ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 7, Issue 4, April 2018

An Extension of the Dot Product: The Idot Product

Ying Liu[1], Shaohui Wang[2]

Department of Engineering Technology, Savannah State University, Savannah, Georgia 31404

Department of Mathematics, Savannah State University, Savannah, Georgia 31404

Abstract: Dot product is an advanced subject of applications in many areas. It is known that the dot product is inadequate for many applications. In this paper, we introduce a new type of dot product, Image Dot Product or Idot Product. In particular, an L-factor for the Idot product is introduced. We also investigate both the binary Idot product, and the Idot product. In addition, there are many open questions to be answered about the Idot product.

Keywords: Dot product, scaler product, image, image recognition, artificial intelligence.

1. INTRODUCTION

The dot product or scale product has unlimited applications in many areas. In the age of AI [3], however, for many computer applications, the dot product is inadequate. It will be necessary to extend the dot product.

The classification problem [3] is to classify an unknown pattern into a set of classes. Given the following set of sample vectors, S = {s_1, s_2, s_3}, where each sample is a 15-dimension binary vector:

	011		111		111
	001		001		001
s1 =	001	s2=	111	s3=	111
	001		100		001
	001		111		111

They look like 3 images of 3-pixel by 5-pixel; furthermore, these are images of characters 1, 2, and 3. These three vectors form a sample vector set.

Assuming another unknown vector, u_1, is given:

	001
	001
u_1=	011
	001
	101.

The question is to classify the unknown vector, u_1, from the sample vector set, S = {s_1, s_2, s_3}. In this particular example, the correct classification is that the unknown vector, u_1, belongs to class, s_1.

There are three types of pixels: object, background, and noise. We will examine each pixel of a vector and each pixel is classified into {object, background, noise}. The noise pixel will appear randomly in the unknown patterns. Let us look at the sample vector, s_1:

011
001
001
001
001.

Note, this vector has 6 “object” pixels represented by “1”, 9 “background” pixels represented by “0”, and 0 “noise” pixels. Let us look at the unknown vector, u_1:

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 7, Issue 4, April 2018

001
001
011
001
101

Note, this vector has 5 “object” pixels, 8 “background” pixels, and 2 “noise” pixels; if we remove the noise pixels, this vector will look like this:

001
001
001
001
001

In pixel matching, not all pixel pairs have an equal weight. It is the collection of the object pixels that determines the classification of the unknown vector. The background pixels play much smaller roles in determining the classification of an unknown vector. As a result, the object matching should have the highest weight: if an object pixel matches a corresponding object pixel, then this match should have the highest weight. The background match should have lower weight: if a background pixel matches a corresponding background pixel, then it should have a lower weight than the object pixel match. The mismatch should have the lowest weight: if an object pixel matches a background pixel, then it should produce the lowest contribution.

Let us classify the unknown vector, u_1, from the sample vectors, $\{s_1, s_2, s_3\}$. We can have several classification algorithms, including classification by shortest distance (K-Means Clustering) [2,4,5,6] and classification by largest projection [1]. We will briefly look at both approaches to address several shortcoming of the dot product.

Let us first look at the classification by minimum distance. This is a simple case of K-Means Clustering [2,4,5,6], where each sample vector is a cluster. The distance is the norm of $(u_1 - s_1)$, $(u_1 - s_2)$, $(u_1 - s_3)$. The minimum distance will determine the classification of the given unknown vector. Let us look at $(u_1 - s_1)$:

011 001 0 1 0
001 001 0 0 0
001 - 011 = 0 -1 0
001 001 0 0 0
001 101 -1 0 0

When an object pixel matches an object pixel, it contributes 1-1 = 0; and when a background pixel matches a background pixel, it contributes 0 – 0 = 0. The dot product cannot discriminate the difference. It gives too much weight to the background match. This overestimates the contribution of the background.

Let us now look at the classification by maximum projection [6]: $(u_1 \cdot s_1)$, $(u_1 \cdot s_2)$, $(u_1 \cdot s_3)$. The maximum projections will determine the classification of the given unknown vector. Let us look at $(u_1 \cdot s_1)$:

011 001 0+0+1+
001 001 0+0+1+
001 · 011 = 0+0+1+ = 5
001 001 0+0+1+
001 101 0+0+1

When an object pixel matches an object pixel, it will contribute $1 \cdot 1 = 1$; and when a background pixel matches a background pixel, it contributes $0 \cdot 0 = 0$. When an object pixel matches a background pixel, it will contribute $1 \cdot 0 = 0$; and when a background pixel matches an object pixel, it contributes $0 \cdot 1 = 0$.

Note, both the mismatch ($1 \cdot 0 = 0 \cdot 1 = 0$) and the correct background match ($0 \cdot 0 = 0$) have the same weight. The dot product cannot take any contribution from the correct background matching. Although the correct background matching ($0 \cdot 0 = 0$) is not as important as correct object matching ($1 \cdot 1 = 1$), the dot product thinks it has the same weight as a mismatch ($1 \cdot 0 = 0 \cdot 1 = 0$). This underestimates the contribution of the background.

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 7, Issue 4, April 2018

In both cases, the dot products are inadequate to complete the vector classification computation; an extension of the dot product will be necessary, which can distinguish three different types of pixel matches:

- Object pixel vs object pixel
- Background pixel vs background pixel
- All other matches.

In this paper, we make an attempt to extend the dot product by introducing a new type of dot product, Image Dot Product or Idot product. In particular, an L-factor is introduced. We first introduce the binary Idot product; then we introduce the Idot product. There are many open questions to be answered about the Idot product.

2. BACKGROUND

The definition of the dot product is well known. Below, we merely introduce the notations used for the rest of this paper.

If $e_1, ..., e_n$ are the standard basis vectors in R_n, then we can write:

$$a = [a_1, \dots, a_n] = \sum a_i e_i,$$
$$b = [b_1, \dots, b_n] = \sum b_i e_i.$$

Without loss of generality, assuming vectors, e_i, are an orthonormal basis,

$$e_i \cdot e_j = \delta_{ij} .$$

Now
$$a \cdot b = \sum a_i b_i .$$

In particular, (1) if vector b is a unit vector, the dot product gives the projection of a vector, a, along the direction of vector, b; (2) The distance between a and b is the norm of (a – b).

3. BINARY IDOT PRODUCT

We will now introduce the binary Idot product. If $e_1, ..., e_n$ are the standard basis vectors in R_n, then we can write:

$$a = [a_1, \dots, a_n] = \sum a_i e_i$$
$$b = [b_1, \dots, b_n] = \sum b_i e_i$$

Without loss of generality, assuming vectors, e_i, are an orthonormal basis,

$$e_i \cdot e_j = \delta_{ij} .$$

Now, the Binary Idot product is defined as:

$$a \cdot b = \sum l_i (i, a_i , b_i)$$

Where $l_i (i, a_i , b_i)$ is called a L-factor, which is a function of the index i, a_i and b_i. In this way, it can discriminate an object pixel from a background pixel. For a binary variable, x = 0, or 1, rewrite the variable as:

$$x = x (\delta_{x1} + \delta_{x0}) .$$

An object pixel, x, is identified by δ_{x1}; and a background pixel is identified by δ_{x0}. We define the L-factor as follows:

$$l_i (i, x, y) = w_{11} (i) \delta_{x1} \delta_{y1} + w_{10} (i) \delta_{x1} \delta_{y0} + w_{01} (i) \delta_{x0} \delta_{y1} + w_{00} (i) \delta_{x0} \delta_{y0}$$

If $w_{11} (i), w_{10} (i) , \dots$, depends on i, we call them local weights. If they are independent of i, we call them global weights. For the rest of this paper, for simplicity, we will deal with global weights only:

$$l_i (x, y) = w_{11} \delta_{x1} \delta_{y1} + w_{10} \delta_{x1} \delta_{y0} + w_{01} \delta_{x0} \delta_{y1} + w_{00} \delta_{x0} \delta_{y0}$$

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 7, Issue 4, April 2018

This is an extension of the standard dot product as follows; rewrite the product of two variables, x and y, as:

$$xy = (x\delta_{x1} + x\delta_{x0})(y\delta_{y1} + y\delta_{y0})$$

$$= x\,(\delta_{x0}\ \ \delta_{x1}) \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \delta y0 \\ \delta y1 \end{pmatrix} y$$

This product does not discriminate the background and object. Now, we can extent this product and define the L-matrix:

$$xy = (\delta_{x0}\ \ \delta_{x1}) \begin{pmatrix} w11 & w12 \\ w21 & w22 \end{pmatrix} \begin{pmatrix} \delta y0 \\ \delta y1 \end{pmatrix}.$$

Here $\delta_{x1}\delta_{y1}$ specifies the object pixel matches and $w_{11}(i)$ is its weight; $\delta_{x0}\delta_{y0}$ specifies the background pixel matches and $w_{00}(i)$ is its weight; and $\delta_{x1}\delta_{y0}$, $\delta_{x0}\delta_{y1}$ specifies mismatches.

Example. Define the L-factor:

$$l_i(x, y) = 0.9\ \delta_{x1}\delta_{y1} + 0.1\ \delta_{x0}\delta_{y0},$$

while the missing weight means that they are 0. The Idot product is:

$$(1, 0, 1, 0) \cdot (1, 0, 0, 1) = 0.9 + 0.1 = 1.$$

Example. Define the L-factor:

$$l_i(x, y) = 1.0\,\delta_{x1}\delta_{y1} - 0.1\,\delta_{x1}\delta_{y0} - 0.1\,\delta_{x0}\delta_{y1} + 0.2\,\delta_{x0}\delta_{y0}$$

then

$$(1, 0, 1, 0) \cdot (1, 0, 0, 1) = 1 + 0.2 - 0.1 - 0.1 = 1.$$

4. IDOT PRODUCT

In the last section, we introduced the binary Idot product. The advantage of binary numbers is that it classifies a pixel into two types: object (1) and background (0).

To extend the Idot product beyond binary vectors, a threshold, T, will be needed to separate a pixel into two types: object (pixel >= T) and background (pixel < T).

Define x as a step function of a pixel value:

$$x(\text{pixel}, T) = \begin{cases} 1, & if\ (\text{pixel} >= T) \\ 0, & if\ (\text{pixel} < T) \end{cases}$$

then we can extend the definition below:

If $e_1, \ldots, e_n$ are the standard basis vectors in R_n, then we can write:

$$a = [a_1, \ldots, a_n] = \sum a_i\ e_i$$
$$b = [b_1, \ldots, b_n] = \sum b_i\ e_i$$

Without loss of generality, assuming vectors, e_i, are an orthonormal basis,

$$e_i \cdot e_j = \delta_{ij}.$$

Now the Idot product is defined as:

$$a \cdot b = \sum a_i\ b_i\ l_i(a_i, b_i, T)$$

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 7, Issue 4, April 2018

Where the L-factor is:

$$l_i(a_i, b_i, T) = w_{11}\delta_{x1}\delta_{y1} + w_{10}\delta_{x1}\delta_{y0} + w_{01}\delta_{x0}\delta_{y1} + w_{00}\delta_{x0}\delta_{y0}$$

where

$$x = \begin{cases} 1, & if\ (ai >= T) \\ 0, & if\ (ai < T) \end{cases}$$

$$y = \begin{cases} 1, & if\ (bi >= T) \\ 0, & if\ (bi < T) \end{cases}$$

There are numerous computer applications using the Idot product. The results are significantly better than the dot product. However, this topic will be beyond the scope of this paper.

5. YET OTHER EXTENSIONS

The above Idot definitions assumed that vectors are of the same dimension. Often the sample vector is smaller than the unknown vector, so this Idot product has to be extended. Continuing from the earlier example, an unknown vector could be:

$$B = \begin{matrix} 0010 \\ 0010 \\ 0110 \\ 0010 \\ 1010 \\ 0000 \end{matrix}$$

To compute the Idot product, $a \cdot B$, assume B has a set of sub-vectors, $B = \{ b_k, k = 0, 1, 2, \ldots\}$, that have the same dimension as vector, a.

The Idot product can be defined as:

$$a \cdot B = \max \{ a \cdot b_k \}, \quad b_k \in B.$$

This is one of many options and this definition emphasizes the maximum sub-vectors matches. There are numerous other alternatives to the above definitions, such as replacing the max in the above definition by sum, average, minimum, counting, … .

6. OPEN QUESTION

A natural question is: given a set of sample binary vectors and a set of classified binary vectors where the classifications are known, what is an algorithm to determine $\{w_{11}, w_{10}, w_{01}, w_{00}\}$? One can ask a similar question for a non-binary case, how to determine $\{w_{11}, w_{10}, w_{01}, w_{00}, T\}$? One can ask a similar question when the sample vector dimension is smaller than the unknown vectors. One can ask a similar question when the weights are local instead of global.

7. CONCLUSION

In this paper, we have illustrated why dot product is inadequate for many applications. We have introduced a new type of dot product, Image Dot Product or Idot product. In particular, an L-factor has been introduced. We have introduced both the binary Idot product and the Idot product. There are many open questions to be answered about weight inference for the Idot product.

ACKNOWNLEDGEMENTS

I would like to thank **Gina Porter** for proof reading of this paper.

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

IJARCCE

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 7, Issue 4, April 2018

REFERENCES

[1] S. Amari, K. Kurata, and H Nagaoka, "Information geometry of Boltzmann machine," IEEE Trans., Neural Network, Vol. 3, No. 2, pp. 260 – 271, 1992.
[2] E.W. Forgy, "Cluster analysis of multivariate data: efficiency versus interpretability of classifications". Biometrics 21, 768–769, 1965.
[3] G. Hinton, G. E., Osindero, S. and Teh, Y., "A fast learning algorithm for deep belief nets," Neural Computation 18, pp 1527-1554, 2006.
[4] S. P. Lloyd, "Least squares quantization in PCM" (PDF). IEEE Transactions on Information Theory 28 (2): 129–137, 1957, 1982.
[5] J. B. MacQueen, Some Methods for classification and Analysis of Multivariate Observations. Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability 1. University of California Press. pp. 281–297, 1967.
[6] H. Steinhaus, "Sur la division des corps matériels en parties". Bull. Acad. Polon. Sci. (in French) 4 (12): 801–804, 1957.