CONTINUOUS NON-NEGATIVE MATRIX FACTORIZATION FOR TIME-DEPENDENT DATA

Lars Omlor [a], *Jean-Jacques Slotine* [b]

[a] Department of Cognitive Neurology, University of Tübingen, Wilhelmstr. 7 ,72074 Tübingen, Germany
phone: +49 (0)7071/ 89130, email: lars.omlor@medizin.uni-tuebingen.de
[b] Nonlinear Systems Laboratory, MIT, 77 Massachusetts Avenue, Cambridge MA 02139-4307, USA
phone: +1 617-253-0490, email: jjs@mit.edu

ABSTRACT

In many signal processing applications such as image analysis or spectral decomposition, non-negativity constrains are necessary to provide a physical reasonable interpretation. This constraint is exploited by non-negative matrix factorization (NMF) methods. The goal of NMF is to find low rank matrices $A \geq 0$ and $B \geq 0$ such that the positive data matrix X can be approximated by $AB \approx X$. Most algorithms for this type of factorization are discrete-step iterative optimization procedures based on gradient descent or Quasi-Newton methods. Here we propose a continuous-time version of NMF based on dynamical systems with positive solutions, which allows time-dependent cost functions, e.g. due to time-dependent data.

1. INTRODUCTION

In many applications, such as image-processing [5] or spectral-deconvolution [6], the requirement that the solutions be non-negative has lead to the development of multiplicative update rules for non-negative constraint optimization. Lee and Seung [9] used these rules to decompose images (pictures of faces) $X \approx AB$ into parts A (lips, eyes, etc.) and the corresponding mixing operator B, by alternating updates for the matrices A and B. This approach has been shown recently to be useful for many applications, such as blind feature extraction [9] or dimension reduction [13]. For minimizing the constrained squared Euclidean distance:

$$E(X,AB) = \|X - AB\|^2 \text{ such that } a_{ij} \geq 0\ ,\ b_{jk} \geq 0\ \ \forall i,j,k \tag{1}$$

the multiplicative algorithm takes the form:

$$a_{ij} \leftarrow a_{ij}\left(\frac{XB^T}{ABB^T}\right)_{ij}\ ,\ b_{jk} \leftarrow b_{jk}\left(\frac{A^TX}{A^TAB}\right)_{jk} \tag{2}$$

Various improvements or alternatives to this simple gradient descent algorithm have been put forward [2], in order to speed up the slow convergence and to avoid the often occurring slow progress zig-zag path of the optimization in flat valleys of the cost function (called zigzagging or jamming).
Implicit in these approaches to non-negative matrix factorization is the assumption of a constant (static) data matrix X. The issue of time-dependent data is usually treated by increasing the dimensionality of X, either by arranging the data observed at different time-points $X(t_i)$ as rows of X, or by treating each time as independent element constituting a single data point (adding time to the columns of X). This treatment of time has three major drawbacks. First, the data for all time-points has to be known in advance, making this kind of algorithm unsuitable for online learning. Second, increasing the dimensionality of X is obviously only feasible for a moderate number of time points. Finally, adding dimensions to X neglects temporal dependencies of the data such as causality.
To address these problems we propose here a time-dependent non-negative model:

$$X(t) \approx A(t) \cdot B(t) \text{ with } a_{ij}(t) \geq 0\ ,\ b_{jk}(t) \geq 0\ \ \forall t$$

which minimizes a differentiable time-dependent cost function $E(t, X(t), A(t), B(t)) \geq 0$. This approach not only encompasses the classical algorithm but also enables NMF to factorize data streams online, e.g. from audio or video. The parameters $A(t), B(t)$ are optimized using simple dynamical systems based on gradient descent and projected gradient descent. This dynamical system approach allows the use of NMF as a network element in multilayered structures, which admits feedback in the form of a time-dependent cost. The performance of the method is illustrated in numerical simulations based on two different data sets. The first data set, composed of unstructured random matrices, is used to compare the convergence rates of different implementations of time-dependent NMF. The second, more specific data set, is composed of music sequences, as audio separation is one of the main application fields of blind source separation methods.

2. DYNAMICAL SYSTEMS AND OPTIMIZATION

One of the most popular methods for minimizing an arbitrary cost function $E(x) \geq 0$ is discrete gradient descent, given by the iterative algorithm:

$$x_{t+1} = x_t - \eta \nabla E(x_t) \tag{3}$$

Here $\eta \geq 0$ denotes the step length taken into the direction of the gradient $\nabla E(x_t)$ of the cost E at point x_t. Taking the limit in the discrete method leads to the continuous-time gradient descent method [1], described by the ordinary differential equation (ODE):

$$\frac{dx(t)}{dt} = \dot{x}(t) = -\eta \nabla E\big(x(t)\big) \tag{4}$$

The solution of this equation is a function $x(t)$ along which the cost E decreases the fastest:

$$\frac{dE\big(x(t)\big)}{dt} = \dot{E}(t) = \big\langle \nabla E\big(x(t)\big), \dot{x}(t)\big\rangle = -\eta \|\nabla E\big(x(t)\big)\|^2 \tag{5}$$

Thus, following the smooth solution $x(t)$ leads to a stationary point of E. The idea can be adjusted for constrained optimization, leading to the projected gradient descent method:

$$\dot{x}(t) = P_C\Big(x(t) - \eta\nabla E\big(x(t)\big)\Big) - x(t) \tag{6}$$

where P_C denotes the projection onto the set of constraints C (e.g. $P_C(x(t)) = \max(x(t), 0)) =: (x(t))_+$ for $C = \mathbb{R}_+$).
One of the major advantages of this continuous treatment is the fact that these methods can be easily adjusted to the case of (explicit or implicit) time-dependent cost functions $E = E(t,x)$.

2.1 Continuous gradient descent for NMF

In the case of classical NMF, neither the cost function E nor the data X are seen as time dependent. However the factorization parameters A,B can be obtained as limit points of trajectories $A(t) = (a_{ij}(t))_{ij}, B(t) = (b_{jk}(t))_{jk}$ satisfying the ODE (4):

$$\dot{a}_{ij}(t) = -\eta_{ij}\frac{\partial E}{\partial a_{ij}(t)}\ , \ \dot{b}_{jk}(t) = -\tilde{\eta}_{jk}\frac{\partial E}{\partial b_{jk}(t)} \tag{7}$$

In the following the explicit notation of time dependency $a_{ij}(t), b_{jk}(t)$ is dropped, as the variables a_{ij}, b_{jk} are always assumed to be possibly dependent on time. The idea behind the multiplicative update rules (2) is to choose special step lengths (gains) $\eta_{ij}, \tilde{\eta}_{jk}$ equal to the diagonally rescaled variables [9]. In particular this implies that independent of E the step lengths have the form $\eta_{ij} = a_{ij}\eta^{'}_{ij}, \tilde{\eta}_{jk} = b_{ij}\tilde{\eta}^{'}_{jk}$ with arbitrary rests $\eta^{'}_{ij}, \tilde{\eta}^{'}_{jk} \geq 0$. For such a choice of parameters it is a priori unclear whether the new step lengths are positive (a_{ij}, b_{ij} could assume negative values). Yet assuming the special step lengths, we get for a_{ij} (and similarly for b_{jk}):

$$\begin{aligned} \dot{a}_{ij} = -a_{ij}\eta^{'}_{ij}\frac{\partial E}{\partial a_{ij}} \underset{a_{ij}\neq 0}{\Rightarrow} \frac{\dot{a}_{ij}}{a_{ij}} = -\eta^{'}_{ij}\frac{\partial E}{\partial a_{ij}} \\ \Rightarrow a_{ij} = \pm a_{ij}(t_0)\exp\left(\int_{t_0}^{t} -\eta^{'}_{ij}\frac{\partial E}{\partial a_{ij}}\right) \end{aligned} \tag{8}$$

Thus, if $a_{ij}(t_0) = 0, b_{jk}(t_0) = 0$, then the solution vanishes for all $t > t_0$. As any solution is continuous, non-negative initial-values guaranty $a_{ij} \geq 0$ and $b_{jk} \geq 0$. It follows from equation (5) that the total time derivative $\dot{E} \leq 0$, and thus system (7) indeed minimizes the cost E.
As an illustration, consider for instance Euclidean cost (1):

$$E = \|X - A(t)B(t)\|^2 \Rightarrow \frac{\partial E}{\partial A(t)} = 2\big(A(t)B(t)B^T(t) - XB^T(t)\big)$$

To obtain the full diagonally rescaled step length we set $\eta^{'}_{ij} = \frac{1}{(ABB^T)_{ij}(t)} \geq 0$ (the equation for $\dot{b}_{jk}$ is analog)

$$\begin{aligned} \dot{a}_{ij} &= -\frac{a_{ij}}{(ABB^T)_{ij}(t)}\frac{\partial E}{\partial a_{ij}} \\ &= -a_{ij} + a_{ij} * \left(\frac{XB^T(t)}{A(t)B(t)B^T(t)}\right)_{ij} \end{aligned} \tag{9}$$

2.1.1 *Explicit time-dependent cost function*

The most general case of cost function E we consider here can be both explicitly dependent on time, and implicitly dependent on time due to time dependent data $X = X(t)$. For notational convenience the implicit dependency can be incorporated into the explicit time dependency, i.e., $E = E\big(t, X(t), A(t)B(t)\big) =: E\big(t, A(t)B(t)\big)$. In this notation the partial derivative $\frac{\partial E}{\partial t}$ denotes the partial derivative of E with respect to the first variable (time) and thus is a composite term including the derivative with respect to the data. Now considering the total time-derivative $\dot{E} = \frac{dE}{dt}$:

$$\dot{E} = \frac{\partial E}{\partial t} + \left[\sum_{ij}\left(\frac{\partial E}{\partial a_{ij}}\right)\dot{a}_{ij} + \sum_{jk}\left(\frac{\partial E}{\partial b_{jk}}\right)\dot{b}_{jk}\right]$$

it is clear that the simple gradient descent (7) needs to be adjusted, since only the last term in the brackets can be guaranteed to be negative. However equation (5) implies that the absolute value of ϑ is dependent on the chosen step size η. Thus it is possible to change η exactly in a way that the new negative term additively compensates $-|\frac{\partial E}{\partial t}|$. This adjusted system is given by the formulas:

$$\begin{aligned} \dot{a}_{ij} &= -a_{ij}\frac{\partial E}{\partial a_{ij}}\left[\eta_{ij} + \frac{\left((\frac{\partial E}{\partial t})_+ + \beta E\right)}{\sum_{ij}\left(\frac{\partial E}{\partial a_{ij}}\right)^2 a_{ij} + \sum_{jk}\left(\frac{\partial E}{\partial b_{jk}}\right)^2 b_{jk} + \varepsilon}\right] \\ \dot{b}_{jk} &= -b_{jk}\frac{\partial E}{\partial b_{jk}}\left[\tilde{\eta}_{jk} + \frac{\left((\frac{\partial E}{\partial t})_+ + \beta E\right)}{\sum_{ij}\left(\frac{\partial E}{\partial a_{ij}}\right)^2 a_{ij} + \sum_{jk}\left(\frac{\partial E}{\partial b_{jk}}\right)^2 b_{jk} + \varepsilon}\right] \end{aligned} \tag{10}$$

The parameter ε is a regularization term preventing a division by zero in the case of zero denominator above. In practice the influence of noise prevents the gradients $\frac{\partial E}{\partial a_{ij}}$ and $\frac{\partial E}{\partial b_{jk}}$ from vanishing. Furthermore A,B are only zero if $X = 0$ or $A_0 = 0, B_0 = 0$. Thus ε can usually be chosen very small, so it has only negligible influence on the convergence. Since the step size in (10) was exactly chosen to compensate the partial time-derivative, negative total time-derivative of the cost function $\dot{E}$ is now easy to prove in the limit $\varepsilon = 0$:

$$\begin{aligned} \dot{E} =& \frac{\partial E}{\partial t} - \sum_{ij}\eta_{ij}\left(\frac{\partial E}{\partial a_{ij}}\right)^2 a_{ij} - \sum_{jk}\tilde{\eta}_{jk}\left(\frac{\partial E}{\partial b_{jk}}\right)^2 b_{jk} \\ & -(\frac{\partial E}{\partial t})_+ - \beta E \leq -\beta E \end{aligned} \tag{11}$$

Note that the solutions of the modified system (10) are also positive, which follows from the same argument as for the original system (7). Inequality (11) implies that (disregarding numerical difficulties) in theory, the cost function should decrease exponentially with exponent β, which is the reason this parameter was introduced.

2.2 Projected gradient descent

Due to the inherent non-negativity constraint in NMF it seems more natural to use the projected gradient (6) instead of the simple gradient method. This replaces the differential equations (7,8) with the new system:

$$\dot{a}_{ij} = \left(a_{ij} - \mu_{ij}\frac{\partial E}{\partial a_{ij}}\right)_+ - a_{ij}$$
$$\dot{b}_{jk} = \left(b_{jk} - \nu_{jk}\frac{\partial E}{\partial b_{jk}}\right)_+ - b_{jk} \qquad (12)$$

Going back to the example of Euclidean cost (1), the same choice of step lengths $\mu_{ij} = \frac{a_{ij}}{(ABB^T)_{ij}}$ as before results in:

$$\dot{a}_{ij} = \left(a_{ij} * \left(\frac{XB^T(t)}{A(t)B(t)B^T(t)}\right)_{ij}\right)_+ - a_{ij}$$

Thus, if the solution of the projected gradient system is positive, this coincides with the result of the simple gradient descent (9). Thus the projected gradient can indeed be seen as the more general method. The positivity of the solution of (12) follows immediately from

$$\dot{a}_{ij} \geq -a_{ij}\ ,\ \dot{b}_{jk} \geq -b_{jk}.$$

Similar to the gradient descent case this implies $\dot{E} \leq 0$, as can be seen from:

$$\left(\frac{\partial E}{\partial a_{ij}}\right)\dot{a}_{ij} = \begin{cases} -\mu_{ij}\left(\frac{\partial E}{\partial a_{ij}}\right)^2 \leq 0 & \text{if } \mu_{ij}\left(\frac{\partial E}{\partial a_{ij}}\right) \leq a_{ij} \\ -a_{ij}\left(\frac{\partial E}{\partial a_{ij}}\right) \leq 0 & \text{if } 0 \leq a_{ij} \leq \mu_{ij}\left(\frac{\partial E}{\partial a_{ij}}\right) \end{cases}$$

and the analogous computation for $\left(\frac{\partial E}{\partial b_{jk}}\right)\dot{b}_{jk}$.

2.2.1 *Explicit time-dependent cost function*

For explicit time-dependent cost functions the projected gradient method can be adjusted in the same manner as the gradient descent. In order to prove that the adjusted (increased) step sizes:

$$\tilde{\mu}_{ij} = \mu_0 + \frac{(\frac{\partial E}{\partial t})_+}{\sum\limits_{\frac{\partial E}{\partial a_{ij}} \leq 0}\left(\frac{\partial E}{\partial a_{ij}}\right)^2 + \sum\limits_{\frac{\partial E}{\partial b_{jk}} \leq 0}\left(\frac{\partial E}{\partial b_{jk}}\right)^2 + \varepsilon}$$
$$\tilde{\nu}_{jk} = \nu_0 + \frac{(\frac{\partial E}{\partial t})_+}{\sum\limits_{\frac{\partial E}{\partial a_{ij}} \leq 0}\left(\frac{\partial E}{\partial a_{ij}}\right)^2 + \sum\limits_{\frac{\partial E}{\partial b_{jk}} \leq 0}\left(\frac{\partial E}{\partial b_{jk}}\right)^2 + \varepsilon} \qquad (13)$$

compensate the term $\frac{\partial E}{\partial t}$ we need to define the following sets of indices:

$$I_1 := \left\{ij \mid 0 \leq \tilde{\mu}_{ij}\left(\frac{\partial E}{\partial a_{ij}}\right) \leq a_{ij}\right\}$$
$$I_2 := \left\{jk \mid 0 \leq \tilde{\nu}_{ij}\left(\frac{\partial E}{\partial b_{jk}}\right) \leq b_{jk}\right\}$$
$$I_3 := \left\{ij \mid \frac{\partial E}{\partial a_{ij}} \leq 0\right\}\ ,\ I_4 := \left\{jk \mid \frac{\partial E}{\partial b_{jk}} \leq 0\right\}$$
$$I_5 := \left\{ij \mid a_{ij} \leq \tilde{\mu}_{ij}\left(\frac{\partial E}{\partial a_{ij}}\right)\right\}\ ,\ I_6 := \left\{jk \mid b_{jk} \leq \tilde{\nu}_{jk}\left(\frac{\partial E}{\partial b_{jk}}\right)\right\}$$

Now

$$\begin{aligned}\dot{E} &= \frac{\partial E}{\partial t} + \sum_{ij}\left(\frac{\partial E}{\partial a_{ij}}\right)\dot{a}_{ij} + \sum_{jk}\left(\frac{\partial E}{\partial b_{jk}}\right)\dot{b}_{jk} \\ &= \frac{\partial E}{\partial t} - \sum_{I_1}\tilde{\mu}_{ij}\left(\frac{\partial E}{\partial a_{ij}}\right)^2 - \sum_{I_2}\tilde{\nu}_{jk}\left(\frac{\partial E}{\partial b_{jk}}\right)^2 - \sum_{I_5}a_{ij}\left(\frac{\partial E}{\partial a_{ij}}\right) \\ &\quad - \sum_{I_6}b_{jk}\left(\frac{\partial E}{\partial b_{jk}}\right) - \sum_{I_3}\tilde{\mu}_0\left(\frac{\partial E}{\partial a_{ij}}\right)^2 - \sum_{I_3}\tilde{\nu}_0\left(\frac{\partial E}{\partial b_{jk}}\right)^2 \\ &\quad - (\frac{\partial E}{\partial t})_+ \leq 0.\end{aligned}$$

The cumbersome splitting in these subsets is necessary as only I_3 and I_4 are independent of $\tilde{\mu}_{ij}$ and $\tilde{\nu}_{jk}$. Thus the new step sizes are larger than the minimal step sizes compensating $\frac{\partial E}{\partial t}$.

3. NUMERICAL SIMULATIONS

For the numerical solution of ODEs quite a zoo of different methods are available [4], depending on the special type of equation and the goal accuracy. One of the most straightforward methods is the simple Euler approximation, which replaces the differential in (4,6) with the finite difference quotient, resulting in the discrete (projected) gradient descent (3). Applying this approximation to the Euclidean cost example (9) results in the classical iterative NMF update rules (2), with the sole difference that A,B are updated as one block instead of alternatingly. This use of the discrete NMF algorithm provides a baseline in the numerical experiments, as it (almost) coincides with the classical algorithm. Since the Euler method is known to be inaccurate and numerically unstable, for all other simulations the four-point Adams-Bashforth method with variable step size was used [4]. If only the limit point of the optimization procedure is of interest, faster integration schemes can be employed [3].

The first test data set consists of twenty, fast changing unstructured nonnegative matrices $X \in \mathbb{R}_+^{100\times 200}$, i.e. X was constructed as the product of uniformly distributed random positive matrices $A \in \mathbb{R}_+^{100\times 3}$ and $B \in \mathbb{R}_+^{3\times 200}$ times a fast changing sinusoid:

$$X(t) = \big(1.5 + \sin(t)\big)X = \big(1.5 + \sin(t)\big)A * B$$

The second and third data set consists of spectrograms:

$$S_\omega(t) = \left|\int e^{-2\pi \imath t\omega}s(t')h(t'-t)dt'\right|^2$$

computed from musical instrument recordings $s(t')$ taken from the music database described in [16], which mainly consists of sound samples excerpted from classical music CD recordings. For a short window the spectrogram $S_\omega(t)$ can be interpreted as the vector of local frequency intensities at time t, and thus is numerically a time-dependent vector with nonnegative entries. The spectrogram is computed from the mean of both stereo-channels with a short hamming-window h of length approximately $100ms$. All signals are sampled at $11025Hz$.

For the second data set, the data matrix $X(t) \in \mathbb{R}_+^{108\times 1024}$ is composed of the power spectra of all 108 recordings (12 instruments with nine examples each) included in the database.

Finally, for the third test data set, signals from two different instruments (clarinet-violin) $s_i(t)$ are mixed with time-varying weights $a_{ij}(t)$ to form linear 2×2 mixtures m_i given by:

$$m_i(t) = \sum_j a_{ij}(t) s_j(t)$$

The random weight functions $a_{ij}(t)$ have fixed autocorrelation to ensure that they are sufficiently smooth, i.e., their change in the examined $100ms$ window is insignificant but their overall variation is not negligible. From the power spectra of the two mixtures two components are extracted, and compared to the power spectra of the original data.

3.1 Results

3.1.1 Fast changing unstructured data

Figure 1 shows the average convergence rate for both gradient descent methods (4,6) used with Euclidean cost function. The convergence rate is measured as the logarithm of the normalized error:

$$\log\left(Q(X(t), A(t)B(t))\right) := \log\left(\frac{\|X(t) - A(t)B(t)\|}{\|X(t)\|}\right)$$

Both the average convergence (lines in figure 1) and the standard deviation (shaded areas in 1) are computed from twenty random examples $X(t)$. The systems that correct for the temporal derivative $\frac{\partial E}{\partial t}$ outperform their uncorrected counterparts, as expected from their increased step size. In addition the projected gradient descent method have a slightly increased convergence rate compared to the systems with steps equal to the discrete non-negative matrix factorization. The slightly erratic behavior of the corrected projected gradient method is a reflection of the aggressive choice of step length (13). To smooth the convergence behavior smaller step increases can be used, but their theoretical derivation involves the solution of an implicit problem as the set of indices I_1, I_2 also depend on the step sizes.

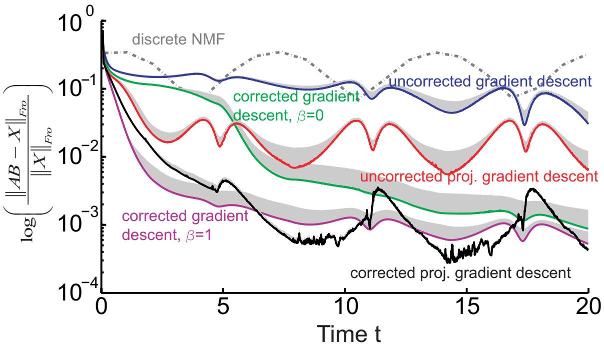

Figure 1: Average convergence (lines) and standard deviation (shaded area) for the uncorrected system (9) and the corrected system (10). The dotted line shows the performance of the discrete NMF update rules (Euler approximation) given the time dependent data $X(t)$, $t = 1, 2, \ldots$.

3.1.2 Music data

Figure 2 shows the factorization performance of both approaches (4) and (6), measured by logarithm of the normalized approximation error $\log\left(Q(t)\right)$ for the complete music data set.

The smaller difference between the two solutions compared with the fast time changing case, can be explained by the fact that for music:

$$\left(\frac{\partial E}{\partial t}\right)_+ \ll \sum_{ij}\left(\frac{\partial E}{\partial a_{ij}}\right)\dot{a}_{ij} + \sum_{jk}\left(\frac{\partial E}{\partial b_{jk}}\right)\dot{b}_{jk}$$

As a baseline Figure 2 also includes the average performance of the discrete update rules (2), i.e. the Euler approximation. The performance of this approximation depends heavily on the variation of the data between time steps, with good convergence for slow change and very erratic behavior for large differences.

The time-averaged approximation performance is $Q_{\text{proj. grad}} = 0.163$, $Q_{\text{grad}} = 0.1868$ for the corrected projected gradient and corrected gradient systems respectively as well as $0.164 = Q_{\text{proj. grad}}$, $0.1868 = Q_{\text{grad}}$ for the uncorrected systems. The baseline discrete time NMF achieves $Q_{\text{NMF}} = 0.472$.

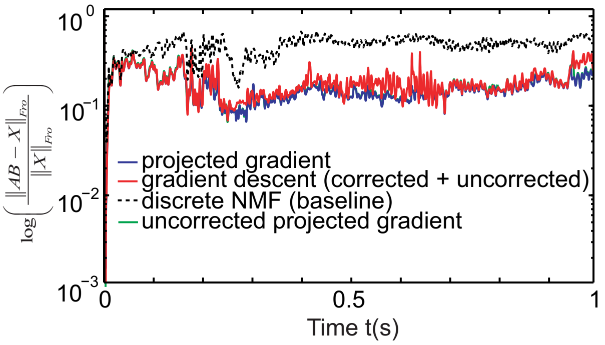

Figure 2: Approximation quality for the (un)corrected solutions for the first second of the music data in comparison to the average performance of discrete NMF.

3.1.3 Time-dependent two by two mixtures

Judging mathematically the quality of sound separation is a nontrivial problem and a variety of measures have been proposed [14]. However most of these measures are hard to interpret and compare. Here we use simple correlation. Specifically, the similarity between the extracted spectra $s_{\omega_{1,2}}(t)$ and the original music spectra $p_{\omega_{1,2}}(t)$ is measured by the Pearson correlation coefficient $c_{ij} = \text{corr}(s_{\omega_i}(t), p_{\omega_j}(t))$ and is shown in figure 3. Note that since the solution of the dynamical system is smooth, the usual ambiguity in the ordering of the extracted sources is largely avoided, i.e. the order determined by the initial values is kept up to points with high correlation between the original spectra $p_{\omega_{1,2}}(t)$. The overall time averaged performance is $c = 0.9$. Thus dynamical NMF with Euclidian cost function performs quite well in separating the different instruments.

This numerical experiment uses the most straightforward application of the simplest case of cost function (Euclidean) allowed in the framework for dynamical NMF. Its performance is bound to be further enhanced if more sophisticated costs like the Itakura-Saito (IS) divergence [12] or a regularizer like sparseness [15] are used. Also, since the scaling ambiguity allows the temporal variability of the weights to be regarded as additional amplitude fluctuation of the sources, a more realistic model for real sound mixtures would replace the weights with time-dependent filters. Using the convolutive non-negative factorization discussed in [11], such a filter-based model is actually a special case of (9).

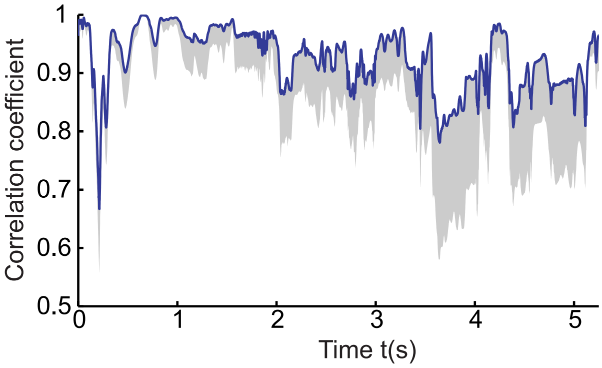

Figure 3: Mean correlation coefficient (continuous line) between the extracted sound spectrograms and the original spectrograms. The shaded area denotes the standard deviation computed in twenty simulations.

4. CONCLUSION

We presented a new approach to non-negative matrix factorization based on dynamical systems. The approach has several advantages. Primarily, the treatment in continuous-time allows the use of time-varying cost functions, including in particular the case of time-dependent data. It is also straightforward to extend the framework to the large variety of modifications developed for non-negative factorizations, such as regularized NMF, non-negative tensor factorization, or convolutive NMF. In fact, using the anechoic NMF [12] update rules, convolutive NMF can be treated as a special case of the algorithm discussed here, thus extending the method to time-varying filters.

In future work, the time-varying cost functions which the method allows may also consist of composite costs functions E of the form, for instance:

$$E = \sin^2(t)E_1 + \cos^2(t)E_2$$

as such composite cost functions have been shown to exhibit promising properties in avoiding local minima [8] and in the development of modular structures reflecting commonalities in the E_i's [7]. In addition, continuous-time Newton (or quasi-Newton) methods for time-varying cost functions along the lines of [10], i.e. state updates designed such that

$$\frac{d}{dt}\,\nabla E(\mathbf{x},t) \;=\; -\lambda \nabla E(\mathbf{x},t)$$

(with λ strictly positive) may also hold promising properties.

Acknowledgements

Supported by EC FP6 project COBOL, Volkswagenstiftung, DFG Forschergruppe Perceptual Graphics and HFSP. We thank Martin A. Giese for his support. Additional support was provided by the Hermann and Lilly-Schilling-Stiftung.

REFERENCES

[1] A.S. Antipin. Minimization of convex functions on convex sets by means of differential equations. *Differential equations*, 30(9):1365–1375, 1994.

[2] M. W. Berry, M. Browne, A. N. Langville, P. V. Pauca, and R. J. Plemmons. Algorithms and applications for approximate nonnegative matrix factorization. *Computational Statistics & Data Analysis*, 52(1):155–173, September 2007.

[3] A.A. Brown and M.C. Bartholomew-Biggs. Some effective methods for unconstrained optimization based on the solution of systems of ordinary differential equations. *J. Optim. Theory Appl.*, 62(2):211–224, 1989.

[4] J.C. Butcher. *Numerical Methods for Ordinary Differential Equations*. Wiley & Sons, June 2008.

[5] M. E. Daube-Witherspoon and G. Muehllehner. An iterative image space reconstruction algorthm suitable for volume ect. *IEEE Transactions on Medical Imaging*, 5(2):61–66, June 1986.

[6] R. Gold. An iterative unfolding method for response matrices. *Argonne National Laboratory Report*, ANL-6984, 1964.

[7] N. Kashtan, A.E. Mayo, T. Kalisky, and U. Alon. An analytically solvable model for rapid evolution of modular structure. *PLoS Comput Biol*, 5(4):e1000355+, April 2009.

[8] N. Kashtan, E. Noor, and U. Alon. Varying environments can speed up evolution. *PNAS*, 104:13711–13716, August 2007.

[9] D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

[10] W. Lohmiller and J.J. Slotine. Contraction analysis of Newton methods for time-varying costs. *MIT-NSL report 090501*, 2009.

[11] L. Omlor and M. A. Giese. Learning of translation-invariant independent components: Multivariate anechoic mixtures. In *ICA 2007*, pages 762–769, 2007.

[12] A. Ozerov and C. Févotte. Multichannel nonnegative matrix factorization in convolutive mixtures. with application to blind audio source separation. In *In Proc. Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP09)*, Taipei, Taiwan, 2009.

[13] S. Tsuge, M. Shishibori, S. Kuroiwa, and K. Kita. Dimensionality reduction using non-negative matrix factorization for information retrieval. *IEEE Int. Conf. on Systems, Man, and Cybernetics*, 2:960–965 vol.2, 2001.

[14] E. Vincent, R. Gribonval, and C. Fevotte. Performance measurement in blind audio source separation. *IEEE Trans. on Audio, Speech, and Language Processing*, 14(4):1462–1469, 2006.

[15] T. Virtanen. Monaural sound source separation by nonnegative matrix factorization with temporal continuity and sparseness criteria. *IEEE Trans. on Audio, Speech, and Language Processing*, 15(3):1066–1074, March 2007.

[16] G. Yu and J.J. Slotine. Audio classification from time-frequency texture. *IEEE ICASSP, Taiwan, 2009.*