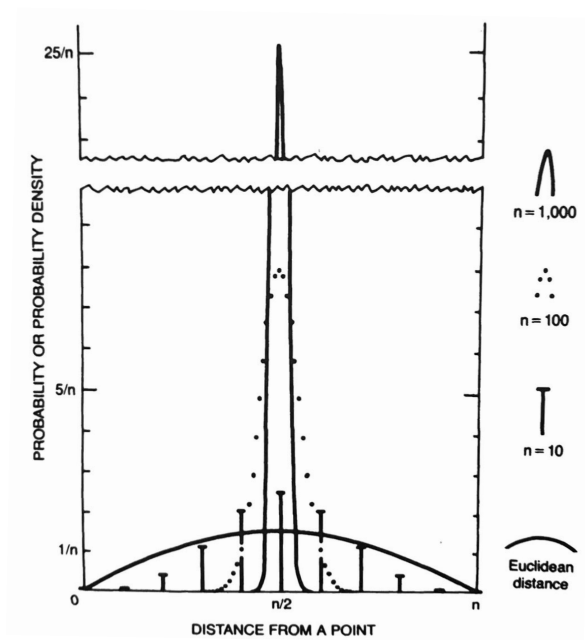
Neuroscience 299: Computing with High-Dimensional Vectors
Assignment 1: Introduction
Due September 8, 1pm

1. *Why high dimensions?* What is the purpose of using high-dimensional vectors? Why might it make sense to select vectors randomly? Explain your answer in about a paragraph.

2. *Plotting pseudo-orthogonality*. For binary vectors $\{0,1\}^N$, let $d(\mathbf{x},\mathbf{y})$, the normalized Hamming between two vectors $\mathbf{x}$, $\mathbf{y}$, be defined as:

$$\mathrm{dist}_{\mathrm{Ham}}(\mathbf{x},\mathbf{y}) = \frac{1}{N}|\mathbf{x}-\mathbf{y}| = \frac{1}{N}(|x_1 - y_1| + \cdots + |x_N - y_N|)$$

The distribution of $\mathrm{dist}_{\mathrm{Ham}}(\mathbf{x},\mathbf{y})$ for i.i.d. random binary vectors will look something like this:



(from Kanerva, *Sparse Distributed Memory,* p. 20)

Plot the distribution of cosine similarity for i.i.d. random bipolar vectors $\{-1,+1\}^N$ in each of the following cases: N = 10^1, 10^2, 10^3, and 10^4. You can obtain the plot by first randomly drawing several thousand pairs of **x** and **y**, calculating their cosine similarity, and fitting the normal distribution to the histogram of the observed cosine similarities. Do you see the results, which resemble the figure above?

3. *Implementing basic HDC/VSA operations.*
(a) For bipolar {-1,+1} vectors, recreate the following three functions corresponding to the basic operations of *HDC/VSA*:

```
def bind(vector1,vector2):
     #your code here
     return bound_vector

def bundle(vector1,vector2):
     #your code here
     return bundled_vector

def permutation(vector1):
     #your code here
     return permuted_vector
```

Note: For now, assume that the bundling operation is simply component-wise addition without any normalization.

(b) Demonstrate that the bundling operation is similarity-preserving, whereas the binding and permutation operations are not.

(c) Assume that your codebook includes 5 random bipolar vectors {**a**,**b**,**c**,**d**,**e**}.
Form the compositional vector **z** of the following form **z**=**a**+**a**+**a**+**b**+**b**+**c**.
Plot the histogram of cosine similarities between **z** and {**a**,**b**,**c**,**d**,**e**} that is averaged over several random initializations of the codebook. What regalities do you observe?