A New Approach to Dictionary-Based Nonnegative Matrix Factorization

Jérémy E. Cohen Nicolas Gillis
Department of Mathematics and Operational Research
Faculté Polytechnique, Université de Mons
Rue de Houdain 9, 7000 Mons, Belgium
nicolas.gillis@umons.ac.be

***Abstract*—In this paper, we propose a new model along with an algorithm for dictionary-based nonnegative matrix factorization. We show its effectiveness on spectral unmixing of hyperspectral images using self dictionary compared to state-of-the-art methods.**

Keywords. dictionary, nonnegative matrix factorization, hyperspectral imaging

I. Introduction

Low-rank matrix approximation problems have become more and more popular and have applications in a wide rang of applications; see, e.g., [1] and the references therein. In general, the problem can be formulated as follows: given a data matrix $X \in \mathbb{R}^{p\times n}$ where each column $X(:,j)$ is a data point in a p-dimensional space and a factorization rank r, the goal is to compute a basis matrix $U \in \mathbb{R}^{p\times r}$ and a weight matrix $V \in \mathbb{R}^{r\times n}$ such that

$$X \approx UV \iff X(:,j) \approx \sum_{k=1}^{k} U(:,k)V(k,j)\ \forall j.$$

There exists many variants of this problem, depending on the way to measure the error and the constraints on the factors (U,V) and/or on the approximation UV.

In this paper, we study nonnegative matrix factorization (NMF) that requires that the factors U and V to be component-wise nonnegative [2]. Moreover, we focus on a particular variant where the columns of the basis matrix U belong to a given dictionary $D \in \mathbb{R}^{p\times d}$, where $d \gg r$ is the number of atoms. Mathematically, this means that $U = D(:,\mathcal{K})$ for some index set $\mathcal{K} \subset \{1,2,\dots,d\}$ with $|\mathcal{K}| = r$. Therefore, using the Frobenius norm to quantify the error, dictionary-based NMF can be formulated as follows

$$\begin{array}{ll} \min_{\mathcal{K},V\geq 0} & ||X - D(:,\mathcal{K})V||_F^2 \\ \text{such that} & \mathcal{K} \subset \{1,2,\dots,d\} \text{ and } |\mathcal{K}| = r. \end{array} \quad (1)$$

An application where the model (1) is particularly useful for hyperspectral unmixing (HU). A hyperspectral image is an image for which there are usually between 100 and 200 channels for each pixel, corresponding to the reflectance (fraction of light reflected by that pixel) at different wavelengths. In other words, for each pixel, there is a vector containing its reflectances at different wavelengths which is its so-called spectral signature. The linear mixing model assumes that the spectral signature of each pixel is a linear combination of the spectral signatures of the constitutive materials, called endmembers, and is a valid model for macroscopic mixture of materials in the scene.

If the resolution of a hyperspectral image is high enough so that for each endmember there exists at least one pixel containing only that endmember, then the so-called pure-pixels assumption is satisfied and HU reduces to solving (1) using self dictionary $D = X$. A dictionary is also available when the endmembers are contained in a hyperspectral library. For more details on HU, we refer the readers to the surveys [3], [4] and the references therein.

A. NMF with self dictionary

In this paper, we will focus on the case with self dictionary $D = X$. As far as we know, there are mainly two types of approaches to tackle (1) in that case:

- *Geometric* approaches that selects the atoms in the dictionary based on some geometric criteria, typically based on the volume of the convex hull of $X(:,\mathcal{K})$. These approaches include for example vertex component analysis (VCA) [5] and the successive projection algorithm (SPA) [6], [7], [8], [9]. They are usually fast, running in $\mathcal{O}(pnr)$ operations. However, they do not always select atoms leading to a small data fitting term $||X - X(:,\mathcal{K})V||_F$ since, most of them do not take it into account directly, as they usually put an emphasis on some geometric properties of $X(:,\mathcal{K})$ (such as having a large volume). In particular, these methods are in general sensitive to outliers.

ISBN 978-0-9928626-7-1

- *Sparse regression* approaches that are based on the following reformulation of (1), which imposes row sparsity constraints on the scores Y:

$$\begin{array}{ll}\min_{Y\in\mathbb{R}^{d\times n}} & ||X-XY||_F^2 \\ \text{such that} & Y \text{ has } r \text{ non-zero rows.}\end{array}$$

  Row-sparsity of Y can be achieved in different ways; in particular using convexifications based on the ℓ_1 norm, e.g., $\ell_{1,2}$ [10], $\ell_{1,\infty}$ [11], or using linear programming [12], [13].
  These methods have the advantage to better model (1) as they take into account the data fitting term explicitly. They usually provide good solutions but are rather costly as an optimization problem in dn variables must be solved. In particular, for $D = X$, we have $d = n$ hence n^2 variables. In HU, n is the order of millions and these approaches are impractical. Hence pixels have to be selected in a preprocessing step [14] (e.g., using a geometric approach). Moreover, the problem solved is an approximation of the original problem, which results may not be as close as desired to the solutions of the non-convex problem.

B. Contribution and outline of the paper

In this paper, we propose a new algorithm to solve the formulation (1). It will combine the advantages of the two types of approaches described above: it is fast, running in $\mathcal{O}(pnr)$ operations, but taking explicitly the data fitting term $||X - X(:,\mathcal{K})V||_F^2$ into account. In Section II, we describe the new algorithm, and in Section III, we show that is competes favorably with state-of-the-art approaches for HU.

II. Proposed algorithm for (1)

Solving the combinatorial model (1) directly is difficult. We propose in this paper to use an alternating strategy:

- Update of V. For fixed $\mathcal{K}$, solving (1) in variable V is a convex optimization problem that can be solved efficiently; it is a nonnegative least squares (NNLS) problem.
- Update of $\mathcal{K}$. The update of $\mathcal{K}$ is difficult, being a combinatorial problem. Moreover, for V fixed, the optimal subset $\mathcal{K}$ is most likely to be unique hence this approach would not be able to modify an initial solution $(V, \mathcal{K})$. To circumvent these difficulties, we introduce an auxiliary variable for the hidden factor $U = D(:,\mathcal{K})$, and consider the problem

$$\begin{array}{ll}\min_{\mathcal{K},U\geq 0,V\geq 0} & ||X-UV||_F^2 + \delta||U - D(:,\mathcal{K})||_F^2 \\ \text{such that} & \mathcal{K}\subset\{1,2,\ldots,d\} \text{ and } |\mathcal{K}| = r,\end{array} \quad (2)$$

  for some penalty parameter $\delta > 0$. Solving for U is again an NNLS problem while for $\mathcal{K}$, it is now a trivial problem to pick the atoms of the dictionary the closest to the columns of matrix U. The parameter δ is progressively increased to ensure convergence of the model (2) to the model (1).

On top of allowing to solve each subproblem effectively, the auxiliary variable U can be used as a correction of the selected atoms since U minimizes the data fitting term while being close to $D(:,\mathcal{K})$ for some $\mathcal{K}$. This way the self-dictionary model allows for some flexibility on the spectra actually contained in the data, which can vary from pixel to pixel due to spectral variability [15].

Note that to update U and V, we do no solve the NNLS subproblems to a high precision, which is not necessary and would be computationally rather costly, but use a few steps of block coordinate descent [16] (we performed 10 iterations).

Algorithm 1 provides the pseudocode for our proposed algorithm.

Algorithm 1 Alternating Optimization for (2)

Input: $X \in \mathbb{R}^{p\times n}$, integer r, `maxiter`.
Output: $U \in \mathbb{R}_+^{p\times r}, V \in \mathbb{R}_+^{r\times n}$ and an index set $\mathcal{K}$ such that $||X - D(:,\mathcal{K})V||_F$ is small and $U \approx D(:,\mathcal{K})$.

1: Choose some initial matrices $U \geq 0, V \geq 0$ and index set $\mathcal{K}$, and initial value of δ.
2: **for** $k = 1 : \texttt{maxiter}$ **do**
3: Solve for V: $\min_{V\geq 0} ||X - D(:,\mathcal{K})V||_F^2$.
4: Solve for U:

$$\min_{U\geq 0} ||X - UV||_F^2 + \delta||U - D(:,\mathcal{K})||_F^2.$$

5: $\mathcal{K} = \emptyset$.
6: **for** $k = 1 : r$ **do**
7: $\mathcal{K} = \mathcal{K} \cup \text{argmax}_k \frac{D(:,k)^T U(:,k)}{||D(:,k)||_2}$.
8: **end for**
9: **if** $||U - D(:,\mathcal{K})||_F > 0.01||U||_F$ **then**
10: Increase δ.
11: **end if**
12: **if** $||U - D(:,\mathcal{K})||_F < 0.05||U||_F$ and K has not changed for 5 iterations **then**
13: return; the algorithm has converged.
14: **end if**
15: **end for**

a) Initialization: Our algorithm tries to solve a highly non-linear and combinatorial problem. This explains why, as we will see, it is sensitive to initialization. In this paper we use the following initialization scheme

1) Compute $\mathcal{K}$ using your favorite algorithm. In the numerical experiments section, we will use random initialization and several state-of-the-art pure-pixel search algorithms.

ISBN 978-0-9928626-7-1 © EURASIP 2017

2) Define $U = D(:, \mathcal{K})$ and, for V, compute the optimal solution of the unconstrained problem and project it onto the nonnegative orthant, that is, use $V = \max\left(0, \text{argmin}_Y ||X - UY||_F^2\right)$ (in Matlab, `V=max(0,U\M)`).
3) Improve (U, V) using an NMF algorithm (this assumes $\delta = 0$). We used 10 iterations of A-HALS [16].
4) Initialize

$$\delta = 0.01 \frac{||X - UV||_F^2}{||U - D(:, \mathcal{K})||_F^2},$$

so that the data fitting term has initially more importance in the objective function. In fact, when δ is large, the algorithm is less likely to be able to update $\mathcal{K}$ since U will be very close to $D(:, \mathcal{K})$.

b) Other related algorithms: There are a number of arbirary choices that were made to design Algorithm 1, we here shortly discuss some variants. First, in the estimation of V, $D(:, \mathcal{K})$ can be swapped with U to minimize the same cost function for the two factors U and V. We noticed however that by doing so, the performance of the algorithm decreased. In particular, when $D(:, \mathcal{K})$ is used in the estimation of V, after some iterations $\mathcal{K}$ should not change much whereas U can still be modified, so that the number of iterations is larger in this modified version. Second, the flexible approach is not mandatory. Indeed, U can be updated using a two-step procedure involving nonnegative least squares followed by a projection on the set of atoms. An advantage of this method is that it does not require to tune or fix any parameter. However we found this simpler version of the algorithm to impose too hard constraints in the first iterations where exploring the set of unconstrained U seems important. This is the reason why δ is introduced, and starts at a relatively small value.

III. Numerical Experiments on hyperspectral images

In this section, we compare Algorithm 1 with three geometric algorithms, namely VCA [5], SPA [6] and SNPA [17], one clustering-based algorithm, H2NMF [18], and a sparse regression framework from [14], referred to as FGNSR (where a subset of 100 and 500 columns is identified using H2NMF).

We will initialize Algorithm 1 with the atoms extracted by the above methods and refer to the corresponding algorithm as d-X, where X is the algorithm; for example, d-VCA stands for Algorithm 1 initialized with VCA. We will also use 10 random initializations (picking r pixels at random as initial endmembers) and report the worst, average and best solution (in terms of reconstruction error), denoted RAND-wo, RAND-av and RAND-be, respectively. For all these numerical experiments, we increase δ using 1.5δ. In order to keep δ bounded, we only increase it if $||U - D(:, \mathcal{K})||_F$ is large; see step 10 of Algorithm 1.

When we report the CPU time of Algorithm 1, we do not report the time for the initialization. The CPU time of FGNSR does not take into account the preprocessing time by H2NMF.

We compare the different approaches on the same data sets as [14] with the same factorization ranks:

- Urban data set with 162 wavelength and 309 $\times$ 309 pixels, $r = 6, 8$.
- San Diego airport with 158 wavelength and 400 $\times$ 400 pixels, $r = 8, 10$.
- The Terrain data set with 166 wavelength and 400 $\times$ 400 pixels, $r = 5, 6$.

Tables I, II and III report the results. For an index set $\mathcal{K}$ extracted by an algorithm, the relative reconstruction error (rel. err.) is given by

$$\min_{V \geq 0} \frac{||X - X(:, \mathcal{K})V||_F}{||X||_F}.$$

In these tables, the relative error is given in **percent**. The lowest reconstruction error is highlighted in bold. In brackets, next to the CPU time, we indicate the number of iterations needed for Algorithm 1 to converge. All tests are preformed using Matlab on a laptop Intel CORE i5-3210M CPU @2.5GHz 6GB RAM.

	$r = 6$		$r = 8$	
	Time (s.)	Rel. err.	Time (s.)	Rel. err.
RAND-wo	0.00	7.87	0.00	11.66
d-RAND-wo	22.46 (13)	5.09	34.87 (18)	5.35
RAND-av	0.02	11.51	0.02	9.60
d-RAND-av	23.91 (13)	4.65	30.77 (15)	4.65
RAND-be	0.00	13.77	0.00	5.54
d-RAND-be	22.01 (11)	4.36	36.18 (19)	4.16
VCA	2.01	18.38	1.86	20.11
d-VCA	26.89 (15)	5.83	29.06 (14)	5.05
SPA	0.30	9.58	0.30	9.45
d-SPA	24.37 (13)	4.67	28.61 (14)	4.62
SNPA	24.34	9.63	36.72	5.64
d-SNPA	23.04 (13)	4.94	27.94 (13)	**3.97**
H2NMF	19.02	5.81	22.35	5.47
d-H2NMF	26.66 (15)	**4.05**	28.92 (14)	4.24
FGNSR-100	2.73	5.58	2.55	4.62
d-FGNSR-100	26.72 (14)	4.36	20.81 (8)	4.04
FGNSR-500	40.11	5.07	39.49	4.08
d-FGNSR-500	25.07 (13)	4.40	26.83 (12)	4.13

TABLE I
NUMERICAL RESULTS FOR THE URBAN DATA SET.

Figure 1 shows the best solution found for the Urban data set with $r = 6$ (with d-H2NMF), and Figure 2 the corresponding abundance maps. We can identify constitutive materials such as trees, roof tops, dirt, grass, and roads.

ISBN 978-0-9928626-7-1 © EURASIP 2017

	$r = 8$		$r = 10$	
	Time (s.)	Rel. err.	Time (s.)	Rel. err.
RAND-wo	0.00	10.63	0.02	11.44
d-RAND-wo	38.55 (9)	4.86	62.23 (13)	4.69
RAND-av	0.02	9.41	0.01	10.39
d-RAND-av	49.50 (14)	4.21	60.80 (13)	3.83
RAND-be	0.02	8.49	0.02	10.71
d-RAND-be	59.17 (18)	**3.57**	57.67 (12)	3.42
VCA	3.51	7.47	3.48	8.83
d-VCA	68.45 (22)	5.15	89.65 (22)	5.79
SPA	0.45	12.62	0.53	7.01
d-SPA	68.45 (22)	4.08	75.88 (17)	3.91
SNPA	64.96	12.84	87.24	7.67
d-SNPA	64.04 (18)	3.75	68.14 (16)	4.45
H2NMF	36.77	4.75	39.48	4.28
d-H2NMF	44.18 (10)	4.13	74.86 (18)	3.36
FGNSR-100	2.55	3.73	2.47	3.40
d-FGNSR-100	43.85 (11)	3.63	78.79 (20)	3.28
FGNSR-500	38.70	4.05	38.28	3.40
d-FGNSR-500	43.88 (11)	3.67	61.62 (14)	**2.95**

TABLE II
NUMERICAL RESULTS FOR THE SAN DIEGO AIRPORT.

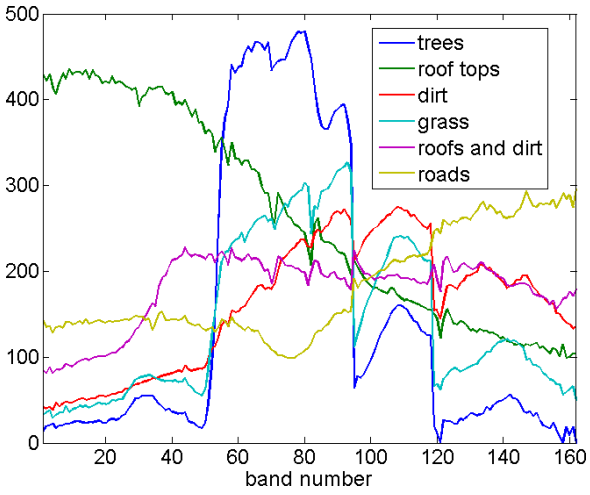

Fig. 1. Spectral signatures of the endmembers exctracted using d-H2NMF for the Urban data set with $r = 6$.

	$r = 5$		$r = 6$	
	Time (s.)	Rel. err.	Time (s.)	Rel. err.
RAND-wo	0.03	16.29	0.02	8.64
d-RAND-wo	26.69 (8)	4.05	29.70 (8)	3.34
RAND-av	0.02	9.75	0.01	7.83
d-RAND-av	38.13 (15)	3.49	36.83 (12)	3.25
RAND-be	0.03	6.59	0.03	9.35
d-RAND-be	32.87 (12)	3.21	38.59 (13)	3.12
VCA	2.95	16.99	2.98	7.54
d-VCA	32.92 (14)	4.25	44.55 (17)	3.25
SPA	0.34	5.89	0.36	4.81
d-SPA	46.38 (20)	3.37	42.85 (15)	3.81
SNPA	32.01	5.76	40.26	4.60
d-SNPA	41.47 (18)	3.88	44.09 (16)	3.70
H2NMF	28.86	5.09	33.68	4.85
d-H2NMF	38.74 (16)	3.52	39.06 (13)	3.30
FGNSR-100	4.23	3.34	2.63	3.21
d-FGNSR-100	28.14 (9)	**3.08**	46.58 (17)	**2.84**
FGNSR-500	40.29	3.68	40.13	3.39
d-FGNSR-500	28.00 (9)	3.12	36.88 (12)	3.22

TABLE III
NUMERICAL RESULTS FOR THE TERRAIN DATA SET.

We observe that

- In all cases but one (FGNSR on the Urban image with $r = 8$, with an increase of 0.05%), Algorithm 1 is able to improve the initial solutions provided by RAND, VCA, SPA, H2NMF and FGNSR.
- In all cases, Algorithm 1 converges in less than 20 iterations. The reason is that we increased δ rather aggressively.
- Even with random initial index sets $\mathcal{K}$, Algorithm 1 provides solutions with small reconstruction errors. In fact, it is rather surprising that even the worst solution generated among 10 random initial sets $\mathcal{K}$ is better than the solutions generated by VCA, SPA and

Fig. 2. Abundance maps identified using d-H2NMF for the Urban data set with $r = 6$.

ISBN 978-0-9928626-7-1 © EURASIP 2017

H2NMF. Moreover, the best solution found for the San Diego airport image with $r = 8$ is from random initialization.

This means that although Algorithm 1 is sensitive to initialization, which was expected, it allows to identify reasonable solutions regardless of the initialization.

- In all cases, Algorithm 1 leads to the best solution compared to the original algorithms.

A reason why Algorithm 1 works remarkably well for these data sets is because the spectral signatures of the pixels are close to one another and form a dense cloud of data points. This allows the index set $\mathcal{K}$ to change progressively between neighboring pixels. Analyzing the behavior of Algorithm 1 in other settings is a direction for further research.

IV. Conclusion

In this paper, we proposed a new algorithm for NMF with self dictionary; see Algorithm 1. Like geometric methods, it is fast, running in $\mathcal{O}(mnr)$ operations, hence can be applied to large problems. Like sparse-regression methods, it takes into account the data fitting term explicitly, hence identifying good atoms in the dictionary with a small approximation error. We illustrate the effectiveness of Algorithm 1 on several hyperspectral images. In all cases, it identifies the set of endmembers providing the lowest reconstruction error. We focused in this paper on this particular variant of dictionary-based low-rank matrix approximations. However, our technique can be applied to the broader class of problems, which will be presented in an extended version of this paper.

Acknowledgments

The authors want to thank Pierre Comon, Rodrigo Cabral-Farias and Miguel A. Veganzones for helpful discussions in the early stages of this work.

The authors acknowledge the support by the F.R.S.-FNRS (incentive grant for scientific research no F.4501.16). NG also acknowledges the support by the ERC (starting grant no 679515).

References

[1] M. Udell, C. Horn, R. Zadeh, and S. Boyd, "Generalized low rank models," *Foundations and Trends in Machine Learning*, vol. 9, no. 1, pp. 1–118, 2016.

[2] 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, 1999.

[3] J. Bioucas-Dias, A. Plaza, N. Dobigeon, M. Parente, Q. Du, P. Gader, and J. Chanussot, "Hyperspectral unmixing overview: Geometrical, statistical, and sparse regression-based approaches," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 5, no. 2, pp. 354–379, 2012.

[4] W. K. Ma, J. M. Bioucas-Dias, T. H. Chan, N. Gillis, P. Gader, A. J. Plaza, A. Ambikapathi, and C. Y. Chi, "A signal processing perspective on hyperspectral unmixing: Insights from remote sensing," *IEEE Signal Processing Magazine*, vol. 31, no. 1, pp. 67–81, 2014.

[5] J. Nascimento and J. Dias, "Vertex component analysis: a fast algorithm to unmix hyperspectral data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 43, no. 4, pp. 898–910, 2005.

[6] M. C. U. Araújo, T. C. B. Saldanha, R. K. H. Galvão, T. Yoneyama, H. C. Chame, and V. Visani, "The successive projections algorithm for variable selection in spectroscopic multicomponent analysis," *Chemometrics and Intelligent Laboratory Systems*, vol. 57, no. 2, pp. 65–73, 2001.

[7] H. Ren and C.-I. Chang, "Automatic spectral target recognition in hyperspectral imagery," *IEEE Trans. on Aerospace and Electronic Systems*, vol. 39, no. 4, pp. 1232–1249, 2003.

[8] T.-H. Chan, W.-K. Ma, A. Ambikapathi, and C.-Y. Chi, "A simplex volume maximization framework for hyperspectral endmember extraction," *IEEE Trans. on Geoscience and Remote Sensing*, vol. 49, no. 11, pp. 4177–4193, 2011.

[9] N. Gillis and S. A. Vavasis, "Fast and robust recursive algorithmsfor separable nonnegative matrix factorization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 4, pp. 698–714, 2014.

[10] E. Elhamifar, G. Sapiro, and R. Vidal, "See all by looking at a few: Sparse modeling for finding representative objects," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. Institute of Electrical & Electronics Engineers (IEEE), 2012. [Online]. Available: http://dx.doi.org/10.1109/cvpr.2012.6247852

[11] E. Esser, M. Moller, S. Osher, G. Sapiro, and J. Xin, "A convex model for nonnegative matrix factorization and dimensionality reduction on physical space," *IEEE Transactions on Image Processing*, vol. 21, no. 7, pp. 3239–3252, 2012.

[12] V. Bittorf, B. Recht, E. Ré, and J. Tropp, "Factoring nonnegative matrices with linear programs," in *Advances in Neural Information Processing Systems (NIPS '12)*, 2012, pp. 1223–1231.

[13] N. Gillis and R. Luce, "Robust near-separable nonnegative matrix factorization using linear optimization," *J. Mach. Learn. Res.*, vol. 15, pp. 1249–1280, 2014.

[14] ——, "A fast gradient method for nonnegative sparse regression with self dictionary," *arXiv:1610.01349*, 2016.

[15] A. Zare and K. Ho, "Endmember variability in hyperspectral analysis: Addressing spectral variability during spectral unmixing," *IEEE Signal Processing Magazine*, vol. 31, no. 1, pp. 95–104, 2014.

[16] N. Gillis and F. Glineur, "Accelerated multiplicative updates and hierarchical ALS algorithms for nonnegative matrix factorization," *Neural Comput.*, vol. 24, no. 4, pp. 1085–1105, 2012.

[17] N. Gillis, "Successive nonnegative projection algorithm for robust nonnegative blind source separation," *SIAM J. Imaging Sci.*, vol. 7, no. 2, pp. 1420–1450, 2014.

[18] N. Gillis, D. Kuang, and H. Park, "Hierarchical clustering of hyperspectral images using rank-two nonnegative matrix factorization," *IEEE Trans. Geosci. Remote Sensing*, vol. 53, no. 4, pp. 2066–2078, 2015.

ISBN 978-0-9928626-7-1 © EURASIP 2017