Hindawi Publishing Corporation
Journal of Applied Mathematics
Volume 2013, Article ID 749429, 11 pages
http://dx.doi.org/10.1155/2013/749429

Research Article

Image Matching Using Dimensionally Reduced Embedded Earth Mover's Distance

Fereshteh Nayyeri and Mohammad Faidzul Nasrudin

Centre for Artificial Intelligence Technology, Faculty of Information Science and Technology, University Kebangsaan Malaysia, 43600 UKM Bangi, Selangor, Malaysia

Correspondence should be addressed to Fereshteh Nayyeri; f.nayyeri@gmail.com

Received 3 July 2013; Accepted 31 October 2013

Academic Editor: Feng Gao

Finding similar images to a given query image can be computed by different distance measures. One of the general distance measures is the Earth Mover's Distance (EMD). Although EMD has proven its ability to retrieve similar images in an average precision of around 95%, high execution time is its major drawback. Embedding EMD into L_1 is a solution that solves this problem by sacrificing performance; however, it generates a heavily tailed image feature vector. We aimed to reduce the execution time of embedded EMD and increase its performance using three dimension reduction methods: sampling, sketching, and Dimension Reduction in Embedding by Adjustment in Tail (DREAT). Sampling is a method that randomly picks a small fraction of the image features. On the other hand, sketching is a distance estimation method that is based on specific summary statistics. The last method, DREAT, randomly selects an equally distributed fraction of the image features. We tested the methods on handwritten Persian digit images. Our first proposed method, sampling, reduces execution time by sacrificing the recognition performance. The sketching method outperforms sampling in the recognition, but it records higher execution time. The DREAT outperforms sampling and sketching in both the execution time and performance.

1. Introduction

One of the interesting problems in database communities is image retrieval from large databases. The fundamental issue is how to design a similarity measure in a manner that shows the concept of similarity between two images, because choosing a proper measure has considerable effects on image retrieval applications. Some of the similarity measures include the Earth Mover's Distance (EMD), Jeffrey's divergence, and Minkowski-form distance [1].

The EMD is a general and flexible metric that has desirable and striking properties for content-based image retrieval [2, 3]. This similarity measure, which applies to weighted point sets, measures the minimum amount of work needed to transform one set into another set by weight transportation. The most significant feature of EMD is that it quantifies perceptual similarity better than other types of distances used for image retrieval [2]. Although EMD can measure the exact distance between images, and by this measure we can retrieve the most similar images from a database, its execution time is problematic, and this similarity measure is very time consuming.

Another method, called embedded EMD to L_1, was proposed to solve the EMD problem. This method maps the image matrix to an L_1 norm; therefore, instead of comparing 2-dimensional matrixes, we can compare 1-dimensional vectors. Although this idea is less time consuming, it produces distortion. Sometimes, an exact computation may be practically infeasible; in this situation, an approximation solution is helpful to find the exact result with some distortion. Both execution time and performance are important factors in image retrieval, and we should attempt to reduce distortion as much as possible. In this paper, we propose two methods to improve the performance of embedded EMD. The first method, sampling, reduces the time but decreases performance. In the next proposed method, sketching, we improve performance by sacrificing the time of execution. Finally, in the last method, by solving the problem of sampling, we improve the performance while reducing the execution time.

Table 1: Relationship between image's size and array's length.

	G_1 n^2	G_2 $n^2/4$	G_3 $n^2/16$	G_4 $n^2/64$	G_5 $n^2/256$
Number of elements in array	256	64	16	4	1
Side length	1	2	4	8	16

The remainder of this paper is organized as follows. In Section 2 we discuss related previous work. In Section 3 we describe our proposed technique. Section 4 provides the details of our proposed methods. Finally, we discuss the results and our conclusion in Sections 5 and 6.

2. Previous Work

The concept of Earth Mover Distance (EMD) was first explored in [4] to measure perceptual shape similarity. The use of EMD for computing similarity between images was later proposed in [5]. Since then, the EMD has become a trendy similarity measure in computer vision; it has been used effectively in various applications including color-based image retrieval systems, texture signatures [6], shape matching [7–9], and music score matching [10]. The EMD performs quite well in comparison with other similarity measures, such as the Jeffrey divergence and the Minkowski-form distance. In addition, the EMD can be used to measure the differences between vector fields [11].

Some authors in [2] have compared the EMD with other similarity measures and evaluated the retrieval performance of each. The results of the comparisons demonstrate that the EMD is more robust than other measures for the purpose of image retrieval because it matches similarity better than other distances.

The main idea behind the EMD metric is as follows. Suppose that each image is a set of colored points in 2-dimensional space. The minimum amount of work needed to transform one set into another set is defined as the distance of two set points. In recent years, a low-distortion embedding of EMD into L_1 has been developed [12]; although the empirical results show that this distortion is much smaller than what had been estimated previously, the embedding steps themselves decrease the complexity of computing similarity between two images. Other authors [9] have reported on the complexities of querying the time and space of an exact EMD versus an embedded EMD for shape similarity. In this work, we demonstrate how to reduce the complexity of the computing correspondence between two images that are mapped to an L_1 norm by dimension reduction.

The most similar work in this area is that of Grauman and Darrell [9], who show a contour matching algorithm that quickly quantifies the minimum weight matching between sets of descriptive local features using the embedding of the Earth Mover's Distance (EMD) into a normed space. Their method achieves an increase in speed of four orders of magnitude over the exact method at the cost of only a 4% reduction in accuracy.

3. Dimension Reductions in L_1

In modern image retrieval applications, the data is sometimes not only very large relative to the physical memory or even to the disk, but also highly sparse. Accordingly, computing the embedded L_1 on large-scale sparse data can be challenging and time consuming. Various projection methods have been suggested for speeding up these computations. Dimension reduction in the L_1 norm has many applications in information retrieval. The authors of [13] show that dimension reduction by sampling from L_1 does not produce poor results. Additionally, by estimating distances in L_1 from random samples, the original L_1 distances can be recovered. Sampling methods become more important with increasingly large collections [14] because we can use the same set of random samples to estimate any L_1 pairwise distances [15], whereas measuring exact pairwise distances is often too time consuming or sometimes infeasible; however, random sampling often performs poorly when most of the samples are zeros [13]. Additionally, in strictly heavy-tailed data, the estimation errors are sometimes very large.

As another choice of random projection, various sketching algorithms have become popular. In general, a sketching algorithm outperforms random sampling, although random sampling is much more flexible [15]. In the sketching method, after scanning the data, we compute specific summary statistics, and then repeat this step k times.

3.1. Procedures of Sampling and Sketching

3.1. Procedures of Sampling and Sketching. Suppose we have a database of n images and we want to compare a particular image with this database. To do so, we need a measurement; this is when we use EMD. Consider that we have 2 images with high similarity, for example, in Figures 1 and 2 apples with spots in different positions.

In this situation, the EMD of two spots in these images is computed as follows.

Euclidean distance between

(i) 1st pixels: $\sqrt{(8-9)^2+(12-8)^2}=\sqrt{15}$,

(ii) 2nd pixels: $\sqrt{(8-9)^2+(13-9)^2}=\sqrt{15}$,

(iii) 3rd pixels: $\sqrt{(9-10)^2+(12-8)^2}=\sqrt{15}$,

(iv) 4th pixels: $\sqrt{(9-10)^2+(13-9)^2}=\sqrt{15}$.

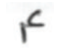
Therefore, the EMD of two spots is $\sqrt{15}+\sqrt{15}+\sqrt{15}+\sqrt{15}=4\sqrt{15}=15.5$.

In the EMD metric, Euclidean distances between all weighted point sets are computed and then the minimum distance between each pair of point sets can be found. There are different methods to solve this type of weighted matching problems; in our case we use the "Hungarian" method [16–19]. This method finds the minimum distances between each pair of points in two images with n points in $O(n^3)$ arithmetic operations; therefore, the typical EMD is very time consuming, which is the biggest drawback for EMD. Another drawback is that when two weighted point sets have unequal total weights, EMD is not an appropriate metric; however,

TABLE 2: Number of vector elements in four methods.

In L_1-vector (EEMD methods)															
	G_1		G_2		G_3		G_4		G_5		G_6		G_7		
	↓		↓		↓		↓		↓		↓		↓		
Elements:	4096	+	1024	+	256	+	64	+	16	+	4	+	1	=	**5461**
In sample and sketch vector															
Elements:	10% of L_1 vector = **546**														
In DREAT vector															
	10% of G_1		10% of G_2		10% of G_3		10% of G_4		10% of G_5		10% of G_6		10% of G_7		
	↓		↓		↓		↓		↓		↓		↓		
Elements:	409	+	102	+	25	+	6	+	1	+	4	+	1	=	**548**

TABLE 3: Some samples of handwritten Persian digit images.

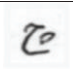
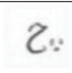
Class	No. of images	Digit	Correct shape	Samples				
				1	2	3	4	5
Class 1	432	Zero	۰					
Class 2	1500	One	۱					
Class 3	1067	Two	۲					
Class 4	256	Three	۳					
Class 5	173	Four (1)	۴					
Class 6	22	Four (2)	٤					
Class 7	180	Five	۵					
Class 8	138	Six (1)	۶					
Class 9	31	Six (2)	٦					
Class 10	713	Seven	۷					
Class 11	150	Eight	۸					
Class 12	657	Nine	۹					

it is desirable for robust matching to allow point sets with different total weights and cardinalities [18]. On the other hand, approximation is a good idea because usually exact computation is practically infeasible and an approximate solution can help to find the exact solution more efficiently.

In implementing EMD, in order to embed two sets of contour features with different total weights, we simulate equal weights by eliminating the appropriate number of random points from the larger weight set. For example, in Figure 2, when points are sampled uniformly from the contours of two images of Persian number 3 with a size of 64 × 64 pixels, the first image has 124 points, while the second image has 131 points. Therefore, the first image has 13 more points than the second one, and 13 points are randomly chosen from its contour to be eliminated.

The next part of the application is implementing embedded EMD into L_1. We formally show how to construct an embedded EMD into L_1. A boundary of $\sqrt{\log n}$ on any L_1 embedding distortion has been defined [20], where n is the number of pixels in the width or height of image (width and height of image are equal). We embed the minimum weight matching of contour features into L_1 via the EMD embedding of [12, 21]. To embed EMD into L_1, we put bitmap image in a grid whose size is twice bigger than that of the original image and shift grid randomly upon the image. Afterwards, we map pixels of the new image (which are all 0 or 1) to elements of an array in a special orientation starting from the first pixel in the left-top bit of the image to its last pixel in the right-bottom bit. The rest of the array should be set after some computation. For example, in the embedding of a 16 × 16 image, G_1 is the first grid and it includes 256 elements, each of which has a side length equal to 1. The first 256 elements of the array are set with these elements. In the next step, we add each of the 4 neighbouring elements in G_1 and place the

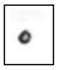
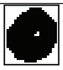
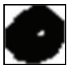
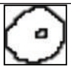
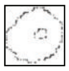
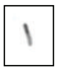
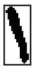
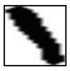
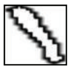
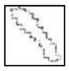
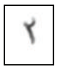
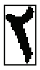
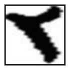
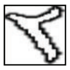
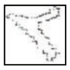
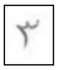
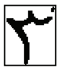
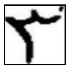
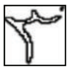
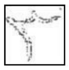
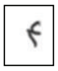
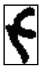
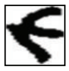
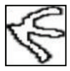
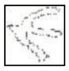
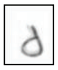
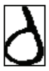
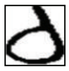
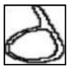
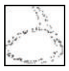
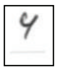
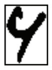
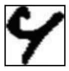
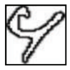
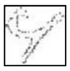
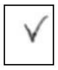
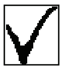
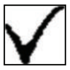
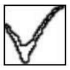
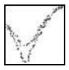
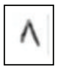
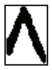
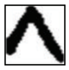
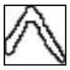
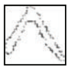
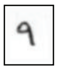
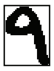
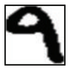
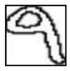
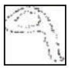
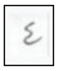
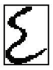
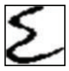
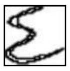
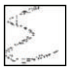
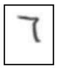
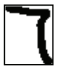
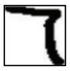
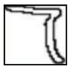
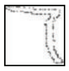
Table 4: Some samples of handwritten Persian letter images.

Letter	Correct shape	samples				
		1	2	3	4	5
Cha	چ					
Ja	ج					
Zha	ژ					
Za	ز					
Ta	ت					
Sa	ث					
Fa	ف					
Gha	ق					

```
function Calculate_$L_1$(image)
begin
    Scale image to $64 \times 64$
    Initialize an image matrix, original_img, of $64 \times 64$
    Initialize an image matrix, imageMtrx, of $(2 \times 64) \times (2 \times 64)$
    Set imageMtrx to original_img
    Initialize $U_1$, a random number between 0 and 64
    Initialize $V_1$, a random number between 0 and 64
    Shift each position in imageMtrx to position $(U_1, V_1)$
    Set $G_1$ array to pixels of imageMtrx
    //Create $L_1$_vector of image matrix including $G_1$ array followed by $G_2$ array,...,
        followed by $G_n$ array (as in Figure 3)
    Initialize $L_1$_vector to null
    $L_1$_vector = $G_1$
    for $i = 1$ to 6
        Set $G_{i+1}$ to sum of each 4-neighbour elements of $G_i$ multiplied by side length
        $L_1$_vector = $L_1$_vector + $G_{i+1}$
    end for
    return $L_1$_vector
end
function Calculate_Embedded EMD(image 1, image 2)
begin
    Initialize $L_1$_vector 1
    Initialize $L_1$_vector 2
    Initialize EEMD
    Set $L_1$_vector 1 to Calculate_$L_1$(image 1)
    Set $L_1$_vector 2 to Calculate_$L_1$(image 2)
    Subtract each pair of corresponding elements of $L_1$_vector 1 and $L_1$_vector 2
    Add all subtractions into EEMD and display it
end
```

Pseudocode 1

Table 5: Preprocessing of some test images.

Original image	1st process Crop white margin		2nd process Resize image	3rd process Get contour		4th process Remove spots randomly
Width: 77 Height: 95			Width: 64 Height: 64	Black spots		Black spots: 150
		Width: 20, height: 23			284	
		Width:13, height: 35			246	
		Width: 18, height: 39			287	
		Width: 34, height: 41			265	
		Width: 22, height: 43			359	
		Width: 33, height: 50			438	
		Width: 29, height: 43			334	
		Width: 36, height: 42			292	
		Width: 29, height: 43			322	
		Width: 28, height: 39			342	
		Width: 25, height: 36			350	
		Width: 34, height: 43			272	

results in the corresponding elements of G_2, which will be the next 64 elements in the array. In the 3rd step, we add each of the 4 neighbouring elements in G_2 and place the results in the corresponding elements of G_3, which will be the next 16 elements in the array. We continue this process until we have just one element, G_5, which will be the last element of the array. In Figure 3, you can observe the embedding of an image's pixels to an array in L_1.

The length of array is the sum of all the grids' lengths. So, in our example length of the array is 341 which is approximately equal to 2×16^2. As a result, length of the embedding vector is $2\Delta^2$. Table 1 shows the relationship between the size of an image, side length, and the number of each grid's elements in an array.

Pseudocode 1 describes the embedded EMD technique.

Finding the EMD of two images with this method has a complexity of $O(n^2)$, because, for the mapping to L_1, the vectors are of length $O(n^2)$. Therefore, finding the L_1 of two vectors, that is, vector A and B as in Figure 4, can be done in $O(n^2)$, which is better than $O(n^3)$ in exact EMD. The L_1 mapping is defined as

$$L_1(A,B) = \left|A_0 - B_0\right| + \left|A_1 - B_1\right| + \cdots + \left|A_i - B_i\right| + \cdots + \left|A_{2x}^2 - B_{2x}^2\right|. \quad (1)$$

Note that the exact EMD has a complexity of $O(n^3)$, which is the complexity of the Hungarian algorithm used for its implementation, and that the embedding EMD to L_1, which computes an approximation instead of the exact EMD, reduces the complexity to $O(n^2)$. We propose two techniques

```
function Calculate_Sampling(image 1, image 2)
begin
    Initialize $L_1$_vector 1
    Initialize $L_1$_vector 2
    Initialize sampling_$L_1$_Vector 1
    Initialize sampling_$L_1$_Vector 2
    Initialize sampling_EMD
    Set $L_1$_vector 1 to Calculate_ $L_1$(image 1)
    Set $L_1$_vector 2 to Calculate_ $L_1$(image 2)
    Select 10% indexes of $L_1$_vector 1 randomly
    Put the elements of selected indexes of $L_1$_vector 1 into sampling_$L_1$_Vector 1
    Put the elements of selected indexes of $L_1$_vector 2 into sampling_$L_1$_Vector 2
    Subtract each pair of corresponding elements in sampling_$L_1$_Vector 1 and sampling_$L_1$_Vector 2
    Add all subtractions into sampling_EMD and display it
end
```

Pseudocode 2

```
function Calculate_Sketching (image 1, image 1)
begin
    Initialize $L_1$_vector 1
    Initialize $L_1$_vector 2
    Initialize Sketching Vector Length as 10% of $L_1$_vector 1 length
    Initilize Sketching_Mtrx as $L_1$_vector 1 length $x$ Sketching Vector Length randomly
    // Sketching_Mtrx is as in Figure 6
    Initialize sketching_Vector 1
    Initialize sketching_Vector 2
    Initialize sketching_EMD
    Set $L_1$_vector 1 to Calculate_$L_1$(image 1)
    Set $L_1$_vector 2 to Calculate_$L_1$(image 2)
    for $i = 1$ to Sketching Vector Length
        Multiply each pair of corresponding elements in row $i$ of Sketching_Mtrx and $L_1$_vector 1
        Put the sum of multiplications in sketching_Vector 1
        Multiply each pair of corresponding elements in row $i$ of Sketching_Mtrx and $L_1$_vector 2
        Put the sum of multiplications in sketching_Vector 2
    end for
    Subtract each pair of corresponding elements in sketching_Vector 1 and sketching_Vector 2
    Add all subtractions into sketching_EMD and display it
end
```

Pseudocode 3

to reduce the complexity of EMD to *O(n)* by using dimension reduction in the L_1, sampling, and sketching. Concept of the dimension reduction technique from n to predetermined N-dimensional space is based on linear transformation, for example, elements of transformation 2-dimensional matrix A to a 1-dimensional vector [22].

Sampling is an option for dimension reduction in any norm (e.g., L_1 or L_2). In fact, using this technique, distances in L_1or $L_2$from random samples can be estimated by a simple scaling [13, 22]. Although it is a simple and popular method to approximate distances, it does not guarantee accuracy. In this method, as it is shown in Figure 5, we randomly pick k (out of D) columns from the image matrix A and image matrix B. We subtract them and set the results as a corresponding element in the sample vector. Finally, we sum all of the elements of the sample vector and call the result the sampling EMD of two images A and B.

In order to get the best or, at least, near to the EEMD method, we tested different sampling rates, for example 5%, 20%, 30%, and above the whole vector. Finally, we found that 10% is the best sampling rate. Therefore, we randomly select 10% of elements from L_1 vector that will generate just 546 elements.

Sampling EMD is displayed in Pseudocode 2.

Sketching is another option for dimension reduction. In this method, after scanning the data, we multiply the original data of image matrix A and image matrix B by a random matrix R which has either a 0 or 1 for each element, and the subtraction of the resulting matrices forms one element of the sketch vector. We repeat this step k times. The sum of all

```
function Calculate_DREAT(image 1, image 2)
begin
    Initialize $L_1$_vector 1
    Initialize $L_1$_vector 2
    Initialize index Vector
    Initialize DREAT_Vector 1
    Initialize DREAT_Vector 2
    Initialize DREAT_EMD
    Set $L_1$_vector 1 to Calculate_$L_1$(image 1)
    Set $L_1$_vector 2 to Calculate_$L_1$(image 2)
    For $i = 1$ to 7
    Select 10% indexes of $G_i$ randomly
    Put the selected indexes in index Vector
    end for
    Select all elements of $L_1$_vector 1 whose indexes are in index Vector
    Put the elements in DREAT_Vector 1
    Select all elements of $L_1$_vector 2 whose indexes are in index Vector
    Put the elements in DREAT_Vector 2
    Subtract each pair of corresponding elements in DREAT_Vector 1 and DREAT_Vector 2
    Add all subtractions into DREAT_EMD and display it
end
```

Pseudocode 4

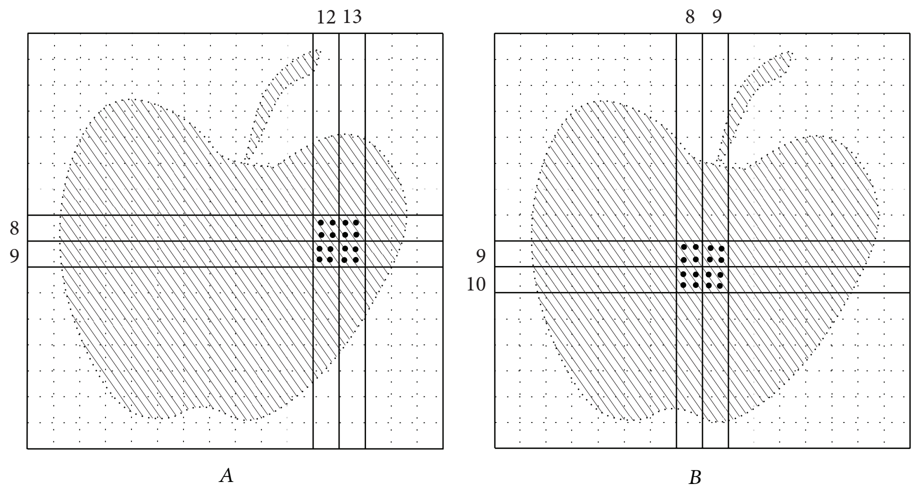


Figure 1: Two figures with high similarity.

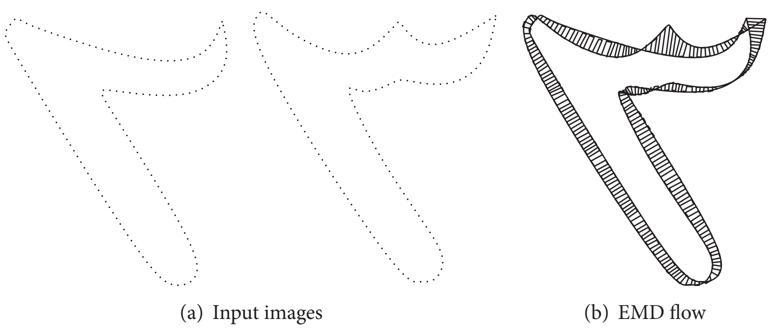


Figure 2: Computation of dissimilarity between two input images in Euclidean space and their corresponding EMD flow.

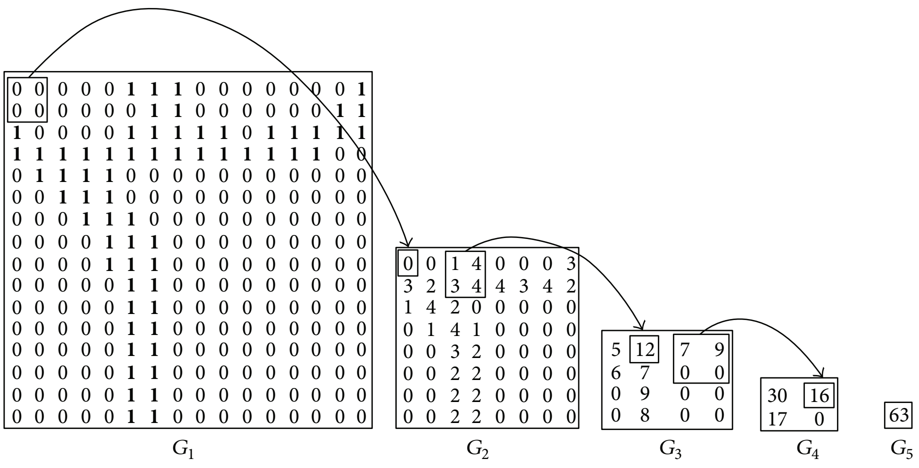


Figure 3: Mapping of a 16 × 16 image into a vector.

Table 6: Example of average precision calculation.

n	Doc no.	Relevance	Precision points
1	588	Yes	$P = 1/1 = 1$
2	589	No	
3	576	Yes	$P = 2/3 = 0.667$
4	590	No	
5	986	Yes	$P = 3/5 = 0.6$
6	592	No	
7	984	No	
8	988	Yes	$P = 4/8 = 0.5$
9	578	Yes	$P = 5/9 = 0.556$
10	985	No	

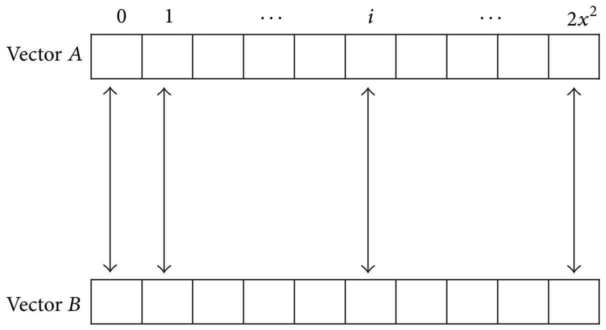


Figure 4: Computing $L_1(A, B)$.

elements of the sketch vector is what we call the sketching EMD. Sketching method is illustrated in Figure 6.

Pseudocode 3 shows sketching method.

3.2. Procedures of DREAT. Based on the sampling and sketching experiments, the images' L_1 vectors are heavily tailed where there are many zero elements in former grids and many nonzero elements in latter grids. In the sampling method, we choose samples of the L_1 vector and apply EMD to the samples instead of the whole vector; therefore, the execution time is reduced. However, the problem is that all elements of vector are sampled at the same rate.

When we go through the vector, most data in the initial sections, such as G_1 and G_2, contain almost all zeros when compared with the latter sections, such as G_3, G_4, and G_5. We considered this fact to be a heavy-tailed vector. As a result, when we apply the sampling method, the vector might by chance contain almost all zeros, which is meaningless. That is the reason why we need to create a method that will select an equal portion of samples from each part of the grid instead of randomly sampling from the whole.

We called the proposed method as the Dimension Reduction in Embedding by Adjustment in Tail (DREAT), a method that hybrids both the sampling and sketching. For example, suppose we want to select 10% of a vector as a sample vector. In the original sampling method we randomly selected elements of the vector, but, in the DREAT method, we selected only 10% of the elements of each grid part, G_n. In this way, we can select the same portion of all parts of the vector, not only among early elements that have many zeros but also among latter elements with large numbers that are required for recognition.

Table 2 shows the comparison between the number of vector elements in sampling, sketching, and DREAT methods. In DREAT method, we select 10% elements from each grid part because it will produce a number of vector elements that are near to the ones produced by the sampling and sketching methods. Therefore, by using similar idea of adjusting the heavy-tailed vector that we used in the sketching method and combining it with the sampling method we can improve the accuracy without increase of the running time.

This can be expressed in Pseudocode 4.

4. Experiments

In this work, we tested 5 methods: exact EMD, embedded EMD, sampling, sketching, and DREAT. Our image dataset includes bitmap images from Amirkabir University of Iran [23]. The images are scanned from handwritten Persian letters

Table 7: Results of 5 methods.

	MAP	Percentage of first correct recognition						
		1st pos.	2nd pos.	3rd pos.	4th pos.	5th pos.	6th pos.	>6th pos.
Exact EMD	0.97	0.99	0.01	—	—	—	—	—
Embedded EMD	0.85	0.90	0.01	0.02	0.03	0.01	0.01	0.02
Sampling	0.59	0.58	0.09	0.02	0.05	0.03	0.01	0.23
Sketching	0.87	0.89	0.04	0.02	0.01	—	0.01	0.03
DREAT	0.91	0.91	0.06	0.02	—	—	—	0.01

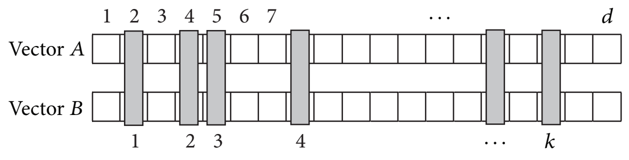


Figure 5: Sampling method on vector *A* and vector *B*.

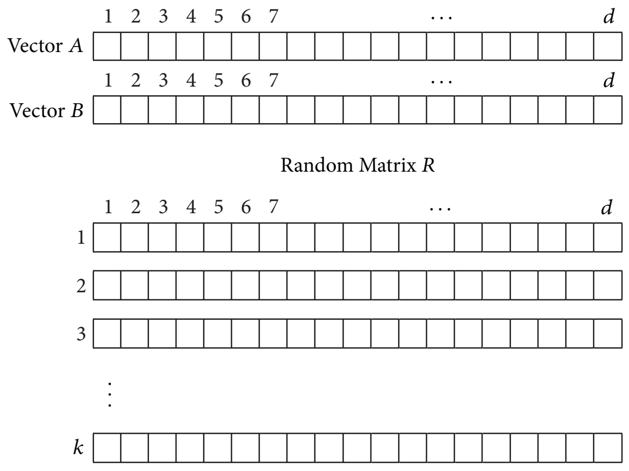


Figure 6: Sketching method on vector *A* and vector *B*.

and digits. The dataset includes 47 classes which are divided into two parts. The first part of the dataset consists of 35 classes including letter images and the second part of the dataset consists of 12 classes including digit images.

In the dataset, each image is named based on a combination of class number and running number. The first part of image name is its class number as listed in Table 3. Similarity is measured by comparing between the class number of a test image and the queried image. In our work, we only use the second part of the dataset that consists of 12 classes of 5319 handwritten Persian digit images. In Table 3, some of digit images are shown.

We did not use the letter images because Persian letters are very similar in handwritten shape even for a human reader. As some samples are shown in Table 4, letters that have two or three dots are similar to other handwritten letters with a dot. For example in the first sample of Table 4, letter "Cha" is very similar to letter "Ja" because the dots of letter "Cha" stick to each other and they look like one dot. Similarly, letter "Zha" is similar to letter "Za" in some cases. In this case, the similarity measurement will produce a high distortion. Since that is not a focus of this work, we excluded all the letters.

We divided our dataset into two parts: reference images and test images. The reference set includes 100 images that we randomly selected from the dataset and tested them on the rest of the dataset. In other terms, we remove these 100 reference images from test images' part. So, we will only find similar images to these reference images not exact ones. For each reference image, we applied the 5 methods and calculated EMD. We then computed the average precision (AP) and placed the results in a table. Finally, we computed the mean average precision (MAP) for all 100 reference images for each method; the results are shown in Table 5.

In each method, some preprocessing steps should be performed. The first step, which is common to all methods, is cropping the white margin of the image. Then, in the next step, the image should be resized to a particular size 64 × 64. As a result of pre-processing, we have some images with the same features.

In the first method, exact EMD, what we need is the contour of the image as well as the same number of spots for all of the images. Therefore, 2 additional steps are necessary in this method: getting the contour of the image and removing the additional spots randomly until the same number of spots is achieved, which should be 150 spots.

However, for the methods of embedded EMD, sampling, and sketching, we need the whole image not just its contour. Therefore, preprocessing steps in these methods are only up to the second step. Preprocessing of some of the images is illustrated in Table 5.

5. Results

We computed the mean average precision (MAP) values for the results of 5 different methods applied to 100 query images. Average precision (AP) is the average of the precision values at the points at which each relevant document is retrieved. Precision is defined as

$$\text{Precision} = \frac{\text{number of relevant documents retrieved}}{\text{total number of documents retrieved}}. \tag{2}$$

For example in Table 6 it can be clearly seen that 5 out of 10 documents are relevant; thus, the AP is computed as

$$\text{AP} : \frac{(1 + 0.667 + 0.6 + 0.5 + 0.556)}{5} = 0.665. \tag{3}$$

Table 8: Time execution of 3 reference images and estimation for 100 images.

	Image query 1:	Image query 2:	Image query 3:	Average for 100 images	Total for 100 images
	mm : ss	mm : ss	mm : ss	mm : ss	hh : mm
Exact EMD	22 : 45	23 : 11	22 : 57	22 : 57	36 : 41
Embedded EMD	9 : 18	9 : 51	9 : 39	9 : 36	16 : 00
Sampling	9 : 10	9 : 13	9 : 11	9 : 11	15 : 18
Sketching	11: 06	11 : 20	11 : 14	11 : 13	18 : 41
DREAT	8 : 45	9 : 09	9 : 02	8 : 58	14 : 56

Information retrieval systems are frequently judged by their mean average precision (MAP). MAP is the average of the average precision values for a set of queries; it is a performance evaluation measure of information retrieval. Using this measure, we are able to retrieve top-ranked images that are mostly relevant.

The results of all of our experiments are presented in Tables 7 and 8. We compute the AP, which is the average precision of relevant retrieved images among the 10 top-ranked images of the test image sets. In Table 7, the average AP of 100 test images for each method is in the second column. In columns 3 to 8 the percentages of first correct recognition in the first to sixth positions are shown. In the last column, the percentages of first correct recognition in the seventh position and beyond are shown.

In Table 8, the execution times of 5 methods for 3 randomly selected test images are shown. In the fifth column we estimate an average execution time, and in the last column we estimate the execution time for 100 test images for each method. As can be seen in this table, exact EMD has the highest execution time, and embedded EMD reduces this time by half. We can reduce this time by using our proposed methods. The last method, DREAT, achieves the lowest execution time.

6. Conclusion

DREAT is a method that hybrids both the sampling and sketching. In this paper, it shows its usefulness in dimension reduction of sparse and heavy-tailed data. As can be seen in the results, the exact EMD has a MAP value of 0.97; the MAP value is the average of relevant retrieved images among the 10 top-ranked images of 100 images. Although this method is excellent for measuring image similarity, its execution time is very high. By using embedded EMD, a MAP value of 0.85 can be achieved in half the time of exact EMD. Our first proposed method, sampling, reduces the time of execution, but it achieves the poorest MAP value of 0.59. Our second method, sketching, improves the MAP to 0.87 by sacrificing the execution time. Our last method, DREAT, has the lowest execution time and produces one of the best MAP values, which is 0.91.

In general, the results show that dimension reduction techniques like those in this paper, are useful for improving the processing time and matching. DREAT, especially, combines sketching and sampling where it converts sketches of the data into conditional random samples online in the estimation stage, with the sample size being determined retrospectively. The improvement to the EEMD is useful for overcoming problems with heavily tailed feature vectors.

References

[1] J. K. K. Samuel, *Lower Bounds for Embedding the Earth Mover Distance Metric into Normed Spaces*, Massachusetts Institute of Technology, Cambridge, Mass, USA, 2005.

[2] Y. Rubner, C. Tomasi, and L. J. Guibas, "Earth mover's distance as a metric for image retrieval," *International Journal of Computer Vision*, vol. 40, no. 2, pp. 99–121, 2000.

[3] J. Xu, Z. Zhang, A. K. H. Tung, and G. Yu, "Efficient and effective similarity search over probabilistic data based on Earth mover's distance," *The VLDB Journal*, vol. 21, no. 4, pp. 535–559, 2012.

[4] S. Peleg, M. Werman, and H. Rom, "Unified approach to the change of resolution: space and gray-level," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 739–742, 1989.

[5] D. Mumford, "Mathematical theories of shape: do they model perception," in *Geometric Methods in Computer Vision*, vol. 1570 of *Proceedings of SPIE*, pp. 2–10, International Society for Optics and Photonics, San Diego, Calif, USA, 1991.

[6] Y. Rubner and C. Tomasi, "Texture metrics," in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, pp. 4601–4607, October 1998.

[7] S. Cohen and L. Guibas, "Earth mover's distance under transformation sets," in *Proceedings of the 7th IEEE International Conference on Computer Vision (ICCV '99)*, pp. 1076–1083, September 1999.

[8] P. Giannopoulos and R. Veltkamp, "A pseudo-metric for weighted point sets," in *Proceedings of the Computer Vision-ECCV*, pp. 89–114, 2002.

[9] K. Grauman and T. Darrell, "Fast contour matching using approximate Earth mover's distance," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '04)*, vol. 1, pp. I220–I227, July 2004.

[10] R. Typke, P. Giannopoulos, R. C. Veltkamp, F. Wiering, and R. van Oostrum, "Using transportation distances for measuring melodic similarity," Tech. Rep. uu-cs-2003-024, Computer Science Department, University of Utrecht, Utrecht, The Netherlands, 2003.

[11] R. Batra and L. Hesselink, "Feature comparisons of 3-D vector fields using Earth mover's distance," in *Proceedings of the IEEE*

Visualization '99, pp. 105–114, IEEE Computer Society Press, October 1999.

[12] P. Indyk and N. Thaper, "Fast image retrieval via embeddings," in *Proceedings of the International Conference on Computer Vision (ICCV '03)*, 2003.

[13] P. Li, T. Hastie, and K. W. Church, *Procedures for Dimension Reduction in l1*, Department of Statistics, Stanford University, 2006.

[14] P. Li and K. W. Church, "Using sketches to estimate associations," in *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pp. 708–715, Association for Computational Linguistics, October 2005.

[15] P. Li, K. W. Church, and T. J. Hastie, "Conditional random sampling: a sketch-based sampling technique for sparse data," in *Advances in Neural Information Processing Systems*, vol. 19, pp. 873–880, 2007.

[16] K. Steiglitz and C. H. Papadimitriou, *Combinatorial Optimization: Algorithms and Complexity*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1982.

[17] C. H. Papadimitriou and U. V. Vazirani, "On two geometric problems related to the travelling salesman problem," *Journal of Algorithms*, vol. 5, no. 2, pp. 231–246, 1984.

[18] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*, Dover Publications, Prentice-Hall, Englewood Cliffs, NJ, USA, 1998.

[19] J. S. Cope and P. Remagnino, "Utilizing the hungarian algorithm for improved classification of high-dimension probability density functions in an image recognition problem," in *Advanced Concepts for Intelligent Vision Systems*, pp. 268–277, Springer, New York, NY, USA, 2012.

[20] A. Naor and G. Schechtman, "Planar earthmover is not in L_1," *SIAM Journal on Computing*, vol. 37, no. 3, pp. 804–826, 2007.

[21] W. Hong and T. S. Chen, "A novel data embedding method using adaptive pixel pair matching," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 1, pp. 176–184, 2012.

[22] S. Łukasik and P. Kulczycki, "An algorithm for sample and data dimensionality reduction using fast simulated annealing," in *Advanced Data Mining and Applications*, vol. 7120 of *Lecture Notes in Computer Science*, pp. 152–161, 2011.

[23] H. Khosravi and E. Kabir, "Introducing a very large dataset of handwritten Farsi digits and a study on their varieties," *Pattern Recognition Letters*, vol. 28, no. 10, pp. 1133–1141, 2007.