Unit Dot Product Similarity

- Have you ever questioned the reasonableness of cosine similarity only considering the angle between two vectors?
- Can the dot product be used to calculate similarity? What problems exist if the dot product is used as the criterion for judging the similarity between two vectors?

In this paper (https://link.springer.com/article/10.1007/s10791-025-09516-2), we discuss the shortcomings of cosine similarity and dot product similarity and propose a novel similarity measure – Unit Dot Product Similarity (UDPS). This new similarity measure perfectly addresses the issues of cosine similarity and dot product similarity while possessing good mathematical properties.

Due to graduation requirements and other reasons, the paper primarily focuses on speech recognition and does not provide a separate explanation for Unit Dot Product Similarity. Furthermore, the first author and corresponding author positions of this paper were occupied by supervisors. Therefore, I am providing a renewed explanation of Unit Dot Product Similarity here. If there are any errors, please feel free to contact me (2447424163@qq.com) for discussion.

Cosine Similarity and Dot Product Similarity

First, let's review cosine similarity and dot product similarity:

$$S_{\text{cosine}}(\mathbf{a},\mathbf{b}) = \cos(\theta) = \frac{\mathbf{a}\cdot\mathbf{b}}{||\mathbf{a}||\,||\mathbf{b}||}$$

$$S_{\text{dot}}(\mathbf{a},\mathbf{b}) = \mathbf{a}\cdot\mathbf{b} = ||\mathbf{a}||\,||\mathbf{b}||\cos(\theta)$$

Here, $\mathbf{a}$ and $\mathbf{b}$ are two N-dimensional vectors not both equal to the zero vector, $\mathbf{a}\in\mathbb{R}^N$, $\mathbf{b}\in\mathbb{R}^N$, θ is the angle between the two N-dimensional vectors $\mathbf{a}$ and $\mathbf{b}$, $\cos$ is the cosine function, $||\cdot||$ is the vector norm, also known as the vector magnitude, typically the L2 norm. S_{cosine} is the cosine similarity, and S_{dot} is the dot product similarity.

Shortcomings

Cosine similarity only considers the angle between two vectors. This means that regardless of whether the vectors are of the same length, as long as their directions are the same, their similarity is the maximum value of 1. For example, the similarity between vector $(1,2)$ and $(1,2)$ is 1, and the similarity between vector $(1,2)$ and $(2,4)$ is also 1. Although $(1,2)$ and $(2,4)$ are different, their similarity is still the maximum value of 1.

The value range of dot product similarity is unbounded, and when the angle remains constant, the larger the vector magnitudes, the larger the dot product value. The similarity between vector $(1,2)$ and $(1,2)$ is 5, and the similarity between vector $(2,4)$ and $(2,4)$ is 20. This means that if the dot product is used as the similarity measure, the similarity between $(2,4)$ and $(2,4)$ is greater than the similarity between $(1,2)$ and $(1,2)$. However, $(1,2)$ and $(1,2)$ are identical, and $(2,4)$ and $(2,4)$ are also identical. From a common-sense perspective, the similarity between $(2,4)$ and $(2,4)$ is not greater than the similarity between $(1,2)$ and $(1,2)$.

What causes the aforementioned shortcomings of dot product similarity?

Dot product similarity is not normalized. Therefore, when two vectors are scaled proportionally, the result is also scaled. So, can we try to solve the aforementioned shortcomings of dot product similarity through normalization?

Solution

In fact, cosine similarity is a normalized form of dot product similarity. It is equivalent to first converting each vector to a unit vector and then taking their dot product:

$$S_{\text{cosine}}(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{a}}{\|\mathbf{a}\|} \cdot \frac{\mathbf{b}}{\|\mathbf{b}\|}$$

However, the problem with this approach is that it ignores the ratio of the magnitudes of the two vectors. As long as the directions of the two vectors are the same, the similarity between two different vectors and the similarity between two identical vectors are both the maximum value of 1. From an intuitive perspective, the similarity between two different vectors should be less than the similarity between two identical vectors. For example, the similarity between $(1, 2)$ and $(2, 4)$ should intuitively be less than the similarity between $(1, 2)$ and $(1, 2)$.

If we can maintain the ratio of the vector magnitudes while normalizing, can we solve the problem of dot product similarity while avoiding the shortcomings of cosine similarity?

A simple method to normalize while preserving the ratio of vector magnitudes is to normalize the sum of the magnitudes of the two vectors to 1:

$$S(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{a}}{\|\mathbf{a}\| + \|\mathbf{b}\|} \cdot \frac{\mathbf{b}}{\|\mathbf{a}\| + \|\mathbf{b}\|}$$

Properties

Vector $\mathbf{a}$ is normalized to $\frac{\mathbf{a}}{\|\mathbf{a}\| + \|\mathbf{b}\|}$, and vector $\mathbf{b}$ is normalized to $\frac{\mathbf{b}}{\|\mathbf{a}\| + \|\mathbf{b}\|}$. After normalization, the sum of the magnitudes of the two vectors is 1:

$$\left\| \frac{\mathbf{a}}{\|\mathbf{a}\| + \|\mathbf{b}\|} \right\| + \left\| \frac{\mathbf{b}}{\|\mathbf{a}\| + \|\mathbf{b}\|} \right\| = 1$$

When $\mathbf{a}$ and $\mathbf{b}$ are scaled proportionally, the magnitudes of the normalized vectors remain unchanged:

$$\frac{k\mathbf{a}}{\|k\mathbf{a}\| + \|k\mathbf{b}\|} = \frac{\mathbf{a}}{\|\mathbf{a}\| + \|\mathbf{b}\|}$$

The ratio of the magnitudes of the normalized vectors is the same as the ratio of the magnitudes of the original $\mathbf{a}$ and $\mathbf{b}$, because during normalization, $\mathbf{a}$ and $\mathbf{b}$ are scaled by the same factor ($\frac{1}{\|\mathbf{a}\| + \|\mathbf{b}\|}$).

Can $S(\mathbf{a}, \mathbf{b})$ solve the shortcomings of cosine similarity and dot product similarity, and can it serve as a better method for calculating similarity?

Let the normalized $\mathbf{a}$ and $\mathbf{b}$ be denoted as $\mathbf{a}^{'}$ and $\mathbf{b}^{'}$ respectively:

$$\mathbf{a}^{'} = \frac{\mathbf{a}}{\|\mathbf{a}\| + \|\mathbf{b}\|}$$

$$\mathbf{b}^{'} = \frac{\mathbf{b}}{\|\mathbf{a}\| + \|\mathbf{b}\|}$$

Since $\|\mathbf{a}^{'}\| + \|\mathbf{b}^{'}\| = 1$, therefore:

$$S(\mathbf{a},\mathbf{b}) = \mathbf{a}^{'} \cdot \mathbf{b}^{'} = \|\mathbf{a}^{'}\| \|\mathbf{b}^{'}\| \cos(\theta) = \|\mathbf{a}^{'}\|(1 - \|\mathbf{a}^{'}\|)\cos(\theta)$$

Where $\mathbf{a}$ and $\mathbf{b}$ are not both zero vectors, θ is the angle between $\mathbf{a}$ and $\mathbf{b}$, and also the angle between $\mathbf{a}^{'}$ and $\mathbf{b}^{'}$. Since $0 \le \|\mathbf{a}^{'}\| \le 1$ and $0 \le \theta \le \pi$, it can be proven that $-\frac{1}{4} \le S(\mathbf{a},\mathbf{b}) \le \frac{1}{4}$. $S(\mathbf{a},\mathbf{b}) = \frac{1}{4}$ if and only if $\mathbf{a}$ and $\mathbf{b}$ are identical. $S(\mathbf{a},\mathbf{b}) = -\frac{1}{4}$ if and only if $\mathbf{a}$ and $\mathbf{b}$ have the same magnitude but opposite directions. Please refer to the paper for the specific proof.

Since

$$\|\mathbf{a}^{'}\| = \begin{cases} \frac{1}{1+\frac{\|\mathbf{b}\|}{\|\mathbf{a}\|}}, & \|\mathbf{a}\| > 0 \\ 0, & \|\mathbf{a}\| = 0 \end{cases}$$

Therefore, the magnitude of $\mathbf{a}^{'}$ depends only on the ratio of the magnitudes of $\mathbf{a}$ and $\mathbf{b}$. When $\mathbf{a}$ and $\mathbf{b}$ have the same magnitude, $\|\mathbf{a}^{'}\| = \frac{1}{2}$. When the magnitude of $\mathbf{a}$ is less than that of $\mathbf{b}$, $0 \le \|\mathbf{a}^{'}\| < \frac{1}{2}$ and $\frac{\|\mathbf{a}\|}{\|\mathbf{b}\|} < 1$. When the magnitude of $\mathbf{a}$ is greater than that of $\mathbf{b}$, $\frac{1}{2} < \|\mathbf{a}^{'}\| \le 1$ and $\frac{\|\mathbf{a}\|}{\|\mathbf{b}\|} > 1$.

When the angle between $\mathbf{a}$ and $\mathbf{b}$ remains unchanged, proportionally scaling $\mathbf{a}$ and $\mathbf{b}$ does not affect the ratio of their magnitudes, so $\|\mathbf{a}^{'}\|$ is unaffected. Since $\cos(\theta)$ remains unchanged, $S(\mathbf{a},\mathbf{b})$ also remains unchanged. However, if the angle remains unchanged while $\frac{\|\mathbf{a}\|}{\|\mathbf{b}\|}$ is adjusted, $\|\mathbf{a}^{'}\|$ will change accordingly, $S(\mathbf{a},\mathbf{b})$ will usually change as well. Therefore, when the angle is unchanged, $S(\mathbf{a},\mathbf{b})$ neither suffers from the issue of dot product similarity where the result scales with proportional scaling of the vectors, nor does it behave like cosine similarity where the similarity remains unchanged regardless of the magnitudes as long as the angle is unchanged.

Furthermore, when the angle between $\mathbf{a}$ and $\mathbf{b}$ is unchanged, since $\|\mathbf{a}^{'}\|(1 - \|\mathbf{a}^{'}\|)$ first increases monotonically and then decreases monotonically, reaching its maximum if and only if $\|\mathbf{a}^{'}\| = \|\mathbf{b}^{'}\| = \frac{1}{2}$, therefore the closer $\frac{\|\mathbf{a}\|}{\|\mathbf{b}\|}$ is to 1, the larger $S(\mathbf{a},\mathbf{b})$ is. This aligns with the intuitive feeling of vector similarity – the closer the magnitudes of two vectors are, the higher their similarity.

When the magnitudes of the two vectors remain unchanged, the larger the angle, the smaller $\cos(\theta)$ becomes. Since $\|\mathbf{a}^{'}\|(1 - \|\mathbf{a}^{'}\|)$ remains unchanged, $S(\mathbf{a},\mathbf{b})$ also becomes smaller. This characteristic is consistent with both dot product similarity and cosine similarity. It also aligns with the intuitive feeling of vector similarity – the larger the angle between two vectors, the lower their similarity.

Therefore, $S(\mathbf{a},\mathbf{b})$ retains the desirable characteristic that similarity decreases as the angle increases, shared by both dot product and cosine similarity, while avoiding their shortcomings.

To make the value range of the similarity the same as many classic similarity measures, Unit Dot Product Similarity is defined as:

Final Definition

$$S_{\text{udot}}(\mathbf{a},\mathbf{b}) = \begin{cases} 4 \cdot S(\mathbf{a},\mathbf{b}) = \frac{4 \cdot \mathbf{a} \cdot \mathbf{b}}{(\|\mathbf{a}\| + \|\mathbf{b}\|)^2}, & \|\mathbf{a}\| + \|\mathbf{b}\| > 0 \\ 0, & \|\mathbf{a}\| = \|\mathbf{b}\| = 0 \end{cases}$$

The value range of $S_{\text{udot}}(\mathbf{a},\mathbf{b})$ is $[-1, 1]$, and its properties are similar to $S(\mathbf{a},\mathbf{b})$, which will not be repeated here.

About Unit Dot Product Similarity and Self-Attention

Can the dot product in self-attention be replaced with Unit Dot Product? How effective is it?

For each attention head, the classic self-attention is calculated as follows:

$$\mathbf{A} = \mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)$$

Where $\mathbf{A} \in \mathbb{R}^{N \times N}$, $Q \in \mathbb{R}^{N \times k}$, $K \in \mathbb{R}^{N \times k}$. $\mathbf{A}_{i,j} = \frac{q_i k_j^T}{\sqrt{d_k}}$, q_i and k_j are the i-th column of Q and the j-th column of K, respectively.

After replacing the scaled dot product of q_i and k_j with Unit Dot Product, $\frac{q_i k_j^T}{\sqrt{d_k}}$ becomes $4 \cdot \frac{q_i k_j^T}{(\|q_i\|+\|k_j\|)^2}$, and $\mathbf{A}$ correspondingly becomes:

$$\mathbf{A} = \mathrm{softmax}\left(4 \cdot \frac{QK^T}{D}\right)$$

Where $D \in \mathbb{R}^{N \times N}$,

$$D_{i,j} = (\|q_i\| + \|k_j\|)^2$$

However, the value range of $[-1, 1]$ is too narrow for $\mathrm{softmax}$ function to function effectively. In order to improve the discrimination ability, a learnable parameter can be multiplied to the original values:

$$\mathbf{A}_{udps} = \mathrm{softmax}\left(4 \cdot \alpha \cdot \frac{QK^T}{D}\right)$$

Where $\alpha \in \mathbb{R}^1$, α is a learnable parameter, whose initial value can be set to around 10. This is the most primitive form of self-attention using Unit Dot Product Similarity. In practical applications, α can also be changed to α^2 to ensure the scaling factor is greater than 0. Multiple attention heads can share one α, or each attention head can have its own independent α.

Furthermore, since the calculation of D can be done in a parallel manner, the actual computation time only increases by about 10%.

Code

The code implementation for Unit Dot Product Similarity and self-attention using Unit Dot Product Similarity is in `main.py`. The code of related automic speech recognition model (improved Transformer and Conformer) is in `wenet-1.0.0-att_udps`, especially in `wenet-1.0.0-att_udps/wenet/transformer/attention.py` and `wenet-1.0.0-att_udps/wenet/transformer/encoder.py`.