



Title	Monophonic sound source separation by non-negative sparse autoencoders
Author(s)	Zen, Keiki; Suzuki, Masahiro; Sato, Haruhiko; Oyama, Satoshi; Kurihara, Masahito
Citation	Systems, Man and Cybernetics (SMC), 2014 IEEE International Conference on, ISBN: 978-1-4799-3840-7, 3623-3626 https://doi.org/10.1109/SMC.2014.6974492
Issue Date	2014
Doc URL	http://hdl.handle.net/2115/65295
Rights	© 2014 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.
Type	proceedings (author version)
File Information	smc2014-zen.pdf



[Instructions for use](#)

Monophonic Sound Source Separation by Non-negative Sparse Autoencoders

Keiki Zen, Masahiro Suzuki, Haruhiko Sato, Satoshi Oyama, Masahito Kurihara
Graduate School of Information Science and Technology
Hokkaido University
Sapporo, Hokkaido 060-0814, Japan
{xen,masa,haru}@complex.ist.hokudai.ac.jp, {oyama,kurihara}@ist.hokudai.ac.jp

Abstract—Monophonic sound source separation is an essential subject on the fields where sound, such as voice, music and noise, is dealt with. In particular, unsupervised approaches to this problem have high versatility in comparison with supervised approaches. Non-negative matrix factorization is the most frequently used algorithm for the monophonic sound source separation without prior knowledge. This is also applied to various applications, including data clustering, face recognition, gene expression classification. However, non-negative matrix factorization cannot be efficiently used in online learning. In order to solve this difficulty, the non-negative sparse autoencoder was proposed in the literature. Although several successful applications have been reported, this is not yet applied to the monophonic sound source separation. This paper shows that the non-negative sparse autoencoder can perform the monophonic sound source separation without prior knowledge in online learning.

Keywords—monophonic sound source separation, unsupervised learning, online learning, non-negative sparse autoencoder, NMF

I. INTRODUCTION

Recently, it has become common to process audio signals in everyday life. People are highly utilizing sound, voice, and music with devices such as smartphones. The audio signal processing supports these various applications. In particular, the sound source separation which extracts an objective source sound from mixed sounds is a fundamental technique used for speech recognition, noise reduction and many other feature extraction problems. The sound source separation is categorized into supervised and unsupervised methods and is also divided into multi-channel and monophonic sound source separation. The unsupervised monophonic sound source separation is the most versatile technique. To put it differently, it is applicable to all sound data that already exist and is available in the environment without enough channels.

Non-negative matrix factorization (NMF) [1] is the most frequently used algorithm for the unsupervised monophonic sound source separation. NMF is a multivariate statistical method that decomposes non-negative data into additive non-negative components. It can be naturally assumed that many data in the real world, such as the number, a frequency, a power spectrum, and a pixel value, are non-negative. Therefore, NMF is applied to a variety of purposes, including data clustering, face recognition, gene expression classification [2]. However, NMF cannot be efficiently used in online learning, which learns one instance at a time. In the standard NMF, to update the model, recalculation should be performed on existing data as well as new data. Although some online NMF algorithms

[3], [4] were proposed in the literature, a shortcoming of iterative encoding persists. This drawback is due to the fact that all matrix factorization approaches regardless of offline or online compute *linear* matrix factorization equation iteratively to achieve *non-linear* encodings algorithmically [5].

To overcome this difficulty, the non-negative sparse autoencoder (NNSAE) [5] was proposed in the literature. The NNSAE is based on the neural network called autoencoder, and hence it has non-linear structure in comparison with NMF based on matrix factorization. Nevertheless, it can decompose non-negative data into additive non-negative components as with NMF. In addition, it is an online learning algorithm, which can handle time-varying data sets efficiently. It is reported that the NNSAE often outperforms NMF in the image processing field which produces parts-based image representations. On the other hand, the NNSAE has not been applied to other applications. It is important to confirm that the NNSAE is more effective than NMF in other application fields.

In this paper, we compare NMF and the NNSAE through experiments and show that the NNSAE can perform in monophonic sound source separation without prior knowledge in online learning.

II. BACKGROUND KNOWLEDGE

A. Non-negative Matrix Factorization (NMF)

NMF is a multivariate statistical method that factorizes a matrix which comprises non-negative vectors into two matrices which also consist of non-negative vectors [1]. This factorization is represented by the following approximate equation

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

where \mathbf{X} is the dataset matrix whose column represents a vectorized instance. The data dimension is V , and the number of the data is N , therefore the size of \mathbf{X} is $V \times N$. The size of \mathbf{W} and \mathbf{H} are $V \times K$ and $K \times N$ respectively. K is the number of components, and it must be defined beforehand. In order to produce a meaningful result, K must be less than V and N . The elements of \mathbf{X} , \mathbf{W} , and \mathbf{H} are non-negative value x_{vn} , w_{vk} , and h_{kn} . The column vector $\mathbf{w}_k = [w_{1k}, \dots, w_{V_k}]^T$ of \mathbf{W} expresses a component called a basis, and each element of the column vector $\mathbf{h}_n = [h_{1n}, \dots, h_{Kn}]^T$ of \mathbf{H} implies the strength of the basis \mathbf{w}_k , and then they attempt to reconstruct $\mathbf{x}_n = [x_{1n}, \dots, x_{Vn}]^T$.

It is necessary to define the distance between two matrices \mathbf{WH} and \mathbf{X} . In NMF, the square Frobenius norm, the

generalized Kullback-Leibler divergence, and the Itakura-Saito distance are widely used [6]. NMF is regarded as the problem of the minimization of these distances. The representative optimization algorithm of NMF is an efficient iterative method based on the multiplicative update rules [7]. In this algorithm, \mathbf{W} and \mathbf{H} are initialized with random values and then each element is updated repeatedly by update rules until all elements converge. The definition of the update rules depends on the defined distance. On the square Frobenius norm distance, the update rules are defined as

$$w_{vk} \leftarrow w_{vk} \frac{\sum_n x_{vn} h_{kn}}{\sum_n \hat{x}_{vn} h_{kn}}, \quad (2)$$

$$h_{kn} \leftarrow h_{kn} \frac{\sum_v x_{vn} w_{vk}}{\sum_v \hat{x}_{vn} w_{vk}}. \quad (3)$$

Although NMF has been applied to a variety of purposes, several open problems remain. In particular, the difficulty regarding online learning should be mentioned. NMF is an offline (batch) learning algorithm, therefore it is not suitable for real-time application and time-varying data sets.

B. Non-negative Sparse Autoencoder (NNSAE)

In order to solve this difficulty, the non-negative sparse autoencoder was proposed by Lemme *et al.* [5]. The autoencoder is a neural network that learns to reconstruct input data, which is frequently used and studied in deep learning (representation learning) [8] in late years. It consists of the input layer, the hidden layer, and the output layer as shown in Fig. 1(a). The colored neuron has an activation function.

The NNSAE introduces the concepts of non-negativity and sparseness to the standard autoencoder. Unlike the autoencoder, it does not have activation functions of the output layer, as shown in Fig. 1(b). The dimension of the input \mathbf{x} and the output $\hat{\mathbf{x}}$ is V , and the dimension of the hidden layer output \mathbf{h} is K . In Fig. 1, the two connection weight matrices \mathbf{W} and \mathbf{W}^T are constrained to be transposed (called tied weights), and the size of \mathbf{W} is $V \times K$. The activation function of the k th hidden layer neuron is the following sigmoid function

$$f_k(x_k) = \frac{1}{1 + e^{-a_k x_k - b_k}} \quad (4)$$

with the two parameters of the slope a_k and the bias b_k . Therefore, \mathbf{h} and $\hat{\mathbf{x}}$ are given by

$$\mathbf{h} = f(\mathbf{W}^T \mathbf{x}), \quad (5)$$

$$\hat{\mathbf{x}} = \mathbf{W} \mathbf{h}, \quad (6)$$

where f is the element-wise calculation using Eq. (4). The non-negativity of the NNSAE is enforced by using an appropriate asymmetry decay function in updating the weights as explained below. First, an error function is defined as the square error function

$$E = \|\mathbf{x} - \hat{\mathbf{x}}\|^2, \quad (7)$$

and then the gradient descent of Eq. (7) yields the following update rule

$$\tilde{w}_{vk} = w_{vk} + \eta(x_v - \hat{x}_v)h_k, \quad (8)$$

$$w_{vk} \leftarrow \tilde{w}_{vk}, \quad (9)$$

where η is a learning rate. In order to guarantee the non-negativity of the weight matrix \mathbf{W} , here the NNSAE utilizes the asymmetry decay function

$$d(\tilde{w}_{vk}) = \begin{cases} -\alpha \tilde{w}_{vk} & (\tilde{w}_{vk} < 0) \\ -\beta \tilde{w}_{vk} & (\tilde{w}_{vk} \geq 0), \end{cases} \quad (10)$$

where α and β are hyperparameters which control the non-negativity. Finally, the update rule is defined as

$$w_{vk} \leftarrow \tilde{w}_{vk} + d(\tilde{w}_{vk}). \quad (11)$$

Typically, the weight decay appears as $-\lambda w_{vk}$ in an update rule when regularization $-\lambda \|\mathbf{W}\|^2$ is added to the square error function. This is equivalent to the Gaussian prior distribution of the network weights, and the asymmetry decay function is equivalent to a skewed prior distribution. We set $\alpha = 1$ and $\beta = 0$ in the following experiments. These yield completely non-negative weight values.

The sparseness of the NNSAE is enforced by an adaptive learning rate and the intrinsic plasticity (IP) mechanism [9]. The learning rate η is defined as the adaptive learning rate

$$\eta = \frac{\tilde{\eta}}{\|\mathbf{h}\|^2 + \epsilon}, \quad (12)$$

where ϵ is a small value to prevent zero division. This adaptive learning rate takes a large value when few neurons of the hidden layer spike. By the IP mechanism, the parameters of the sigmoid function are updated by

$$\Delta b_k = \eta_{IP} \left(1 - \left(2 + \frac{1}{\mu} \right) h_k + \frac{1}{\mu} h_k^2 \right), \quad (13)$$

$$\Delta a_k = \eta_{IP} \frac{1}{a_k} + g_k \Delta b_k, \quad (14)$$

where η_{IP} is a learning rate, and g_i is input of the i th neuron, and μ is the mean of a desirable output distribution. As the training time elapses, the output distribution is optimized to the desirable exponential distribution. In this way, the hidden layer neurons output \mathbf{h} obtains the sparseness.

III. EXPERIMENTS AND ANALYSIS

We performed two experiments using mixed instrumental sounds. In Experiment 1, we executed the monophonic sound source separation with the NNSAE and NMF. In Experiment 2, we changed the mixed sound to a different one which is composed of other instruments during training.

A. Experimental Setup

First, we prepared tones of four instruments, guitar, piano, bass and trumpet, and then we sounded them at random period and generated two mixed sound data. Sound 1 comprises guitar and piano tones, and Sound 2 consists of bass and trumpet tones. Second, we converted Sound 1 and Sound 2 into spectrograms by the short-time Fourier transform. The size of the spectrograms are shown in Table I. The frequency points corresponds to the number of the data dimension V , and the time points is the number of the data N . The parameters used in the NNSAE and NMF are shown in Table II.

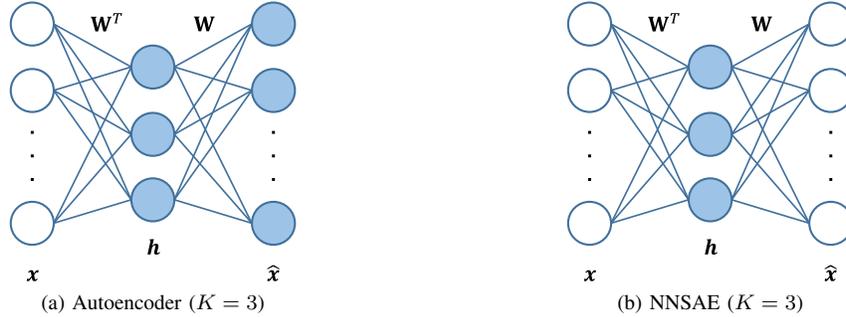


Fig. 1. The network structure of the autoencoder and the NNSAE with the input \mathbf{x} and the output $\hat{\mathbf{x}}$ in the case of $K = 3$. \mathbf{x} and $\hat{\mathbf{x}}$ have the same dimension. \mathbf{h} is the output of the hidden layer neurons. The two connection weight matrices \mathbf{W} and \mathbf{W}^T are constrained to be transposed to each other. The colored neuron has an activation function. The NNSAE does not have activation functions in the output layer.

TABLE I. THE SIZE OF THE SPECTROGRAMS

	Frequency points	Time points
Sound 1	513	6870
Sound 2	513	3883

TABLE II. THE PARAMETERS OF THE NNSAE AND NMF

NNSAE									NMF
K	α	β	$\tilde{\eta}$	a_0	b_0	μ	η_{IP}	Batch size	K
2	1	0	0.01	1	-3	0.2	0.0001	20	2

B. Results

In Experiment 1, we executed the monophonic sound source separation of Sound 1 by the NNSAE and NMF. The results of separation are shown in Fig. 2. On each sub figure 2(a) and 2(b), the left column is for spectrograms, and the right column is for activation time. Fig. 2 shows that the NNSAE generated the same separation as NMF with respect to the frequency and the activation. We also present the reconstruction error over time in Fig. 3(a). It shows that the NNSAE converges at almost the same reconstruction error as NMF. In addition, Fig. 3(b) shows the change of the sparseness over time in Experiment 1. The sparseness is defined by the following formula [10]

$$s(\mathbf{w}) = \left(\sqrt{V} - \frac{\sum_{i=1}^V |w_i|}{\sqrt{\sum_{i=1}^V w_i^2}} \right) (\sqrt{V} - 1)^{-1}, \quad (15)$$

where V is the dimension of \mathbf{w} . The value of (15) becomes large when \mathbf{w} is a sparse vector having few positive values and many zeros. This concept is introduced to capture the intuition that the characteristics of data will be well represented by sparse components. Fig. 3(b) reports the fact that the NNSAE converges at almost the same sparseness of \mathbf{W} as NMF.

In Experiment 2, we changed the mixed sound to a different one during training. First, Sound 2 consisting of bass and trumpet tones was used to train the NNSAE, and then Sound 1 consisting of guitar and piano tones was successively given to the NNSAE. The results of performance indexes are shown in Fig. 4. Fig. 4(a) shows the error versus epoch, and Fig. 4(b) shows the sparseness versus epoch. Sound 2 had been given to the NNSAE until the 25th epoch and Sound 1 was given to

after the 26th epoch.

From Figures 2 and 3, the NNSAE and NMF gave almost the same separation results. Hence, it can be said that the NNSAE has the capability for the monophonic sound source separation as with NMF. Furthermore, Fig. 3 shows that the NNSAE can be performed at a lower cost than NMF. This implies strongly that the NNSAE is a more efficient algorithm than NMF. As shown in Fig. 4, the NNSAE can rapidly adapt to the new mixed sound data. This shows that the NNSAE can perform the monophonic sound source separation in the online learning setting.

IV. CONCLUSION

In this paper, we applied the NNSAE to the monophonic sound source separation, which has been mainly handled by NMF. In Experiment 1, the NNSAE successfully performed the monophonic sound source separation as with NMF. In Experiment 2, we changed the instruments used in the mixed sound during training. Through the two experiments, we found that the NNSAE is superior to NMF in the following aspects: (1) The NNSAE has the capability for the monophonic sound source separation as with NMF and performs it at a lower cost than standard NMF. (2) The NNSAE can perform the monophonic sound source separation in online learning, while NMF is limited to batch learning.

REFERENCES

- [1] 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.
- [2] A. Cichocki, R. Zdunek, A. H. Phan, and S. Amari, *Nonnegative matrix and tensor factorizations: applications to exploratory multi-way data analysis and blind source separation*. John Wiley & Sons, 2009.
- [3] B. Cao, D. Shen, J.-T. Sun, X. Wang, Q. Yang, and Z. Chen, "Detect and track latent factors with online nonnegative matrix factorization." in *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI)*, vol. 7, 2007, pp. 2689–2694.
- [4] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *The Journal of Machine Learning Research*, vol. 11, pp. 19–60, 2010.
- [5] A. Lemme, R. F. Reinhart, and J. J. Steil, "Online learning and generalization of parts-based image representations by non-negative sparse autoencoders," *Neural Networks*, vol. 33, pp. 194–203, 2012.

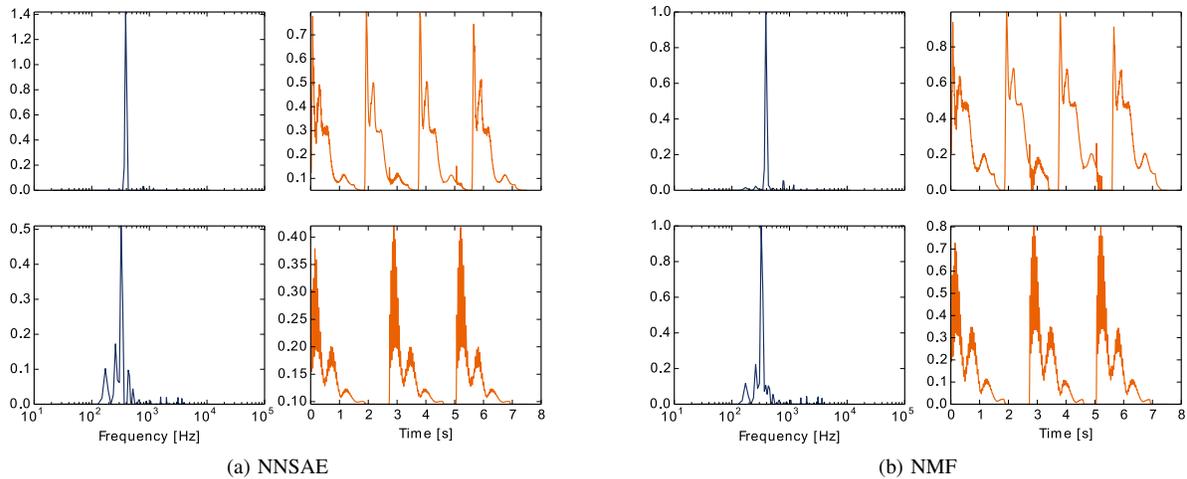


Fig. 2. The separations of Sound 1 in Experiment 1. On each sub figure (a) and (b), the left column is for spectrograms, and the right column is for activation time.

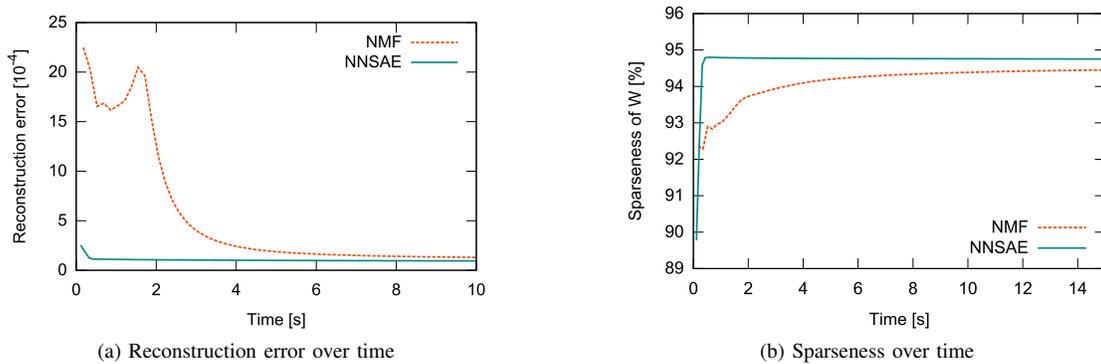


Fig. 3. The performance in Experiment 1. The sub figures (a) is the reconstruction error over time. The sub figures (b) is the sparseness over time.

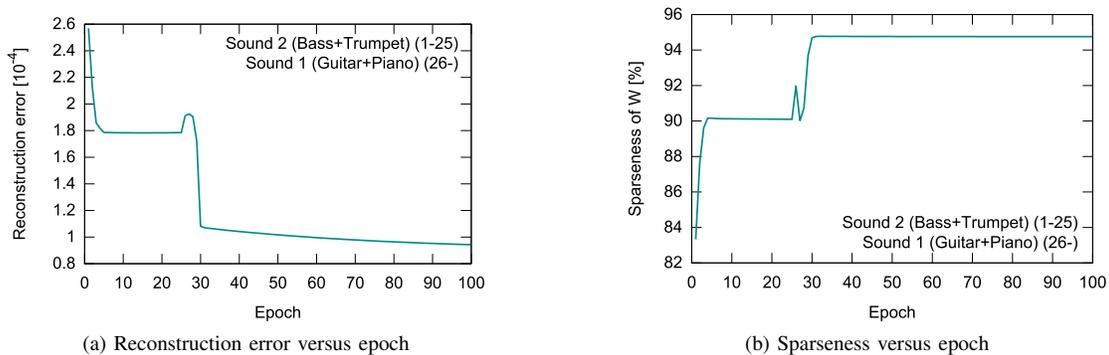


Fig. 4. The performance in Experiment 2. The sub figures (a) is the reconstruction error versus epoch. The sub figures (b) is the sparseness versus epoch.

[6] M. Nakano, H. Kameoka, J. Le Roux, Y. Kitano, N. Ono, and S. Sagayama, "Convergence-guaranteed multiplicative algorithms for nonnegative matrix factorization with β -divergence," in *Proceeding of the 2010 IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, 2010, pp. 283–288.

[7] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Advances in Neural Information Processing Systems 13*. MIT Press, 2001, pp. 556–562.

[8] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[9] J. Triesch, "A gradient rule for the plasticity of a neuron's intrinsic excitability," in *Artificial Neural Networks: Biological Inspirations—ICANN 2005*. Springer, 2005, pp. 65–70.

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