Almost orthogonal vectors in high dimensionalities

Inar Timiryasov

2024-01-08

How many almost orthogonal vectors can we have in high dimensions?

TODO

- Intro about vectors in high dimensions and how can we store information
- Use mamba and vect2text as motivation
- write the usual proof of orthogonality based on normal distribution

In this post we will discuss the following question: how many approximately orthogonal vectors can we have in high dimensions?

Why even ask this question?

Suppose we have vectors in $\mathbb{R}^d$. These vectors could be, e.g. the activations in the residual stream of a transformer, or a state space in the state space models like mamba. By definition, there will be d orthogonal vectors. By orthogonal we mean that $ v_i v_j = 0$ for $i \neq j$. But what if we require that $v_i v_j $ for $i \neq j$? How many such vectors can we have? Quite surprisingly, there are many such vectors. In fact, there are exponentially many such vectors.

One cue to suspect that it might be true is the so-called concentration of measure phenomenon, which says that most of the mass of a sphere in $\mathbb{R}^d$ is close to the equator. Imagine a sphere and one vector v_1 pointing to the North. Vectors that are almost orthogonal to v_1 will be close to the equator. And since most of the mass of the sphere is close to the equator, there could be room for many such vectors. But how many exactly?

Spherical caps

A spherical cap is a portion of a sphere that is cut off by a plane.

```python
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

def plot_sphere_with_uniform_color(radius=1, z_max=np.sqrt(2)/2):
    """
    Plots a sphere with a uniformly colored spherical cap using Matplotlib.
    Displays the Cartesian coordinate axes and the opening angle theta.

    Parameters:
    radius (float): Radius of the sphere.
    z_max (float): Maximum z-coordinate for the cap, scaled between 0 and 1.

    Returns:
    Matplotlib figure
    """

    # Define the azimuthal angle phi and theta
    phi = np.linspace(0, 2 * np.pi, 100)
    theta = np.linspace(0, np.pi, 100)
    phi, theta = np.meshgrid(phi, theta)

    # Parametric equations for the sphere
    x = radius * np.sin(theta) * np.cos(phi)
    y = radius * np.sin(theta) * np.sin(phi)
    z = radius * np.cos(theta)

    # Create a 3D plot
    fig = plt.figure(figsize=(8, 6))
    ax = fig.add_subplot(111, projection='3d')

    # Plot the sphere
    ax.plot_surface(x, y, z, color='blue', alpha=0.2)

    # Calculate the angle for the cap based on z_max
    theta_max = np.arccos(z_max)
    cap_theta = np.linspace(0, theta_max, 50)
    cap_phi, cap_theta = np.meshgrid(phi[0], cap_theta)
```

```python
    # Parametric equations for the cap
    cap_x = radius * np.sin(cap_theta) * np.cos(cap_phi)
    cap_y = radius * np.sin(cap_theta) * np.sin(cap_phi)
    cap_z = radius * np.cos(cap_theta)
    ax.plot_surface(cap_x, cap_y, cap_z, color='orange', alpha=1.0)

    # Add Cartesian coordinate axes with labels
    axis_length = radius * 1.4
    ax.quiver(0, 0, 0, axis_length, 0, 0, color='black', arrow_length_ratio=0.05)
    ax.quiver(0, 0, 0, 0, axis_length, 0, color='black', arrow_length_ratio=0.05)
    ax.quiver(0, 0, 0, 0, 0, axis_length, color='black', arrow_length_ratio=0.05)
    ax.text(axis_length*1.15, 0, 0, "X", color='black')
    ax.text(0, axis_length*1.1, 0, "Y", color='black')
    ax.text(0, 0, axis_length*1.1, "Z", color='black')

    # Display the opening angle theta
    edge_x = radius * np.sin(theta_max)
    edge_z = radius * np.cos(theta_max)
    ax.plot([0, edge_x], [0, 0], [0, edge_z], color='orange', linestyle='dashed')
    ax.text(edge_x / 2, 0, edge_z / 2+0.05, r'$\theta$', fontsize=16, color='orange')

    # Setting the aspect ratio
    ax.set_box_aspect([1,1,1])  # Aspect ratio is 1:1:1
    ax.view_init(elev=20, azim=40)

    ax.set_xticks([])
    ax.set_yticks([])
    ax.set_zticks([])

    return fig

# Plot the sphere with the cap
fig = plot_sphere_with_uniform_color()
plt.show()
```

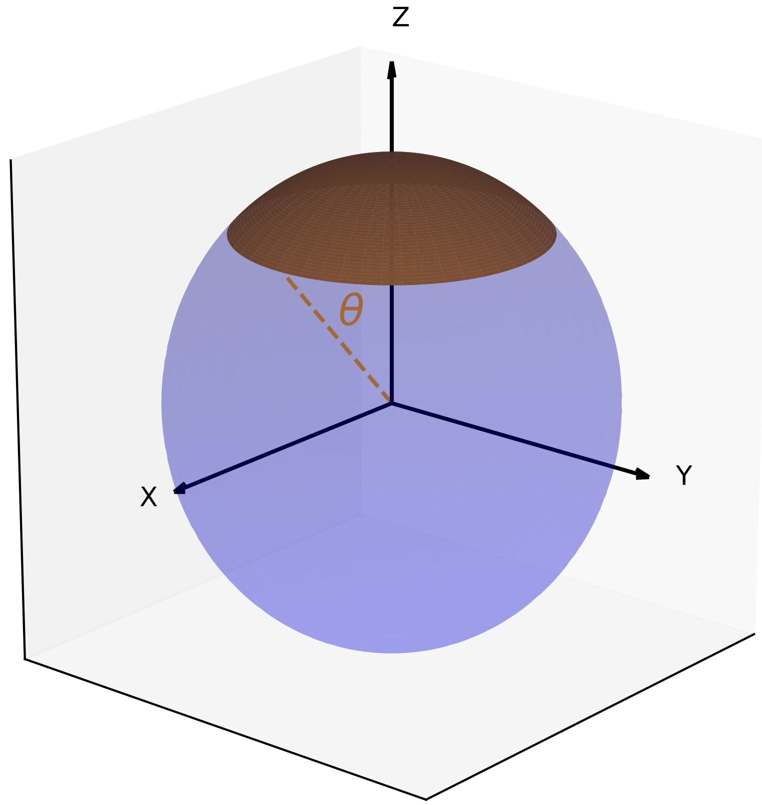

Figure 1: A sperical cap.

An area of a spherical cap in 3D is given by

$$A = 2\pi r^2(1 - \cos\theta)$$

where r is the radius of the sphere and θ is the polar angle.

If we have two vectors with their own spherical caps with opening angles

We can now imagine that almost orthogonal vectors all define their own spherical caps with $\theta = \pi/4 - \epsilon/2$. How many such caps can we place on a sphere in d dimensions?

A very simple estimate could be obtained as follows. Lets compute the area of a spherical cap with the opening angle α. This can be done relatively straightforwardly by introducing spherical coordinates in D dimensions (see wiki). The ratio of the area of a spherical cap to the area of the whole sphere.

$$r_D(\alpha) = \frac{\int_0^\alpha \sin^{D-2}(\theta) d\theta}{2\int_0^{\pi/2} \sin^{D-2}(\theta) d\theta}.$$

But this is not the whole story. High dimensions are weird, the area of "empty spaces" between the caps also grows quickly (a nice vide by 3Blue1Brown on a related topic here). So we need to take this into account.

Spherical codes

It turns out that this precise question has been asked by Shannon himself in 1959. He considered code words $w_m = (s_1, s_2, \dots, s_n)$ where s_i are real numbers and the power of code words is the same so s_i lie on a sphere. He asked how many such code words can we have given a certain error tolerance. In other words, he asked how many points can we place on a sphere such that $w_m \cdot w_k \le \epsilon$ for $m \ne k$. Shannon's paper "Probability of Error for Optimal Codes in a Gaussian Channel" is a feast of analysis and geometrical reasoning in high dimensions, but, as he puts it himself, "It might be said that the algebra involved is in several places unusually tedious". I don't show the results here since it requires introducing literally one page of definitions.

Shannon's bound was further improved by Kabatiansky and Levenshtein in 1978 paper. Their paper is also super nice to read (especially if one knows Russian) and the main result can be state in a rather accessible form.

The number of vectors with the angle between them exceeding θ is bounded by, for large n,

$$M(n, \theta) \le 2^{n\,C(\theta)},$$

where

$$C(\theta) = \frac{1+\sin\theta}{2\sin\theta} \log \frac{1+\sin\theta}{2\sin\theta} - \frac{1-\sin\theta}{2\sin\theta} \log \frac{1-\sin\theta}{2\sin\theta}.$$

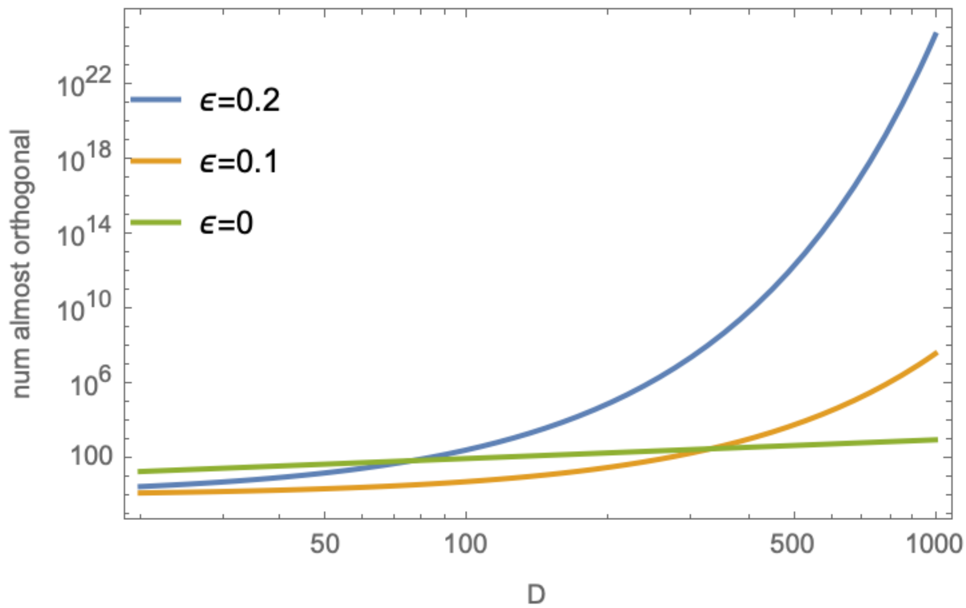

Figure 2: orthogonal-vectors

References

- Terrence Tao presents a rather straightforward derivation of a slightly weaker bound where $\epsilon = 1/\sqrt{n}$ https://terrytao.wordpress.com Perhaps he was interested in compressed sensing at that time.
- [Kabatiansky, Levenshtein](https://www.mathnet.ru/links/9bcb05981fd711e6ffcc3a4fe45bf559/ppi1518.pd
- [Blogpost by Le Scao on Johnson-Lindenstrauss lemma](https://tevenlescao.github.io/blog/fastpages/jupy Lemma-+-Linformer.html
- More than approximately orthonormal vectors in also here
- Why does the surface area of the hypersphere go to zero as the number of dimensions goes to infinity?
- Spherical code mathworld.wolfram.com; https://en.wikipedia.org/wiki/Spherical_code; https://en.wikipedia.org/wiki/Kissing_number a lot of examples
- SPHERE PACKING BOUNDS VIA SPHERICAL CODES by HENRY COHN AND YUFEI ZHAO paper link