

Eviter la malédiction de pré-image : application à la factorisation en matrices non négatives à noyaux

Paul HONEINE, Fei ZHU

Institut Carles Delaunay (UMR CNRS 6281)
Université de Technologie de Troyes, Troyes, France
paul.honeine@utt.fr, fei.zhu@utt.fr

Résumé – Les méthodes à noyaux reposent sur une transformation non linéaire des échantillons. L’estimation de la transformation inverse, dite pré-image, est omniprésente dans divers domaines d’application. Sa résolution est d’une grande utilité, en ouvrant la voie à de nouveaux domaines d’application des méthodes à noyaux, notamment en reconnaissance de formes et débruitage de signaux/images. En illustrant cette malédiction de pré-image avec le problème de factorisation en matrices non négatives, la présente communication propose de surmonter cette difficulté par la mise en oeuvre d’un nouveau paradigme de modèles non linéaires. Différents algorithmes de résolution sont présentés et une étude comparative montre la pertinence de l’approche proposée sur le problème de démixage en imagerie hyperspectral.

Abstract – Kernel-based methods rely on a nonlinear transformation of available data. The so-called pre-image problem is the estimation of the inverse transformation. Solving this problem is of great interest in many fields, including pattern recognition and denoising problems. In this paper, we propose to overcome the pre-image problem by defining a novel model. This approach is illustrated on the nonnegative matrix factorization problem. The relevance of the proposed approach is illustrated on the unmixing problem in hyperspectral imagery.

1 Introduction

En apprentissage statistique et automatique, les méthodes dites à noyaux puisent leur force dans deux résultats clés que sont le Théorème de Représentation [25, 24] et le coup du noyau [1] (dit *kernel trick* en anglais). Le premier démontre que la solution d’un problème d’apprentissage régularisé est linéaire dans les paramètres, selon le modèle

$$\psi = \sum_{j=1}^n \alpha_j \phi(\mathbf{x}_j),$$

pour un ensemble d’apprentissage $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$, où $\phi(\cdot)$ est une transformation non linéaire de l’espace d’observation \mathcal{X} à un autre espace \mathcal{H} . Le second stipule qu’on n’a pas besoin d’exhiber la transformation $\phi(\cdot)$, puisque seuls les produits scalaires dans l’espace transformé \mathcal{H} sont nécessaires pour estimer les paramètres α_j . Ces produits scalaires sont obtenus par l’utilisation d’un noyau $\kappa(\mathbf{x}_i, \mathbf{x}_j)$, qui définit implicitement la transformation non linéaire $\phi(\cdot)$.

Les motivations historiques des méthodes à noyaux, à savoir la classification et la régression, ne nécessitent pas la connaissance explicite de ψ . C’est le cas par exemple en détection et classification, où la discrimination d’une observation \mathbf{x} est donnée par une règle de décision qui consiste à comparer $\sum_{j=1}^n \alpha_j \kappa(\mathbf{x}_j, \mathbf{x})$ à un seuil. Ainsi nul n’a besoin ni d’explicitement la transformation induite par le noyau, ni de représenter les images de échantillons d’apprentissage.

Toutefois, rien n’empêche de s’intéresser à ψ en tant qu’une caractéristique dans l’espace \mathcal{H} , ou encore à son équivalent

dans l’espace observable \mathcal{X} . Le retour inverse, de l’espace transformé à l’espace des observations, est alors primordial. A titre illustratif avec le débruitage de signaux/images, bien que le débruitage se fasse dans l’espace transformé, par exemple en projetant sur un sous-espace pertinent identifié par l’ACP-à-noyaux, le résultat final devra apparaître dans l’espace des observations. Ainsi, un signal est-il débruité en un signal dans le même espace initial, c’est à dire l’espace des observations. Au delà du débruitage, le retour inverse ouvre la voie à de nouveaux domaines d’application des méthodes à noyaux, dont la modélisation autoregressive de séries temporelles [14], le débruitage d’images [15], le démixage en imagerie hyperspectrale [22], ou encore l’auto-localisation de capteurs dans un réseau sans fil [9].

Malheureusement, il s’avère que le retour, de l’espace \mathcal{H} à l’espace \mathcal{X} , est un problème inverse mal-posé. Le problème dit de pré-image consiste à déterminer l’élément \mathbf{x}^* de l’espace des observations \mathcal{X} dont l’image $\phi(\mathbf{x}^*)$ est une bonne approximation de la caractéristique étudiée ψ . Plusieurs équipes de recherche ont étudié la résolution du problème de pré-image, notamment une forme explicite proposée dans [11]. Voir [12] pour une revue complète. Ces méthodes de résolution opèrent en dernière étape, une fois que la caractéristique ψ a été déterminée par une méthode d’apprentissage classique.

Le présent article montre que l’on peut éviter le problème de pré-image, en l’intégrant dans l’algorithme d’apprentissage initial ; il s’agit alors de proposer un problème d’optimisation global, avec une résolution en une seule étape. Dans la suite, nous démontrons cette approche dans le cadre de la factorisa-

tion en matrices non négatives (NMF pour *nonnegative matrix factorization*). La suite de cet article est organisée ainsi. La section suivante présente la malédiction de pré-image au travers des tentatives classiques de *kernéliser* la NMF. A la section 3, nous proposons une nouvelle formulation qui permet d'éviter cette malédiction. La pertinence de cette approche est illustrée par des simulations à la section 4.

2 La malédiction de pré-image illustrée par le problème de NMF

La factorisation en matrices non négatives (NMF) est très connue en traitement du signal et des images [18, 6]. Elle a été appliquée avec succès pour la classification d'images [2], la reconnaissance de l'expression du visage [4], la reconnaissance d'objets [21, 26], ou encore en bioinformatique [7, 16, 19].

Elle consiste à approximer une matrice non négative par deux matrices non négatives de faible rang [18]. Pour une matrice \mathbf{X} non négative, la factorisation est de la forme

$$\mathbf{X} \approx \mathbf{A}\mathbf{S}, \quad (1)$$

sous contraintes $\mathbf{A} \geq 0$ et $\mathbf{S} \geq 0$. Soient $\mathbf{X} = [\mathbf{x}_1 \ \mathbf{x}_2 \ \cdots \ \mathbf{x}_n]$, $\mathbf{A} = [\mathbf{a}_1 \ \mathbf{a}_2 \ \cdots \ \mathbf{a}_m]$, et $s_{j,i}$ la (j, i) -ème composante de la matrice \mathbf{S} . La NMF consiste à estimer $\mathbf{a}_j \geq 0$ et $s_{j,i} \geq 0$, pour tout $j = 1, \dots, m$ et $i = 1, \dots, n$, tels que

$$\mathbf{x}_i \approx \sum_{j=1}^m s_{j,i} \mathbf{a}_j. \quad (2)$$

Par souci de simplification, nous utilisons par convention $i = 1, 2, \dots, n$, $j = 1, 2, \dots, m$, avec $m < n$. Avec la notation proposée ici, nous avons un ensemble d'apprentissage $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$, avec $\mathbf{x}_i \in \mathcal{X}$ où \mathcal{X} désigne l'espace des observations.

Plusieurs tentatives ont été faites pour développer des méthodes NMF non linéaires, dans le cadre des méthodes à noyaux [28, 8, 20]. L'approche qui a été considérée consiste à appliquer une fonction non linéaire $\phi(\cdot)$ aux colonnes de \mathbf{X} , transformant ainsi chaque \mathbf{x}_i en $\phi(\mathbf{x}_i)$. Soient $\kappa(\cdot, \cdot)$ le noyau associé à cette transformation non linéaire, et \mathcal{H} l'espace associé. Écrit dans cet espace, le modèle NMF devient

$$\phi(\mathbf{x}_i) \approx \sum_{j=1}^m s_{j,i} \mathbf{a}_j^\phi. \quad (3)$$

Ici, les éléments \mathbf{a}_j^ϕ appartiennent à l'espace \mathcal{H} . Soit $\mathbf{X}^\phi = [\phi(\mathbf{x}_1) \ \phi(\mathbf{x}_2) \ \cdots \ \phi(\mathbf{x}_n)]$, le modèle (3) devient

$$\mathbf{X}^\phi \approx [\mathbf{a}_1^\phi \ \mathbf{a}_2^\phi \ \cdots \ \mathbf{a}_m^\phi] \mathbf{S}.$$

Le modèle présenté ici a été investi dans [28, 8, 20]. Malheureusement, ce modèle souffre d'une faiblesse majeure, héritée des méthodes à noyaux : nous n'avons pas accès aux éléments de \mathcal{H} . Cette malédiction de pré-image est illustrée ici avec les éléments \mathbf{a}_j^ϕ qui appartiennent à \mathcal{H} .

Il existe un autre inconvénient majeur du modèle (3) : il n'est pas direct d'imposer la non-négativité des éléments dans l'espace fonctionnel \mathcal{H} , et en particulier \mathbf{a}_j^ϕ . Par conséquent, la contrainte $\mathbf{a}_j^\phi \geq 0$ devrait être abandonnée. Seuls les coefficients $s_{j,i}$ peuvent être non négatifs. Dans ce cas, il ne s'agit plus du problème NMF, mais plutôt du problème semi-NMF, où seule la contrainte $\mathbf{S} \geq 0$ est imposée, comme présenté dans [20]. Pour surmonter cette difficulté, les auteurs de [23] proposent d'approximer le noyau par une fonction associée à une transformation non négative, ce qui nécessite de résoudre un autre problème d'optimisation en pré-traitement, avant l'application de la NMF. En outre, le problème de pré-image doit être résolu par la suite.

Pour toutes ces raisons, l'application de la factorisation en matrices non négatives dans l'espace transformé a été limitée jusqu'ici à des problèmes de classification, ou encore au noyau homogène [3]. Dans la suite, nous démontrons que nous pouvons estimer les matrices \mathbf{A} et \mathbf{S} dans l'espace des observations, sans souffrir de la malédiction du problème de pré-image.

3 Méthode NMF-à-noyaux

Nous proposons une nouvelle méthode NMF-à-noyaux, où les matrices résultantes sont définies dans l'espace d'entrée, et donc sans la nécessité de résoudre le problème de pré-image. A cette fin, nous considérons le modèle de factorisation suivant :

$$\mathbf{X}^\phi \approx \mathbf{A}^\phi \mathbf{S},$$

où $\mathbf{A}^\phi = [\phi(\mathbf{a}_1) \ \phi(\mathbf{a}_2) \ \cdots \ \phi(\mathbf{a}_m)]$. Par conséquent, nous avons le modèle suivant :

$$\phi(\mathbf{x}_i) \approx \sum_{j=1}^m s_{j,i} \phi(\mathbf{a}_j).$$

Cela signifie que nous estimons les éléments \mathbf{a}_j directement dans l'espace des observations \mathcal{X} , par opposition au modèle (3) où les éléments \mathbf{a}_j^ϕ appartiennent à \mathcal{H} . La contrainte de non-négativité est imposée à la matrice \mathbf{S} et aux vecteurs \mathbf{a}_j , pour tout $j = 1, 2, \dots, m$.

Afin d'estimer tous les \mathbf{a}_j et $s_{j,i}$, nous considérons une technique de moindres carrés alternés pour résoudre le problème d'optimisation suivant :

$$\min_{s_{j,i}, \mathbf{a}_j} \frac{1}{2} \sum_{i=1}^n \left\| \phi(\mathbf{x}_i) - \sum_{j=1}^m s_{j,i} \phi(\mathbf{a}_j) \right\|_{\mathcal{H}}^2. \quad (4)$$

En développant l'expression ci-dessus, le problème d'optimisation devient :

$$\min_{s_{j,i}, \mathbf{a}_j} \sum_{i=1}^n \left(- \sum_{j=1}^m s_{j,i} \kappa(\mathbf{a}_j, \mathbf{x}_i) + \frac{1}{2} \sum_{j=1}^m \sum_{j'=1}^m s_{j,i} s_{j',i} \kappa(\mathbf{a}_j, \mathbf{a}_{j'}) \right).$$

Soit \mathcal{J} la fonction coût dans cette expression. Sa dérivée par rapport à $s_{j,i}$ est donnée par l'expression suivante :

$$\nabla_{s_{j,i}} \mathcal{J} = -\kappa(\mathbf{a}_j, \mathbf{x}_i) + \sum_{j'=1}^m s_{j',i} \kappa(\mathbf{a}_j, \mathbf{a}_{j'}),$$

et son gradient par rapport au vecteur \mathbf{a}_j est :

$$\nabla_{\mathbf{a}_j} \mathcal{J} = \sum_{i=1}^n s_{j,i} \left(-\nabla_{\mathbf{a}_j} \kappa(\mathbf{a}_j, \mathbf{x}_i) + \sum_{j'=1}^m s_{j',i} \nabla_{\mathbf{a}_j} \kappa(\mathbf{a}_j, \mathbf{a}_{j'}) \right). \quad (5)$$

Ici, $\nabla_{\mathbf{a}_j} \kappa(\mathbf{a}_j, \cdot)$ désigne le gradient du noyau par rapport à son argument \mathbf{a}_j . Ses expressions peuvent être facilement obtenues pour les différents noyaux. Voir plus loin pour le cas du noyau gaussien. Mais avant, nous décrivons deux algorithmes itératifs pour résoudre le problème NMF-à-noyaux, en alternant l'estimation de $s_{j,i}$ et \mathbf{a}_j . Dans la suite, nous désignons par $s_{j,i,t+1}$ et $\mathbf{a}_{j,t+1}$ leurs estimations à l'itération t .

3.0.1 Algorithme avec une mise à jour additive

Dans le premier algorithme, nous proposons une règle de mise à jour additive pour résoudre le problème d'optimisation. Chaque itération est basée, d'abord sur une descente de gradient, en alternant sur $s_{j,i}$ et \mathbf{a}_j , et ensuite sur une fonction de redressement pour imposer la non-négativité.

En utilisant une descente de gradient, le coefficient $s_{j,i,t}$ est actualisé à l'itération t selon

$$s_{j,i,t+1} = s_{j,i,t} - \eta_{j,i,t} \nabla_{s_{j,i,t}} \mathcal{J},$$

où le pas $\eta_{j,i,t}$ peut prendre des valeurs différentes pour chaque couple (j, i) . En remplaçant $\nabla_{s_{j,i,t}} \mathcal{J}$ par son expression, on obtient la règle de mise à jour suivante :

$$s_{j,i,t+1} = s_{j,i,t} - \eta_{j,i,t} \left(\sum_{j'=1}^m s_{j',i,t} \kappa(\mathbf{a}_{j,t}, \mathbf{a}_{j',t}) - \kappa(\mathbf{a}_{j,t}, \mathbf{x}_i) \right). \quad (6)$$

Une procédure similaire est appliquée pour estimer les vecteurs \mathbf{a}_j . La règle de mise à jour obtenue à l'itération t est alors

$$\mathbf{a}_{j,t+1} = \mathbf{a}_{j,t} - \eta_{j,t} \nabla_{\mathbf{a}_j} \mathcal{J}, \quad (7)$$

où $\eta_{j,t}$ désigne le pas selon la j -ème direction et $\nabla_{\mathbf{a}_j} \mathcal{J}$ est le gradient donné par l'expression (5).

Afin d'imposer la non-négativité des matrices, les valeurs négatives obtenues par les règles susmentionnées sont mises à zéro. Pour cette rectification, il suffit de remplacer chaque élément $s_{j,i,t+1}$ par $\max\{s_{j,i,t+1}, 0\}$. De même pour les éléments des vecteurs $\mathbf{a}_{j,t+1}$.

3.0.2 Algorithme avec une mise à jour multiplicative

La règle de mise à jour additive est une procédure simple. Toutefois, la convergence est généralement lente, et dépend directement de la valeur du pas utilisée. Pour surmonter ces problèmes, nous proposons une règle de mise à jour multiplicative, dans le même esprit que celle proposée dans [18] pour la méthode NMF conventionnelle.

Afin de proposer une règle de mise à jour multiplicative de $s_{j,i,t}$ à l'itération t , le pas $\eta_{j,i,t}$ est choisi de telle sorte que le premier et le troisième termes du membre de droite de l'équation (6) s'annulent, c'est à dire $\eta_{j,i,t} =$

$s_{j,i,t} / \sum_{j'=1}^m s_{j',i,t} \kappa(\mathbf{a}_{j,t}, \mathbf{a}_{j',t})$. Par conséquent, en substituant cette expression du pas dans (6), on obtient la règle de mise à jour suivante :

$$s_{j,i,t+1} = s_{j,i,t} \times \frac{\kappa(\mathbf{a}_{j,t}, \mathbf{x}_i)}{\sum_{j'=1}^m s_{j',i,t} \kappa(\mathbf{a}_{j,t}, \mathbf{a}_{j',t})}. \quad (8)$$

Une procédure similaire est appliquée pour estimer les vecteurs $\mathbf{a}_{j,t}$, pour $j = 1, 2, \dots, m$. L'astuce est que l'expression du gradient (5) peut toujours être décomposée selon $\nabla_{\mathbf{a}_j} \mathcal{J} = P - Q$, où P et Q sont deux termes non négatifs. Cette décomposition est connue dans la littérature par la méthode *split gradient* [17]. Il est évident que cette décomposition n'est pas unique. Pourtant, nous pouvons décrire une mise à jour multiplicative pour une fonction noyau donnée, comme indiqué dans la suite avec le noyau gaussien.

Notons qu'une propriété fondamentale de la méthode NMF-à-noyaux proposée est que la NMF classique en est un cas particulier, avec le noyau linéaire $\kappa(\mathbf{a}_j, \cdot) = \cdot^\top \mathbf{a}_j$ et dont son gradient $\nabla_{\mathbf{a}_j} \kappa(\mathbf{a}_j, \cdot) = \cdot$.

Exemple : le noyau gaussien

Le noyau gaussien $\kappa(\mathbf{a}_j, \cdot) = \exp(-\frac{1}{2\sigma^2} \|\mathbf{a}_j - \cdot\|^2)$ admet un gradient selon $\nabla_{\mathbf{a}_j} \kappa(\mathbf{a}_j, \cdot) = -\frac{1}{\sigma^2} \kappa(\mathbf{a}_j, \cdot) (\mathbf{a}_j - \cdot)$. Il est facile de déterminer la règle de mise à jour de $s_{j,i}$, aussi bien selon la règle additive que multiplicative. Pour la mise à jour de \mathbf{a}_j , on obtient la règle additive suivante :

$$\mathbf{a}_{j,t+1} = \mathbf{a}_{j,t} - \eta_{j,i,t} \left(+ \frac{1}{\sigma^2} \sum_{i=1}^n s_{j,i,t} \kappa(\mathbf{a}_{j,t}, \mathbf{x}_i) (\mathbf{a}_{j,t} - \mathbf{x}_i) - \frac{1}{\sigma^2} \sum_{i=1}^n \sum_{j'=1}^m s_{j,i,t} s_{j',i,t} \kappa(\mathbf{a}_{j,t}, \mathbf{a}_{j',t}) (\mathbf{a}_{j,t} - \mathbf{a}_{j',t}) \right).$$

Pour l'algorithme multiplicatif, nous écrivons le gradient selon une soustraction de deux termes non négatifs. Ceci est possible puisque toutes les matrices sont non négatives, ainsi que les valeurs du noyau. Nous obtenons la règle de mise à jour suivante :

$$\mathbf{a}_{j,t+1} = \mathbf{a}_{j,t} \otimes \frac{\sum_{i=1}^n s_{j,i,t} \left(\mathbf{x}_i \kappa(\mathbf{a}_{j,t}, \mathbf{x}_i) + \sum_{j'=1}^m s_{j',i,t} \mathbf{a}_{j',t} \kappa(\mathbf{a}_{j,t}, \mathbf{a}_{j',t}) \right)}{\sum_{i=1}^n s_{j,i,t} \left(\mathbf{a}_{j,t} \kappa(\mathbf{a}_{j,t}, \mathbf{x}_i) + \sum_{j'=1}^m s_{j',i,t} \mathbf{a}_{j',t} \kappa(\mathbf{a}_{j,t}, \mathbf{a}_{j',t}) \right)},$$

où \otimes désigne le produit matriciel de Hadamard, et la division utilisée ici est aussi une opération matricielle terme à terme.

4 Expérimentations

Nous avons considéré des expérimentations sur le problème de démixage en imagerie hyperspectrale. Le tableau 1 présente une étude comparative sur deux images connues dans la littérature (Cuprite et Moffett), en utilisant les erreurs de reconstruction dans l'espace des observations et dans l'espace transformé. Voir [29] pour plus de détails.

TABLE 1 – Erreurs de reconstruction ($\times 10^{-2}$), dans l'espace des observations et dans l'espace transformé, pour les images Cuprite et Moffett.

		Cuprite		Moffett	
		ErrRec	ErrRec $^\phi$	ErrRec	ErrRec $^\phi$
[10]	VCA+FCLS	3.20	-	15.61	-
[5]	K-Hype	2.12	-	5.27	-
[27]	GBM-sNMF	0.98	-	2.09	-
[13]	MinDisCo	1.65	-	2.92	-
[8]	ConvexNMF	1.61	-	2.58	-
[20]	KconvexNMF	-	20.80	-	35.95
[28]	KsNMF	-	1.38	-	2.30
[23]	MercerNMF	-	2.74	-	2.77
le présent travail	Linéaire \oplus	0.96	0.96	2.90	2.90
	Linéaire \otimes	0.93	0.93	0.73	0.73
	Gaussien \oplus	2.16	0.94	2.12	0.98
	Gaussien \otimes	1.05	0.50	1.24	0.45

5 Conclusion

Dans le présent article, nous avons montré que la malédiction de pré-image peut être évitée, en proposant un nouveau paradigme de modèles non linéaires. La prochaine étape de nos travaux de recherche sera orientée vers une extension de cette approche à d'autres méthodes à noyaux, notamment l'ACP.

Références

[1] M. Aizerman, E. Braverman, and L. Rozonoer. Theoretical foundations of the potential function method in pattern recognition learning. *Automation and Remote Control*, 25 :821–837, 1964.

[2] G. Buchsbaum and O. Bloch. Color categories revealed by non-negative matrix factorization of munsell color spectra. *Vision Research*, 42(5) :559–63, 2002.

[3] I. Buciu, N. Nikolaidis, and I. Pitas. Nonnegative matrix factorization in polynomial feature space. *IEEE Transactions on Neural Networks*, 19(6) :1090–1100, 2008.

[4] 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*, volume 1, pages 288–291, Cambridge, UK, 2004.

[5] 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*, 61(2) :480–492, Jan. 2013.

[6] P. Comon and C. Jutten, editors. *Handbook of Blind Source Separation : Independent Component Analysis and Applications*. Academic Press, March 2010.

[7] K. Devarajan. Nonnegative Matrix Factorization : An Analytical and Interpretive Tool in Computational Biology. *PLoS Comput Biol*, 4(7), July 2008.

[8] C. Ding, T. Li, and M. I. Jordan. Convex and Semi-Nonnegative Matrix Factorizations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(1) :45–55, Nov. 2010.

[9] M. Essoloh, C. Richard, H. Snoussi, and P. Honeine. Distributed localization in wireless sensor networks as a pre-image problem in a reproducing

kernel hilbert space. In *Proc. 16th European Conference on Signal Processing*, pages 1–5, Lausanne, Switzerland, August 2008.

[10] D. Heinz and C.-I. Chang. Fully constrained least squares linear spectral mixture analysis method for material quantification in hyperspectral imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 39(3) :529–545, Mar. 2001.

[11] P. Honeine and C. Richard. A closed-form solution for the pre-image problem in kernel-based machines. *Journal of Signal Processing Systems*, 65(3) :289–299, December 2011.

[12] P. Honeine and C. Richard. Preimage problem in kernel-based machine learning. *IEEE Signal Processing Magazine*, 28(2) :77–88, March 2011.

[13] A. Huck, M. Guillaume, and J. Blanc-Talon. Minimum dispersion constrained nonnegative matrix factorization to unmix hyperspectral data. *IEEE Transactions on Geoscience and Remote Sensing*, 48(6) :2590–2602, Jun. 2010.

[14] M. Kallas, P. Honeine, C. Francis, and H. Amoud. Kernel autoregressive models using yule-walker equations. *Signal Processing*, 93(11) :3053–3061, November 2013.

[15] 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*, 46(11) :3066–3080, November 2013.

[16] P. M. Kim and B. Tidor. Subsystem identification through dimensionality reduction of large-scale gene expression data. *Genome Res.*, 13(7) :1706–1718, 2003.

[17] H. Lantéri, C. Theys, C. Richard, and D. Mary. Regularized split gradient method for nonnegative matrix factorization. In *ICASSP*, pages 1133–1136, 2011.

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

[19] T. Li and C. Ding. The relationships among various nonnegative matrix factorization methods for clustering. In *Proceedings of the Sixth International Conference on Data Mining, ICDM '06*, pages 362–371, Washington, DC, USA, 2006. IEEE Computer Society.

[20] Y. Li and A. Ngom. A new kernel non-negative matrix factorization and its application in microarray data analysis. In *IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology, CIBCB*, pages 371–378, San Diego, CA, USA, May 9-12 2012.

[21] W. Liu and N. Zheng. Non-negative matrix factorization based methods for object recognition. *Pattern Recognition Letters*, 25(8) :893–897, 2004.

[22] N. H. Nguyen, J. Chen, C. Richard, P. Honeine, and C. Theys. Supervised nonlinear unmixing of hyperspectral images using a pre-image method. In *New Concepts in Imaging : Optical and Statistical Models*, In Eds. D. Mary, C. Theys, and C. Aime, volume 59 of *EAS Publications Series*, pages 417–437. EDP Sciences, 2013.

[23] B. Pan, J. Lai, and W.-S. Chen. Nonlinear nonnegative matrix factorization based on Mercer kernel construction. *Pattern Recognition*, 44(10-11) :2800 – 2810, 2011.

[24] B. Schölkopf, R. Herbrich, and R. Williamson. A generalized representer theorem. Technical Report NC2-TR-2000-81, Royal Holloway College, Univ. of London, UK, 2000.

[25] G. Wahba. *Spline Models for Observational Data*. Society for Industrial and Applied Math (SIAM), Philadelphia, 1990.

[26] S. Wild, J. Curry, and A. Dougherty. Improving non-negative matrix factorizations through structured initialization. *Pattern Recognition*, 37(11) :2217–2232, Nov. 2004.

[27] N. Yokoya, J. Chanussot, and A. Iwasaki. Nonlinear unmixing of hyperspectral data using semi-nonnegative matrix factorization. *IEEE Transactions on Geoscience and Remote Sensing*, 52(2) :1430–1437, Feb. 2014.

[28] D. Zhang, Z. Zhou, and S. Chen. Non-negative matrix factorization on kernels. In *Lecture Notes in Computer Science*, volume 4099, pages 404–412. Springer, 2006.

[29] F. Zhu, P. Honeine, and M. Kallas. Kernel nonnegative matrix factorization without the curse of the pre-image. *ArXiv (arXiv :1407.4420)*, Submitted on 16 Jul 2014.