supported on the \( n \)-dimensional unit sphere and separated by a unit-norm hyperplane \( w \) with margin \( \gamma = \min_{x \in \mathcal{X}} |\langle x, w \rangle| \). Let \( \Phi \in \mathbb{R}^{d \times n} \) be a matrix whose rows are sampled from the uniform distribution over the \( n \)-dimensional unit-sphere. Define the encoding of a point \( x \) by \( \phi(x) = g(\Phi x) \), where \( g \) is a quantizer. There with probability \( 1 - \delta \) there exists a classifier which depends on at most \( k \) coordinates provided:

\[
d \geq k \frac{\log \delta}{\log 1 - \theta}
\]

where \( \theta = O((1 - \frac{1}{k})^n) \), when \( n \) is large.

To prove the theorem we first use the following simple Lemma:

**Lemma 24** Suppose there exists \( \Phi_i \) such that \( \langle \Phi_i, w \rangle \geq 1 - \gamma^2 \). Then \( \langle \Phi_i, x \rangle \) is positive for any \( x \in \mathcal{X}^{(+)} \) and negative for any \( x \in \mathcal{X}^{(-)} \)

**Proof** Consider some \( x \in \mathcal{X}^{(+)} \). Then we have:

\[
\langle \Phi_i, x \rangle = \langle w, x \rangle + \langle \Phi_i - w, x \rangle \geq \gamma + \langle \Phi_i - w, x \rangle
\]

Using the Cauchy-Schwarz inequality and recalling \( \|x\| = 1 \), we wish to ensure:

\[
\gamma - \|\Phi_i - w\| > 0 \Rightarrow \|\Phi_i - w\| < \gamma
\]

which, by the law of cosines, is equivalent to:

\[
\langle \Phi_i, w \rangle > 1 - \frac{\gamma^2}{2}
\]

Unfortunately, the probability of sampling such a direction is on the order of \( \gamma^n \) meaning that the method is only viable when the data is very well separated in low-dimensional space which is seldom the case in practice. However, we might instead hope to sample \( k \) vectors that are weakly correlated with \( w \) and exploit their cumulative effect on \( x \). We
define a vector to be $\rho$-correlated with $w$ if $\langle \Phi_i, w \rangle \geq \rho$. We are now in a position to prove the theorem.

**Proof** We wish to find some collection of $k \ll d$ vectors such that:

$$\frac{\langle \sum_i \Phi_i, w \rangle}{\| \sum_i \Phi_i \|} > 1 - \gamma^2$$

We can upper bound the denominator as:

$$\| \sum_i \Phi_i \|^2 = \sum_{i=1}^k \| \Phi_i \|^2 + \sum_{i \neq j} \langle \Phi_i, \Phi_j \rangle \leq k \mu$$

where $\mu$ is the incoherence of $\Phi^T$. So then:

$$\frac{\langle \sum_i \Phi_i, w \rangle}{\| \sum_i \Phi_i \|} \geq \rho (k + k^2 \mu)^{-1/2}$$

And so we must have:

$$\rho > \sqrt{k^{-1} + \mu(1 - \gamma^2/2)}$$

The probability of sampling one such direction is $\theta = \frac{1}{2} I(\sin^2(\psi); \frac{n-1}{2}, \frac{1}{2})$, where $\psi = \cos^{-1} \rho$ and $I$ is the incomplete Beta-function (Li, 2011). And so to ensure we sample at least $k$ $\rho$-correlated directions with probability $1 - \delta$, we should choose:

$$d > k \frac{\log \delta}{\log 1 - \theta}$$

We can obtain a more explicit relationship with $k$ using some approximations to the above. When $\gamma$ is small, $\rho$ will be dominated by $\sqrt{k^{-1} + \mu}$. To obtain an estimate for $\mu$, fix some $\Phi_i$. Noting that we can simulate a random direction from the $n$-sphere by sampling $z \sim N(0, I_n)$ and normalizing, we have:

$$\langle \Phi_i, \Phi_j \rangle \sim \frac{\langle \Phi_i, z \rangle}{\| z \|^2} \sim N(0, 1/\| z \|^2)$$

Observing $\| z \|^2 = n(1 \pm \epsilon)$, where $\epsilon \to 0$ in $n$, we can obtain a reasonable approximation to the above as $\langle \Phi_i, \Phi_j \rangle \sim N(0, 1/n)$. By a Gaussian tail bound and the union bound we have, with high probability:

$$\mu = O \left( \sqrt{(\log d)/n} \right)$$

Then, assuming we are in the setting $n = O(k^2 \log d)$, we have $\sqrt{k^{-1} + \mu} \approx \sqrt{2/k}$. Using the approximations $\cos^{-1}(x) = \pi/2 - x$, for $x$ small and $\sin(x) = 1 - x^2/2$ for $x$ near $\pi/2$, we have: $\sin^2 \cos^{-1} \rho \approx 1 - 2/k$, and so finally we obtain:

$$\theta = O((1 - \frac{1}{k})^n)$$
This result is very appealing because it does not depend on the margins in low-dimensional space and thus saves us from the exponentially bad scaling in the case that \( k = 1 \). Furthermore, by considering multiple bits in \( \phi(x) \) it is additionally possible to discriminate classes that are not linearly separable in low-dimensional space. In summary, the random projection method in tandem with the Winnow algorithm is particularly well suited to the HD setting. Intuitively, the HD representation allows us to transform a dense low-dimensional problem where learning may have been difficult, for instance because the data had a small margin, into a high-dimensional representation where we can exploit sparsity to simplify learning.

7. Conclusion

To conclude, we lay out several research directions related to HD computing we believe it would be of particular interest to further explore. There are several interesting open problems related to encoding. Our analysis established preservation of only the most basic forms of structure in data. Can encoding procedures satisfying the desiderata of HD computing be designed that capture other more exotic forms of structure? The quantized random Fourier feature construction discussed in Section 5 presents one such option, but is only applicable to structure that can be captured using a shift-invariant kernel on a Euclidean space. For instance, can we devise encoding methods that exploit low-dimensional manifold structure in the data?

Several recent works have claimed, based on empirical evidence, that HD computing evinces one-shot learning (Thrun, 1996; Lake et al., 2011) in which a single labeled example is needed to learn a generalizable classifier (Burrello et al., 2018; Imani et al., 2017a; Rahimi et al., 2018). However, this work has focused on settings in which specialized hand-crafted features could be extracted, and it is not clear to us that existing encoding procedures would lead to one-shot classifiers absent such outside information. We would be interested to explore whether the HD representation makes one-shot learning easier in any broader sense. We expect this will necessitate the use of more sophisticated encoding procedures that can learn salient properties of a given domain. For this latter point we see dictionary learning as a promising avenue for developing adaptive encoding procedures. Dictionary learning is a well studied problem and can be solved using online and neurally plausible methods (Arora et al., 2015; Mairal et al., 2010) and would thus seem to be a promising avenue to address the limitations of existing encoding procedures without sacrificing the simplicity and neural plausibility of existing HD based methods.

References

Sanjeev Arora, Rong Ge, Tengyu Ma, and Ankur Moitra. Simple, efficient, and neural algorithms for sparse coding. Proceedings of The 28th Conference on Learning Theory, COLT 2015, 40:113–149, 2015.

Andrew R Barron. Universal approximation bounds for superpositions of a sigmoidal function. IEEE Transactions on Information theory, 39(3):930–945, 1993.
A.J. Bell and T.J. Sejnowski. An information-maximization approach to blind separation and blind deconvolution. *Neural Computation*, 7(6):1129–1159, 1995.

Christopher M Bishop. *Pattern recognition and machine learning*. Springer, 2006.

Burton H Bloom. Space/time trade-offs in hash coding with allowable errors. *Communications of the ACM*, 13(7):422–426, 1970.

Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge University Press, 2004.

Andrei Broder and Michael Mitzenmacher. Network applications of bloom filters: A survey. *Internet mathematics*, 1(4):485–509, 2004.

Alessio Burrello, Kaspar Schindler, Luca Benini, and Abbas Rahimi. One-shot learning for ieeeg seizure detection using end-to-end binary operations: Local binary patterns with hyperdimensional computing. In *2018 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, pages 1–4. IEEE, 2018.

Sophie JC Caron, Vanessa Ruta, LF Abbott, and Richard Axel. Random convergence of olfactory inputs in the drosophila mushroom body. *Nature*, 497(7447):113–117, 2013.

Maurice J Chacron, Andre Longtin, and Leonard Maler. Efficient computation via sparse coding in electrosensory neural networks. *Current Opinion in Neurobiology*, 21(5):752–760, 2011.

Krzysztof M Choromanski, Mark Rowland, and Adrian Weller. The unreasonable effectiveness of structured random orthogonal embeddings. In *Advances in Neural Information Processing Systems*, pages 219–228, 2017.

G. Cybenko. Approximations by superpositions of sigmoidal functions. *Mathematics of Control, Signals, and Systems*, 2(4):303–314, 1989.

Sanjoy Dasgupta, Timothy C Sheehan, Charles F Stevens, and Saket Navlakha. A neural data structure for novelty detection. *Proceedings of the National Academy of Sciences*, 115(51):13093–13098, 2018.

David L Donoho, Michael Elad, and Vladimir N Temlyakov. Stable recovery of sparse overcomplete representations in the presence of noise. *IEEE Transactions on information theory*, 52(1):6–18, 2005.

E Paxon Frady, Denis Kleyko, and Friedrich T Sommer. A theory of sequence indexing and working memory in recurrent neural networks. *Neural computation*, 30(6):1449–1513, 2018.

Kunihiro Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics*, 36(4):193–202, 1980.

Stephen I Gallant and T Wendy Okaywe. Representing objects, relations, and sequences. *Neural Computation*, 25(8):2038–2078, 2013.
Saransh Gupta, Mohsen Imani, and Tajana Rosing. Felix: Fast and energy-efficient logic in memory. In *2018 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 1–7. IEEE, 2018.

Thang Huynh and Rayan Saab. Fast binary embeddings and quantized compressed sensing with structured matrices. *Communications on Pure and Applied Mathematics*, 73(1):110–149, 2020.

Mohsen Imani, Deqian Kong, Abbas Rahimi, and Tajana Rosing. Voicehd: Hyperdimensional computing for efficient speech recognition. In *2017 IEEE International Conference on Rebooting Computing (ICRC)*, pages 1–8. IEEE, 2017a.

Mohsen Imani, Abbas Rahimi, Deqian Kong, Tajana Rosing, and Jan M Rabaey. Exploring hyperdimensional associative memory. In *2017 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, pages 445–456. IEEE, 2017b.

Mohsen Imani, Tarek Nassar, Abbas Rahimi, and Tajana Rosing. Hdna: Energy-efficient dna sequencing using hyperdimensional computing. In *2018 IEEE EMBS International Conference on Biomedical & Health Informatics (BHI)*, pages 271–274. IEEE, 2018.

Mohsen Imani, John Messerly, Fan Wu, Wang Pi, and Tajana Rosing. A binary learning framework for hyperdimensional computing. In *2019 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 126–131. IEEE, 2019a.

Mohsen Imani, Justin Morris, Samuel Bosch, Helen Shu, Giovanni De Micheli, and Tajana Rosing. Adapthd: Adaptive efficient training for brain-inspired hyperdimensional computing. In *2019 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, pages 1–4. IEEE, 2019b.

Mohsen Imani, Justin Morris, John Messerly, Helen Shu, Yaobang Deng, and Tajana Rosing. BRIC: Locality-based encoding for energy-efficient brain-inspired hyperdimensional computing. In *Proceedings of the 56th Annual Design Automation Conference 2019*, pages 1–6. 2019c.

Mohsen Imani, Sahand Salamat, Saransh Gupta, Jiani Huang, and Tajana Rosing. Fach: Fpga-based acceleration of hyperdimensional computing by reducing computational complexity. In *Proceedings of the 24th Asia and South Pacific Design Automation Conference*, pages 493–498. ACM, 2019d.

Laurent Jacques, Jason N Laska, Petros T Boufounos, and Richard G Baraniuk. Robust 1-bit compressive sensing via binary stable embeddings of sparse vectors. *IEEE Transactions on Information Theory*, 59(4):2082–2102, 2013.

Pentti Kanerva. *Sparse distributed memory*. MIT press, 1988.

Pentti Kanerva. The spatter code for encoding concepts at many levels. In *International Conference on Artificial Neural Networks*, pages 226–229. Springer, 1994.

Pentti Kanerva. A family of binary spatter codes. In *ICANN’95, Proceedings International Conference on Artificial Neural Networks*, volume 1, pages 517–522, 1995.
Pentti Kanerva. Hyperdimensional computing: An introduction to computing in distributed representation with high-dimensional random vectors. *Cognitive computation*, 1(2):139–159, 2009.

Yeseong Kim, Mohsen Imani, Niema Moshiri, and Tajana Rosing. Geniehd: Efficient dna pattern matching accelerator using hyperdimensional computing. In *2020 Design, Automation & Test in Europe Conference & Exhibition (DATE - To Appear)*. IEEE, 2020.

Yeseong Kim et al. Efficient human activity recognition using hyperdimensional computing. In *IoT*, page 38. ACM, 2018.

Denis Kleyko, Abbas Rahimi, Dmitri A Rachkovskij, Evgeny Osipov, and Jan M Rabaey. Classification and recall with binary hyperdimensional computing: Tradeoffs in choice of density and mapping characteristics. *IEEE Transactions on Neural Networks and Learning Systems*, 29(12):5880–5898, 2018.

Brenden Lake, Ruslan Salakhutdinov, Jason Gross, and Joshua Tenenbaum. One shot learning of simple visual concepts. In *Proceedings of the annual meeting of the cognitive science society*, volume 33, 2011.

Simon D Levy and Ross Gayler. Vector symbolic architectures: A new building material for artificial general intelligence. In *Proceedings of the 2008 conference on artificial general intelligence 2008: Proceedings of the first AGI conference*, pages 414–418. IOS Press, 2008.

Shengqiao Li. Concise formulas for the area and volume of a hyperspherical cap. *Asian Journal of Mathematics and Statistics*, 4(1):66–70, 2011.

Nick Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine learning*, 2(4):285–318, 1988.

Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research*, 11(1), 2010.

Schmuck Manuel, Luca Benini, and Abbas Rahimi. Hardware optimizations of dense binary hyperdimensional computing: Rematerialization of hypervectors, binarized bundling, and combinational associative memory. *ACM Journal on Emerging Technologies in Computing Systems*, 2019.

Nicolas Y Massé, Glenn C Turner, and Gregory SXE Jefferis. Olfactory information processing in drosophila. *Current Biology*, 19(16):R700–R713, 2009.

B.A. Olshausen and D.J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381:607–609, 1996.

Bruno A Olshausen and David J Field. Sparse coding of sensory inputs. *Current Opinion in Neurobiology*, 14(4):481–487, 2004.

Anna Pagh, Rasmus Pagh, and S Srinivasa Rao. An optimal bloom filter replacement. pages 823–829, 2005.
Yaniv Plan and Roman Vershynin. Dimension reduction by random hyperplane tessellations. *Discrete & Computational Geometry*, 51(2):438–461, 2014.

Tony A Plate. Holographic reduced representations. *IEEE Transactions on Neural networks*, 6(3):623–641, 1995.

Dmitri A. Rachkovskij. Representation and processing of structures with binary sparse distributed codes. *IEEE Transactions on Knowledge and Data Engineering*, 13(2):261–276, 2001.

Dmitriy A Rachkovskiy, Sergey V Slipchenko, Ernst M Kussul, and Tatyana N Baidyk. Sparse binary distributed encoding of scalars. *Journal of Automation and Information Sciences*, 37(6), 2005a.

Dmitriy A Rachkovskiy, Sergey V Slipchenko, Ivan S Misuno, Ernst M Kussul, and Tatyana N Baidyk. Sparse binary distributed encoding of numeric vectors. *Journal of Automation and Information Sciences*, 37(11), 2005b.

Maxim Raginsky and Svetlana Lazebnik. Locality-sensitive binary codes from shift-invariant kernels. In *Advances in neural information processing systems*, pages 1509–1517, 2009.

Abbas Rahimi, Sohum Datta, Denis Kleyko, Edward Paxon Frady, Bruno Olshausen, Pentti Kanerva, and Jan M Rabaey. High-dimensional computing as a nanoscalable paradigm. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 64(9):2508–2521, 2017a.

Abbas Rahimi, Artiom Tchouprina, Pentti Kanerva, José del R Millán, and Jan M Rabaey. Hyperdimensional computing for blind and one-shot classification of eeg error-related potentials. *Mobile Networks and Applications*, pages 1–12, 2017b.

Abbas Rahimi, Pentti Kanerva, Luca Benini, and Jan M Rabaey. Efficient biosignal processing using hyperdimensional computing: Network templates for combined learning and classification of eeg signals. *Proceedings of the IEEE*, 107(1):123–143, 2018.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in Neural Information Processing systems*, pages 1177–1184, 2008.

F. Rosenblatt. The Perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408, 1958.

Walter Rudin. *Fourier analysis on groups*, volume 121967. Wiley Online Library, 1962.

D.E. Rumelhart, J.L. McClelland, and the PDP Research Group. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Volume 1: Foundations*. MIT Press, 1986.

Sahand Salamat, Mohsen Imani, Behnam Khaleghi, and Tajana Rosing. F5-hd: Fast flexible fpga-based framework for refreshing hyperdimensional computing. In *Proceedings of the 2019 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, pages 53–62, 2019.
Dan D Stettler and Richard Axel. Representations of odor in the piriform cortex. *Neuron*, 63(6):854–864, 2009.

Sebastian Thrun. Is learning the n-th thing any easier than learning the first? In *Advances in neural information processing systems*, pages 640–646, 1996.

Glenn C Turner, Maxim Bazhenov, and Gilles Laurent. Olfactory representations by drosophila mushroom body neurons. *Journal of Neurophysiology*, 99(2):734–746, 2008.

Martin J Wainwright. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge University Press, 2019.

Rachel I Wilson. Early olfactory processing in drosophila: mechanisms and principles. *Annual Review of Neuroscience*, 36:217–241, 2013.