

Nonnegative Matrix Factorization with Earth Mover's Distance Metric

Roman Sandler and Michael Lindenbaum
Computer Science dept. Technion,
Haifa 32000, Israel
{romats, mic}@cs.technion.ac.il

Abstract

Nonnegative Matrix Factorization (NMF) approximates a given data matrix as a product of two low rank nonnegative matrices, usually by minimizing the L_2 or the KL distance between the data matrix and the matrix product. This factorization was shown to be useful for several important computer vision applications.

We propose here a new NMF algorithm that minimizes the Earth Mover's Distance (EMD) error between the data and the matrix product. We propose an iterative NMF algorithm (EMD NMF) and prove its convergence. The algorithm is based on linear programming. We discuss the numerical difficulties of the EMD NMF and propose an efficient approximation.

Naturally, the matrices obtained with EMD NMF are different from those obtained with L_2 NMF. We discuss these differences in the context of two challenging computer vision tasks – texture classification and face recognition – and demonstrate the advantages of the proposed method.

1. Introduction

The Nonnegative Matrix Factorization (NMF) is a representation of a nonnegative matrix as a product of two nonnegative matrices. The factorization becomes useful and interesting when the multiplied matrices are of low rank, implying usually that the factorization is approximate. In this case, the decomposition is useful for signal representation as an additive combination of a small number of atomic signals (part-based representation). The factorization and the first algorithm for finding it were introduced by Paatero [18]. An efficient multiplicative update algorithm was proposed by Lee and Seung [12, 13]. Different aspects of this latter algorithm were analyzed and many improvements were proposed [2, 6, 9, 8, 5, 26]. The NMF technique has been applied to many applications in the fields of object and face recognition, action recognition, and segmentation [26, 23, 21].

The basic algorithm proposed by Lee and Seung [13]

gets a matrix H^* and tries to find a pair of low rank matrices H and W satisfying

$$\min_{H,W} \text{Dist}_\phi(H^*, HW), \quad (1)$$

where the distance ϕ is either the Frobenius norm or the Kullback-Leibler distance. Although these distances have nice mathematical properties (e.g., bounded reconstruction error for the Frobenius norm [8]), they are not always the best choice for signal comparison. Therefore, some variations, adding a bias to desirable properties such as locality, were suggested [16, 11]. The obvious nonuniqueness of the factorization was also discussed [6, 2] and usually resolved by some problem specific bias.

In this work we propose to factorize the given matrix using a different metric: the Earth Mover's Distance (EMD) [19]. That is, we consider here the minimization (1) where ϕ is the EMD metric. The EMD was proposed by Werman et al. [25] and generalized by Rubner [19]. As demonstrated in many studies (e.g., [25, 19, 15, 7]), the EMD should be preferred for many signal comparison tasks, where the error mechanism is not modeled well by additive noise but is rather a complex local deformation of the original signal or the signal descriptor.

Unlike many recent contributions, the NMF algorithm for this metric differs greatly from the multiplicative update algorithm [12] and its variations [2]. It is based on linear programming steps, and as such is more closely related to the techniques presented in [9]. In the first part of the paper we formulate the minimization task, propose the LP based algorithm, and suggest an efficient numeric scheme.

We then examine two vision tasks that benefit from the proposed factorization: extracting the descriptors of individual textures from multiple texture images and learning a database for face recognition from unaligned facial images with pose changes and different facial expressions. In both cases we consider naturally deformed samples of a signal and reconstruct parts which appear to be the meaningful original signals.

The main contributions of this paper are:

1. A new type of NMF problem, NMF with EMD metric, is stated. The problem is solved using a linear-programming based iterative algorithm.
2. An efficient WEMD-based [22] mechanism that significantly accelerates the linear-programming based algorithm is proposed.
3. New approaches, based on the proposed factorization, are suggested for two challenging computer vision problems:

Unsupervised texture classification, for which the EMD NMF reconstructed both texture descriptors and mixing coefficients much better than analogous L_2 based algorithm.

Face recognition, for which it is shown to find meaningful basis face components, and outperformed the previous state-of-the-art NMF based algorithms using much smaller bases.

This paper continues as follows: the formal definition of NMF with the EMD metric as well as the linear programming based algorithm are presented in section 2. Methods for implementing the proposed factorization are discussed in section 3. Experiments with two actual vision tasks are discussed in section 4.

2. EMD NMF

Consider M nonnegative signatures¹ of length N . The signatures are represented in a matrix form, $H^* \in \mathfrak{R}^{N \times M}$, where the i -th signature is the column H_i^* . The matrix H^* may be decomposed into a product of $H \in \mathfrak{R}^{N \times K}$ and $W \in \mathfrak{R}^{K \times M}$, where H is interpreted as a basis of K signatures and W contains the corresponding coefficients. In most cases, however, a low dimensional approximation is more meaningful than exact factorization. Then, the desired factorization H, W , is a solution of

$$\min_{H, W} \|H^* - HW\|_{EMD} \quad (2)$$

for small K value.

2.1. Earth Mover's Distance

The EMD is motivated by the following intuitive observation: The distance between two signatures which may be considered as small local deformations of each other should be less than that of other signature pairs which differ in non-neighboring bins. Therefore, an intuitive metric would be some sum of the changes required to transform one signature into the other with low cost given to local deformations

¹Following [19], we use the term signature to denote a signal descriptor. A histogram is a simple case of a normalized signature, but a normalized signal descriptor (as used in this paper) is not necessarily a histogram.

and high cost to nonlocal ones. Formally, the EMD distance between two signatures is formulated as a linear program (3, 4) which aims to minimize the flow $f(i, j)$ between the bins of the source signature (i) and the bins (j) of the target signature for a given inter-bin flow cost $d(i, j)$; see [19]. The cost parameter $d(i, j)$ specifies the inter-bin flow cost for each pair of source and target bins. EMD is a metric when $d(i, j)$ is a metric as well; thus, we consider here only this type of cost function and denote it the underlying metric.

$$EMD(h^s, h^t) = \frac{\sum_{i,j} f(i, j)d(i, j)}{\sum_{i,j} f(i, j)}, \quad (3)$$

where $f(i, j)$ is a solution of:

$$\begin{aligned} \min_f \quad & \sum_{i,j} f(i, j)d(i, j) \\ \text{s.t.} \quad & f(i, j) \geq 0, \\ & \sum_j f(i, j) \leq h_i^s, \\ & \sum_i f(i, j) \leq h_j^t, \\ & \sum_{i,j} f(i, j) = \min \left(\sum_i h_i^s, \sum_j h_j^t \right). \end{aligned} \quad (4)$$

Here, the EMD between two matrices with M columns is defined as a sum of EMDs between each column in the source matrix and the corresponding column in the target matrix:

$$\|H^s - H^t\|_{EMD} = \sum_{m=1}^M EMD(H_m^s, H_m^t). \quad (5)$$

2.2. A two phase LP-based algorithm

The general problem of NMF is nonconvex and has a unique solution only for limited cases [6]. However, if one of the variable matrices H or W is given, the problem becomes linear. Thus, by consecutively fixing either H or W , one can find a local minimum for (1) by solving a sequence of convex tasks. This approach is also applicable to the case at hand by a simple reformulation of the EMD linear programming problem. As a result, the local minimum of EMD NMF is found by solving a sequence of linear programming tasks.

Consider $h^s = H_m^*$ and $h^t = (HW)_m$. Note that if each entry of h^t is larger than the relative entry in h^s , then the EMD becomes zero due to $d(i, i) = 0$ in any underlying metric. To avoid this trivial solution, we force $\sum_i h_i^s = \sum_j h_j^t$. To deal with the nonuniqueness of the solution, we also chose to normalize the columns of both H^* and H so that each column sums to 1. These two constraints imply that the columns of W sum to 1 as well.

With these normalizations, the linear programming constraints associated with the EMD between H_m^* and HW_m (eq. 4) become

$$\begin{aligned} f_m(i, j) &\geq 0, \\ \sum_j f_m(i, j) &= H^*(i, m), \\ \sum_i f_m(i, j) &= \sum_k H(j, k)W(k, m). \end{aligned} \quad (6)$$

Note that the constraint $\sum_{i,j} f_m(i, j) = 1$ is satisfied automatically since $\sum_{i,j} f_m(i, j) = \sum_i H^*(i, m) = 1$. Therefore the matrix EMD distance is just

$$\|H^* - HW\|_{EMD} = \sum_m \sum_{i,j} f_m(i, j)d(i, j). \quad (7)$$

Note that if we know H , both $f_m(i, j)$ and the matrix W minimizing it may be found as:

$$\arg \min_{f, W} \sum_m \sum_{i,j} f_m(i, j)d(i, j) \quad \text{s.t. (6)}. \quad (8)$$

Analogously, if we know W , we can find both $f_m(i, j)$ and the matrix H minimizing it as:

$$\arg \min_{f, H} \sum_m \sum_{i,j} f_m(i, j)d(i, j) \quad \text{s.t. (6)}. \quad (9)$$

Thus, given some initial guess for H or W , we can improve the solution by the following two phase Algorithm 1.

Algorithm 1 NMF EMD

Input: The objective matrix $H^* \in \mathfrak{R}^{N \times M}$ and an initial guess for the basis $H^0 \in \mathfrak{R}^{N \times K}$.

- 1: Find W^0 using (8).
- 2: $k = 0$
- 3: **repeat**
- 4: $k = k + 1$
- 5: Find H^k using (9).
- 6: Find W^k using (8).
- 7: **until**

$$\epsilon > \left| \|H^* - H^k W^k\|_{EMD} - \|H^* - H^{k-1} W^{k-1}\|_{EMD} \right|$$

Output: W^k and H^k .

2.3. Convergence

Theorem 1.1. *Algorithm 1 converges to a local minima*

Proof. 1. **Feasibility:** First note that Algorithm 1 is a sequence of LP processes. We should show that a feasible solution exists for every one of them. The minimization (8) gets a pair H^*, H^k of normalized matrices. Any normalized matrix W^k ensures that

$\sum_i H_m^* = \sum_j (HW)_m$ and thus implies that a feasible solution exists. This follows from EMD being a transportation problem, which has a feasible solution when $\sum_i h_i^s = \sum_j h_j^t$ [10]. An identical argument shows the existence of a feasible solution for minimization (9).

2. Linear programming, by definition, minimizes the flow cost and, due to (7), minimizes $\|H^* - HW\|_{EMD}$. Thus, applying (9) finds globally optimal H^k for a given W^{k-1} and applying (8) finds globally optimal W^k for a given H^k .
3. Since the objective in (9) and in (8) is the same, $\|H^* - H^k W^{k-1}\|_{EMD} \leq \|H^* - H^{k-1} W^{k-1}\|_{EMD}$ and $\|H^* - H^k W^k\|_{EMD} \leq \|H^* - H^k W^{k-1}\|_{EMD}$.
4. From the above it follows that every cycle of Algorithm 1 monotonically decreases the distance $\|H^* - H^k W^k\|_{EMD}$. This distance is lower-bounded, and therefore the algorithm converges (to a local minimum).

3. A more efficient algorithm

It is possible to find a local minima of (5) by iterative application of (9) and (8) starting from some reasonable guess for H . Linear programming is a well-studied problem and plenty of freeware and commercial solvers are available. However, for (9) the dimension of the problem is MN^2 . This means that even for a relatively small problem of factorizing 100 signatures of 256 bins (image in 16×16 resolution), the LP optimization problem operates about 6 million variables. This makes even the specification of the problem (construction of the constraint matrix) a challenging task with today's solvers.

Most of the variables arise from the need to calculate the flow $f_m(i, j)$ in order to estimate the EMD between the signatures. The actual variables of interest are H and W , which are only a small fraction of the variables in both (8) and (9).

3.1. A gradient based approach

The task of finding H^k and W^k in each step of Algorithm 1 is:

$$\begin{aligned} H^k &= \arg \min_H \sum_m EMD(H_m^*, (HW^{k-1})_m) \\ W_m^k &= \arg \min_W EMD(H_m^*, (H^k W)_m). \end{aligned} \quad (10)$$

Given both H and W , the error (7) can be calculated by solving M independent, relatively small LP problems. We can solve both minimizations in (10) with some gradient based optimization over possible H (or W) values. We are

guaranteed to find the globally optimal solutions for each optimization because tasks (8) and (9) are convex.

Unfortunately, the complexity of a single precise EMD computation is $O(N^3 \log N)$. Thus, the gradient based approach is expected to be complex as well.

3.2. WEMD approximation

Much effort was devoted to speeding up the EMD calculation. For some underlying metrics it is easier than for others. For example, the match distance [25], which is the EMD between 1D histograms with a specific underlying metric, can be calculated as an L_1 distance between the cumulative versions of the histograms. A short survey of other methods suggested for faster EMD calculation may be found in [22].

Shirdhonkar and Jacobs [22] proposed an efficient way to calculate the EMD between two signatures for some common underlying metrics $d(i, j)$. They proved that the result of optimization (4) is approximated very well by:

$$d(h^t, h^s)_{WEMD} = \sum_{\lambda} \alpha_{\lambda} |\bar{w}_{\lambda}(h^t - h^s)|, \quad (11)$$

where $\bar{w}_{\lambda}(h^t - h^s)$ are the wavelet transform coefficients of the n dimensional difference $h^s - h^t$ for all shifts and scales λ , and α_{λ} are scale dependent coefficients. The different underlying metrics are characterized by the choice of different scale weighting and different wavelet kernels. Note that we are looking for a local minima of some calculated EMD values and not for the EMD values themselves. Empirically we found that the local minima of EMD and WEMD are generally co-located, and thus the accuracy of the WEMD approximation of the actual EMD is less important for our goal.

Using the approximation (11) in (10) reduces the computational complexity of EMD to be linear. However, gradient methods naturally require knowledge of the gradient for the optimization variables. In the case of linear programming, the gradient may be found from the solution of the dual problem; therefore, it is a byproduct of EMD calculation. Unfortunately, for the WEMD we need to calculate the gradient separately. This gradient is:

$$\nabla d_{WEMD} = \sum_{\lambda} \alpha_{\lambda} \cdot \text{sign}(\bar{w}_{\lambda}(h^t - h^s)) \cdot \nabla \bar{w}_{\lambda}(h^t), \quad (12)$$

where the explicit expression for the gradient $\nabla \bar{w}_{\lambda}(h^t)$, with respect to either W or H , is lengthy but straightforward. The complexity of the gradient (12) computation for H is $O(N^2 K)$. Note, however, that many additives remain constant between the iterations and significant acceleration was gained by a smart calculation of the gradient.

3.3. Actual implementation

The two minimization problems (10) are not of equivalent complexity. W^k optimization is separable and may be solved for each column m separately with K variables. H^k is not separable and all $K \times N$ variables should be optimized at the same time.

We tested two optimization strategies. One is constrained optimization ($H \geq 0, W \geq 0$) of the distance (11). Another is unconstrained optimization with high penalty for negative variable values:

$$\arg \min_x \sum_m d(H_m^*, HW_m)_{WEMD} + \Phi(x), \quad (13)$$

where x is either W or H according to the relevant iteration and $\Phi(x)$ is a quadratic penalty term for $x < 0$. The latter unconstrained optimization appears to be more precise and faster.

Still, EMD NMF iterations are more complex than those of L_2 NMF. Using Matlab on Intel Core 2 Quad 2.5 GHz, one full H iteration for $M = 256, N = 32, K = 3$ (corresponding to the texture experiment described in section 4.1) takes around 30 seconds. One full H iteration for $M = 200, N = 1024, K = 40$ (corresponding to the face recognition experiment described in section 4.2) may take up to 20 minutes.

4. Experiments

We now turn to test the performance of the proposed algorithm in the context of two challenging tasks. The NMF is useful especially when the analyzed data is a mixture of data from several sources. The use of the EMD metric is preferable over L_2 when bin dependent changes in signatures are more likely than independent ones. Here we consider two typical examples.

4.1. Texture descriptor estimation

A texture mosaic contains several types of textures in random arrangements; see examples from [17] in Figure 1. We consider the task of estimating the texture descriptors associated with each texture class of the mosaic. We also would like to classify the textures in each mosaic location, at least roughly (e.g., for consecutive segmentation). To that end, we consider the texture in nonoverlapping square image patches (blocks). The texture in each block is a positive mixture of the basic textures. Therefore the NMF suggests itself as an analysis tool.

Unfortunately, the textures in the database [17] exhibit a lot of spatial variation. Even for relatively large blocks, the average texture descriptor in the block differs significantly from the average descriptor for the whole texture patch. Nor are the mosaics large enough to render descriptor distribu-

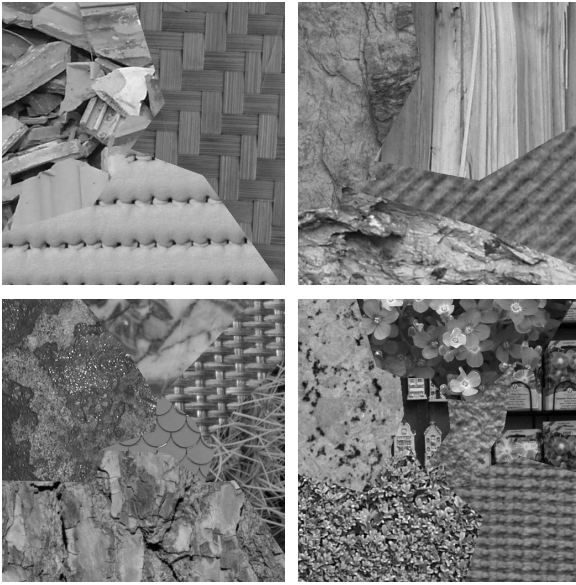


Figure 1. Examples of texture mosaics. The mosaic borders change randomly, resulting in random combinations of the textures in the sample rectangles. Here, the images contain 3, 4, 6, and 7 textures. Note the high local variability of the textures.

tion methods (e.g., [14]) effective. The EMD metric better compensates for the variability of the texture descriptor within the same texture than does L_2 [19, 3]. Therefore, EMD NMF is expected to be more accurate than L_2 NMF in estimation of the texture descriptors and the mixing coefficients thereof.

We assume that each texture class is associated with some vector descriptor h_i^{true} in each location of this texture. Then the K descriptors associated with a mosaic image are $H^{true} = (h_1^{true}, \dots, h_K^{true})$. Ideally, the mean texture descriptor in the j -th image block should be $h_j^* = H^{true} w_j^{true}$, where w_j^{true} is the vector of true fractions of the j -th block area associated with each texture class. Then, by taking very small blocks and applying any NMF algorithm, we would obtain both the class descriptors and the correct segmentation. As already mentioned, the actual descriptors vary a lot within their classes, and therefore $H w_j$ is only a rough approximation of h_j^* , making the fine segmentation proposition unrealistic.

We applied the NMF to the texture mosaics by:

1. Converting the image to some feature vector representation. Following the findings in [19], we chose to work with the Gabor features, and thus each location is represented by a feature vector of Gabor responses.
2. Dividing the image into M nonoverlapping rectangular blocks and calculating the mean feature vector h_j^* for each block. We denote all the sampled mean block descriptors $H^* = (h_1^* | \dots | h_M^*)$.

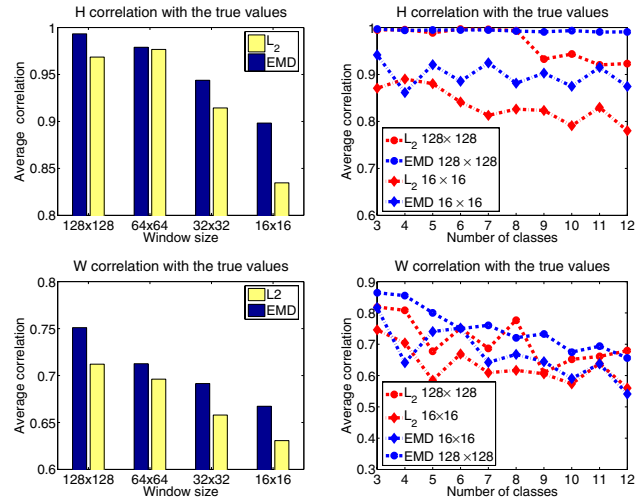


Figure 2. The accuracy of texture descriptor estimation. The first row shows the reconstruction quality of the basis descriptors and the second row shows the reconstruction quality of the mixing coefficients. The left column shows the average (over different K -s) reconstruction quality for the different sizes of the sampling blocks and the right column demonstrates the reconstruction quality as a function of the number of texture classes for two sizes of the sampling blocks.

3. Finding the factorization $H^* \approx HW$.

The results of the factorization are the approximated representative texture descriptors $H = (h_1 | \dots | h_K)$ and the approximated fraction of each texture in each block $W = (w_1 | \dots | w_M)$.

We used 90 online generated mosaics [17]. Each test was repeated for combinations of two parameters: the number of textures in the mosaic ($K = 3, \dots, 12$ textures) and the number of blocks $M = 16, 64, 256, 1024$ (number of columns in H^*). The blocks tessellate the image. Therefore, M specifies the block size $128 \times 128, 64 \times 64, 32 \times 32$, and 16×16 pixels respectively. In each test the K parameter was set to the number of texture classes in the image.

We compared the estimated H and W matrices with the actual matrices H^{true} and W^{true} using the following correlation measure:

$$Q_a(A, A^{true}) = \frac{1}{K} \sum_{i=1}^K \frac{\langle \vec{a}_i, a^{true}_i \rangle}{\|\vec{a}_i\| \|a^{true}_i\|}. \quad (14)$$

The estimated $Q_h = q(H, H^{true})$ and $Q_w = q(W^T, (W^{true})^T)$ values for the different test parameters are shown in Figure 2. The columns/rows are assigned to the respective ones in the true matrices by sequential greedy assignment, which maximizes Q_a . Note that as the block size increases, the descriptors H^* are evaluated over a bigger area and are thus more precise for both metrics.

The analysis of the graphs in Figure 2 points out two significant differences in the metrics' behavior. Both perform

comparably when a significant amount (64 samples) of relatively reliable (64×64 blocks) data is available. When the number of sample vectors is small or when the samples are less reliable, the performance of EMD NMF is much better than that of L_2 NMF. Note also that the performance of the H reconstruction does not depend on the number of classes with the EMD metric, but decreases with a larger K for the L_2 metric. This finding also supports the observation that EMD is more robust when ideal data is not available.

In addition to the mean of the column/row correlations (14), we also measured their standard deviation. We found that the EMD NMF is generally associated with much smaller (in 30-50%) standard deviation than the L_2 NMF. The intuitive explanation is that while the L_2 NMF estimations of H and W are either very accurate or very inaccurate, the EMD NMF estimations are generally more stable. Together with the average correlation results, this makes the EMD NMF estimations for both H and W more reliable than those of L_2 NMF.

4.2. Face recognition

Face representation is a common test case for the NMF algorithms [12, 12, 16, 26]. Traditional NMF algorithms measure the differences between the faces with translation-sensitive L_2 related metrics, and thus require a good alignment between the facial features. It was shown that when the NMF is forced to prefer spatially limited basis components, these L_2 based algorithms perform better and provide perceptually reasonable parts, especially for databases containing different poses [16, 11]. Here we show that the use of NMF with the EMD metric yields different, but still perceptually meaningful components. We found that these components are even more efficient for face classification.

4.2.1 The EMD NMF components

Unlike the L_2 distance, the EMD is not very sensitive to small misalignments, facial expressions, and pose changes. The basis components provided by the EMD NMF are facial archetypes, each of which looks like a slightly deformed face. Each facial feature (e.g., the shape of the head, the haircut, or the shape of the nose) associated with some archetype is shared by several people. The set of face images associated with the same person, and with different poses and expressions, are usually close (in the EMD sense) to a common facial prototype. This prototype is usually a convex combination of a small number of archetypes. In practice, every face image is a combination of a few archetypes with relatively high coefficients (the prototype) and some other archetypes with much lower coefficients.

To better illustrate this structure, we start by considering a simple image set of 4 faces: two parents, their daughter, and another, male, non-family member (six images of

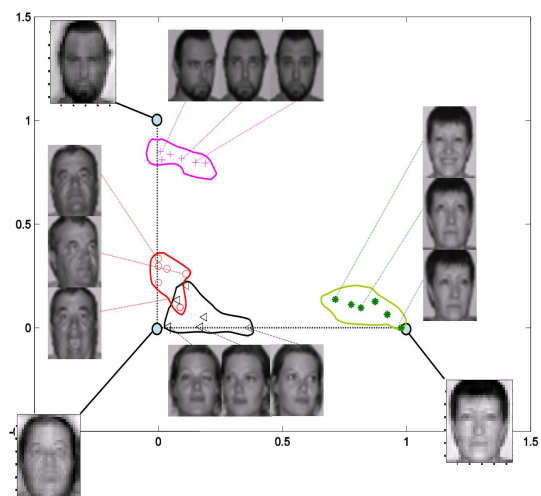


Figure 3. Facial space for 4 people. The two-dimensional (w_1, w_2) convex subspace is projected onto the triangle with corners in $(1, 0)$, $(0, 1)$, and $(0, 0)$. The corners of the triangle represent the basis face archetypes obtained by EMD NMF. The inner points show the actual facial images weighted in this basis.

each person; see examples in Figure 3). The people in the database share several features. The males have rougher facial features, while the female faces are smoother. The daughter shares facial features with both of her parents, especially with her father. A weight diagram associated with the EMD NMF ($K = 3$) of the set is shown in Figure 3. The 3 weights associated with every image and the EMD NMF may be plotted in 2D because $w_1 + w_2 + w_3 = 1$. See Figure 3, where the input faces are plotted as (w_1, w_2) points. The $k=3$ archetypes correspond to the $(1, 0)$, $(0, 1)$, and $(0, 0)$ points. The archetypes and some inputs images are shown as well. Note the similarity between the father (red circles) and the daughter (black triangles): both are represented mainly by the archetype in $(0, 0)$. However, the father shares some male facial features with the archetype in $(0, 1)$. The daughter, on the other hand, shares many facial features with her mother's archetype, located in $(1, 0)$. The significant changes in facial appearance caused by pose and expression are represented by insignificant translations in the obtained subspace.

Interestingly, the representation of visual objects as a combination of object-like archetypes was suggested as a plausible model for object recognition in the human visual system [4, 24].

4.2.2 Face recognition algorithm

We now describe a straightforward recognition algorithm, based on EMD-NMF and 1-NN in the coefficient space. Let $\{(I_j, C_j) \mid j = 1, \dots, L\}$ be the training set (I_j is an image, and C_j is the corresponding class label).

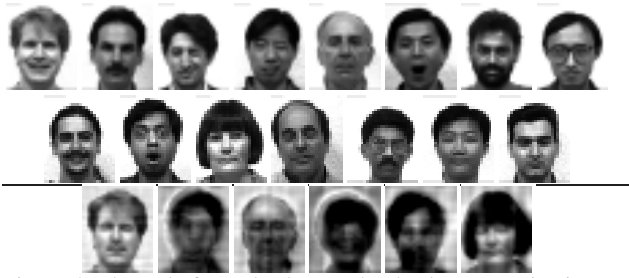


Figure 4. The Yale faces database. The database contains images of 15 people, and we considered 8 images for each person. The first two rows show examples of the database images. The last row shows the basis images obtained with EMD NMF.

Training:

Input: $\{(I_j, C_j) \mid j = 1, \dots, L\}$

- 1: Normalize every image I_j so that $\|I_j\|_1 = 1$.
- 2: Decompose the matrix I (with columns I_j), by EMD NMF, $I = HW$.
- 3: Normalize every column w_j so that $\|w_j\|_2 = 1$.

Output: H, W

Test:

Input: I_t, H, W .

- 1: Normalize the test image I_t so that $\|I_t\|_1 = 1$.
- 2: Approximate I_t as a convex combination of H 's columns, with weights

$$w_t = \arg \min_w EMD(I_t, Hw).$$
- 3: Normalize w_t so that $\|w_t\|_2 = 1$.
- 4: Find $j^* = \arg \max_j \langle w_j, w_t \rangle$.

Output: C_{j^*} .

4.2.3 Face recognition experiment

We tested the EMD NMF based recognition algorithm on the popular Yale [1] and ORL [20] face databases. We follow the experimental procedure of [26], so that we can relate our results to those compared in [26] using the ORL database. Therefore, the face images are downsampled so that their longer side is 32 pixels. Moreover, as observed in [26], the recognition performance depends to a small extent on the partition of the database into the training and test sets. Following [26] and the approaches cited there, we provide the best results obtained in several training/test partitions.

In contrast to [26], we did not tightly align the faces by forcing the eye positions to coincide. Both databases contained images that were only roughly aligned. We did not touch the ORL database and, in the Yale database, we only centered the faces. This was necessary to avoid a situation in which the identification might be significantly assisted by the position of the face.

The Yale face database contains fewer people than ORL, but is more challenging for recognition. We used a subset of it containing a set of images corresponding to the same

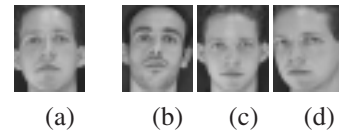


Figure 5. Typical recognition error in ORL database. When the test face image (a) is in a significantly different pose from those of the same person in the training set, the most similar person in the same pose (b) may be erroneously identified. The second-most similar identifications (c,d) are correct.

lighting direction. Even with this restriction, the recognition task is not easy due to the high variability of expressions and to the possible presence of glasses. This implies that even for the best partition of the database into training and test sets, the test faces always differ significantly from their closest training examples. Four images were used to represent every person in the training set. A relatively high recognition rate of 86.6% was achieved using only 6 basis archetypes (representing 15 people). The archetypes obtained in this test are shown in Figure 4 together with examples of the faces they represent. Increasing the number of archetypes to 15 (one per person) increased the recognition rate to 95%. All the misses are due to glasses appearing in the test image but not in the corresponding training images.

It is interesting to observe that the proposed algorithm does not behave like a nearest neighbor algorithm with EMD metric. When a representative archetype for each person was computed as the image minimizing the sum of EMD distances over the corresponding training images, and 1-NN (with EMD metric) was used for recognition, accuracy was only 73.3%. This advantage of the EMD NMF based algorithm could be predicted also from the weight diagram in Figure 3, where, clearly, the father's images are closer to the daughter's mean image than to his own mean image (in weight space) and can be recognized only by the additional components.

The ORL database contains images of 40 people and is somewhat easier. As in [26], five images were used to represent every person in the training set. The recognition accuracy naturally changes with basis size K . For K equal to or larger than the number of classes (people), the EMD NMF algorithm outperforms all the NMF based algorithms

Table 1. Classification accuracies of different algorithms on the ORL database and the corresponding basis sizes cited from [26].

Algorithm	NMF	LNMF	NGE	PCA	LDA	MFA
Basis Size	158	130	121	105	39	48
Accuracy (%)	74.0	87.5	95.5	85.5	94.5	95.5

Table 2. Classification accuracy of EMD NMF on the ORL database for different basis sizes.

Basis Size	2	5	8	10	20	30	40	50
Accuracy (%)	8.5	70.5	87.5	94.5	90.5	95.0	96.5	97.0

considered in [26], which often use much larger bases; see Table 1. Even with much lower basis dimension, the proposed algorithm achieves very high, competitive, accuracy.

Analyzing the (few) recognition errors, we found that they are associated with poses which are significantly different from those in the training set; see Figure 5.

5. Conclusions

A new type of NMF problem, NMF with EMD metric, is proposed. The problem is solved with a linear programming based iterative algorithm. A WEMD [22] based optimization technique is proposed for fast implementation of the proposed algorithm. Algorithms based on the proposed EMD NMF outperformed previous NMF based algorithms in the context of two challenging computer vision tasks.

It seems that the main advantage of the new approach is its enhanced robustness. Consider, for example, the task of identifying a set of basis descriptors from mixture measurements. When the given measurements closely approximate linear combinations of the hidden descriptors, then the L_2 NMF technique suffices to extract the basis with high accuracy. When the mixtures are, however, mixtures of deformed descriptors, this is no longer the case. Nonetheless, the deformed descriptors may be close, in the EMD sense, to the original descriptors. Then, the mixture of deformed descriptors is EMD close to the mixture of original descriptors (with the same weights). This lower sensitivity to deformations allows the EMD NMF to succeed when the L_2 NMF does not. Note that this situation is typical when we approximate a histogram from a small sample mixture.

6. Acknowledgements

This work was supported by the Israeli Science Foundation. We thank Dr. Michael Zibulevsky and Dr. Boris Bachelis for valuable discussions.

References

- [1] P. N. Bellhumer, J. Hespanha, and D. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *PAMI*, 17(7):711–720, 1997.
- [2] M. Berry, M. Browne, A. Langville, P. Pauca, and R. Plemmons. Algorithms and applications for approximate nonnegative matrix factorization. *Computational Statistics and Data Analysis*, 52(1):155–173, September 2007.
- [3] R. E. Broadhurst. Statistical estimation of histogram variation for texture classification. In *Texture Analysis and Synthesis Workshop, ICCV*, pages 25–30, 2005.
- [4] H. H. Bühlhoff and S. Edelman. Psychophysical support for a two-dimensional view interpolation theory of object recognition. In *PNAS*, volume 89, pages 60–64, January 1992.
- [5] I. Dhillon and S. Sra. Generalized nonnegative matrix approximations with Bregman divergences. In *NIPS*, volume 18, pages 283–290, 2006.
- [6] D. Donoho and V. Stodden. When does non-negative matrix factorization give a correct decomposition into parts? In *NIPS*, 2003.
- [7] K. Grauman and T. Darrel. Fast contour matching using approximate earth mover’s distance. In *CVPR*, volume 1, pages 220–227, 2004.
- [8] T. Hazan and A. Shashua. Analysis of l2-loss for probabilistically valid factorizations under general additive noise. Technical Report 2007-13, The Hebrew University, 2007.
- [9] M. Heiler and C. Schnörr. Learning sparse representations by non-negative matrix factorization and sequential cone programming. *J. Mach. Learn. Res.*, 7:1385–1407, 2006.
- [10] F. S. Hillier and G. J. Lieberman. *Introduction to Operations Research*. McGraw-Hill Science/Engineering/Math, 2005.
- [11] P. Hoyer. Non-negative matrix factorization with sparseness constraints. *J. Mach. Learn. Res.*, 5:1457–1469, 2004.
- [12] D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, October 1999.
- [13] D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. *NIPS*, 13:556–562, 2001.
- [14] T. Leung and J. Malik. Representing and recognizing the visual appearance of materials using three-dimensional tex-tons. *IJCV*, 43(1):29–44, June 2001.
- [15] E. Levina and P. Bickel. The earth mover’s distance is the mallows distance: some insights from statistics. In *ICCV*, volume 2, pages 251–256, 2001.
- [16] S. Li, X. Hou, H. Zhang, and Q. Cheng. Learning spatially localized, parts-based representation. In *CVPR*, pages 207–212, 2001.
- [17] S. Mikeš and M. Haindl. Prague texture segmentation data generator and benchmark, 2006.
- [18] P. Paatero and U. Tapper. Positive matrix factorization: a non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5(2):111–126, 1994.
- [19] Y. Rubner. *Perceptual Metrics for Image Database Navigation*. PhD thesis, Stanford University, 1999.
- [20] F. Samaria and A. Harter. Parameterisation of a stochastic model for human face identification. In *Proceedings of 2nd IEEE Workshop on Applications of Computer Vision*, Sarasota FL, December 1994. IEEE.
- [21] R. Sandler and M. Lindenbaum. Unsupervised estimation of segmentation quality using nonnegative factorization. In *CVPR*, pages 1–8, 2008.
- [22] S. Shirdhonkar and D. Jacobs. Approximate earth mover’s distance in linear time. In *CVPR*, pages 1–8, 2008.
- [23] C. Thureau and V. Hlavac. Pose primitive based human action recognition in videos or still images. In *CVPR*, pages 1–8, 2008.
- [24] S. Ullman. *High-level Vision: Object Recognition and Visual Cognition*. The MIT Press, Cambridge, MA, 1996.
- [25] M. Werman, S. Peleg, and A. Rosenfeld. A distance metric for multidimensional histograms. In *CVGIP*, volume 32, pages 328–336, 1985.
- [26] J. Yang, S. Yang, Y. Fu, X. Li, and T. Huang. Non-negative graph embedding. In *CVPR*, pages 1–8, 2008.