Incremental Learning in the Non-negative Matrix Factorization

Sven Rebhan, Waqas Sharif, Julian Eggert

2008

Preprint:

This is an accepted article published in Proceedings of the 15th International Conference on Neural Information Processing. The final authenticated version is available online at: https://doi.org/[DOI not available]

Incremental Learning in the Non-negative Matrix Factorization

Sven Rebhan[1], Waqas Sharif[2], and Julian Eggert[1]

[1] Honda Research Institute Europe GmbH
Carl-Legien-Strasse 30, 63073 Offenbach am Main, Germany
[2] Darmstadt University of Technology (Control Theory and Robotics Lab)
Landgraf-Georg-Strasse 4, 64283 Darmstadt, Germany

Abstract. The non-negative matrix factorization (NMF) is capable of factorizing strictly positive data into strictly positive activations and base vectors. In its standard form, the input data must be presented as a batch of data. This means the NMF is only able to represent the input space contained in this batch of data whereas it is not able to adapt to changes afterwards. In this paper we propose a method to overcome this limitation and to enable the NMF to incrementally and continously adapt to new data. The proposed algorithm is able to cover the (possibly growing) input space *without* putting further constraints on the algorithm. We show that using our method the NMF is able to approximate the dimensionality of a dataset and therefore is capable to determine the required number of base vectors automatically.

1 Introduction

The NMF has been introduced by Lee and Seung [1,2] as an unsupervised factorization method for decomposing multi-variant data under the constraint of non-negativity. By only allowing the additive combination of components, the method generated a parts-based representation. In its standard form, the NMF works as a batch algorithm, i.e. the whole dataset is presented at once and has to cover the desired input space entirely. Changes in that space can only be incorporated by restarting the learning process from scratch, using the new and the old data samples together to represent the new input space. This requires an enormous amount of memory and computational effort, because all data samples seen so far have to be stored, and the base vectors have to be recomputed once a new sample is presented. An additional drawback of the NMF and batch algorithms in general is that we have to specify the number of base vectors a priori. This poses a hard problem for many real-world datasets, because the intrinsic dimensionality of those datasets is not known.

Cao et al. developed an online variant of the NMF [3] trying to overcome the mentioned limitations. In their approach it is possible to add new data samples to the representation later, making it possible to adapt to temporally changing data. Nevertheless, it is necessary to put an orthogonality constraint on the learning process in order to obtain a proper representation of the input space.

M. Köppen et al. (Eds.): ICONIP 2008, Part II, LNCS 5507, pp. 960–969, 2009.

Furthermore, both the initial number of vectors and the number of additionally available vectors for new learning cycles still have to be specified beforehand.

We propose a method to learn the available data sequentially and hereby to incrementally adapt the base vectors to represent the input space. By doing so, the algorithm is able to autonomously approximate the intrinsic dimensionality of the input data, rendering it unnecessary to specify the number of base vectors in advance. By assuming to only see one new data sample at a time we, contrary to [3], do not need to put further constraints on the learning process.

In Sect. 2 we briefly review the standard NMF algorithm and its update procedure, before we describe the idea of the incremental NMF (iNMF) in Sect. 3. Then we present our results (Sect. 4) using the bar dataset and the Essex face94 dataset [4]. Finally, we discuss the results and give an outlook on future work.

2 Standard NMF Review

The NMF algorithm as described by Lee and Seung [1,2] is based on a distance measure between the input dataset $\mathbf{V}$ and the reconstruction $\mathbf{R}$. We here focus on the Euclidian distance measure

$$F(\mathbf{W},\mathbf{H}) = ||\mathbf{V} - \mathbf{R}||^2 \ , \tag{1}$$

where the reconstruction is calculated by the linear superposition of the base vectors $\mathbf{W}$ weighted with their corresponding activation $\mathbf{H}$

$$\mathbf{V} \approx \mathbf{R} = \mathbf{W}\mathbf{H} \ . \tag{2}$$

Lee and Seung have shown that one can derive the following multiplicative update rules for $\mathbf{W}$ and $\mathbf{H}$ to minimize the error function of Eq. 1[1]

$$\mathbf{H} \leftarrow \mathbf{H} \odot \frac{\mathbf{W}^T\mathbf{V}}{\mathbf{W}^T\mathbf{R}} \tag{3}$$

$$\mathbf{W} \leftarrow \mathbf{W} \odot \frac{\mathbf{V}\mathbf{H}^T}{\mathbf{R}\mathbf{H}^T} \tag{4}$$

By alternately performing the update steps for $\mathbf{H}$ and $\mathbf{W}$, the algorithm is able to find at least a locally optimal solution for Eq. 1 (see [2]). The final update schema for the NMF reads as follows:

1. Calculate the reconstruction according to Eq. 2.
2. Update the activations using Eq. 3.
3. Calculate the reconstruction according to Eq. 2.
4. Update the base vectors using Eq. 4.
5. Repeat step 1 to 4 until convergence.

[1] $\odot$ and $\div$ denote componentwise operations: $\mathbf{a} \odot \mathbf{b} := A_i \cdot B_i, \ \forall i$.

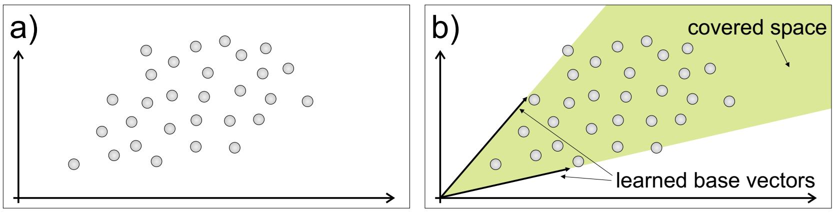

Fig. 1. While learning a dataset a), the base vectors are adapted to cover at least the space spanned by the data samples b)

The matrix $\mathbf{V}$ contains a set of data samples $\mathbf{v}_i$ as column vectors, representing the input space that should be covered by the base vectors (see Fig. 1a for an example). The available number of base vectors is mainly responsible for the learning result and must be defined a priori. If the number of the base vectors is greater or equal to the intrinsic dimensionality of the data, the NMF learns base vectors that cover at least the space spanned by the data samples (see Fig. 1b). If their number is too low, the NMF will not be able to derive sensible base vectors and will end up with a high reconstruction error for all input data. Another problem related to the NMF and batch algorithms in general is that it is impossible to adjust the representation of the input space to subsequent and new information, after the learning process took place.

3 Incremental NMF Algorithm (iNMF)

To overcome the mentioned limitations, we propose to incrementally extend the representation of the input space and enable the algorithm to add further base vectors if necessary. The learning process, as schematically described in Fig. 2, starts with only one base vector. In each learning cycle the incremental NMF(iNMF) only sees one data sample at a time. For the first data sample we perform a NMF learning (see section 2) with the single base vector. The resulting base vector, as shown in Fig. 2b, represents all points along its extension, lying on a straight line and hence covering a 1D-space. To start the next learning cycle using a new data sample $\mathbf{d}(t)$, we have to make sure that the already acquired information is preserved. Here we exploit the fact that the learned base vector represents all information about the previously seen data samples (see [5] and Eq. 2), which is the basic idea of our method. So all we need to do is to append the new data sample $\mathbf{d}(t)$ to the previously learned base vectors $\mathbf{W}(t-1)$ in order to obtain the new NMF input vector $\mathbf{V}(t)$

$$\mathbf{V}(t) = \{\mathbf{d}(t), \mathbf{W}(t-1)\} . \tag{5}$$

With the new $\mathbf{V}(t)$ we now perform a new learning cycle equivalent to the NMF update schema (see section 2). For the example shown in Fig. 2c, both the new sample and the previous base vector are already covered by the set of base vectors $\mathbf{W}(t-1)$, so there is no need for adaptation.

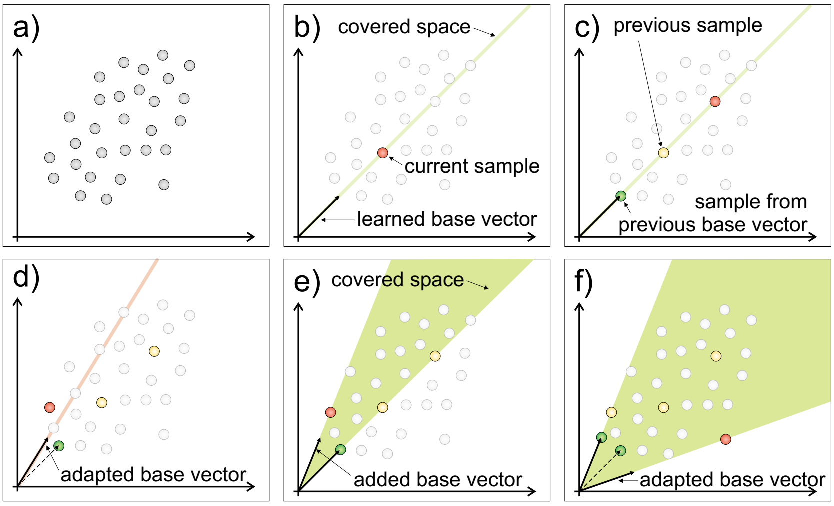

Fig. 2. The learning and adding of base vectors in the iNMF shown as: a) the available data samples. b) the learned base vector from the first data sample covering a 1D-space. c) a new data sample and the learned base vector as a virtual sample (containing the first sample shown). Both data points are inside the covered 1D-space, allowing for the reconstruction using a single base vector. d) the third data sample being outside the covered 1D-space. Even adapting the available base vector leads to a large reconstruction error. e) by adding and adapting a base vector both the virtual sample and the new data sample can be reconstructed. As a result a 2D-space is covered. f) by adapting the two base vectors a larger region of the 2D-space can be covered. Here the intrinsic number of dimensions (two) is found by the iNMF.

If we provide a data sample beyond the covered space, a reconstruction of both the new data sample and the previous base vector is not longer possible. In Fig. 2d this is due to the fact that the data samples span a 2D-space, but only a 1D-space can be covered with a single vector. Although the NMF tries to adapt the existing vector to minimize the reconstruction error $F(\mathbf{W}, \mathbf{H})$ (see Eq. 1), a high error will remain. We know that the NMF is able to reconstruct all data samples in an n-dimensional space if the number of base vectors is at least equal to the dimensionality of that space (see [5] and Fig. 2 for illustration). So a high reconstruction error indicates that the dimensionality of the space spanned by the data samples must be higher than the number of currently available base vectors. Because we provide the iNMF only with one new data sample at a time, we know that the dimensionality also must have increased by one and hence requires one additional base vector. By providing this vector it is possible to span a 2D-space and to reach a good reconstruction quality by repeating the NMF learning cycle (see Fig. 2e). The dataset used for our illustration (see Fig. 2a) is 2D, it will be possible to reconstruct all further data samples provided to the iNMF by adapting the two available base vectors as shown in Fig. 2f. By successively applying our method, the iNMF algorithm is able to approximate the dimensionality of the input data by itself. We can formulate the update schema for the iNMF in the following pseudo code:

1. Randomly initialize a single base vector $\mathbf{w}_0$ and set $\mathbf{W}(t-1) = \{\}$.
2. Take the next data sample $\mathbf{d}(t)$ and construct the input vector $\mathbf{V}(t) = \{\mathbf{d}(t), \mathbf{W}(t-1)\}$ as described in Eq. 5.
3. Set $\mathbf{W}(t) = \mathbf{W}(t-1)$ to initialize the base vectors.
4. Initialize the activation matrix $\mathbf{H}$ randomly.
5. Perform a NMF learning cycle:
   (a) Calculate the reconstruction according to Eq. 2.
   (b) Update the activations using Eq. 3.
   (c) Calculate the reconstruction according to Eq. 2.
   (d) Update the base vectors using Eq. 4.
   (e) Repeat step 5a to 5d until convergence.
6. If the reconstruction error $F(\mathbf{W}, \mathbf{H})$ (Eq. 1) is low, continue with step 2. Otherwise provide an additional base vector $\mathbf{W}(t) = \{\mathbf{W}(t-1), \mathbf{W}_{new}\}$ and a corresponding activation. Continue with step 4.

4 Results

4.1 Bar Dataset

The bar dataset we have used consists of 162 images with a size of 32x32 pixel. Each of the images is a superposition of up to four horizontal and vertical bars. A horizontal bar can be applied to four different horizontal positions; the vertical bar to four different vertical positions. Complete overlapping of two bars is not allowed. We permute the original dataset (see e.g. Fig 3) in order to destroy a potential order (from single bars to many bars) in the dataset and remove all images containing only a single bar. This prevents the iNMF from focusing on the

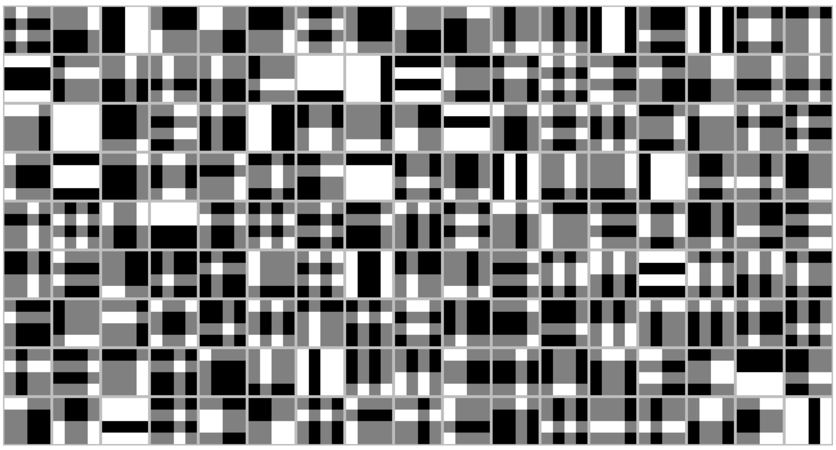

Fig. 3. Here you can see the 153 images of the permuted bar dataset

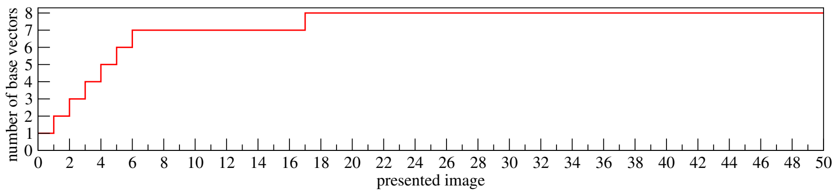

Fig. 4. After about 20 images, the iNMF reaches the final number of base vectors, which are adapted afterwards. Due to visibility reasons, we only show the first 50 images

Fig. 5. While provided with the first ten images (top), the iNMF algorithm generates seven base vectors (bottom). The first seven input images are simply stored one-to-one in the base vectors. After this phase, the base vectors are adapted and become less complex and sparser.

trivial cases where solely the initial eight vectors are sufficient to learn the input space. Figure 4 shows the massive necessity to add base vectors within the first 7 images. During this phase the algorithm copies the data samples one-to-one into base vectors as shown in Fig. 5. Afterwards, the number of vectors stays constant for about 10 input images. Here, the reconstruction of the input data is achieved by adapting the already existing vectors and thus removing redundancy in the set of base vectors. This can be interpreted as a specialization of the vectors, resulting in a lower complexity, lower redundancy and higher sparsity.

If we compare the base vectors after presenting 20 images with the final base vectors (see Fig. 6), we see that the final shape of the base vectors has already emerged. Merely the amplitude of some pixels is different. So already at this early point the input space is covered to a very large degree. We have tested ten different permutations of the bar dataset and found that the final number of vectors is reached between the 9th and 76th image. This also implies that the dimensionality of the input dataset was correctly estimated. In all cases the iNMF algorithm is able to find the correct base vectors out of the randomly

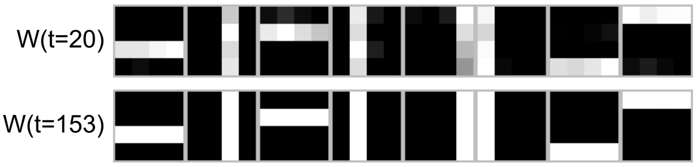

Fig. 6. A comparison between the vectors after 20 images (top) and the final vectors after 153 images (bottom) shows that the final shape of the base vectors has already emerged after 20 images

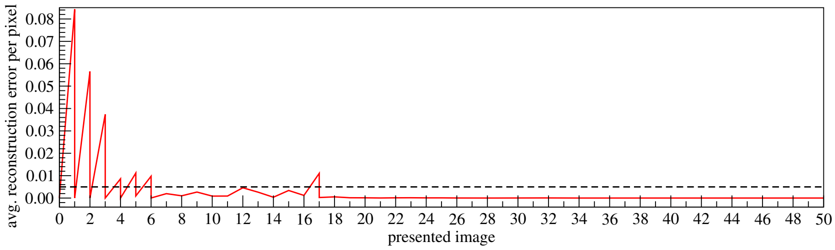

Fig. 7. The error function over the presented images shows characteristic peaks if the number of base vectors is insufficient. A vector is added if the error exceeds the threshold = 0.005 (dashed), increasing the reconstruction quality drastically. We only show the first 50 images here.

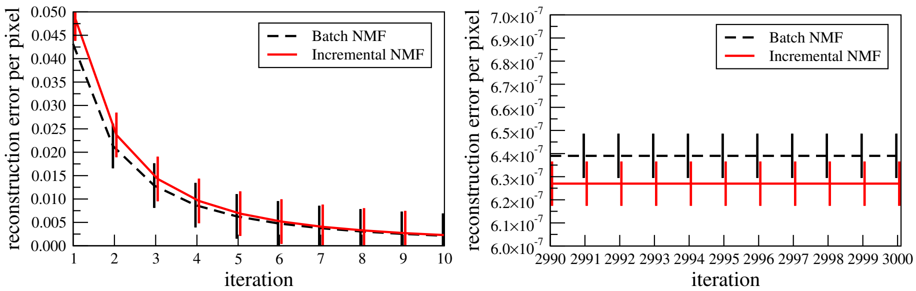

Fig. 8. Mean and standard deviation (scaled by 3 for visualization) in the first and last 10 steps of the bar dataset reconstruction. The iNMF and the NMF perform comparably good.

ordered samples that dot not contain the initial vectors. A typical development of the error function over the presented images is shown in Fig. 7. In the diagram one can see sharp rises of the reconstruction error, which are characteristic for an insufficient number of base vectors. If such a peak is detected, a new base vector will be added and we can observe a drastic drop in the reconstruction error. This results from the fact that the iNMF algorithm is now able to place the new base vector pointing into the direction of the current sample and thus leading to a nearly perfect reconstruction. In our case we have chosen a fixed threshold to detect the critical reconstruction error. The threshold itself can be varied in a large range (from 0.0001 to 0.01), assuring an easy handling.

4.2 Comparison to Batch NMF

To compare the iNMF to the NMF we apply both algorithms to the bar dataset (see Fig. 3). Afterwards we use the final base vectors of both algorithms to reconstruct the dataset to evaluate how much information is stored in the base vectors. During this phase we do not adapt the base vectors, but keep them fixed. Each algorithm is started with 10 randomly chosen initial activations. In Fig. 8 you can see the resulting mean and standard deviation comparison of the reconstruction quality between the NMF and the iNMF. As you can see, the incrementally learned base vectors of the iNMF exhibit a comparable reconstruction quality on the dataset with respect to the base vectors learned by the batch NMF algorithm.

4.3 Essex Face94 Dataset

The Essex face dataset [4] contains images of 152 individuals at 20 different postures with a size of 180 by 200 pixels. For our experiments we have only used a part of the dataset comprising 780 face images including 19 female and 20 male individuals at 20 different postures (see Fig. 9). For the test on the face dataset the same error threshold was chosen as for the bar data.

The effect of adding a base vector at the 260th image is shown as an example in Fig. 10. Here you can see a drastic improvement in the reconstruction quality after adding a new base vector. The older base vectors W_0 to W_3 are already very sparse and specialized for certain face regions. The new base vector first points directly in the direction of the current image. After a few iterations, the vector also gets less complex and sparser.

Fig. 9. The part of the Essex face94 dataset we have used includes the 39 individuals depicted here. For each individual 20 different postures are available.

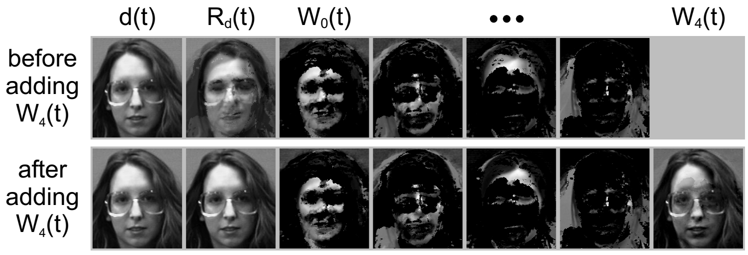

Fig. 10. The reconstruction error before adding a base vector (top) is very high. After adding the base vector $\mathbf{w}_4(t)$ (bottom), the reconstruction quality increases drastically.

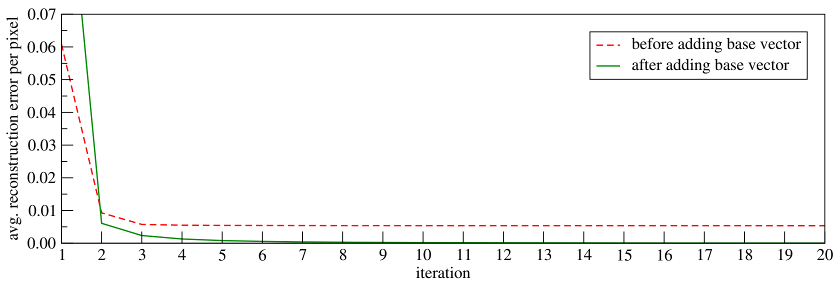

Fig. 11. The error for the insufficient number of base vectors stays high (dashed), whereas after adding a vector (solid) it drops to nearly zero

In Fig. 11 we show the error function before and after adding the base vector. Both curves show a similar development, but only with the additional vector a sufficiently good reconstruction quality can be reached, whereas before, the error could not fall below a certain value.

Figure 12 depicts the final six base vectors and six randomly selected data samples **d** with their corresponding reconstruction **R**. It strikes that the vec-

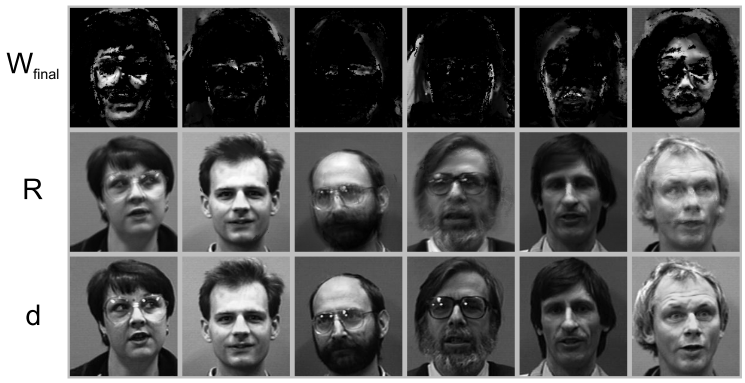

Fig. 12. Here you can see the final base vectors generated by the iNMF (top) and the reconstruction (middle) of selected faces (bottom)

tors represent "parts" of a face. Furthermore, one can see that the information about the input space is preserved to a large degree in the base vectors, as the reconstructions show. In this case, the iNMF also finds the number of base vectors by itself, starting with only one vector.

5 Discussion

In this paper we have shown that the incremental NMF is able to handle both artificial and real-world data in a sequential manner. We exploit the fact that the base vectors themselves already represent all previously shown data samples. Starting from only one single base vector the iNMF decides autonomously if and when an additional base vector is required to reconstruct the input space. The presented algorithm overcomes the need to choose the number of base vectors a priori. This eases the use of the iNMF, especially if the intrinsic dimensionality of the dataset is not known, which is often the case for real-world data. Nevertheless the reconstruction quality of the iNMF is comparable to the NMF (see Sect. 4.2). Furthermore it is possible to perform both online computation and long-term adaptation and at the same time reduce the memory and computational requirements. These reduced requirements make the processing of huge real-world datasets possible in the first place. Since the proposed principle is not tailored to the NMF only, we propose to investigate ways to transfer the presented incremental schema to other factorization methods.

References

1. Lee, D.D., Seung, H.S.: Learning the parts of objects by non-negative matrix factorization. Nature 401, 788–791 (1999)
2. Lee, D.D., Seung, H.S.: Algorithms for non-negative matrix factorization. In: NIPS, pp. 556–562 (2000)
3. Cao, B., Shen, D., Sun, J.-T., Wang, X., Yang, Q., Chen, Z.: Detect and track latent factors with online nonnegative matrix factorization. In: IJCAI, pp. 2689–2694 (2007)
4. University of Essex: Essex face94 dataset, `http://cswww.essex.ac.uk/mv/allfaces/faces94.html`
5. Donoho, D., Stodden, V.: When does non-negative matrix factorization give a correct decomposition into parts? In: NIPS (2003)