

# Energy efficient cosine similarity measures according to a convex cost function

Cem Emre Akbaş<sup>1</sup> · Osman Günay<sup>3</sup> · Kasım Taşdemir<sup>2</sup> · A. Enis Çetin<sup>1</sup>

Received: 8 February 2016 / Revised: 28 June 2016 / Accepted: 20 July 2016 / Published online: 17 August 2016  
© Springer-Verlag London 2016

**Abstract** We propose a new family of vector similarity measures. Each measure is associated with a convex cost function. Given two vectors, we determine the surface normals of the convex function at the vectors. The angle between the two surface normals is the similarity measure. Convex cost function can be the negative entropy function, total variation (TV) function and filtered variation function constructed from wavelets. The convex cost functions need not to be differentiable everywhere. In general, we need to compute the gradient of the cost function to compute the surface normals. If the gradient does not exist at a given vector, it is possible to use the sub-gradients and the normal producing the smallest angle between the two vectors is used to compute the similarity measure. The proposed measures are compared experimentally to other nonlinear similarity measures and the ordinary cosine similarity measure. The TV-based vector product is more energy efficient than the ordinary inner product because it does not require any multiplications.

**Keywords** Cosine similarity measures · Convex cost functions ·  $\ell_1$  norm

✉ Cem Emre Akbaş  
akbas@ee.bilkent.edu.tr  
Osman Günay  
gunayosman@gmail.com  
Kasım Taşdemir  
kasim.tasdemir@agu.edu.tr  
A. Enis Çetin  
cetin@bilkent.edu.tr

<sup>1</sup> Department of Electrical and Electronics Engineering, Bilkent University, 06800 Çankaya, Ankara, Turkey

<sup>2</sup> Department of Computer Engineering, Gül University, 38080 Kocasinan, Kayseri, Turkey

<sup>3</sup> ASELSAN A.Ş., 06370, Yenimahalle, Ankara, Turkey

## 1 Introduction

Inner product of two vectors is used in many big data analysis, machine learning and signal processing algorithms [1, 2]. In this article, we define new “vector products” and construct cosine similarity measures using the new vector products. Some of the vector products that we introduce in this article do not require any multiplications. As a result, they lead to energy efficient cosine similarity measures because multiplication requires more energy than addition and subtraction.

It is well known that the cosine similarity between two vectors  $\mathbf{x}_1$  and  $\mathbf{x}_2$  is computed using the inner product of the two vectors divided by the  $\ell_2$ -norms of the vectors:

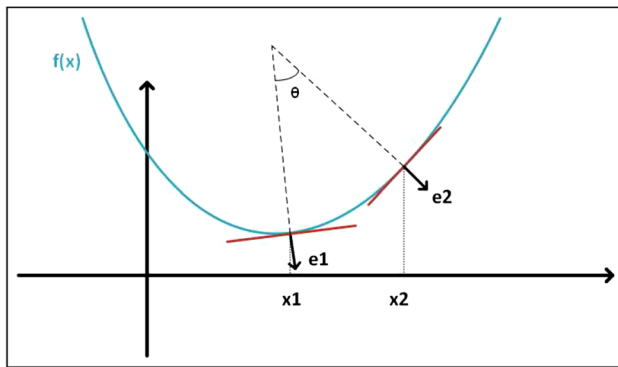
$$\cos(\mathbf{x}_1, \mathbf{x}_2) = \frac{\langle \mathbf{x}_1, \mathbf{x}_2 \rangle}{\|\mathbf{x}_1\| \|\mathbf{x}_2\|}, \quad (1)$$

We want to determine the similarity of two vectors according to an associated convex cost function  $f$ . In Fig. 1, the main idea behind the new cost measure is graphically described. Given a convex cost function  $f$ , tangent lines and the surface normals  $\mathbf{e}_1$  and  $\mathbf{e}_2$  at  $(\mathbf{x}_1, f(\mathbf{x}_1))$  and  $(\mathbf{x}_2, f(\mathbf{x}_2))$  are determined, respectively, and the proposed similarity measure is defined as the cosine similarity between the surface normals (or surface tangents) of the two vectors  $\mathbf{x}_1$  and  $\mathbf{x}_2$  on the convex cost function  $f$  as follows:

$$C(\mathbf{x}_1, \mathbf{x}_2) = \langle \mathbf{e}_1, \mathbf{e}_2 \rangle, \quad (2)$$

where  $\mathbf{e}_1$  and  $\mathbf{e}_2$  are the unit surface normal vectors of the convex cost function  $f$  at  $\mathbf{x}_1$ , and  $\mathbf{x}_2$ , respectively. We call the cosine measure  $C(\mathbf{x}_1, \mathbf{x}_2)$ , Bregman angle between  $\mathbf{x}_1$ , and  $\mathbf{x}_2$ .

This new measure is inspired by the well-known Bregman divergence [3–6], which is based on the surface tangent of



**Fig. 1** Angle between  $\mathbf{e}_1$  and  $\mathbf{e}_2$  is the similarity value between the two vectors  $\mathbf{x}_1$  and  $\mathbf{x}_2$

a given cost function  $f$ . The Bregman divergence  $D(\mathbf{x}_1, \mathbf{x}_2)$  between the two vectors  $\mathbf{x}_1$  and  $\mathbf{x}_2$  is the “vertical” distance between the cost function  $f$  and the tangent line at  $\mathbf{x}_2$  evaluated at the vector  $\mathbf{x}_1$ :

$$D(\mathbf{x}_1, \mathbf{x}_2) = f(\mathbf{x}_1) - f(\mathbf{x}_2) - \nabla f(\mathbf{x}_2)^T (\mathbf{x}_1 - \mathbf{x}_2) \quad (3)$$

For example, when  $f(\mathbf{x}) = \|\mathbf{x}\|^2$ , then the Bregman divergence reduces to Euclidean or the square distance between the two vectors, i.e.,  $D(\mathbf{x}_1, \mathbf{x}_2) = \|\mathbf{x}_1 - \mathbf{x}_2\|^2$ . Similarly, when the cost function  $f$  is the Euclidean distance, the new cosine measure is the same as the ordinary cosine similarity measure defined in Eq. (1).

Our main motivation is to introduce a new family of similarity functions to be used in various applications where the inner product of two vectors is desired. We provide a comparative experimental analysis of the proposed similarity measures to determine their performance of different datasets.

The paper is organized as follows. In Sect. 2, we describe the Bregman angle-based cosine similarity measures. It is possible to select the cost function as the well-known total variation (TV) function. In this case, the resulting TV-based cosine similarity measure can be implemented without performing any multiplications. In Sect. 3, we describe  $\ell_1$ -norm based “cosine-like” similarity measures. The vector product operations of the  $\ell_2$ -norm based measures can be also computed without performing any multiplications. As a result, computationally efficient similarity measures are realized. In Sect. 4, experimental results are presented.

## 2 Bregman angle similarity measure

Given a convex function  $f(\mathbf{x})$ , the unit surface normal is defined as follows:

$$\mathbf{e} = \frac{[\nabla f(\mathbf{x}), -1]}{\|[\nabla f(\mathbf{x}), -1]\|} \quad (4)$$

In Sect. 2.1, we use the surface normals of the convex function to construct vector similarity measures for various convex functions such as the negative entropy and the TV function. In Sect. 2.2, we use the surface gradients of the convex function to construct vector similarity measures in a similar manner. These measures are obviously related to each other.

### 2.1 Similarity measure based on surface normals

The general form of the proposed similarity measure based on surface normals is defined as follows:

$$C(\mathbf{x}_1, \mathbf{x}_2) = \frac{\langle \nabla f(\mathbf{x}_1), \nabla f(\mathbf{x}_2) \rangle + 1}{\sqrt{\langle \nabla f(\mathbf{x}_1), \nabla f(\mathbf{x}_1) \rangle + 1} \sqrt{\langle \nabla f(\mathbf{x}_2), \nabla f(\mathbf{x}_2) \rangle + 1}} \quad (5)$$

When the cost function is the well-known negative entropy function  $f(\mathbf{x}) = \sum_i \mathbf{x}(i) \log(\mathbf{x}(i))$ , the surface normals are given by:

$$\begin{aligned} \mathbf{E}_i &= \left[ \frac{\partial f(\mathbf{x}_i)}{\partial \mathbf{x}_i(1)}, \dots, \frac{\partial f(\mathbf{x}_i)}{\partial \mathbf{x}_i(N)}, -1 \right] \\ &= [\log(\mathbf{x}_i(1)) + 1, \dots, -1], \quad i = 1, 2 \end{aligned} \quad (6)$$

and unit normals are:

$$\mathbf{e}_i = \frac{\mathbf{E}_i}{\|\mathbf{E}_i\|}, \quad i = 1, 2 \quad (7)$$

The cosine similarity based on the negative entropy function between the vectors is then defined as follows:

$$C(\mathbf{x}_1, \mathbf{x}_2) = \frac{\sum_i (\log(\mathbf{x}_1(i)) + 1)(\log(\mathbf{x}_2(i)) + 1) + 1}{\sqrt{\sum_i (\log(\mathbf{x}_1(i)) + 1)^2 + 1} \sqrt{\sum_i (\log(\mathbf{x}_2(i)) + 1)^2 + 1}} \quad (8)$$

Since the entropy function is only defined for positive values, we can use the modified entropy function introduced in [7] to account for negative values:

$$f(\mathbf{x}) = \sum_i \left( |\mathbf{x}(i)| + \frac{1}{e} \right) \log \left( |\mathbf{x}(i)| + \frac{1}{e} \right) + \frac{1}{e} \quad (9)$$

In this case, the Bregman angle measure can be obtained from the following surface normals:

$$\mathbf{E}_i = \left[ \text{sign}(\mathbf{x}_i(1)) \left( \log \left( |\mathbf{x}_i(1)| + \frac{1}{e} \right) + 1 \right), \dots, -1 \right] \quad (10)$$

for  $i = 1$  and  $2$ .

Entropy function-based cost functions require the computation of logarithms. Therefore, they are computationally more expensive than the ordinary cosine similarity measure.

A well-known convex cost function is the total variation (TV) function [8]:

$$TV(\mathbf{x}) = \sum_i |x_{i+1} - x_i| \tag{11}$$

The surface normal vector  $SN(TV(\mathbf{x}))$  of the TV function is given by:

$$SN(TV(\mathbf{x})) = \left[ \frac{\partial TV(\mathbf{x})}{\partial x_1}, \frac{\partial TV(\mathbf{x})}{\partial x_2}, \dots, \frac{\partial TV(\mathbf{x})}{\partial x_N}, -1 \right]$$

$$= [(\text{sign}(x_2 - x_1)), (\text{sign}(x_2 - x_1) - \text{sign}(x_3 - x_2)), \dots, (\text{sign}(x_N - x_{N-1})), -1], \tag{12}$$

where  $\text{sign}(\cdot)$  is the signum function. We can easily construct a vector similarity measure from the above vector. It turns out that we get the best experimental results using the TV function.

Similarly, for  $f(\mathbf{x}) = \|\mathbf{x}\|^2$  the distance function becomes:

$$C(\mathbf{x}_1, \mathbf{x}_2) = \frac{\sum_i 4\mathbf{x}_1(i)\mathbf{x}_2(i) + 1}{\sqrt{\sum_i 4\mathbf{x}_1(i)^2 + 1}\sqrt{\sum_i 4\mathbf{x}_2(i)^2 + 1}} \tag{13}$$

When we remove the last entry from the surface normals, the Bregman cosine similarity becomes the ordinary cosine similarity.

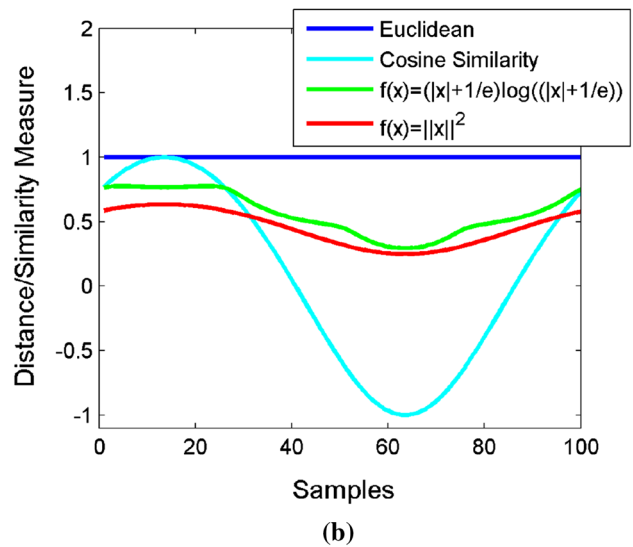
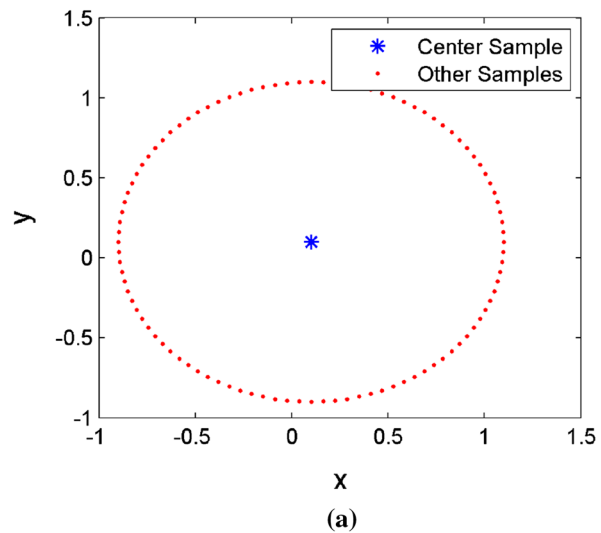
Figures 2 and 3 present two examples to compare the proposed similarity measures for two extreme cases of sample distributions. When the samples are defined over a circle, the Euclidean distance is the same for all samples, but cosine similarity and Bregman angle can distinguish between samples at different angles according to the center sample. When the samples are defined on a line, the ordinary cosine similarity cannot separate the samples, but proposed Bregman angle measures produce different results.

**2.2 Similarity measures based on surface gradients**

When surface gradients are used instead of surface normals, the similarity measure reduces to:

$$C_t(\mathbf{x}_1, \mathbf{x}_2) = \frac{\langle \nabla f(\mathbf{x}_1), \nabla f(\mathbf{x}_2) \rangle}{\sqrt{\langle \nabla f(\mathbf{x}_1), \nabla f(\mathbf{x}_1) \rangle} \sqrt{\langle \nabla f(\mathbf{x}_2), \nabla f(\mathbf{x}_2) \rangle}} \tag{14}$$

Bregman distance uses surface gradients of the convex cost function. Therefore, we can also use the surface gradients to define another cosine similarity measure. Given two vectors  $\mathbf{x}_1$  and  $\mathbf{x}_2$ , we compute the gradient vectors  $\mathbf{t}_1$  and  $\mathbf{t}_2$



**Fig. 2** Distance similarity measures for concentric distribution of samples. **a** Distribution of samples. **b** Distance/similarity measures

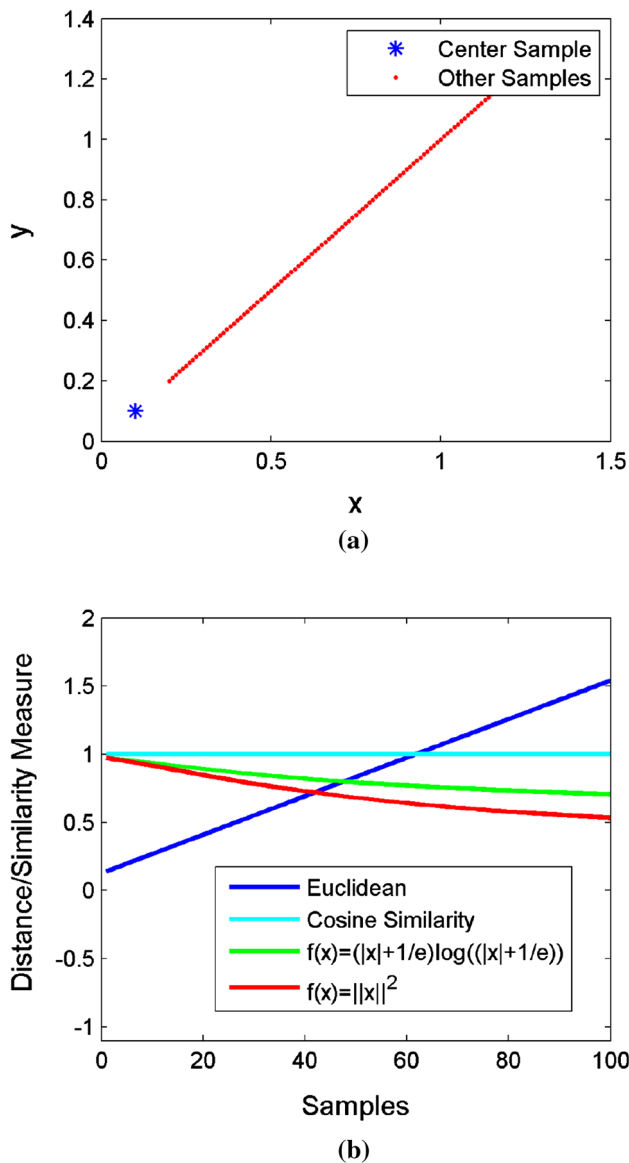
of the cost function  $f$  at  $\mathbf{x}_1$  and  $\mathbf{x}_2$  and the angle between  $\mathbf{t}_1$  and  $\mathbf{t}_2$  is the cosine similarity measure.

For the negative entropy function, the vector similarity measure becomes:

$$C_t(\mathbf{x}_1, \mathbf{x}_2) = \frac{\sum_i (\log(\mathbf{x}_1(i)) + 1)(\log(\mathbf{x}_2(i)) + 1)}{\sqrt{\sum_i (\log(\mathbf{x}_1(i)))^2 + 1}\sqrt{\sum_i (\log(\mathbf{x}_2(i)))^2 + 1}} \tag{15}$$

This is similar to the Eq. (8), but the dimension of the inner product is smaller than Eq. (8).

When the cost function is the Euclidean distance function,  $C_t$  becomes the same as the ordinary cosine similarity measure between the two vectors.



**Fig. 3** Distance similarity measures for linear distribution of samples. **a** Distribution of samples. **b** Distance/similarity measures

### 3 Multiplication-free cosine similarity measures<sup>1</sup>

Another well-known convex cost function is the  $\ell_1$  norm function  $f(\mathbf{x}) = \|\mathbf{x}\|_1$ . Its gradient leads to a widely used correlation measure based on the sign information of input vectors:

$$SG(\mathbf{x}) = [\text{sign}(x_1), \text{sign}(x_2), \dots, \text{sign}(x_N)]^T \tag{16}$$

In this case, the cosine similarity function is normalized by the  $\ell_2$  norms of the above gradient vectors:

<sup>1</sup> This work is partially presented in [9].

$$c_0(\mathbf{x}, \mathbf{y}) \triangleq \frac{\sum_{i=1}^N (SG(\mathbf{x}))_i \times (SG(\mathbf{y}))_i}{\|SG(\mathbf{x})\|_2 \times \|SG(\mathbf{y})\|_2} \tag{17}$$

This similarity vector is too simple for many applications. We recently introduced a related family of multiplication-free vector products [10–14]. It is based on an additive operator whose sign is the same as the multiplication. Let  $a$  and  $b$  be two real numbers. The new operator  $\odot$  is defined as follows:

$$a \odot b = \text{sign}(a \times b) \cdot (|a| + |b|), \tag{18}$$

where

$$\text{sign}(a \times b) = \begin{cases} 1, & \text{if } a \cdot b > 0, \\ 0, & \text{if } a \cdot b = 0, \\ -1, & \text{if } a \cdot b < 0. \end{cases} \tag{19}$$

The operator  $\odot$  is basically a summation operation. However, the sign of the result of  $a \odot b$  is the same as  $a \times b$ . Therefore, the  $\odot$  operator behaves like the multiplication operation.

We define a new “vector product” of two  $N$ -dimensional vectors  $\mathbf{x}_1$  and  $\mathbf{x}_2$  based on  $\odot$  in  $R^N$  as follows:

$$\langle \mathbf{x}_1 \odot \mathbf{x}_2 \rangle = \sum_{i=1}^N x_1(i) \odot x_2(i), \tag{20}$$

where  $x_1(i)$  and  $x_2(i)$  are the  $i$ -th elements of the vectors  $\mathbf{x}_1$  and  $\mathbf{x}_2$ , respectively. Notice that the vector product of a vector  $\mathbf{x}$  with itself reduces to a scaled  $\ell_1$  norm of  $\mathbf{x}$  as follows:

$$\langle \mathbf{x} \odot \mathbf{x} \rangle = \sum_{i=1}^N x(i) \odot x(i) = 2 \sum_{i=1}^N |x(i)| = 2\|\mathbf{x}\|_1. \tag{21}$$

Based on the new vector product, we define several vector similarity measures between the two vectors  $x$  and  $y$  as follows:

$$c_1(\mathbf{x}, \mathbf{y}) \triangleq \frac{\langle \mathbf{x} \odot \mathbf{y} \rangle}{\|\mathbf{x}\|_1 + \|\mathbf{y}\|_1}, \tag{22}$$

where  $\|\mathbf{x}\|_1$  and  $\|\mathbf{y}\|_1$  are the  $\ell_1$  norms of vectors  $\mathbf{x}$  and  $\mathbf{y}$ , respectively. Similar to the ordinary cosine similarity measure, the numerator contains the vector product of the two vectors and the vector product is normalized by the sum of the  $\ell_1$  norms of the two vectors  $\mathbf{x}$  and  $\mathbf{y}$ . When  $\mathbf{x} = \mathbf{y}$ ,  $c_1(\mathbf{x}, \mathbf{y}) = 1$  because  $\langle \mathbf{x} \odot \mathbf{x} \rangle = 2\|\mathbf{x}\|_1$ .

It is also possible to define other related measures using relation operators:

$$c_2(\mathbf{x}, \mathbf{y}) \triangleq \frac{\sum_{i=1}^N \text{sign}(x_i \times y_i) \cdot \max(|x_i|, |y_i|)}{(\|\mathbf{x}\|_1 + \|\mathbf{y}\|_1)/2}, \tag{23}$$

where  $\text{sign}(x_i \times y_i)$  is defined as in (19) and the vectors  $\mathbf{x} = [x_1, x_2, \dots, x_N]^T$  and  $\mathbf{y} = [y_1, y_2, \dots, y_N]^T$ , respectively. A related fourth cosine similarity measure is defined as follows:

$$c_3(\mathbf{x}, \mathbf{y}) \triangleq \frac{\sum_{i=1}^N \text{sign}(x_i \times y_i) \cdot \min(|x_i|, |y_i|)}{(\|\mathbf{x}\|_1 + \|\mathbf{y}\|_1)/2}, \quad (24)$$

where the maximum operation in (23) is replaced by the minimum operation. The similarity measure  $c_1$  can also be normalized in a different manner as follows:

$$c_4(\mathbf{x}, \mathbf{y}) \triangleq \frac{\sum_{i=1}^N \text{sign}(x_i \times y_i) \cdot (|x_i| + |y_i|)}{2 \cdot \sum_{i=1}^N \max(|x_i|, |y_i|)} \quad (25)$$

Clearly, when  $\mathbf{x} = \mathbf{y}$ ,  $c_4(\mathbf{x}, \mathbf{x}) = 1$ . Similarly,  $c_2$  and  $c_3$  can also be normalized as in  $c_4$ .

In all similarity measures defined in Eqs. (22–25), the term  $\text{sign}(x_i \times y_i)$  is common and provides the correlation information between the two vectors  $\mathbf{x}$  and  $\mathbf{y}$ . As a result, the computational complexity and power consumption due to signal analysis can be decreased significantly.

Obviously,  $c_i(\mathbf{x}, \mathbf{y}) = c_i(\mathbf{y}, \mathbf{x})$ ,  $i = 1, 2, 3, 4$  in all three definitions [13–16]. Also, when  $\mathbf{x}=\mathbf{y}$ , they all produce the same result, i.e.,  $c_1(\mathbf{x}, \mathbf{y}) = c_2(\mathbf{x}, \mathbf{y}) = c_3(\mathbf{x}, \mathbf{y}) = c_4(\mathbf{x}, \mathbf{y}) = 1$ . Multiplication requires more power than addition and relational max or min operations. Since all four measures can be computed without performing any multiplications, they are all low-power vector similarity measures.

## 4 Experimental results

In the first experiment, we compare the performance of the Bregman angle measure with cosine similarity measure on a gesture phase segmentation dataset. The gesture phase segmentation dataset [17] was made available by UC Irvine Machine Learning Repository. The dataset contains features extracted from 7 videos with people performing various gestures. The dataset contains 5 classes and 1747 gesture phase data each having 18 attributes. In this paper, simulation examples are carried out using the first two classes which contain total of 202 instances.

First, input vectors are multiplied by  $10^7$  in order to improve classification performance of nonlinear similarity measures. Then, means of input vectors are subtracted from themselves to get zero-mean input vectors. In all experiments, leave-one-out strategy is followed. The size of the test set is one, and the training set contains the remaining data. The test set is circulated to cover all instances. 2 class 1-nearest neighbor classification is performed using the new Bregman similarity measures, and the similarity measures defined in Sect. 3. Classification accuracies are given in

Table 1. The best results are obtained by TV-based measure and  $c_4$  defined in Sect. 3.

In this dataset, the gradient-based similarity function described in Eq. (15) produces slightly lower results than the surface normal-based similarity measure.

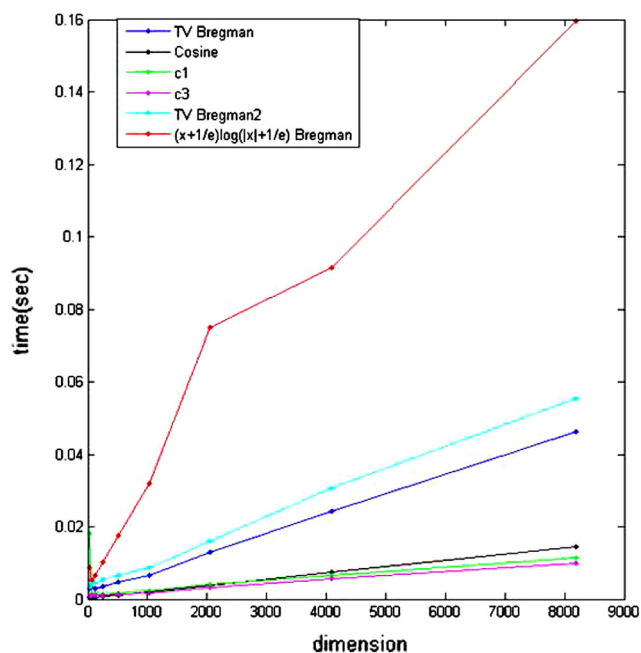
In the second experiment, we used the KTH-TIPS database that contains 800 images for 10 different classes of colored textures [18]. We use half of the images for each class as the training set and the rest as the test set. We use 1-neighbor k-NN classifier and eight different distance/similarity measures. To extract features from the images, we used the dual-tree complex wavelet transform (DT-CWT) as texture features and histograms in HSV color space as color features. Dual-tree complex wavelet transform tree is recently developed to overcome the shortcomings of conventional wavelet transform, such as shift variance and poor directional selectivity [19]. To obtain wavelet features, we divide images into four non-overlapping blocks and calculate the energies and variances of six different sub-bands (oriented at  $\pm 15, \pm 45, \pm 75$ ) for each block. The combined feature vectors of all blocks are used as the texture feature of the image. The results for this test are shown in Table 2. As shown in Table 2, the proposed measures have similar performance to the cosine similarity measure. The measure  $c_0$  does not pro-

**Table 1** Classification accuracies (percentage) for the 2 class 1-nearest neighbor classification with 8 different similarity measures for gesture phase segmentation dataset

Similarity/distance measure	Classification accuracy (%)
Bregman angle (negative entropy)	97.5
Bregman angle (TV)	99.0
Cosine similarity	98.0
$c_0$	80.7
$c_1$	84.2
$c_2$	84.7
$c_3$	98.5
$c_4$	99.0

**Table 2** Classification accuracies (percentage) for KTH-TIPS dataset

Similarity/distance measure	Classification accuracy (%)
Cosine similarity	(380/400) 95.0
Bregman angle (entropy)	(379/400) 94.75
Bregman angle (l2-norm)	(380/400) 95.0
Bregman angle (TV)	(314/400) 78.5
$c_1$	(369/400) 92.25
$c_2$	(231/400) 57.75
$c_3$	(383/400) 95.75
$c_4$	(382/400) 95.5



**Fig. 4** Running time of proposed Bregman cosine similarity measures, multiplication-free cosine similarity measures and the ordinary cosine similarity measure of MATLAB. The horizontal axis represents the size of the vector. Hundred similarity measurements are calculated for each dimension

duce meaningful results in this dataset. It is not included in Table 2. The measure  $c_3$  produces the best result.

In the third experiment, we compare running times of proposed Bregman cosine similarity measures, multiplication-free cosine similarity measures presented in [9] and the ordinary cosine similarity measure.

As seen in Fig. 4, multiplication-free cosine similarity measures are faster than the ordinary cosine similarity measure. This is because they are normalized by the  $\ell-1$  norms of the vectors. Run-times of Bregman angle measures are comparable to the ordinary cosine similarity measure.

**Table 3** First ten songs with the highest cosine similarity values for query = “we will rock you” using 7 different similarity measures

Cosine	$c_0$	$c_1$	$c_2$	$c_3$	$c_4$	Bregman angle (TV)
#921	#901	#921	#921	#921	#921	#921
#144	#921	#144	#144	#144	#144	#144
#336	#144	#336	#336	#378	#844	#336
#378	#336	#378	#378	#336	#378	#277
#673	#378	#673	#673	#258	#673	#378
#258	#673	#258	#244	#673	#258	#901
#166	#166	#901	#901	#812	#901	#258
#901	#258	#661	#661	#661	#661	#661
#166	#166	#166	#612	#612	#166	#166
#612	#104	#612	#166	#166	#612	#612

The first column is the ordinary cosine similarity measure. Numbers denote song IDs. Song #921 is the desired result “Queen—We Will Rock You”

In the fourth experiment, we present classification performance of multiplication-free cosine similarity measures, TV-based Bregman similarity measure and the ordinary cosine similarity measure in a song lyrics dataset (Table 3).

TF-IDF is a computationally efficient method to compute similarity of two text documents [20]. The TF-IDF matrix is computed as in [20]. Ordinary TF-IDF methods use the cosine similarity measure to compare TF-IDF vectors of documents. In this case, we compute the TF-IDF scores using the new proposed similarity measures. We also use the new multiplication-free similarity measures defined in Eqs. (22)–(25) in order to further reduce the computational cost of query retrieval.

In this example, 964 randomly chosen song lyrics are used. The query is entered and TF-IDF vector of query is constructed. Then, TF-IDF values of each keyword is computed for each song separately in order to construct 964 TF-IDF vectors. Mean value of each TF-IDF vector is subtracted from itself in order to obtain zero-mean vectors. Similarity values between TF-IDF vector of each song and the TF-IDF vector of query is computed using 7 different similarity measures. First ten songs with highest similarity values are presented as query results.

The fifth experiment is also on the song lyrics dataset. In this experiment, the fourth experiment is repeated 964 times with different queries (Table 5). The  $i$ -th query is the title of  $i$ -th song; therefore, the desired first place output is  $i$ -th song ( $i = 1, 2, \dots, 964$ ). If the desired song is retrieved at the first place, it is counted as a correct retrieval. Otherwise, it is counted as an incorrect retrieval.

As shown in Tables 3, 4 and 5, query retrieval performance of multiplication-free cosine similarity measures are comparable to the ordinary cosine similarity, while they are all computationally cheaper than the ordinary cosine similarity. In this dataset, the measure  $c_1$  produces the best result. TV-based Bregman angle similarity measure also produces comparable results to the ordinary cosine similarity. All pro-

**Table 4** First ten songs with the highest cosine similarity values for query = “something good can work” using 7 different similarity measures

Cosine	$c_0$	$c_1$	$c_2$	$c_3$	$c_4$	Bregman angle (TV)
#963	#771	#963	#963	#44	#963	#963
#44	#44	#44	#771	#963	#556	#44
#771	#166	#556	#44	#771	#44	#33
#556	#171	#771	#556	#608	#771	#771
#241	#817	#241	#89	#241	#188	#241
#817	#556	#498	#241	#556	#646	#556
#646	#241	#817	#646	#103	#241	#801
#103	#602	#646	#817	#389	#817	#817
#389	#389	#389	#177	#522	#184	#646
#184	#103	#103	#389	#955	#509	#103

The first column is the ordinary cosine similarity measure. Numbers denote song IDs. Song #963 is the desired result “Two Door Cinema Club—Something Good Can Work”

**Table 5** Query retrieving accuracy of 6 different similarity measures

Similarity/distance measure	Classification accuracy (%)
Cosine similarity	(927/964) 96.2
$c_1$	(932/964) 96.7
$c_2$	(930/964) 96.5
$c_3$	(917/964) 95.1
$c_4$	(905/964) 93.9
Bregman angle (TV)	(925/964) 96.0

**Table 6** Classification accuracies (percentage) for SUSY dataset

Similarity/distance measure	Classification accuracy (%)
Cosine similarity	(3327281/499e4) 66.67
Bregman angle (entropy)	(3279653/499e4) 65.72
Bregman angle (l2-norm)	(3334363/499e4) 66.81
Bregman angle (TV)	(3038327/499e4) 60.89
$c_0$	(2866967/499e4) 57.45
$c_1$	(3010165/499e4) 60.32
$c_2$	(2726263/499e4) 54.63
$c_3$	(3423296/499e4) 68.60
$c_4$	(3428785/499e4) 68.71

posed measures, except for  $c_0$  and  $c_3$ , can retrieve the desired song in the first place.

In the final part of the experiments, we use large datasets from UC Irvine Machine Learning Repository to test the performance of the proposed similarity measures. For these datasets, we used k-NN with  $k = 3$ . The first dataset is SUSY, that has 5 million 18-element feature vectors obtained using Monte Carlo simulations. The features correspond to exotic particles and background signals. The aim is to separate particles from the background [21]. In the original paper, 4.5 million samples are used for the training and 500 thousand samples are used for testing. We examine a more difficult

**Table 7** Classification accuracies (percentage) for forest covertype dataset

Similarity/distance measure	Classification accuracy (%)
Cosine similarity	(368671/569600) 64.73
Bregman angle (Entropy)	(371395/569600) 65.20
Bregman angle (l2-norm)	(368666/569600) 64.72
Bregman angle (TV)	(285922/569600) 50.19
$c_0$	(284774/569600) 49.99
$c_1$	(287201/569600) 50.42
$c_2$	(244722/569600) 42.96
$c_3$	(365169/569600) 64.11
$c_4$	(363544/569600) 63.82

scenario by using first 10 thousand samples for the training and the rest for testing. The classification accuracies for this experiment is shown in Table 6. In this dataset,  $c_4$  measure obtains the best results.

Both  $c_3$  and  $c_4$  are more energy efficient than the ordinary cosine similarity measure because they do not require any multiplications.

The second dataset is forest covertype dataset that has 581012 54-element feature vectors describing seven different types of forest cover. In the original paper, 58 % success rate is obtained using linear discriminant analysis and 70 % success rate is obtained using neural networks [22]. We use the same setup, which has 11340 training samples. The classification accuracies for this experiment is shown in Table 7. In this dataset, *Bregman Angle(Entropy)* measure obtains the best results.

### 5 Conclusion

In this paper, we introduced a family of vector similarity measures. Each measure is defined according to a given convex

cost function. The angle between the two surface normals or surface gradients is used to construct the similarity measures. When the cost function is the ordinary Euclidean function, the surface tangent-based similarity measure reduces to the ordinary cosine measure. It is experimentally observed that TV function-based vector similarity measure produces the best results in a dataset containing human gesture data. But it is not possible to single out a similarity measure that provides good performance on all datasets. The performance of a similarity function depends on the type of feature vectors in the dataset. The theoretical analysis of the proposed functions is a topic of the future work.

Some of the similarity measures introduced in this paper are energy efficient measures. They can be computed without performing any multiplications.

## References

- Rajaraman, A., Ullman, J.D.: Mining of Massive Datasets. Cambridge University Press, Cambridge (2012)
- Zadeh, R.B., Goel, A.: Dimension independent similarity computation. *J. Mach. Learn. Res.* **14**(1), 1605–1626 (2013)
- Banerjee, A., Merugu, S., Dhillon, I.S., Ghosh, J.: Clustering with bregman divergences. *J. Mach. Learn. Res.* **6**, 1705–1749 (2005)
- Bregman, L.: The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR Comput. Math. Math. Phys.* **7**(3), 200–217 (1967)
- Censor, Y., Lent, A.: An iterative row-action method for interval convex programming. *J. Optim. Theory Appl.* **34**(3), 321–353 (1981)
- Çetin, A.E.: Reconstruction of signals from fourier transform samples. *Signal Process.* **16**(2), 129–148 (1989)
- Kose, K., Gunay, O., Çetin, A.E.: Compressive sensing using the modified entropy functional. *Digit. Signal Process.* **24**, 63–70 (2014)
- Rudin, L.I., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. *Phys. D* **60**(1–4), 259–268 (1992)
- Akbaş, C.E., Bozkurt, A., Arslan, M., Aslanoglu, H., Çetin, A.E.: L1 norm based multiplication-free cosine similarity measures for big data analysis. In: 2014 International Workshop on Computational Intelligence for Multimedia Understanding (IWCIM), pp. 1–5 (2014)
- Suhre, A., Keskin, F., Ersahin, T., Çetin Atalay, R., Ansari, R., Çetin, A.E.: A multiplication-free framework for signal processing and applications in biomedical image analysis. In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1123–1127 (2013)
- Tuna, H., Onaran, I., Çetin, A.E.: Image description using a multiplier-less operator. *Signal Process. Lett. IEEE* **16**(9), 751–753 (2009)
- Yorulmaz, O., Pearson, T.C., Çetin, A.E.: Detection of fungal damaged popcorn using image property covariance features. *Comput. Electron. Agric.* **84**, 47–52 (2012)
- Duman, K., Çetin, A.E.: Target Detection in Sar Images Using Codifference and Directional Filters, pp. 76,990S–76,990S-10 (2010)
- Habiboglu, Y.H., Gunay, O., Çetin, A.E.: Covariance matrix-based fire and flame detection method in video. *Mach. Vis. Appl.* **23**(6), 1103–1113 (2012)
- Kleinberg, J., Tardos, E.: Approximation algorithms for classification problems with pairwise relationships: metric labeling and markov random fields. In: 40th Annual Symposium on Foundations of Computer Science, pp. 14–23 (1999)
- Charikar, M.S.: Similarity estimation techniques from rounding algorithms. In: Proceedings of the Thiry-fourth Annual ACM Symposium on Theory of Computing, ser. STOC '02. New York: ACM, pp. 380–388 (2002)
- Madeo, R.C.B., Wagner, P.K., Peres, S.M.: UCI Machine Learning Repository (2014). [Online]. <http://archive.ics.uci.edu/ml>
- Hayman, E., Caputo, B., Fritz, M., Eklundh, J.-O.: On the significance of real-world conditions for material classification. In: Pajdla, T., Matas, J. (Eds.) European Conference on Computer Vision (ECCV), pp. 253–266 (2004)
- Selesnick, I., Baraniuk, R., Kingsbury, N.: The dual-tree complex wavelet transform. In: Zoltowski, M.D. (Ed.) IEEE Signal Processing Magazine, vol. 22, no. 6, pp. 123–151 (2005)
- Ramos, J.: Using tf-idf to determine word relevance in document queries. In: Proceedings of the First Instructional Conference on Machine Learning (2003)
- Baldi, P., Sadowski, P., Whiteson, D.: Searching for exotic particles in high-energy physics with deep learning. *Nat. Commun.* **5**(4308), 1 (2014)
- Blackard, J.A., Dean, D.J.: Comparative accuracies of artificial neural networks and discriminant analysis in predicting forest cover types from cartographic variables. *Comput. Electron. Agric.* **24**, 131–151 (1999)