2014 IEEE INTERNATIONAL WORKSHOP ON MACHINE LEARNING FOR SIGNAL PROCESSING, SEPT. 21–24, 2014, REIMS, FRANCE

KERNEL NONNEGATIVE MATRIX FACTORIZATION WITHOUT THE PRE-IMAGE PROBLEM

Fei Zhu, Paul Honeine

Institut Charles Delaunay (CNRS)
Université de technologie de Troyes, France
fei.zhu@utt.fr; paul.honeine@utt.fr

Maya Kallas

Centre de Recherche en Automatique de Nancy (CNRS)
Université de Lorraine, France
maya.kallas@univ-lorraine.fr

ABSTRACT

The nonnegative matrix factorization (NMF) is widely used in signal and image processing, including bio-informatics, blind source separation and hyperspectral image analysis in remote sensing. A great challenge arises when dealing with nonlinear NMF. In this paper, we propose an efficient nonlinear NMF, which is based on kernel machines. As opposed to previous work, the proposed method does not suffer from the pre-image problem. We propose two iterative algorithms: an additive and a multiplicative update rule. Several extensions of the kernel-NMF are developed in order to take into account auxiliary structural constraints, such as smoothness, sparseness and spatial regularization. The relevance of the presented techniques is demonstrated in unmixing a synthetic hyperspectral image.

Index Terms— Kernel machines, nonnegative matrix factorization, reproducing kernel Hilbert space, pre-image problem, unmixing problem, hyperspectral data

1. INTRODUCTION

The nonnegative matrix factorization (NMF) has been gaining wide popularity in signal and image processing [1]. It consists in approximating a nonnegative matrix with two low-rank nonnegative matrices [2, 3]. By allowing a dimensionality reduction with non-negativity constraints, it provides a physical interpretation to the factorization. The NMF has been successfully applied to image classification [4], face expression recognition [5], objet recognition [6] and gene expression data [7]. For the hyperspectral image unmixing problem, the NMF provides a decomposition suitable for physical interpretation, where the high-rank matrix consists of pixels and the low-rank ones represent endmembers and abundances [8, 9].

First proposed in [10], the NMF has gained popularity thanks to the work of Lee and Seung [3, 11], with a multiplicative update rule investigated to solve the optimization problem. In [12], an alternating nonnegativity constrained least squares method is applied. Smoothness is a frequently considered constraint. It enables less spiky vectors either by penalizing their ℓ_2-norm [13], or by reducing the fluctuation between neighboring values within a vector [14], or by considering a weighted-average term [15]. Sparseness, which allows the uniqueness and enhances interpretation, is imposed either with projections [16] or with ℓ_1 regularization [17].

The NMF and most of its variants are based on a linear mixing assumption. Kernel machines offer an elegant framework to derive nonlinear techniques based on linear ones, by mapping the data using some nonlinear function to a feature space, and applying the linear algorithm on the mapped data. The key idea is the kernel trick, where a kernel function allows to evaluate the inner product between transformed data without the need of an explicit knowledge of the mapping function.

Recently, a few attempts have been made to apply NMF in a feature space, in order to derive a nonlinear variant [18, 19]. Unfortunately, the results obtained in the feature space cannot be exploited (except for a decisional problem such as classification). One needs to get back from the feature space to the input space. This is the curse of the pre-image problem, a major drawback inherited from kernel machines [20, 21]. This ill-posed problem yields an even more difficult problem when dealing with the non-negativity of the result [22, 23].

In this paper, we revisit the NMF within the framework of kernel machines. We propose a novel kernel-NMF that does not suffer from the curse of the pre-image problem, by solving the problem in the input space. This is accomplished thanks to the nature of the kernel function. We derive two iterative algorithms: a gradient-based additive update rule and a multiplicative update rule. It turns out that the linear kernel yields the conventional linear NMF. We also investigate the cases of the polynomial and Gaussian kernels. Within the proposed framework for kernel-NMF, several extensions are derived by incorporating different constraints. The unconstrained (basic) kernel-NMF and its extensions are applied to unmix the synthetic hyperspectral image.

This work was supported by the French ANR, grand HYPANEMA: ANR-12BS03-0033.

On applying the NMF in a feature space

Recently, several attempts have been made to derive nonlinear, kernel-based, NMF. These methods originate in mapping, with a nonlinear function $\Phi(\cdot)$, the columns of the L by T investigated matrix $\boldsymbol{X}$, thus transforming each column $\boldsymbol{x}_t$ into $\Phi(\boldsymbol{x}_t)$ for $t = 1, \ldots, T$. Let $\kappa(\cdot, \cdot)$ be the reproducing kernel associated with the nonlinear map, and $\mathcal{H}$ the induced feature space. Written in the feature space, the NMF model is

$$\Phi(\boldsymbol{x}_t) \approx \sum_{n=1}^{N} a_{nt}\, \boldsymbol{e}_n^{\Phi}. \tag{1}$$

Here, the elements $\boldsymbol{e}_n^{\Phi}$ lie in the feature space $\mathcal{H}$. Let $\boldsymbol{X}^{\Phi} = \big[\Phi(\boldsymbol{x}_1), \Phi(\boldsymbol{x}_2), \cdots, \Phi(\boldsymbol{x}_T)\big]$, then we obtain in matrix form $\boldsymbol{X}^{\Phi} \approx \big[\boldsymbol{e}_1^{\Phi}, \boldsymbol{e}_2^{\Phi}, \cdots, \boldsymbol{e}_N^{\Phi}\big]\, \boldsymbol{A}$. Essentially, all kernel-based NMF proposed so far have been considering this model [18, 19]. Unfortunately, the model (1) suffers from an important weakness, inherited from kernel machines: one has no access to the elements in the feature space, but only to their inner products with the kernel function. The fact that the elements $\boldsymbol{e}_n^{\Phi}$ lie in the feature space $\mathcal{H}$ leads to several drawbacks, as shown next.

The aforementioned model satisfies: $\langle \Phi(\boldsymbol{x}_{t'}), \Phi(\boldsymbol{x}_t) \rangle \approx \sum_{n=1}^{N} a_{nt} \langle \Phi(\boldsymbol{x}_{t'}), \boldsymbol{e}_n^{\Phi} \rangle$, for all $t, t' = 1, \ldots, T$. Here, the left-hand-side is equivalent to $\kappa(\boldsymbol{x}_{t'}, \boldsymbol{x}_t)$. Unfortunately, the inner product $\langle \Phi(\boldsymbol{x}_{t'}), \boldsymbol{e}_n^{\Phi} \rangle$ cannot be evaluated using the kernel function. To circumvent this difficulty, one should restrict the form of $\boldsymbol{e}_n^{\Phi}$, as given in [18] by writing them as a linear combination of $\Phi(\boldsymbol{x}_t)$. By rearranging the coefficients of the linear combination in a matrix $\boldsymbol{W}$, the problem takes the form $\boldsymbol{X}^{\Phi} \approx \boldsymbol{X}^{\Phi} \boldsymbol{W} \boldsymbol{A}$. However, this expression is quiet different from the conventional NMF problem. Another downside is that one cannot impose the non-negativity of the elements in the feature space, namely $\boldsymbol{e}_n^{\Phi}$. Thus, the constraint $\boldsymbol{e}_n^{\Phi} \geq 0$ should be dropped. Only the coefficients a_{nt} can be set to nonnegative. This yields to the semi-NMF problem, where only the constraint $\boldsymbol{A} \geq 0$ is forced [19].

The most important drawback is that one has no access to the elements $\boldsymbol{e}_n^{\Phi}$. Having a given matrix $\boldsymbol{X}$, only the matrix $\boldsymbol{A}$ is determined. To estimate a matrix $\boldsymbol{E}$ such as $\boldsymbol{X}^{\Phi} \approx \boldsymbol{E}\boldsymbol{A}$, one needs to solve the so-called pre-image problem [20]. This ill-posed problem consists of estimating an input vector whose image, by the nonlinear map $\Phi(\cdot)$, is as close as possible to a given element in the feature space. In other words, one determines each column $\boldsymbol{e}_n$ of $\boldsymbol{E}$ by solving $\Phi(\boldsymbol{e}_n) \approx \boldsymbol{e}_n^{\Phi}$, for all $n = 1, \ldots, N$. This problem is obvious in all previous work on kernel-based NMF [24]. Including the non-negativity constraint to the pre-image problem is a challenging problem, as investigated recently in [23].

For all these reasons, applying the nonnegative matrix factorization in the feature space has been limited so far. Next, we propose a new kernel-NMF, where the resulting matrices are defined in the input space, and therefore without the pain of solving the pre-image problem.

2. THE PROPOSED KERNEL-NMF

We consider the following matrix factorization problem:

$$\boldsymbol{X}^{\Phi} \approx \boldsymbol{E}^{\Phi} \boldsymbol{A}.$$

where $\boldsymbol{E}^{\Phi} = [\Phi(\boldsymbol{e}_1), \Phi(\boldsymbol{e}_2), \cdots, \Phi(\boldsymbol{e}_N)]$. The non-negativity constraint is imposed to $\boldsymbol{A}$ and $\boldsymbol{e}_n$ for all $n = 1, \ldots, N$. Therefore, we have the following model:

$$\Phi(\boldsymbol{x}_t) \approx \sum_{n=1}^{N} a_{nt}\, \Phi(\boldsymbol{e}_n).$$

This means that we are estimating the elements $\boldsymbol{e}_n$ directly in the input space, as opposed to the model given in (1) where the elements $\boldsymbol{e}_n^{\Phi}$ lie in the feature space.

To estimate all $\boldsymbol{e}_n$ and a_{nt}, we consider a simple alternating to minimize the following cost function:

$$J = \frac{1}{2} \sum_{t=1}^{T} \|\Phi(\boldsymbol{x}_t) - \sum_{n=1}^{N} a_{nt}\, \Phi(\boldsymbol{e}_n)\|_{\mathcal{H}}^2, \tag{2}$$

where $\|\cdot\|_{\mathcal{H}}$ is the norm in the feature space $\mathcal{H}$, i.e., $\|\cdot\|_{\mathcal{H}}^2 = \langle \cdot, \cdot \rangle$. By expanding the above expression, the optimization problem becomes:

$$\min_{a_{nt}, \boldsymbol{e}_n} \sum_{t=1}^{T} \Big(-\sum_{n=1}^{N} a_{nt} \kappa(\boldsymbol{e}_n, \boldsymbol{x}_t) + \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} a_{nt} a_{mt} \kappa(\boldsymbol{e}_n, \boldsymbol{e}_m) \Big),$$

where $\kappa(\boldsymbol{x}_t, \boldsymbol{x}_t)$ is removed from the expression since it is independent of a_{nt} and $\boldsymbol{e}_n$. By taking its derivative with respect to a_{nt}, we obtain the following expression: $\nabla_{a_{nt}} J = -\kappa(\boldsymbol{e}_n, \boldsymbol{x}_t) + \sum_{m=1}^{N} a_{mt}\, \kappa(\boldsymbol{e}_n, \boldsymbol{e}_m)$. By taking the gradient of J with respect to the vector $\boldsymbol{e}_n$, we obtain:

$$\nabla_{\boldsymbol{e}_n} J = \sum_{t=1}^{T} a_{nt} \Big(-\nabla_{\boldsymbol{e}_n} \kappa(\boldsymbol{e}_n, \boldsymbol{x}_t) + \sum_{m=1}^{N} a_{mt}\, \nabla_{\boldsymbol{e}_n} \kappa(\boldsymbol{e}_n, \boldsymbol{e}_m) \Big). \tag{3}$$

Here, $\nabla_{\boldsymbol{e}_n} \kappa(\boldsymbol{e}_n, \cdot)$ denotes the gradient of the kernel with respect to its argument $\boldsymbol{e}_n$. Expressions can be easily derived using most valid kernels, as given in Section 2.3. But before, we derive two iterative algorithms for solving the above kernel-NMF, by alternating the estimation of a_{nt} and $\boldsymbol{e}_n$.

2.1. Additive update rule

By using a gradient descent scheme, we update a_{nt} according to $a_{nt} = a_{nt} - \eta_{nt}\, \nabla_{a_{nt}} J$, where the stepsize η_{nt} can take different values for each pair (n, t). Replacing $\nabla_{a_{nt}} J$ with its expression, we get the following update rule:

$$a_{nt} = a_{nt} - \eta_{nt} \Big(\sum_{m=1}^{N} a_{mt}\, \kappa(\boldsymbol{e}_n, \boldsymbol{e}_m) - \kappa(\boldsymbol{e}_n, \boldsymbol{x}_t) \Big). \tag{4}$$

A similar procedure is applied to estimate the elements $\boldsymbol{e}_n$. The obtained update rule is given by

$$\boldsymbol{e}_n = \boldsymbol{e}_n - \eta_n \nabla_{\boldsymbol{e}_n} J, \tag{5}$$

where the stepsize η_n can depend on n. To impose the nonnegativity of the matrices, the negative values obtained by the above update rules are set to zero at each iteration.

2.2. Multiplicative update rule

The additive update rule is a simple procedure, however, the convergence is generally slow, and is directly related to the stepsize value used. In order to overcome these issues, we propose a multiplicative update rule.

To derive a multiplicative update rule for a_{nt}, the stepsize η_{nt} in (4) is chosen such that the first and the third terms in its right-hand-side cancel. Therefore, we get:

$$a_{nt} = a_{nt} \times \frac{\kappa(\boldsymbol{e}_n, \boldsymbol{x}_t)}{\sum_{m=1}^{N} a_{mt}\, \kappa(\boldsymbol{e}_n, \boldsymbol{e}_m)}. \tag{6}$$

A similar procedure is applied to estimate the elements $\boldsymbol{e}_n$, for $n = 1, \ldots, N$. The trick is that the expression of the gradient (3) can always be decomposed as $\nabla_{\boldsymbol{e}_n} J = P - Q$, where P and Q have nonnegative entries. This is called the split gradient method [25]. It is obvious that this decomposition is not unique. Still, one can provide a multiplicative update for a given kernel function, as shown next.

2.3. Kernels

In this section, expressions from the three most known kernels are provides.

2.3.1. Back to the conventional linear NMF

A key property of the proposed kernel-NMF framework is that the conventional NMF is a special case, when the linear kernel is used with $\kappa(\boldsymbol{e}_n, \boldsymbol{z}) = \boldsymbol{z}^\top \boldsymbol{e}_n$, for any vector $\boldsymbol{z}$ from the input space. By substituting $\nabla_{\boldsymbol{e}_n} \kappa(\boldsymbol{e}_n, \boldsymbol{z}) = \boldsymbol{z}$ in the above expressions, we get the additive update rules

$$\left\{ \begin{array}{ll} a_{nt} & = a_{nt} - \eta_{nt} \Big(\sum_{m=1}^{N} a_{mt}\, \boldsymbol{e}_m^\top \boldsymbol{e}_n - \boldsymbol{x}_t^\top \boldsymbol{e}_n \Big); \\ \boldsymbol{e}_n & = \boldsymbol{e}_n - \eta_n \sum_{t=1}^{T} a_{nt} \Big(-\boldsymbol{x}_t + \sum_{m=1}^{N} a_{mt}\, \boldsymbol{e}_m \Big), \end{array} \right.$$

as well as the multiplicative update rules

$$\left\{ \begin{array}{ll} a_{nt} & = a_{nt} \times \dfrac{\boldsymbol{x}_t^\top \boldsymbol{e}_n}{\sum_{m=1}^{N} a_{mt}\, \boldsymbol{e}_m^\top \boldsymbol{e}_n}; \\ \boldsymbol{e}_n & = \boldsymbol{e}_n \otimes \dfrac{\sum_{t=1}^{T} a_{nt}\, \boldsymbol{x}_t}{\sum_{t=1}^{T} a_{nt} \sum_{m=1}^{N} a_{mt}\, \boldsymbol{e}_m}. \end{array} \right. \tag{7}$$

In the latter expression for updating $\boldsymbol{e}_n$, the element-wise operations are used with the division and multiplication, the latter being the Hadamard product given by $\otimes$.

2.3.2. The polynomial kernel

The polynomial kernel is defined as $\kappa(\boldsymbol{e}_n, \boldsymbol{z}) = (\boldsymbol{z}^\top \boldsymbol{e}_n + \mathrm{c})^d$. Here, c is a nonnegative constant. Concerning the most common quadratic polynomial kernel with $d = 2$, the additive update rules are

$$\left\{ \begin{array}{l} a_{nt} = a_{nt} - \eta_{nt} \Big(\sum_{m=1}^{N} a_{mt} (\boldsymbol{e}_m^\top \boldsymbol{e}_n + \mathrm{c})^2 - (\boldsymbol{x}_t^\top \boldsymbol{e}_n + \mathrm{c})^2 \Big); \\ \boldsymbol{e}_n = \boldsymbol{e}_n - \eta_n \sum_{t=1}^{T} a_{nt} \Big(-2 (\boldsymbol{x}_t^\top \boldsymbol{e}_n + \mathrm{c}) \boldsymbol{x}_t \\ \qquad + 2 \sum_{m=1}^{N} a_{mt} (\boldsymbol{e}_m^\top \boldsymbol{e}_n + \mathrm{c}) \boldsymbol{e}_m \Big), \end{array} \right.$$

and the multiplicative update rules are

$$\left\{ \begin{array}{ll} a_{nt} & = a_{nt} \times \dfrac{(\boldsymbol{x}_t^\top \boldsymbol{e}_n + \mathrm{c})^2}{\sum_{m=1}^{N} a_{mt}\, (\boldsymbol{e}_m^\top \boldsymbol{e}_n + \mathrm{c})^2}; \\ \boldsymbol{e}_n & = \boldsymbol{e}_n \otimes \dfrac{\sum_{t=1}^{T} a_{nt} (\boldsymbol{x}_t^\top \boldsymbol{e}_n + \mathrm{c}) \boldsymbol{x}_t}{\sum_{t=1}^{T} a_{nt} \sum_{m=1}^{N} a_{mt} (\boldsymbol{e}_m^\top \boldsymbol{e}_n + \mathrm{c}) \boldsymbol{e}_m}. \end{array} \right. \tag{8}$$

2.3.3. The Gaussian kernel

The Gaussian kernel is defined by $\kappa(\boldsymbol{e}_n, \boldsymbol{z}) = \exp(\frac{-1}{2\sigma^2} \| \boldsymbol{e}_n - \boldsymbol{z} \|^2)$, with the gradient $\nabla_{\boldsymbol{e}_n} \kappa(\boldsymbol{e}_n, \boldsymbol{z}) = -\frac{1}{\sigma^2} \kappa(\boldsymbol{e}_n, \boldsymbol{z}) (\boldsymbol{e}_n - \boldsymbol{z})$. The update rules of a_{nt} can be easily derived, in both additive and multiplicative cases. For the estimation of $\boldsymbol{e}_n$, the additive rule is

$$\begin{aligned} \boldsymbol{e}_n = \boldsymbol{e}_n - \eta_n \Big(& + \tfrac{1}{\sigma^2} \sum_{t=1}^{T} a_{nt}\, \kappa(\boldsymbol{e}_n, \boldsymbol{x}_t) (\boldsymbol{e}_n - \boldsymbol{x}_t) \\ & - \tfrac{1}{\sigma^2} \sum_{t=1}^{T} \sum_{m=1}^{N} a_{nt} a_{mt}\, \kappa(\boldsymbol{e}_n, \boldsymbol{e}_m) (\boldsymbol{e}_n - \boldsymbol{e}_m) \Big), \end{aligned}$$

and the multiplicative algorithm is

$$\boldsymbol{e}_n = \boldsymbol{e}_n \otimes \frac{\sum_{t=1}^{T} a_{nt} \Big(\boldsymbol{x}_t\, \kappa(\boldsymbol{e}_n, \boldsymbol{x}_t) + \sum_{m=1}^{N} a_{mt}\, \boldsymbol{e}_n\, \kappa(\boldsymbol{e}_n, \boldsymbol{e}_m) \Big)}{\sum_{t=1}^{T} a_{nt} \Big(\boldsymbol{e}_n\, \kappa(\boldsymbol{e}_n, \boldsymbol{x}_t) + \sum_{m=1}^{N} a_{mt}\, \boldsymbol{e}_m\, \kappa(\boldsymbol{e}_n, \boldsymbol{e}_m) \Big)}, \tag{9}$$

where the division is component-wise.

3. EXTENSIONS OF KERNEL-NMF

Our approach for kernel-NMF can be easily extended to include constraints on the endmember and abundance matrix when taking into account the structure of the investigated data, *i.e.*, the hyperspectral image. When flattening such three-dimensional image into a matrix $\boldsymbol{X}$, the t-th column of $\boldsymbol{X}$ is filled with the (i, j)-th spectrum from the original image of size a by b, with $i = \lceil \frac{t}{b} \rceil$ and $j = t - (i-1)b$. Recall that $\boldsymbol{E} = [\boldsymbol{e}_1, \boldsymbol{e}_2, \cdots, \boldsymbol{e}_N]$ represents the endmember matrix, where column $\boldsymbol{e}_n$ is the n-th estimated endmember. The n-th row $\boldsymbol{a}_n$ of the abundance matrix $\boldsymbol{A} = [\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots, \boldsymbol{a}_N]^\top$ is used to construct the abundance map, denoted $\mathbf{M}_n$. In the following, we outline several extensions. It is worth noting

that we can certainly acquire more desirable unmixing performances with the proposed extensions, simply because that they will return to basic kernel-NMF when the regularization parameter is null. Due to space limitation, only the multiplicative updates are given in the following.

3.1. Kernel-NMF with constraints on the endmembers

In this subsection, constraints are added on the endmembers. Since the derivatives with respect to a_{nt} remain unchanged, update rules for a_{nt} are identical to the ones given in Section 2. Update rules for $\boldsymbol{e}_n$ are given next, for three different regularizations.

3.1.1. Smoothness with 2-norm regularization

In the similar spirit of [13], we introduce an input-space penalty term $\frac{\lambda}{2}\sum_{n=1}^{N}\|\boldsymbol{e}_n\|^2$ to the cost function (2), thus making each endmember less spiky. The balance between the reconstruction accuracy and the smoothness term is adjustable with the positive parameter λ. Using the split gradient method [25], we get the corresponding multiplicative update rules for each kernel function. It turns out that one gets similar expressions as in the unconstrained case, with (7), (8) or (9), where the term $\lambda\,\boldsymbol{e}_n$ is added to the denominator.

We can also consider a similar constraint in the feature space by including the term $\frac{\lambda_{\mathcal{H}}}{2}\sum_{n=1}^{N}\|\boldsymbol{e}_n\|^2_{\mathcal{H}}$ to (2). This leads to update rules similar to the above expressions, where $\lambda\,\boldsymbol{e}_n$ is replaced by $\lambda_{\mathcal{H}}\nabla_{\boldsymbol{e}_n}\kappa(\boldsymbol{e}_n,\boldsymbol{e}_n)$.

3.1.2. Smoothness with fluctuation regularization

Following [14], we add $\frac{\gamma}{2}\sum_{n=1}^{N}\sum_{l=2}^{L-1}|e_{ln}-e_{(l-1)n}|$ to (2) in order to force small fluctuations between neighboring values within each endmember. The derivative of the penalty term with respect to e_{ln} equals to:

$$\begin{cases} +\gamma & \text{if } e_{ln} < e_{(l-1)n} \quad \text{and} \quad e_{ln} < e_{(l+1)n}; \\ -\gamma & \text{if } e_{ln} > e_{(l-1)n} \quad \text{and} \quad e_{ln} > e_{(l+1)n}; \\ 0 & \text{otherwise.} \end{cases}$$

Incorporating the above expression into $\nabla_{\boldsymbol{e}_n}J$ (see (3)), we easily get the corresponding modified update rules.

3.1.3. Smoothness with weighted-average regularization

Another smoothness regularization raised by Chen and Cichocki [15] aims to reduce the difference between e_{ln} and a weighted average $\overline{e}_{ln}=\alpha\overline{e}_{(l-1)n}+(1-\alpha)e_{ln}$. The penalty term $\frac{\rho}{2L}\sum_{n=1}^{N}\|(\mathbf{I}-\mathbf{T})\boldsymbol{e}_n\|^2$ describes the sum of such effect over n, and its gradient with respect of $\boldsymbol{e}_n$ takes the form $\rho\,\mathbf{Q}\,\boldsymbol{e}_n$. The additive update rule of the endmembers is easy to derive using the descent gradient method. The multiplicative update rule depends on the used kernel, with expressions similar to (7), (8) and (9), by adding the term $\rho\,\mathbf{Q}\,\boldsymbol{e}_n$ to the denominator.

3.2. Kernel-NMF with constraints on the abundances

As for the constraints on the abundances, it turns out that the added penalty has no influence on the update rules of the endmembers. Update rules of a_{nt} are given next for sparseness and spatial regularizations.

3.2.1. Sparseness regularization

Sparseness has been proved to be very attractive in many disciplines, namely by penalizing with the ℓ_1-norm of the weight coefficients [16]. Introducing a sparseness term of abundance matrix $\mu\sum_{n=1}^{N}a_{nt}$ to the cost function (2), each spectrum $\boldsymbol{x}_t$ tends to be represented by using a few endmembers. The corresponding multiplicative update rule is obtained as:

$$a_{nt}=a_{nt}\times\frac{\kappa(\boldsymbol{e}_n,\boldsymbol{x}_t)}{\sum_{m=1}^{N}a_{mt}\,\kappa(\boldsymbol{e}_n,\boldsymbol{e}_m)+\mu}.$$

3.2.2. Spatial regularization

In [15], the authors take a one-direction spatial effect into consideration. Here, we extend the idea to the abundance map $\mathbf{M}_n$ (constructed by $\boldsymbol{a}_n$), $n=1,2,...,N$. For any inner element $\mathbf{M}_n(i,j)$ belonging to the n-th abundance map, the spatial effects from its geographical neighboring directions will count. We use the term $\sum_{n=1}^{N}R_{\mathrm{n}}$ to reflect this spatial effect, where R_{n} denotes the cost brought by the abundance map $\boldsymbol{a}_n$. The multiplicative update rule of this variant is:

$$a_{nt}=a_{nt}\times\frac{\kappa(\boldsymbol{e}_n,\boldsymbol{x}_t)}{\sum_{m=1}^{N}a_{mt}\,\kappa(\boldsymbol{e}_n,\boldsymbol{e}_m)+\mathbf{G}(i,j)}.$$

Here, $\mathbf{G}$ is a matrix that controls the spatial effect. The expression of this matrix is not given here due to space limitation.

4. EXPERIMENTAL RESULTS

The relevance of the proposed kernel-NMF and its extensions is studied on a synthetic image generated from the generalized bilinear model with endmembers Water, Cedar and Brick Building chosen from the USGS Digital Spectral Library. We apply seven metrics to assess the unmixing performance, in both input and feature spaces:

- Reconstruction error:
  $\mathrm{RE}=\sqrt{\frac{1}{TL}\sum_{t=1}^{T}\|\boldsymbol{x}_t-\sum_{n=1}^{N}a_{nt}\boldsymbol{e}_t\|^2}$
  $\mathrm{RE}^{\phi}=\sqrt{\frac{1}{TL}\sum_{t=1}^{T}\|\phi(\boldsymbol{x}_t)-\sum_{n=1}^{N}a_{nt}\phi(\boldsymbol{e}_t)\|^2_{\mathcal{H}}}$
- Root mean square error for endmember estimation:
  $\mathrm{RMSE_E}=\sqrt{\frac{1}{NL}\sum_{n=1}^{N}\|\boldsymbol{e}_n-\widehat{\boldsymbol{e}_n}\|^2}$
  $\mathrm{RMSE_E^{\phi}}=\sqrt{\frac{1}{NL}\sum_{n=1}^{N}\|\phi(\boldsymbol{e}_n)-\phi(\widehat{\boldsymbol{e}_n})\|^2_{\mathcal{H}}}$

- Spectral angle mapper:
  $\text{SAM} = \frac{1}{N}\sum_{n=1}^{N} \arccos\left(\frac{\boldsymbol{e}_n^\top \widehat{\boldsymbol{e}_n}}{\|\boldsymbol{e}_n\|\,\|\widehat{\boldsymbol{e}_n}\|}\right)$
  $\text{SAM}^\phi = \frac{1}{N}\sum_{n=1}^{N} \arccos\left(\frac{\kappa(\boldsymbol{e}_n,\widehat{\boldsymbol{e}_n})}{\sqrt{\kappa(\boldsymbol{e}_n,\boldsymbol{e}_n)}\sqrt{\kappa(\widehat{\boldsymbol{e}_n},\widehat{\boldsymbol{e}_n})}}\right)$
- Root mean square error for abundance estimation:
  $\text{RMSE}_\text{A} = \sqrt{\frac{1}{NT}\|\boldsymbol{A} - \widehat{\boldsymbol{A}}\|^2}$

For the studied image, experiments are firstly conducted with the unconstrained (basic) kernel-NMF multiplicative algorithm to determine the parameter of each kernel. The values of the parameters $\mathsf{c} = 0.5$ and $\sigma = 6.0$ are then fixed for all kernel-NMF extensions. In the search for appropriate regularization parameter for the different kernel-NMF extensions, the candidate regularization parameter set $\{0, 2^{-5}, 2^{-4}, \ldots, 2^{20}\}$ is used. The number of iterations is fixed to 100. The results are given in Table 1. As it can be observed, the Gaussian kernel enables the optimal results in the feature space, while the polynomial kernel mainly improves the results in the input space. Additionally, the kernel-NMF outperforms the conventional linear NMF. Moreover, extensions are more efficient for appropriate regularization parameters.

5. CONCLUSION

In this paper, we presented a new kernel-based NMF, where the matrices are estimated in the input space. This approach circumvents the curse of the pre-image problem. Several extensions including regularization constraints were proposed within this framework. The relevance of these techniques was illustrated on synthetic data. As for future work, extensive experiments are conducted on real hyperspectral images. We also aim to extend this work to provide pre-image-free methods for principal component analysis and tensor decomposition.

6. REFERENCES

[1] P. Comon and C. Jutten, *Handbook of Blind Source Separation: Independent Component Analysis and Applications*, Academic Press, 1st edition, 2010.

[2] P. Paatero and U. Tapper, "Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values," *Environmetrics*, vol. 5, no. 2, pp. 111–126, 1994.

[3] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization.," *Nature*, vol. 401, no. 6755, pp. 788–791, Oct. 1999.

[4] G. Buchsbaum and O. Bloch, "Color categories revealed by non-negative matrix factorization of munsell color spectra.," *Vision Research*, vol. 42, no. 5, pp. 559–63, 2002.

[5] I. Buciu and I. Pitas, "Application of non-negative and local non negative matrix factorization to facial expression recognition," in *17th International Conference on Pattern Recognition*, Cambridge, UK, 2004, vol. 1, pp. 288–291.

[6] S. Wild, J. Curry, and A. Dougherty, "Improving non-negative matrix factorizations through structured initialization," *Pattern Recognition*, vol. 37, no. 11, pp. 2217–2232, Nov. 2004.

[7] J.-P. Brunet, P. Tamayo, T. R Golub, and J. P Mesirov, "Metagenes and molecular pattern discovery using matrix factorization," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 101, no. 12, pp. 4164–4169, 2004.

[8] D.C. Heinz and C.-I Chang, "Fully constrained least squares linear mixture analysis for material quantification in hyperspectral imagery," *IEEE Trans. on Geoscience and Remote Sensing*, vol. 39, pp. 529–545, 2001.

[9] N. Yokoya, J. Chanussot, and A. Iwasaki, "Nonlinear unmixing of hyperspectral data using semi-nonnegative matrix factorization," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 52, no. 2, pp. 1430–1437, Feb 2014.

[10] D.J. Leggett, "Numerical analysis of multicomponent spectra," *Analytical Chemistry*, vol. 49, pp. 276—281, 1977.

[11] D. D. Lee and H. S. Seung, "Algorithms for Non-negative Matrix Factorization," in *Advances in Neural Information Processing Systems 13*, Todd K. Leen, Thomas G. Dietterich, and Volker Tresp, Eds. Apr. 2001, pp. 556–562, MIT Press.

[12] H. Kim and H. Park, "Nonnegative matrix factorization based on alternating nonnegativity constrained least squares and active set method," *SIAM Journal on Matrix Analysis and Applications*, vol. 30, no. 2, pp. 713–730, July 2008.

[13] J. Piper, V. Paul Pauca, Robert J. Plemmons, and Maile Giffin, "Object characterization from spectral data using nonnegative factorization and information theory," in *In Proc. Amos Technical Conf., Maui, HI*, 2004.

[14] T. Virtanen, "Sound source separation using sparse coding with temporal continuity objective," *Proc. ICMC*, vol. 3, pp. 231–234, 2003.

[15] Z. Chen and A. Cichocki, "Nonnegative matrix factorization with temporal smoothness and/or spatial decorrelation constraints," in *Laboratory for Advanced Brain Signal Processing, RIKEN, Tech. Rep*, 2005, vol. 68.

[16] P. O. Hoyer, "Non-negative matrix factorization with sparseness constraints," *Journal Machine Learning Research*, vol. 5, pp. 1457–1469, Dec. 2004.

[17] H. Kim and H. Park, "Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis," *Bioinformatics*, vol. 23, no. 12, pp. 1495–1502, June 2007.

[18] C. Ding, T. Li, and M. I. Jordan, "Convex and Semi-Nonnegative Matrix Factorizations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 1, pp. 45–55, Nov. 2010.

[19] Y. Li and A. Ngom, "A new kernel non-negative matrix factorization and its application in microarray data analysis," in

Table 1: Unmixing performance

			Reconstruction		Endmember				Abundance
			RE ($\times 10^{-2}$)	RE$^{\phi}$ ($\times 10^{-2}$)	RMSE$_{\mathrm{E}}$ ($\times 10^{-2}$)	RMSE$_{\mathrm{E}}^{\phi}$ ($\times 10^{-2}$)	SAM ($\times 10^{-2}$)	SAM$^{\phi}$ ($\times 10^{-2}$)	RMSE$_{\mathrm{A}}$ ($\times 10^{-2}$)
	FCLS [26, 8]		3.08	-	17.83	-	33.21	-	45.93
	Semi-NMF [27, 9]		1.90	-	26.11	-	35.56	-	30.51
	K-Hype [27, 28]		3.20	-	26.11	-	35.56	-	24.23
	MinDisCo [29]		1.76	-	23.93	-	70.78	-	19.06
Kernel-NMF and its extensions	J	Lin-Multi	1.50	1.50	28.20	28.20	37.63	37.63	26.25
		Poly-Multi	6.52	64.21	24.91	201.37	34.24	89.73	47.27
		Gauss-Multi	2.06	0.42	8.78	1.44	28.57	28.22	18.59
	$J_{\text{2-norm}}$	Lin-Multi	1.84	1.84	23.21	23.21	28.67	28.67	34.46
		Poly-Multi	8.66	67.69	5.62	72.57	29.25	31.86	38.85
		Gauss-Multi	2.03	0.41	8.79	1.44	28.54	28.57	18.39
	$J^{\mathcal{H}}_{\text{2-norm}}$	Poly-Multi	3.28	22.28	28.44	254.44	33.47	69.52	41.22
	J_{fluct}	Lin-Multi	1.49	1.49	28.06	28.06	36.28	36.28	26.37
		Poly-Multi	13.51	104.63	21.02	187.36	31.87	72.9	41.24
		Gauss-Multi	2.04	0.42	8.39	1.37	26.85	26.49	17.73
	J_{averaged}	Lin-Multi	1.52	1.52	25.50	25.50	28.60	28.60	28.32
		Poly-Multi	10.36	109.61	15.14	90.43	26.45	83.29	43.51
		Gauss-Multi	2.00	0.41	8.41	1.38	26.82	26.78	17.49

IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology, CIBCB, San Diego, CA, USA, May 9-12 2012, pp. 371–378.

[20] P. Honeine and C. Richard, "Preimage problem in kernel-based machine learning," *IEEE Signal Processing Magazine*, vol. 28, no. 2, pp. 77–88, 2011.

[21] P. Honeine and C. Richard, "A closed-form solution for the pre-image problem in kernel-based machines," *Journal of Signal Processing Systems*, vol. 65, no. 3, pp. 289–299, December 2011.

[22] M. Kallas, P. Honeine, C. Richard, C. Francis, and H. Amoud, "Non-negative pre-image in machine learning for pattern recognition," in *Proc. 19th European Conference on Signal Processing*, Barcelona, Spain, 29 Aug. - 2 Sept. 2011.

[23] M. Kallas, P. Honeine, C. Richard, C. Francis, and H. Amoud, "Non-negativity constraints on the pre-image for pattern recognition with kernel machines," *Pattern Recognition*, vol. 46, no. 11, pp. 3066 – 3080, 2013.

[24] B.-B. Pan, J.-H. Lai, and W.-S. Chen, "Nonlinear nonnegative matrix factorization based on Mercer kernel construction," *Pattern Recognition*, vol. 44, no. 10-11, pp. 2800 – 2810, 2011, Semi-Supervised Learning for Visual Content Analysis and Understanding.

[25] H. Lantéri, C. Theys, C. Richard, and D. Mary, "Regularized split gradient method for nonnegative matrix factorization," in *ICASSP*, 2011, pp. 1133–1136.

[26] M.E. Winter, "N-FINDR: an algorithm for fast autonomous spectral end-member determination in hyperspectral data: an algorithm for fast autonomous spectral end-member determination in hyperspectral data," *Proc. of SPIE: Imaging Spectrometry V*, vol. 3753, no. 10, 1999.

[27] Jose M. P. Nascimento and Jose M. B. Dias, "Vertex component analysis: A fast algorithm to unmix hyperspectral data," *IEEE Trans. Geoscience and Remote Sensing*, vol. 43, no. 4, pp. 898–910, 2004.

[28] J. Chen, C. Richard, and P. Honeine, "Nonlinear unmixing of hyperspectral data based on a linear-mixture/nonlinear-fluctuation model," *IEEE Transactions on Signal Processing*, vol. 61, no. 2, pp. 480–492, 2013.

[29] A. Huck, M. Guillaume, and J. Blanc-Talon, "Minimum dispersion constrained nonnegative matrix factorization to unmix hyperspectral data," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 48, no. 6, pp. 2590–2602, June 2010.