Non-negative matrix factorization extended by sparse code shrinkage and by weight sparsification

Botond Szatmáry[1], Barnabás Póczos[1], Julian Eggert[2], Edgar Körner[2], and András Lőrincz[1, 3]

Abstract. Properties of a novel algorithm called non-negative matrix factorization (NMF), are studied. NMF can discover substructures and can provide estimations about the presence or the absence of those, being attractive for completion of missing information. We have studied the working and learning capabilities of NMF networks. Performance was improved by adding sparse code shrinkage (SCS) algorithm to remove structureless noise. We have found that NMF performance is considerably improved by SCS noise filtering. For improving noise resistance in the learning phase, weight sparsification was studied; a sparsifying prior was applied on the NMF weight matrix. Learning capability versus noise content was measured with and without sparsifying prior. In accordance with observation made by others on independent component analysis, we have also found that weight sparsification improved learning capabilities in the presence of Gaussian noise.

1 Introduction

In most pattern recognition problem noise filtering is of central issue. The separation of noise from data is, however, problem-dependent. In information theory, noise is considered structure-free, i.e. of maximal entropy. For continuous variables, Gaussian distribution of unit variance has the maximal entropy. The recently introduced sparse code shrinkage (SCS) algorithm [3], aims to separate Gaussian noise from structured components by minimizing mutual information. This novel approach can be considered as a generalization of wavelet denoising [3]. The generalization concerns the process of the learning of the underlying basis set given an ensemble of inputs and then performing *denoising* via thresholding, similarly to in the case of wavelet bases. SCS originates from independent component analysis (ICA, also called blind source separation, or de-mixing), which has about a ten years long history now [5, 2, 1, 9, 8]. The objective of ICA algorithms is to optimize information transfer for linearly mixed inputs [4]. ICA removes higher order correlations from components. ICA transformed information has limited power in pattern completion problems that assume correlations between components.

Decomposition of multivariate data into correlating sub-structures can be useful in pattern completion problems, e.g. when occlusion may occur. The objective of learning is to seek sub-parts in individual inputs of an ensemble of inputs to enable inferencing. A powerful recent technique is non-negative matrix factorization (NMF, [6, 7]), which aims to find sub-structures in a given set of inputs. NMF assumes that each input is built from non-negative components and is mixed by a matrix having non-negative matrix elements.

In this paper a combination of SCS and NMF algorithms is proposed for filtering noisy inputs. Weight sparsification using a sparsifying prior is applied to the NMF matrix to improve learning capabilities for noisy inputs. In Sec. 2 the joined architecture is described. Simulation results on a the two-bar problem are presented in Sec. 3. Conclusions are drawn in Sec. 4. For completeness and for the reproduction purposes a short summary of the algorithms and the derivation of the learning rules are provided in the Appendix. Detailed derivations can be found in the cited literature.

2 Architecture

SCS is a bottom-up filter, which assumes that inputs are mixed from independent sources. In this case, the SCS filter is capable to recover the original sources. Moreover, these original sources can be found even if the mixed inputs are corrupted by additive Gaussian noise. NMF, on the other hand, can be seen as a top-down generative algorithm that searches for positively correlated components in the inputs under the condition that both the sources and the mixing matrix can have only non-negative elements. NMF optimizes the internal representation to minimize the reconstruction error between input and generated (reconstructed) input. NMF provides positive magnitudes of positive components to be superimposed for reconstruction. In turn, NMF discovers positive substructures, which can be superimposed. For faces, for example, NMF provides barely overlapping components, such as eyes, mouth and nose [6, 7].

The two algorithms can be merged into a single architecture as shown in Fig. 1, where $\mathbf{W}$ denotes the bottom-up SCS transformation that together with denoising produces the sparse components, whereas $\mathbf{Q}$ is the NMF generative matrix. I_{NMF} denotes the NMF procedure. The architecture depicted in Fig. 1 will be referred to as the 'loop'.

We investigated the learning capabilities and working performance of the proposed joined architecture. The loop exhibited good parameter-free performance with the following settings:

First, bottom-up learning The sparse components were computed using the non-linear SCS method. The bottom-up demixing matrix $\mathbf{W}$ is learned on noise-free inputs, whereas the SCS shrinkage function is estimated from noise covered inputs in this phase.

Second, top-down learning: Inputs are filtered by the bottom-up matrix $\mathbf{W}$ and the SCS non-linearity is applied. The non-linear outputs are multiplied by the pseudo-inverse $\mathbf{W}^+$ of matrix $\mathbf{W}$ (which is equal to $\mathbf{W}^T$ in our case [3]). In other words, the inputs are projected into the SCS subspace defined by the row vectors of matrix $\mathbf{W}$ and sparsification is performed on the projections. Afterwards,

[1] Department of Information Systems, Eötvös Loránd University, Pázmány Péter sétány 1/C, 1117 Budapest, Hungary
[2] Future Technology Research, Honda R&D Europe, Carl-Legien-Strasse 30, 63073 Offenbach, Germany
[3] Corresponding author

these sparse components are transformed back to the original input space. The reconstructed inputs are shifted to the positive range. These bounded and non-negative inputs are subsequently used in batch mode to compute the NMF basis set (or NMF matrix, $\mathbf{Q}$).

Working phase: Inputs are projected into the SCS subspace as before. Sparsification occurs, sparsified components are projected back to the input space. NMF components $\mathbf{h}$ are developed using the NMF iteration procedure ($\mathbf{I}_{NMF}$) keeping matrix $\mathbf{Q}$ unchanged. The iteration minimizes the mean square of the reconstruction error. Reconstructed input $\hat{\mathbf{x}}$ can be computed by multiplying the NMF matrix $\mathbf{Q}$ with the NMF vector $\mathbf{h}$ from the right.
For further details, see Appendix.

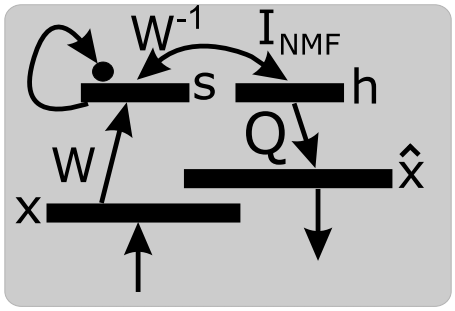

Figure 1. Graphical representation of the algorithm
$\mathbf{x}, \mathbf{s}, \mathbf{h}$ and $\hat{\mathbf{x}}$: input, shrunk (denoised) ICA components, hidden (NMF) variables, and reconstructed input, respectively. $\mathbf{W}$, $\mathbf{W}^+$, $\mathbf{Q}$ and $\mathbf{I}_{NMF}$ denote demixing matrix, pseudoinverse of the demixing matrix, NMF matrix and NMF iteration, respectively. Arrow: linear transformation, arrow with black dot: linear transformation with component-wise non-linearity (the shrinkage kernel), lines with two arrow-heads: iteration. The algorithm was utilized in a two phase mode (see text for details).

3 Results

3.1 The bar example

Orthogonal bars were used for demonstration purposes. The orientation of bars can be either horizontal or vertical. The task was to identify and reconstruct the bars on gray-scale images while (i) each input is composed of two or more bars, (ii) inputs are covered by additive Gaussian noise of zero mean but with varying variance in different experiments. We studied the effect of the additive noise on the quality of the reconstruction. The 'original' inputs consisted of white bars represented by 1's for each vector component, whereas the background was black having 0 values in the appropriate vector component. Overlapping components of bars were not added but assumed value 1. Noisy inputs were constructed as follows: Zero mean Gaussian noise with various standard deviations (STD) was added to the values of the noise-free inputs. These noise corrupted inputs were shifted and clipped to the positive range. The amount of shifting was determined so that clipping preserved 90% of the noise. An example of a noise-free input and its noise added version are shown in Fig. 2(A) and (B), respectively.

3.2 Experimental demonstrations

Simulation results on reconstruction properties are shown in Figs. 2 and 3. Figure 2(B) depicts one of the inputs used for training and testing. The SCS, NMF, and combined SCS and NMF reconstructed inputs are shown in Fig. 2(C)-(E). NMF (Fig. 2(D)) fails with such high noise content, it can be seen that reconstructed input has a high noise content. Instead, the combined method 'predicts' higher amplitudes for pixel values corresponding to the 1's of the original (noise-free) input, leading to a better reconstruction.

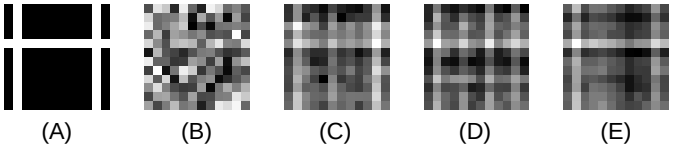

Figure 2. Input and it's reconstructed forms
(A) Perfect input without noise. Input size: 12×12. (B) Noise-free input covered with zero mean 1.5 STD Gaussian noise. (C) Reconstructed input (RI) with SCS. (D) RI with NMF. (E) RI with combination of SCS and NMF. Note the improved reconstruction for the combined method compared to single SCS or single NMF algorithms. The number of basis vectors for both algorithms (SCS, NMF) was set to 24.

Reconstructing capabilities of the different methods are shown in Fig. 3 as a function of noise. As it can be seen on this figure, performance of NMF can be improved by both SCS pre-filtering and by weight matrix sparsification.

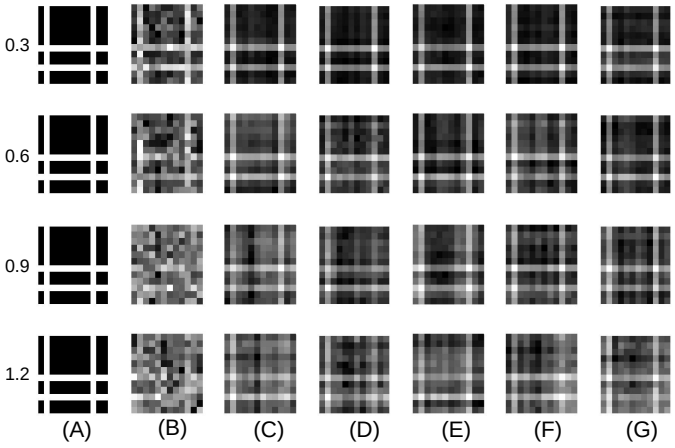

Figure 3. Reconstruction abilities of the different methods
Reconstructed inputs produced by different methods for different noise content. Standard deviation of Gaussian noise is shown on the leftmost side. (A): noise-free input, (B): noise covered input, (C): SCS reconstruction, (D): NMF reconstruction, (E): using SCS pre-filtering, (F): NMF reconstruction with NMF matrix trained with weight sparsification, (G): NMF reconstruction using SCS pre-filtering and with NMF matrix trained with weight sparsification.

Additional information can be gained by examining the basis sets of the different methods (Fig. 4). Fig. 4(A) depicts the ICA basis vectors trained on noisy inputs. ICA basis vectors reveal the underlying line-like structure. ICA, as expected, fails to discover the positive components. Instead, ICA finds basis vectors of different signs. Moreover, ICA fails to find the single line structure if more bars are present in each input. (This effect is not shown here). NMF is much less sensitive to the number of bars presented simultaneously in the inputs. NMF, on the other hand, is noise sensitive. The high noise content of the NMF basis set (Fig. 4(B)) ex-

plains the poor performance of NMF reconstruction shown in Fig. 2. NMF trained on inputs that have undergone SCS pre-filtering before NMF training represent the original noise-free inputs better (Fig. 4(C)). The NMF basis set is also improved somewhat by weight sparsification (Fig. 4(D)). Further improvement can be seen when both SCS pre-filtering and NMF weight sparsification are applied together (Fig. 4(E)). In this last figure, as expected, small elements of the NMF matrix have disappeared; the figure seems 'cleaner'. It is, however, unexpected that the larger elements of the NMF matrix become more balanced, the variance of pixel grey level within each bar became lower upon weight sparsification.

Quantitative dependencies on noise are shown in Fig. 5. Root mean square (RMS) reconstruction error versus noise is depicted for NMF and for the combined NMF and SCS method. Figure 5(A) shows the noise dependence of the reconstruction error for a perfect single bar input covered with varying amount of noise. NMF (SCS+NMF) results are shown by dashed (dotted) lines. Another plot shows the analogous results for two bar inputs (Fig. 5(B)). The performance is improved by the combined method in both cases. SCS has attractive noise resistance properties even if it is unable to find the components of the underlying structure, so SCS can improve noise resistance of NMF. Note that the RMS error is small but non-zero for inputs with zero noise. This is because of the non-linearity in the construction of the inputs – pixel values of overlapping bars are not added.

There is a dependence of performance on the input dimension and on the number of hidden components. It was found that the noise filtering capabilities of SCS have a more pronounced positive effect on the NMF algorithm if the number of hidden components is smaller relative to the dimension of the input.

It may be important to emphasize that SCS was trained on noise-free inputs. In turn, it may seem that noise-filtering remains a problem for the combined method. This is not the case, however. In our architecture, the task of SCS is not the learning of the hidden variables, because this is done by the NMF algorithm. In our architecture the task of SCS is simply the estimation and the removal of the Gaussian noise. To this end, the learning and the removal of the noise content, a feature that SCS exhibits, is satisfactory. Finally, we note that the mathematical theorems concerning the convergence of learning and the stability of the iterative procedures of both algorithms are left intact in this loop structure by construction.

4 Conclusions

Non-negative matrix factorization was studied in this paper. NMF has attractive properties for pattern decomposition and, in turn, has potentials for correlation based pattern completion. However, it was found that NMF is noise sensitive. Two methods were studied to overcome this difficulty. Sparse code shrinkage denoising, which was used for pre-filtering, and weight weight sparsification, which was applied to improve the quality of the basis vectors.

The combined method has attractive properties because of two reasons:

- Gaussian noise can be separated by the SCS algorithm and, in turn, the noise sensitivity of the NMF method is lowered.
- Efficient correlation based pattern completion can not be expected either from ICA or from SCS, because components of these methods have minimized second and higher order correlations. SCS followed by the NMF algorithm has good chances to discover substructures and, in turn, the combined method has good chances in pattern completion tasks.

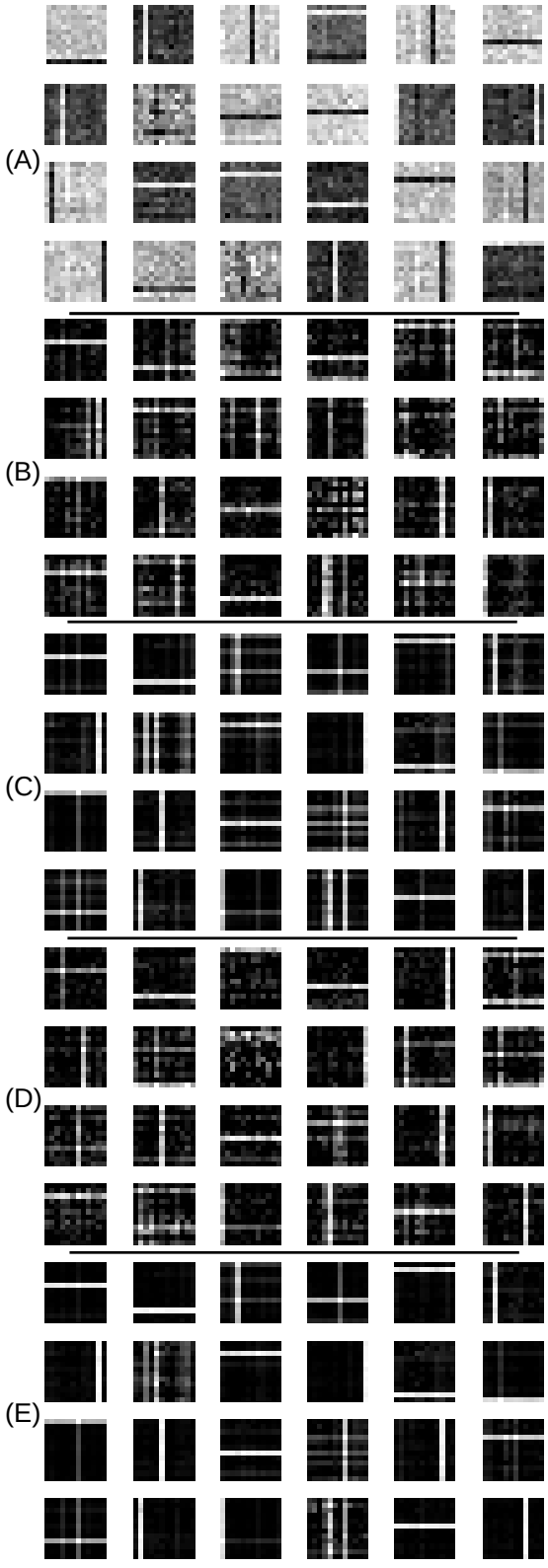

Figure 4. Learned sets of basis vectors
Sets of learned basis vectors for (A): ICA basis set, (B): NMF basis set, (C): NMF basis set trained on inputs pre-filtered by SCS, (D): basis set of NMF augmented with weight sparsification, (E): NMF basis set trained on inputs pre-filtered by SCS and augmented with weight sparsification. STD of Gaussian noise equals 1.5 for all cases.

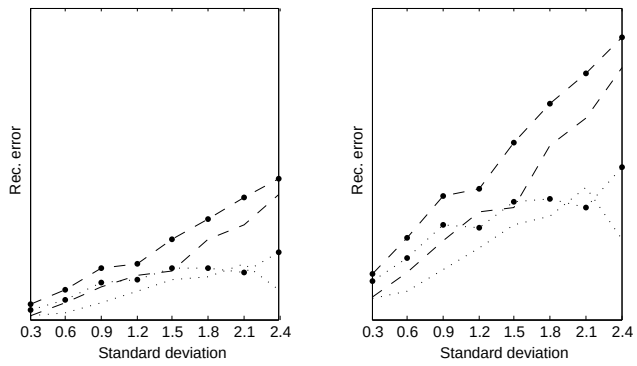

Figure 5. Reconstruction errors
Markers denote different methods. Line with point marker: NMF: dashed line, SCS and NMF: dotted line. Line without marker: NMF with prior: dashed line, SCS and NMF with prior: dotter line. (A) Root mean square reconstruction error for single bar inputs versus STD of noise. (B) Dependence of the root mean square reconstruction error on standard deviation of the noise for double-bar inputs. In both cases, performance of the combined method is significantly better and the usage of sparsifying prior decreases the reconstruction error.

Weight sparsification was shown to improve the learning properties of the NMF algorithm.

Appendix

Weight Sparsification

Given a non-negative matrix $\mathbf{V}$, find non-negative matrix factors $\mathbf{Q}$ and $\mathbf{H}$ such that:

$$\mathbf{V} \approx \mathbf{QH}, \tag{1}$$

where (following the notation of [7]) $\mathbf{V} \in \mathbb{R}_+^{n\times m}$, $\mathbf{Q} \in \mathbb{R}_+^{n\times r}$, $\mathbf{H} \in \mathbb{R}_+^{r\times m}$.

Consider the following cost function:

$$F(\mathbf{Q},\mathbf{H}) = \exp\left(\frac{1}{2}\sum_{ij}\left(\mathbf{V}_{ij} - (\mathbf{QH})_{ij}\right)^2\right), \tag{2}$$

[7] presents an algorithm that optimizes (2) and proves its convergence to a local minima. We extend the mentioned cost function with a sparse exponential prior on the weight matrix $\mathbf{Q}$ and representation matrix $\mathbf{H}$:

$$\begin{aligned} F(\mathbf{Q},\mathbf{h}) &= \exp\left(\frac{1}{2}\sum_{ij}\left(\mathbf{V}_{ij} - (\mathbf{QH})_{ij}\right)^2\right)\cdot \\ &\cdot \exp\left(\sum_{ij}\gamma_1\mathbf{Q}_{ij}\right)\cdot\exp\left(\sum_{ij}\gamma_2\mathbf{H}_{ij}\right), \end{aligned}$$

where $\gamma_1,\ \gamma_2 \in \mathbb{R}_+$.

For simplicity we can optimize the logarithm of cost function (3) (the logarithm is a monotone function). The modified cost function is then

$$\begin{aligned} F = \log F(\mathbf{Q},\mathbf{h}) &= \frac{1}{2}\sum_{ij}\left(\mathbf{V}_{ij} - (\mathbf{QH})_{ij}\right)^2 \\ &+ \sum_{ij}\gamma_1\mathbf{Q}_{ij} + \sum_{ij}\gamma_2\mathbf{H}_{ij} \end{aligned} \tag{3}$$

Using the derivative of (3) with respect to $\mathbf{H}$:

$$\frac{\partial F}{\partial \mathbf{H}} = -\mathbf{Q}^T\mathbf{V} + \mathbf{Q}^T\mathbf{QH} + \gamma_2 \tag{4}$$

and the derivative of (3) with respect to $\mathbf{Q}$:

$$\frac{\partial F}{\partial \mathbf{Q}} = -\mathbf{VH}^T + \mathbf{QHH}^T + \gamma_1 \tag{5}$$

the following gradient algorithm can be derived:

(a) $t = 1$;
(b) $\mathbf{h}(t) = \mathbf{h}(t) + \alpha_1 \quad \mathbf{Q}^T\mathbf{Qh}\,(t) - \mathbf{Q}^T\mathbf{v}\,(t) + \gamma_2$
(c) $\mathbf{h}(t) = \oplus(\mathbf{h}(t))$
(d) $\mathbf{Q} = \mathbf{Q} + \alpha_2 \quad \mathbf{Qh}\,(t)\,\mathbf{h}\,(t)^T - \mathbf{v}(t)\mathbf{h}(t)^T + \gamma_1$
(e) $\mathbf{Q} = \oplus(\mathbf{Q})$
(f) if convergence go back to (b), else go to (g)
(g) $t = t + 1$, back (b)

where function $\oplus$ cuts the negative values, α_1, α_2 are learning parameters, $\mathbf{h}(t)$ and $\mathbf{v}(t)$ are the t^{th} column vector of $\mathbf{H}$ and $\mathbf{V}$, $\mathbf{A}^T$ denotes the transpose of matrix $\mathbf{A}$. Convergence of the algorithm is guaranteed by an appropriate choice of α_1, α_2.

Lee and Seung [7] give a more compact update rule for $\mathbf{Q}$ and $\mathbf{H}$. We extended their derivation for the modified optimization function with sparse exponential prior:

Definition 4.1. $G(h,h')$ is an auxiliary function for $F(h)$ if $G(h,h') \geq F(h)$, $G(h,h) = F(h)$

Lemma 4.2. *If G is an auxiliary function for F, then F is non-increasing under the update*

$$h^{t+1} = arg\min_h G(h,h^t) \tag{6}$$

Proof. $F(h^{t+1}) \leq_{definition} G(h^{t+1},h^t) \leq_{eq.(6)} G(h^t,h^t) = F(h^t)$ □

Lemma 4.3. *If $K(\mathbf{h}^t)$ is a diagonal matrix $K_{ij}(\mathbf{h}^t) = \delta_{ij}(\mathbf{Q}^T\mathbf{Qh}^t)_i/\mathbf{h}_i^t$, ($\delta$ is the Kronecker delta), then*

$$G(\mathbf{h},\mathbf{h}^t) = F(\mathbf{h}^t) + (\mathbf{h}-\mathbf{h}^t)^T\partial F(\mathbf{h}^t) + \frac{1}{2}(\mathbf{h}-\mathbf{h}^t)^T K(\mathbf{h}^t)(\mathbf{h}-\mathbf{h}^t) \tag{7}$$

is an auxiliary function for

$$F(\mathbf{h}) = \frac{1}{2}\sum_i(\mathbf{v}_i - \mathbf{Q}_i\mathbf{h})^2 + \gamma_2\sum_i\mathbf{h}_i$$

Proof. $G(\mathbf{h},\mathbf{h}) = F(\mathbf{h})$ is obvious.

To prove $G(\mathbf{h},\mathbf{h}) \leq F(\mathbf{h})$ we have to compare the Taylor series of $F(\mathbf{h})$:

$$F(\mathbf{h}) = F(\mathbf{h}^t) + (\mathbf{h}-\mathbf{h}^t)^T\partial F(\mathbf{h}^t) + \frac{1}{2}(\mathbf{h}-\mathbf{h}^t)^T(\mathbf{Q}^T\mathbf{Q})(\mathbf{h}-\mathbf{h}^t)$$

$$\text{using } \frac{\partial^2 F}{\partial^2 \mathbf{H}} = \mathbf{Q}^T\mathbf{Q}$$

with equation (7) to find that $G(\mathbf{h},\mathbf{h}) \leq F(\mathbf{h})$ is equivalent to

$$\begin{aligned} G(\mathbf{h},\mathbf{h}^t) - F(\mathbf{h}) &\geq 0 \\ (\mathbf{h}-\mathbf{h}^t)^T(K(\mathbf{h}^t) - \mathbf{Q}^T\mathbf{Q})(\mathbf{h}-\mathbf{h}^t) &\geq 0 \end{aligned}$$

To prove positive semidefiniteness, consider the matrix: $M_{ij}(\mathbf{h}^t) = \mathbf{h}_i^t \ K \ \mathbf{h}^t \ - \mathbf{Q}^T\mathbf{Q}\ _{ij} \mathbf{h}_j^t$ which is just a re-scaling of the components of $\mathbf{K} - \mathbf{Q}^T\mathbf{Q}$. Then $\mathbf{K} - \mathbf{Q}^T\mathbf{Q}$ is semidefinite if and only if M is, and

$$\mathbf{v}^T\mathbf{M}\mathbf{v} = \sum_{ij} \mathbf{v}_i\mathbf{M}_{ij}\mathbf{v}_j =$$

$$\sum_{ij} \mathbf{v}_i\mathbf{h}_i\delta_{ij}(\mathbf{Q}^T\mathbf{Q}\mathbf{h})_i/\mathbf{h}_i\mathbf{h}_j\mathbf{v}_j - \sum_{ij} \mathbf{v}_i\mathbf{h}_i(\mathbf{Q}^T\mathbf{Q})_{ij}\mathbf{h}_j\mathbf{v}_j =$$

$$\sum_{i} \mathbf{v}_i\mathbf{h}_i(\mathbf{Q}^T\mathbf{Q}\mathbf{h})_i/\mathbf{h}_i\mathbf{h}_i\mathbf{v}_i - \sum_{ij} \mathbf{v}_i\mathbf{h}_i(\mathbf{Q}^T\mathbf{Q})_{ij}\mathbf{h}_j\mathbf{v}_j =$$

$$\sum_{i} \mathbf{v}_i\mathbf{h}_i(\mathbf{Q}^T\mathbf{Q})_i\mathbf{h}\mathbf{v}_i - \sum_{ij} \mathbf{v}_i\mathbf{h}_i(\mathbf{Q}^T\mathbf{Q})_{ij}\mathbf{h}_j\mathbf{v}_j =$$

$$\sum_{ij} \mathbf{v}_i^2\mathbf{h}_i(\mathbf{Q}^T\mathbf{Q})_{ij}\mathbf{h}_j - \sum_{ij} \mathbf{v}_i\mathbf{h}_i(\mathbf{Q}^T\mathbf{Q})_{ij}\mathbf{h}_j\mathbf{v}_j =$$

$$\sum_{ij} (\mathbf{Q}^T\mathbf{Q})_{ij}\mathbf{h}_i\mathbf{h}_j \ \ \frac{1}{2}\mathbf{v}_i^2 + \frac{1}{2}\mathbf{v}_j^2 - \mathbf{v}_i\mathbf{v}_j \ \ =$$

$$\frac{1}{2}\sum_{ij} (\mathbf{Q}^T\mathbf{Q})_{ij}\mathbf{h}_i\mathbf{h}_j \ (\mathbf{v}_i - \mathbf{v}_j)^2 \geq 0$$

□

Replacing $G(\mathbf{h}, \mathbf{h}^t)$ in equation (6) by (7) yields the following update rule:

$$\frac{\partial G(\mathbf{h}, \mathbf{h}^t)}{\partial \mathbf{h}} = 0$$

$$\nabla F(\mathbf{h}^t) + K(\mathbf{h}^t)\mathbf{h} - \frac{1}{2}K(\mathbf{h}^t)\mathbf{h}^t - \frac{1}{2}K(\mathbf{h}^t)\mathbf{h}^t = 0$$

$$\Longrightarrow K(\mathbf{h}^t)\mathbf{h} = K(\mathbf{h}^t)\mathbf{h}^t - \nabla F(\mathbf{h}^t)$$

$$\Longrightarrow \mathbf{h}^{t+1} = \mathbf{h}^t - K(\mathbf{h}^t)^{-1}\nabla F(\mathbf{h}^t)$$

$$\Longrightarrow \mathbf{h}_i^{t+1} = \mathbf{h}_i^t \frac{(\mathbf{Q}^T\mathbf{v})_i - \gamma_2}{(\mathbf{Q}^T\mathbf{Q}\mathbf{h}^t)_i} \qquad (8)$$

Similar update rule for $\mathbf{Q}$ can be derived by reversing the roles of $\mathbf{Q}$ and $\mathbf{H}$ (in function G and F) and using a diagonal matrix $K_{ij}(\mathbf{Q}^t) = \delta_{ij}(\mathbf{Q}^t\mathbf{H}\mathbf{H}^T)_i/\mathbf{Q}_i^t$:

$$\mathbf{Q}_{ij}^{t+1} = \mathbf{Q}^t - K(\mathbf{Q}^t)^{-1}\nabla F(\mathbf{Q}^t) =$$

$$\mathbf{Q}_{ij}^t \frac{(\mathbf{V}\mathbf{H}^T)_{ij} - \gamma_1}{(\mathbf{Q}^t\mathbf{H}\mathbf{H}^T)_{ij}} \qquad (9)$$

One can only estimate the optimal γ_1 and γ_2 in equation (8) and (9) to preserve the non-negativity of $\mathbf{h}$ and $\mathbf{Q}$. Time (input) varying γ_1 and γ_2 can also be given:

$$\gamma_1 = \frac{1}{k_1}(\mathbf{V}\mathbf{H}^T)_{ij}, \ \gamma_2 = \frac{1}{k_2}(\mathbf{Q}^T\mathbf{v})_i,$$

where $k_1, k_2 \in (1, \infty)$.

One Loop of the combined algorithm

The combined algorithm in one loop can be summarized as follows. Let $\mathbf{x} \in \mathbb{R}^n$ denote the input to the system.

1.(a) *Learning phase:* Estimation of the sparse coding transformation $\mathbf{W}$ and shrinkage function g.

(b) *Working phase:* Loading the sparse code transformation matrix $\mathbf{W}$ and estimating the shrinkage function g.

2. Computation of the projection on the sparsifying basis: $\mathbf{s} = g(\mathbf{W}\mathbf{x})$, where g is the estimated shrinkage function.
3. Estimation of the denoised inputs: $\mathbf{x}_{SCS} = \mathbf{W}^T\mathbf{s}$.

4.(a) *Learning phase:* NMF basis set $\mathbf{Q}$ and the hidden variables $\mathbf{h}$ are estimated using the denoised $\mathbf{x}_{SCS}$ inputs. In the NMF iteration, the cost function was applied with an additional sparsification prior on the NMF matrix:

$$\mathbf{J} = \exp(\|\mathbf{X} - \mathbf{Q}\mathbf{H}\|_{frob}^2) \cdot \exp(\gamma \cdot \|\mathbf{Q}\|_{frob}^1),$$

where columns of matrix $\mathbf{H}$ represent the hidden representation vectors, columns of matrix $\mathbf{X}$ represent the denoised $\mathbf{x}_{SCS}$ inputs, γ defines the strength of the prior ($\gamma = 0$ means no prior). Subscript 'frob' indicates Frobenius norm defined as the sum of the squared matrix components: $\|\mathbf{Q}\|_{frob}^p = \sum_{ij}(\mathbf{Q}_{ij})^p$.

The batch learning rules for the basis set and hidden variables were used:

$$\mathbf{H}_{ij} \leftarrow \mathbf{H}_{ij}\frac{(\mathbf{Q}^T\mathbf{X})_{ij}}{(\mathbf{Q}^T\mathbf{Q}\mathbf{H})_{ij}}, \ \mathbf{Q}_{ij} \leftarrow \mathbf{Q}_{ij}\frac{(\mathbf{X}\mathbf{H}^T)_{ij} - \gamma}{(\mathbf{Q}\mathbf{H}\mathbf{H}^T)_{ij}}$$

(b) *Working phase:* Using the NMF basis set ($\mathbf{Q}$) computed in batch mode, the hidden variables ($\mathbf{h}$) can be estimated. Input: $\mathbf{x}_{SCS}$.

5. Estimation of the reconstructed input $\hat{\mathbf{x}}$ by multiplying the hidden variable $\mathbf{h}$ with the NMF basis matrix $\mathbf{Q}$.

5 Acknowledgments

Helpful discussions with Ata Kaban are gratefully acknowledged. This work was partially supported by Hungarian National Science Foundation (Grant No. OTKA 32487) and by Honda R&D Europe GmbH, Future Technology Research, Offenbach am Main, Germany

REFERENCES

[1] A.J. Bell and T.J. Sejnowski, 'An information-maximization approach to blind separation and blind deconvolution', *Neural Computation*, **7**, 1129–1159, (1995).

[2] P. Comon, 'Independent component analysis - A new concept?', *Signal Processing*, **36**, 287–314, (1994).

[3] A. Hyvärinen, 'Sparse code shrinkage: Denoising of nongaussian data by maximum likelihood estimation', *Neural Computation*, **11**, 1739–1768, (1999).

[4] A. Hyvärinen, 'Survey on independent component analysis', *Neural Computing Surveys*, **2**, 94–128, (1999).

[5] C. Jutten and J. Herault, 'Blind separation of sources, Part I: An adaptive algorithm based on neuromimetic architecture', *Signal Processing*, **24**, 1–10, (1991).

[6] D.D. Lee and H.S. Seung, 'Learning the parts of objects by non-negative matrix factorization', *Nature*, **401**, 788–791, (1999).

[7] D.D. Lee and H.S. Seung, 'Algorithms for non-negative matrix factorization', in *Advances in Neural Processing Systems*, volume 13, Morgan Kaufmann, San Mateo, CA, (2001). in press.

[8] S.-I. Amari, 'Natural gradient works efficiently in learning', *Neural Computation*, **10**, 251–276, (1998).

[9] S.-I. Amari, A. Cichocki, and H.H. Yang, 'A new learning algorithm for blind signal separation.', in *Advances in Neural Information Processing Systems*, 757–763, Morgan Kaufmann, San Mateo, CA, (1996).