AN ITERATIVE HARD THRESHOLDING APPROACH TO ℓ_0 SPARSE HELLINGER NMF

Ken O'Hanlon and Mark B. Sandler

Centre for Digital Music
Queen Mary University of London

ABSTRACT

Performance of Non-negative Matrix Factorisation (NMF) can be diminished when the underlying factors consist of elements that overlap in the matrix to be factorised. The use of ℓ_0 sparsity may improve NMF, however such approaches are generally limited to Euclidean distance. We have previously proposed a stepwise ℓ_0 method for Hellinger distance, leading to improved sparse NMF. We extend sparse Hellinger NMF by proposing an alternative Iterative Hard Thresholding sparse approximation method. Experimental validation of the proposed approach is given, with a large improvement over NMF methods when learning is performed on a large dataset.

Index Terms— Non-negative matrix factoriation, sparsity, Hellinger distance

1. INTRODUCTION

Given a matrix with non-negative elements, $\mathbf{S} \in \mathbb{R}^{M \times N}$, Non-negative Matrix Factorisation (NMF) [1] seeks to find a dictionary matrix $\mathbf{A} \in \mathbb{R}^{M \times K}$ and an activation matrix $\mathbf{X} \in \mathbb{R}^{K \times N}$ such that

$$\mathbf{S} \approx \mathbf{A}\mathbf{X} \tag{1}$$

where $\mathbf{A}$ and $\mathbf{X}$ also consist completely of non-negative entries. NMF is a popular tool, particularly in audio processing, where it is used to factorise spectrograms. Many different algorithms have been proposed for NMF, including the popular multiplicative updates [1] [2] [3], coordinate descent [4], alternating non-negative least squares [5] and ADMM [6]. Typically, these algorithms alternately update one factor matrix while keeping the other fixed. A range of cost functions have been considered for NMF [1] [2], including generalised cost functions, such as the β-divergence [7] and α-divergence [3].

NMF is considered to derive a "parts-based" representation [1]. However the parts learnt may not be meaningful. A major obstacle in performing NMF is the issue of separability [8], which basically requires that the generating factors of the matrix $\mathbf{S}$ be independent in order to be recovered. Furthermore, in order to guarantee separation it is required that all combinations of the generating elements, or dictionary are present in the mixture matrix, $\mathbf{S}$ [8]. In a non-negative framework such requirements reduce to non-overlapping elements, i.e. $[A]_{m,k} > 0 \implies [A]_{m,j} = 0 \, \forall j \neq k$.

It is considered that sparsity may be beneficial for NMF [9]. In particular, we consider that sparsity may reduce the negative effects of overlap, and counter the tendency of NMF to learn a cone that spans datapoints [8], rather than meaningful atoms, as illustrated in Fig.1. Sparsity considers an ℓ_0 penalisation, where $\|\mathbf{x}\|_0 = |\mathbf{x} \neq 0|$. However, estimation of ℓ_0 sparse problems is difficult, and is generally approximated through convex ℓ_1 relaxation [10] or greedy methods [11]. Such approaches come with guaranteed performance, in incoherent settings, which can be mostly dismissed in the non-negative framework [12]. Sparse NMF approaches have tended to consider the ℓ_1 penalty [13] [14]. Non-negative sparse dictionary learning methods such as NN-K-SVD [15] and NMF-ℓ_0 [16] use non-negative sparse coders such as NNLS and NN-OMP [12] to perform ℓ_0 approximation, but only consider sparse penalised Euclidean distance, which may not be optimal in many cases [17]. However, few methods have considered ℓ_0 sparsity for NMF with cost functions other than Euclidean distance. We previously proposed a greedy Hellinger Sparse Coding (HSC) algorithm, similar to OMP [11], using a nearest neighbour selection approach [18], which we found to be effective for several tasks when incorporated into NMF [19]. However, this greedy approach selects a predefined number of atoms in each activation vector, and stepwise approaches may not be suitable with correlated dictionaries [12].

In this paper we propose an alternative ℓ_0 approach based on an iterative hard thresholding approach. Again, the Hellinger distance is used as we find that it is amenable to such an approach, unlike other popular NMF cost functions. In the next section, we introduce the Hellinger distance, and iterative thresholding algorithms, before describing the proposed approach, which employs a Newton co-ordinate descent algorithm, and thresholding based on an auxiliary function. Experimental results demonstrate the proposed approach improves over a range of NMF algorithms. In particular, when a larger dataset is used, the proposed approach is seen to perform as well as a supervised approach, while standard NMF fails to improve.

This research was funded by ESPRC Grant EP/J010375/1, ESPRC Platform Grant EP/K009559/1, and AHRC Grant AH/L006820/1.

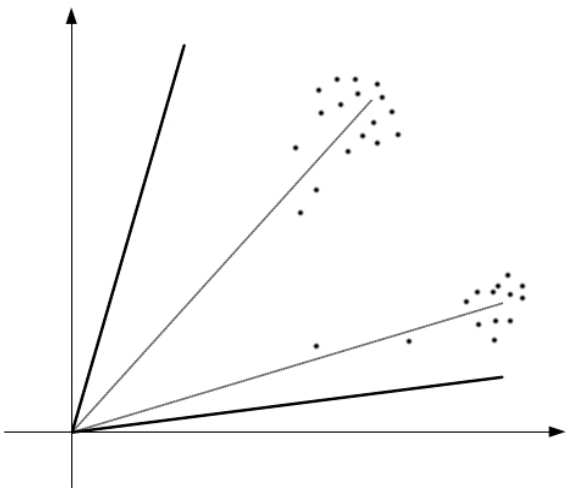

Fig. 1. Synthetic low dimensional dictionary recovery problem. Dots represent datapoints. The dark lines represent the atoms learnt by NMF, which can estimate all datapoints correctly. The brighter lines represent atoms learnt with 1-sparse NMF with each datapoint assigned to one atom only.

2. BACKGROUND

2.1. Iterative Thresholding Algorithms

Iterative thresholding algorithms are sparsity inducing methods that consider two separate steps; a gradient step in which a signal estimate is formed; and a thresholding step in which a hard or soft threshold is applied to the coefficient vector. Iterative Soft Thresholding (IST) methods [20], a particular case of proximal gradient descent, typically consider an ℓ_1 penalty generally applied to Euclidean distance and have been shown to be fast, accurate algorithms for ℓ_1 approximation.

Iterative Hard Thresholding (IHT) algorithms [21] [22] consider ℓ_0 approximation in the case where the number of active atoms is known. In this case thresholding is employed to select an active set, of length equal either to the required sparsity[21] or a multiple thereof [22]. A recent variant of IHT, called MIST-ℓ_0 [23] performs hard thresholding, derived on a majorisation-minimisation algorithm. MIST-ℓ_0 differs from other IHT methods as the active set is not required to be of a fixed size, and can vary relative to the signal and sparsity parameter.

2.2. Hellinger distance

The (squared) Hellinger distance,

$$\mathcal{C}_H(s|z) = (\sqrt{s} - \sqrt{z})^2 \tag{2}$$

is a member of the family of α-divergences

$$\mathcal{C}_\alpha(\mathbf{s}|\mathbf{z}) = \frac{1}{\alpha(\alpha-1)} \sum_m \alpha s_m(1-\alpha)z_m - s_m^\alpha z_m^{1-\alpha} \tag{3}$$

which includes the popular Kullback-Leibler divergence (KL). We observed that KL and and Hellinger perform similarly [18], and both cost share similar properties such as linear scaling. Unlike KL, Hellinger distance is symmetric, which led us to propose a greedy Hellinger nearest neighbour sparse coder in [18], and is bounded, in particular by the Total Variation or ℓ_1 distance [24].

3. PROPOSED APPROACH

An IHT method to approximate penalised Hellinger distance

$$\mathcal{C}_{H\ell_0}(\mathbf{s}|\mathbf{z}) = \sum_m (\sqrt{s_m} - \sqrt{z_m})^2 + \lambda\|\mathbf{x}\|_0. \tag{4}$$

where $\mathbf{z} = \mathbf{A}\mathbf{x}$, is now proposed, and incorporated into a NMF algorithm. The proposed method uses iterative hard thresholding in a similar manner to the MIST-ℓ_0 approach, whereby the number of atoms need not be predetermined, and thresholding is performed based on an auxiliary function.

Similar to other iterative shrinkage algorithms, separate gradient step and thresholding steps are employed. For the gradient step, it is considered that inactive atoms may re-enter the active set, precluding multiplicative update (MU) approaches. A co-ordinate descent algorithm for KL [4] that employs univariate Newton steps to update the factor matrices is adapted. For the Hellinger distance this step is given by

$$[X]_{k,n} \leftarrow [X]_{k,n} - 2\frac{\mathbf{a}_k^T\mathbf{1} - \mathbf{a}_k^T(\frac{\mathbf{s}_n^{[0.5]}}{\mathbf{z}_n^{[0.5]}})}{\mathbf{a}_k^T \mathrm{diag}(\frac{\mathbf{s}_n^{[0.5]}}{\mathbf{z}_n^{[1.5]}})\mathbf{a}_k} \tag{5}$$

where $\mathbf{Z} = \mathbf{A}\mathbf{X}$, $\mathbf{s}_n$ denotes the nth column of $\mathbf{S}$, $\mathbf{s}^{[a]}$ denotes elementwise exponentiation of $\mathbf{s}$ to the power of a; the numerator is the Hellinger gradient and the denominator is the Hessian, which can be quickly calculated as $\mathbf{a}_k^{[2]T}[\mathbf{s}_n^{[0.5]} \oslash \mathbf{z}_n^{[1.5]}]$, where $\oslash$ denotes elementwise division. Non-negativity is enforced by taking a reduced stepsize, $\eta = [X]_{k,n}$, when necessary.

3.1. Hard Thresholding

For the thresholding step, a majorisation-minimisation (MM) approach is considered. MM approaches consider use of an auxiliary function, $\mathcal{G}(\mathbf{x}, \hat{\mathbf{x}})$ where $\hat{x}$ is referred to as an auxiliary variable, that is typically expressed by the current estimate. The auxiliary function, by definition, upper bounds the cost function, $\mathcal{C}(\mathbf{x}) = \mathcal{C}(\mathbf{s}|\mathbf{D}\mathbf{x}) \leq \mathcal{G}(\mathbf{x}, \hat{\mathbf{x}})$ with equality when $\mathbf{x} = \hat{\mathbf{x}}$. Optimisation of the auxiliary function then results in optimisation of the cost function as

$$\mathcal{C}(\mathbf{x}) \leq \mathcal{G}(\mathbf{x}, \hat{\mathbf{x}}) \leq \mathcal{G}(\hat{\mathbf{x}}, \hat{\mathbf{x}}) = \mathcal{C}(\hat{\mathbf{x}}) \tag{6}$$

A variable-wise auxiliary function considers that [25]

$$\mathcal{G}(\mathbf{x}, \hat{\mathbf{x}}) = \sum_k \mathcal{G}(x_k, \hat{\mathbf{x}}) + cst. \tag{7}$$

an example of which is given for the penalised Hellinger (4)

$$\mathcal{G}(x_k, \hat{\mathbf{x}}) = x_k \mathbf{a}_k^T \mathbf{1} - 2\mathbf{a}_k^T \left[\frac{\sqrt{\mathbf{s}}}{\sqrt{\hat{\mathbf{z}}}}\right] \sqrt{x_k}\sqrt{\hat{x}_k} + \lambda \mathbb{I}(x_k \neq 0) \tag{8}$$

where $\mathbb{I}$ is a binary indicator function and $\hat{\mathbf{Z}} = \mathbf{A}\hat{\mathbf{X}}$. At the current estimate, $\hat{x} \neq 0$, (8) is given by

$$\mathcal{G}(\hat{x}_k, \hat{\mathbf{x}}) = \mathbf{a}_k^T \mathbf{1} \hat{x}_k - \hat{x}_k 2\mathbf{a}_k^T \left[\frac{\sqrt{\mathbf{s}}}{\sqrt{\hat{\mathbf{z}}}}\right] + \lambda \tag{9}$$

while after a hard threshold $(x_k = 0)$ (8) is expressed

$$\mathcal{G}(\mathbf{0}_k, \hat{\mathbf{x}}) = 0 \tag{10}$$

from which the difference in auxiliary function

$$\eta(\tau) = \mathcal{G}(\hat{x}_k, \hat{\mathbf{x}}) - \mathcal{G}(\mathbf{0}_k, \hat{\mathbf{x}}) = \hat{x}_k \mathbf{a}_k^T \left[1 - \frac{2\sqrt{\mathbf{s}}}{\sqrt{\hat{\mathbf{z}}}}\right] + \lambda \tag{11}$$

leading to the thresholding operator

$$\mathcal{H}(x_k) = \begin{cases} x_k & \text{if } \lambda < x_k \mathbf{a}_k^T \left[\frac{2\sqrt{\mathbf{s}}}{\sqrt{\hat{\mathbf{z}}}} - 1\right] \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

When the threshold operator (12) is activated, the ℓ_0 approximated Hellinger distance (4) is optimised. As an auxiliary function is used, some elements for which thresholding may lead to further optimisation of (4), may not be thresholded. However, even in this case the auxiliary and cost functions are not increased. As the thresholding operator is performed relative to the variablewise separable auxiliary function, thresholding can be performed in parallel for all variables.

For comparison, we note the difficulty in employing such an approach for KL-divergence in which case a variable-wise auxiliary function is given by

$$\mathcal{G}_{KL}(x_k, \hat{\mathbf{x}}) = x_k \mathbf{a}_k^T \mathbf{1} - \hat{x}_k \mathbf{a}_k^T \left[\frac{\mathbf{s}}{\hat{\mathbf{z}}}\right] \log x_k \tag{13}$$

leading to a difference in auxiliary function, similar to (11)

$$\eta^{KL}(\tau) = \infty. \tag{14}$$

A notable feature of the α-divergence is linear scaling whereby $\mathcal{C}_\alpha(as|az) = a\mathcal{C}_\alpha(s|z)$. In particular for KL-divergence this is explained in terms of dispersion parameters of the Poisson distribution which leads to a natural selection of $\lambda = 1$ [14]. We have empirically observed that similar parametrisation of λ holds for other α divergences as well. For the ℓ_0 penalised Hellinger case, the linear scaling is considered by selecting, for each column

$$\lambda_n = \delta \times \sum_m [S]_{m,n} \tag{15}$$

noting that the Hellinger distance is upper bounded by the $\ell_1$norm. δ can be tuned relative to the desired sparsity level, or empirically determined.

Algorithm 1 HIT-NMF

Input $\mathbf{S} \in \mathbb{R}^{M \times N}$
Initialise $\mathbf{A} \in \mathbb{R}^{M \times K}$; $\mathbf{X} \in \mathbb{R}^{K \times N}$
Do
  HIT-ℓ_0
    Update Coeffcients using (5) $\forall\{k, n\}$
    Threshold using (12)
  Update Dictionary using (16) $\forall\{m, k\}$
Until Stopping Condition

3.2. HIT-NMF

The proposed HIT-NMF algorithm is outlined in Algorithm 1. After inputting the matrix to be factorised, the factor matrices are randomly initialised. We also find it useful to perform some iterations of multiplicative update (MU) NMF as part of the initialisation, as this helps the algorithm to converge. The algorithm then enters an iterative loop. First the coefficients are updated using the univariate Newton step (5), with subsequent thresholding. These updates may be performed several times; in practice we perform two iterations. The dictionary is then updated, also using a univariate Newton step, similar to (5)

$$[A]_{m,k} \leftarrow [A]_{m,k} - 2\frac{\mathbf{1}\mathbf{x}^{k^T} - \left[\frac{\mathbf{s}^{m\,[0.5]}}{\mathbf{z}^{m\,[0.5]}}\right]\mathbf{x}^{k^T}}{\mathbf{x}^k \left[\frac{\mathbf{s}^{m\,[0.5]}}{\mathbf{z}^{m\,[1.5]}}\right]\mathbf{x}^{k^T}} \tag{16}$$

where $\mathbf{x}^k$ refers to the kth row of $\mathbf{X}$. Other update strategies, such as several MU iterations, are possible. Normalisation of the dictionary and corresponding scaling of the activations may be performed, although this is not absolutely necessary as the thresholding operator (12) is invariant to scale.

4. EXPERIMENTS

Experiments were run on a dataset of piano music signals, from which it is hoped to learn templates that represent the spectra of notes. This presents a difficult task for NMF algorithms; while the spectra of piano notes have a fixed structure the problem of separability is evident as spectral overlap is pronounced due to harmonic structure and the logarithmic pitch scale. Indeed, in the first paper in which NMF was suggested for Automatic Music Transcription (AMT) [17] the authors consider that each note might have to be played in isolation at least once so that an atom representative of that note may be found. A standard dataset from the MAPS [26] database is used. This consists of thirty segments, each 30 s long of classical pieces that are recorded live from robotic playback on a Disklavier piano. Each piece is sampled at 44.1 kHz and ERBT spectrograms [27], with dimension $M = 512$, are produced for each signal, with 23 ms time frames used. These are logarithmic frequency scales that are seen to be superior for AMT relative to the STFT [28].

To analyse the NMF outputs, some post-processing is performed. A pitch estimate is assigned to each output atom. The fundamental frequency of the pth pitch on the piano scale is expressed as $f_o^p = 2^{\frac{p-49}{12}} \times 440$, with the expected frequency of the rth harmonic partial is given by $f_r = rf_0$. Pitch is estimated by weighted addition of the coefficients of harmonic partials, and sidelobes of a note. Given f_r is most closely associated with the ρth dimension of the frequency spectrum, the strength of the pth pitch in the kth atom is

$$S_k^p = \frac{1}{R^p} \sum_{r=1}^{R^p} \frac{1}{\sqrt{r}} \sum_{e=\rho-1:\rho+1} [A]_{e,k}$$

where R^p is the number of partials considered, which is set to a maximum of 10. The strongest pitch is assigned to the kth atom, as $\hat{p}^k = \arg\max_p S_k^p$. Similar to [27], a pitch salience matrix $\mathbf{H}$ is calculated by gathering the activations of a collection of atoms $\mathbf{A}(p)$ sharing the pth pitch label

$$[H]_{p,n} = \|\mathbf{A}(p)\mathbf{x}_n(p)\|_2. \tag{17}$$

Thresholding is applied to $\mathbf{H}$ [27] with a thresholding parameter, γ applied to the maximum pitch salience in order to determine the threshold $\zeta = \gamma \times \max_{p,n}[H]_{p,n}$ with all elements of $\mathbf{H}$ with values less than ζ set to zero. Analysis of AMT is then performed by denoting the true positives, tp, false positives, fp, and false negatives fn, from which the $\mathcal{F}$-measure

$$\mathcal{F} = \frac{2|tp|}{2|tp| + |fp| + |fn|} \times 100\%.$$

is determined for the optimal value of γ.

Several NMF algorithms were compared, including MU methods for KL, Itakuro-Saito(IS), and Hellinger NMF (H-NMF). Sparse NMF algorithms include NMF-ℓ_0 [16], a Euclidean distance algorithm with NN-OMP [12] used for ℓ_0 approximation, and the Hellinger distance based approaches, HSC-NMF [18] and HIT-NMF, proposed here. For HIT-NMF a value of $\delta = 0.02$ (15) was used, and 25 iterations were performed. Further comparison is made with semi-supervised Harmonic-NMF, (SS-β-NMF) [27] considered state-of-the-art for NMF-based AMT, and supervised β-NMF, (S-β-NMF) using a fixed dictionary, learnt offline from isolated notes recorded in the same environment as the dataset. A value of $\beta = 0.5$, reported to be optimal [27], is used.

Two experimental setups were employed; in the first case NMF was performed on each of the 30 pieces individually, using a dictionary of 88 atoms, randomly initialised, noting that 88 is the number of keys on a grand piano. In the second case all pieces were factorised simultaneously, with $\mathbf{S} \in \mathbb{R}^{512\times 38760}$, with dictionary size varied in multiples of 88. A final regression with supervised β-NMF with $\beta = 0.5$ is performed with all learnt dictionaries before estimation of (17) to enable a fair comparison of dictionary learning capabilities..

Results are given in Table. 1. Performance of all the multiplicative update NMF algorithms is seen to be similar, with

	Dictionary Size			
	88	*176*	*264*	$\mathcal{S}$
H-NMF	58.5	55.6	53.2	60.2
KL-NMF	58.8	53.7	53.3	60.0
IS-NMF	59.4	57.8	54.8	61.7
NMF-ℓ_0 [16]	61.4	66.1	67.4	65.2
HSC-NMF[18]	70.2	72.5	73.3	66.4
HIT-NMF	**71.4**	**74.2**	**74.7**	**67.5**
SS-β-NMF[27]	65.8			67.7
S-β-NMF	74.4			

Table 1. AMT results in $\mathcal{F}$-measure for NMF and sparse NMF methods when learning is performed on a large dataset with the dictionary size given on top, and when learning is performed separately on each piece, denoted on top by $\mathcal{S}$. Results for supervised β-NMF with fixed optimal dictionary and for semi-supervised harmonic NMF [27] also given.

IS-NMF performing slightly better than the others. In terms of the presented data, the NMF methods perform worse when presented with the full dataset, and deteriorate when the dictionary size is increased. Coordinate descent approaches for KL [4] and Hellinger cost functions resulted in similar results. The sparse NMF methods perform better than standard NMF approaches in all cases. NMF-ℓ_0 performs well on the separated dataset, but doesn't improve much when presented with the full dataset, deteriorating in one case. The Hellinger based methods perform better in all cases, improving significantly when presented with the full dataset, with further improvement with the larger dictionaries. HIT-NMF improves relative to HSC-NMF[18] in all cases, performing similar to harmonically constrained SS-β-NMF when learning on the separate pieces. When learning is performed on the full dataset, HIT-NMF improves by 7% relative to SS-β-NMF, and performs similar to the supervised β-NMF.

5. CONCLUSIONS

A sparse NMF method was proposed for Hellinger distance employing an iterative thresholding approach, and was seen to improve upon other unsupervised NMF and sparse NMF methods. In particular, when presented with a larger dataset for learning the standard NMF methods failed to improve, indicating an unsuitability for such problems. On the contrary, HIT-NMF and HSC-NMF improved with more data, and were seen to perform similar to the supervised algorithm, thereby justifying the ℓ_0-penalised Hellinger distance approach. However, some alterations in the algorithmic approach may be considered. The Newton coordinate descent approach was slow relative to the multiplicative updates, and parallel updates may be preferable, while further steps will be explored to ensure monotonicity of the combined coefficient update and threshold step.

6. REFERENCES

[1] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Advances in Neural Information Processing Systems (NIPS 14)*, Denver, 2000, pp. 556–562.

[2] C. Fevotte, N. Bertin, and J.-L. Durrieu, "Nonnegative matrix factorization with the Itakura-Saito divergence. With application to music analysis," *Neural Computation*, vol. 21, no. 3, pp. 793–830, March 2009.

[3] A. Cichocki, R. Zdunek, and S. Amari, "Csiszar's divergences for non-negative matrix factorization: Family of new algorithms," in *6th International Conference on Independent Component Analysis and Blind Signal Separation (ICA)*, Florida, 2006, pp. 32–39.

[4] C. J. Hsieh and I. S. Dhillon, "Fast coordinate descent methods with variable selection for non-negative matrix factorization," in *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, San Diego, 2011, pp. 1064–1072.

[5] P. Paatero and U. Tapper, "Positive matrix factorisation: A non-negative factor model with optimal utilization of error," *Environmetrics*, vol. 5, pp. 111–126, 1994.

[6] D. L. Sun and C. Fevotte, "Alternating direction method of multipliers for non-negative matrix factorization with the beta-divergence," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 6201–6205.

[7] A. Cichocki, S. Amari, R. Zdunek, R. Kompass, G. Hori, and Z. He, "Extended smart algorithms for non-negative matrix factorization," *Lecture notes in Artificial Intelligence, 8th International Conference on Artificial Intelligence and Soft Computing (ICAISC)*, vol. 4029, pp. 548–562, 2006.

[8] D. Donoho and V. Stodden, "When does non-negative matrix factorisation give a correct decomposition into parts?," in *Advances in Neural Information Processing Systems 16*, 2004.

[9] J. Rapin, J. Bobin, A. Larue, and J.-L. Starck, "Sparse and non-negative BSS for noisy data," *IEEE Transactions on Signal Processing*, vol. 61, no. 22, pp. 5620–5632, 2013.

[10] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Journal on Scientific Computing*, vol. 20, pp. 33–61, December 1998.

[11] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," in *Proceedings of the 27th Annual Asilomar Conference on Signals, Systems, and Computers*, Pacific Grove, CA, 1993, vol. 1, pp. 40–44.

[12] A. M. Bruckstein, M. Elad, and M. Zibulevsky, "On the uniqueness of non-negative sparse solutions to underdetermined systems of equations," *IEEE Transactions on Information Theory*, vol. 54, no. 11, pp. 4813–4820, November 2008.

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

[14] V. Y. F. Tan and C. Fevotte, "Automatic relevance determination in nonnegative matrix factorization with the beta-divergence," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 7, pp. 1592 – 1605, July 2013.

[15] M. Aharon, M. Elad, and A. M. Bruckstein, "K-SVD and its non-negative variant for dictionary design," in *Proceedings of the SPIE conference (Wavelets XI)*, Baltimore, 2005, pp. 327–339.

[16] R. Peharz, M. Stark, and F. Pernkopf, "Sparse nonnegative matrix factorisation using ℓ_0 constraints," in *Proc. of the IEEE International Workshop on Machine Learning for Signal Processing*, Kittila, 2010, pp. 83–88.

[17] P. Smaragdis and J. C. Brown, "Non-negative matrix factorization for polyphonic music transcription," in *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, New Paltz, 2003, pp. 177–180.

[18] K. O'Hanlon, M. D. Plumbley, and M. Sandler, "Non-negative matrix factorisation incorporating greedy Hellinger sparse coding applied to polyphonic music transcription," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing(ICASSP)*, Brisbane, 2015.

[19] K. O'Hanlon, M. D. Plumbley, and M. B. Sandler, "Sparse NMF using Powered Euclidean distances," Submitted to IEEE JSTSP, 2015.

[20] M. Zibulevsky and M. Elad, "L1-L2 Optimisation in signal and image processing: Iterative shrinkage and beyond ," *IEEE Signal Processing Magazine*, vol. 27, no. 3, pp. 76–88, May 2010.

[21] T. Blumensath and M. E. Davies, "Iterative hard thresholding for compressed sensing," *Applied and Computational Harmonic Analysis*, vol. 27, no. 3, pp. 265–274, November 2009.

[22] W. Dai and O. Milenkovic, "Subspace pursuit for compressive sensing signal reconstruction," *IEEE Transactions on Information Theory*, vol. 55, no. 5, pp. 2230–2249, May 2009.

[23] Goran Marjanovic, Magnus O Ulfarsson, and Alfred O Hero III, "Mist: $l0$ sparse linear regression with momentum," *arXiv preprint arXiv:1409.7193*, 2014.

[24] T. Steersman, "On the total variation and Hellinger distance between signed measures; an application to product measures," *Proceedings of the American Mathematical Society*, vol. 88, no. 4, August 1983.

[25] C. Fevotte and J. Idier, "Algorithms for nonnegative matrix factorization with the beta-divergence," *Neural Computation*, vol. 23, no. 9, pp. 2421–2456, September 2011.

[26] V. Emiya, R. Badeau, and B. David, "Multipitch estimation of piano sounds using a new probabilistic spectral smoothness principle," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 6, pp. 1643–1654, August 2010.

[27] E. Vincent, N. Bertin, and R. Badeau, "Adaptive harmonic spectral decomposition for multiple pitch estimation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 3, pp. 528–537, March 2010.

[28] K. O'Hanlon and M. D. Plumbley, "Row-weighted decompositions for automatic music transcription," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Vancouver, 2013, pp. 16–20.