Learning convolutive features for storage and transmission between networked sensors

Ruairí de Fréin
†KTH - Royal Institute of Technology, Stockholm, Sweden
web: https://robustandscalable.wordpress.com

in: Neural Networks (IJCNN), 2015 International Joint Conference on. See also BIBTEX entry below.

BIBTEX:

```
@article{rdefrein15Learning,
author={Ruair\'{i} de Fr\'{e}in},
booktitle={Neural Networks (IJCNN), 2015 International Joint Conference on},
title={Learning convolutive features for storage and transmission between networked sensors},
year={2015},
pages={1-8},
keywords={compressed sensing;learning (artificial intelligence);
matrix decomposition;CNMF;SNMF;bandwidth saving;compression ratio;
convolutive feature learning;convolutive nonnegative matrix factorization;
integer sequence compression literature;learned convolutive factors;
machine learning;networked sensing systems;networked sensors;
signal processing methods;storable NMF;Noise;},
doi={10.1109/IJCNN.2015.7280827},
url={http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7280827&isnumber=7280295},
month={July},}
```


document created on: November 5, 2015
created from file: rdefreinIJCNN15_onlineprepreint.tex
cover page automatically created with CoverPage.sty
(available at your favourite CTAN mirror)

Learning Convolutive Features for Storage and Transmission between Networked Sensors

Ruairí de Fréin
KTH Royal Institute of Technology, Sweden
rdefrein@gmail.com

***Abstract*—Discovering an efficient representation that reflects the structure of a signal ensemble is a requirement of many Machine Learning and Signal Processing methods, and gaining increasing prevalence in sensing systems. This type of representation can be constructed by Convolutive Non-negative Matrix Factorization (CNMF), which finds parts-based convolutive representations of non-negative data. However, convolutive extensions of NMF have not yet considered storage efficiency as a side constraint during the learning procedure. To address this challenge, we describe a new algorithm that fuses ideas from the 1) parts-based learning and 2) integer sequence compression literature. The resulting algorithm, Storable NMF (SNMF), enjoys the merits of both techniques: it retains the good-approximation properties of CNMF while also taking into account the size of the symbol set which is used to express the learned convolutive factors and activations. We argue that CNMF is not as amenable to transmission and storage, in networked sensing systems, as SNMF. We demonstrate that SNMF yields a compression ratio ranging from 10:1 up to 20:1, depending on the signal, which gives rise to a similar bandwidth saving for networked sensors.**

I. Introduction

Independent Component Analysis (ICA) has gained popularity as a method of reducing the redundancy of data by projecting it onto its independent components [1]. The underpinning principle of many members of this class of learning algorithms is to maximize a statistical measure such as independence [2], or non-Gaussianity [3]. Non-negative Matrix Factorization (NMF) can be a more suitable alternative when the data is element-wise non-negative (NN). NMF approximates element-wise NN data, which is expressed in matrix form, $\mathbf{V} \in \mathbb{R}^{\geq 0, M \times N}$, as the product of two element-wise NN matrices $\mathbf{W} \in \mathbb{R}^{M \times R}$ and $\mathbf{H} \in \mathbb{R}^{R \times N}$ [4], [5]. By their own nature, some problems do not involve negative values. Instead of maximizing measures of independence or non-Gaussianity, NMF finds the hidden parts [6] that generate the data, learning a *parts-based representation* –a term made popular in [5].

Given that the only constraint NMF enforces is that the data and factors are element-wise NN, and its optimization routine is non-convex, it has been difficult to prove when NMF gives the correct solution. Vavasis asserts in [7] that NMF is NP-hard; However, a polynomial-time local search heuristic exists. Donoho and Stodden showed that NMF only gives the exact and unique solution for certain classes of parts-based data-sets [6]. Nevertheless, the popularity of NMF in recent years has grown unabated due to the simplicity of its updates and the success of its application in areas such as bioinformatics [8] and computer vision [9]. In this paper, we consider Donoho and Stodden's parts-based explanation for the success of NMF and posit that there exists, in some sense, an even simpler representation of the data. We show that increasing the simplicity of NMF representations could be used to extend the application of NMF to domains where resources, such as storage and bandwidth, are constrained. Many advances in the compression literature rely on the fact that simpler representations for a data-set are typically more compressible. For non-adaptive signal analysis, the application of the Discrete Cosine Transform (Wavelets) in lossy jpeg (jpeg2000) encoding is a case in point [10], [11]. In terms of performing adaptive analysis, the use of a sliding dictionary of functions by Lempel and Ziv [12], has given rise to one of the most commonly used loss-less compression algorithms (gzip). We exploit the fact that parts-based representations are generally simpler representations, that simpler representations can be more compressible, and that if we interleave the parts-based representation learning with compression, we can improve compression without significantly affecting NMF.

Application domain: Our contribution, SNMF, is timely. The Internet-of-Things (IoT) imagines a world of pervasive connected devices, for example, mobile phones, sensors, home area appliances, laptops, tablets [13]. Most of these devices come equipped with acoustic sensors. These devices need to be able to communicate, cooperate and learn in order to achieve their designated goals, whilst also sharing constrained edge-network resources with other applications and services. Applications include acoustic scene monitoring, event detection, ambient assisted living, accident detection and speaker identification. In this connected world, which relies on a number of potentially bandwidth restricted inter-connected networks, these connected devices will produce gigantic quantities of data to be transferred or stored over the Internet. One challenge, illustrated in Fig. 1, is to learn acoustic features at the edge of the network, be-it on an individual sensor, a local gateway point or server, and to be able to transfer selected features between sensors in order to: 1) improve local classification or detection using a low-bit rate, or 2) to store more features on the sensors or connected devices themselves. The focus of this paper is on putting the first building block in place: learning storable features which are as accurate as CNMF's features. Higher level functions such as networked-classification and separation will be reported separately.

Spectrograms have traditionally been the touch-stone representation for speech [14] in speech enhancement; monaural Blind Source Separation (BSS) tasks [15]; and classification and identification of auditory objects. In combination with spectral magnitude analysis of audio, our contribution, SNMF, discovers auditory objects and their activation patterns in a

This work was supported by an ELEVATE Irish Research Council International Career Development Fellowship co-funded by Marie Cure Actions award: ELEVATEPD/2014/62. Dr de Fréin is with the Telecommunications Software and Systems Group, Waterford, Ireland

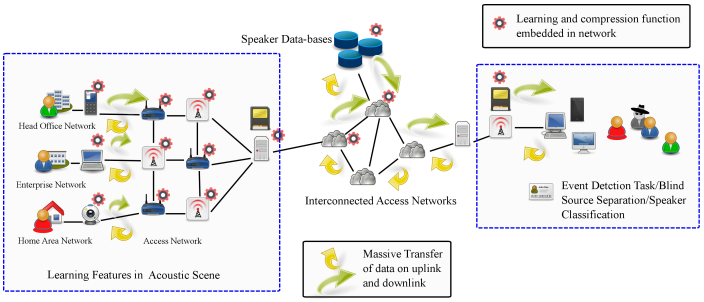

Fig. 1. Learning (learning agents are indicated by a cog) is performed on many connected devices in this IoT scenario. For example, on sensors (PDA, PC, Camera), gateway servers and in cloud data-bases. This vast amount of data transfer is indicated by up-link and down-link arrows. Features may also be stored locally (indicated by a flash drive symbol) on the sensing device etc. On the RHS, features learned in other parts of the network (LHS) may be shared and used to perform speaker identification or event classification. The bottleneck in this scenario is bandwidth - the more features available to each classifier, the greater the potential classification discrimination power/accuracy.

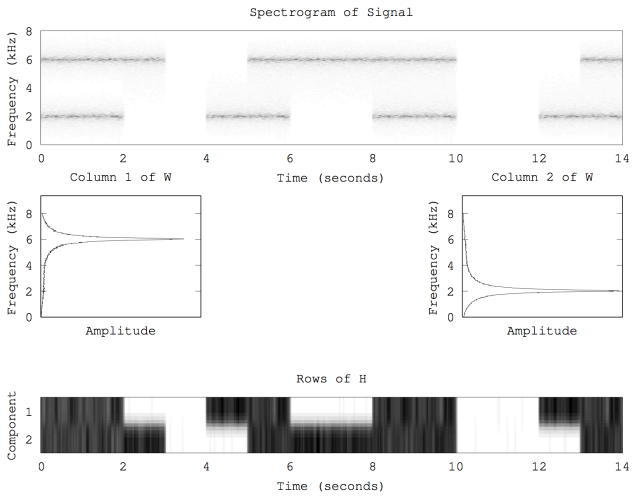

Fig. 2. Spectrogram of a signal composed of band-limited noise bursts (top row). NMF of this signal: The two learned basis functions are illustrated in row two; The activations of these basis functions are illustrated in row three.

manner which is similar in spirit to Non-negative Matrix Factor Deconvolution [16]; however we consider the problem of storage/transmission as well. SNMF builds on *instantaneous* quantized and Integer Matrix Factorization (IMF), [17], [18].

Contribution: We show that the elements of the matrices $\mathbf{W}$ and $\mathbf{H}$ learned by SNMF, are chosen from a set of symbols, which typically has a small cardinality. Moreover, restricting the elements of $\mathbf{W}$ and $\mathbf{H}$ to this symbol set does not significantly affect the accuracy of the decomposition. The motivation for SNMF is that the smaller the symbol set, the more likely the solution is to be amenable to compression: it is *simpler*. Putting this together, we demonstrate that compression can be achieved without significant loss of performance. With regard to the CNMF literature [16], the symbol set cardinality of the factors $\boldsymbol{W}$ and $\boldsymbol{H}$ has not been considered as a problem constraint. In short, we exploit Occam's razor three times. 1) we learn a low rank representation; 2) we learn parsimonious activation functions of the underlying parts by using a convolutive generative model; 3) we express the factorization using a simpler symbol set which has a small cardinality.

Organization: Section II reviews NMF and CNMF. We describe the difference between these mixing models using synthetic audio captured at acoustic sensors. Section III contributes SNMF. Section IV analyzes the performance of SNMF on a noise burst data-set, an evolving spectrogram data-set and speech from the TIMIT database. We demonstrate that an increase in storage/transmission is possible with SNMF, without incurring a significant cost in decomposition accuracy, by learning spectral features on speech captured on a single channel acoustic sensor. We conclude by discussing the benefits of SNMF in a networked sensors scenario in Section V.

II. Sensing and Learning Set-up: NMF

Consider the scenario in Fig. 1. A single channel acoustic sensor captures an audio signal at 16kHz in the "Learning Features in Acoustic Scene" area of the network and computes a magnitude spectrogram $\boldsymbol{V}$, either on the sensing platform, or some other intermediate gateway to the rest of the network. We use a Hamming window with a time window of 1024 samples and the signals are sampled at 16kHz in the remainder of this paper. The time windows have an overlap of 512 samples. Note that this set-up may be optimized for near-real-time sensing and learning. An alternative analysis function (to the discrete Fourier transform) may be used depending on the statistics of the signal ensemble. The matrix $\boldsymbol{V} \in \mathbb{R}^{M \times N}$ is a frequency-by-time matrix, where $M = 513$ is the number of retained (for learning) frequency bins (due to the symmetry of the discrete Fourier transform), and N is the number of time windows captured during the period the sensor is capturing audio. The sensor captures audio intermittently either on a duty cycle in a monitoring scenario, opportunistically (based on a voice detection or activity protocol), or in response to an interaction with a user. It has been argued in the literature [16] that each component in the NMF of $\boldsymbol{V}$ corresponds to an auditory object. We focus on the problem of learning these objects using the convolutive extension of NMF, namely CNMF, which allows these objects to have a temporal structure.

A. Non-negative Matrix Factorization

Given the sensed NN magnitude spectrogram $\mathbf{V} \in \mathbb{R}^{M \times N}$, NMF approximates $\mathbf{V}$ as the product of two element-wise NN matrices, $\tilde{\mathbf{V}} = \mathbf{WH}$. The columns of $\boldsymbol{W} \in \mathbb{R}^{M \times R}$ are audio features and the rows of $\boldsymbol{H} \in \mathbb{R}^{R \times N}$ are the activation patterns of these features in $\boldsymbol{V}$. The goal of NMF is to learn an accurate low-rank representation, that is, $R \ll M, N$. Approximation error is measured below using a generalized version of the Kullback Leibler Divergence (KLD).

$$D(\mathbf{V}||\tilde{\mathbf{V}}) = ||\mathbf{V} \otimes \log \frac{\mathbf{V}}{\tilde{\mathbf{V}}} - \mathbf{V} + \tilde{\mathbf{V}}|| \qquad (1)$$

Element-wise operations are frequently used. Hadamard multiplication and division are denoted $\otimes$ and $\frac{x}{y}$. Multiplicative updates, [4], [18] and [19], have played a crucial role in the uptake of NMF in recent years. Their appeal lies in the fact that NN constraints do not need to be explicitly *programmed* into the solver. Furthermore, the multiplicative algorithms presented by Lee and Seung are guaranteed to converge monotonically. One of the aims of NMF is to learn a *low-rank* solution. When $R < M$, $\mathbf{W}$ is under-determined: NMF reveals low rank features of the data.

B. Convolutive Non-negative Matrix Factorization

The instantaneous NMF mixing model does not capture the temporal extent of features. This limits its suitability for many real-world audio signals. For example, multiple observations of speech over nearby intervals of time have temporal relationships. These temporal relationships motivate the use of a convolutive generative model to extend ICA and NMF mixing models for speech [20], [16] and [21]. NMF models the sensed magnitude spectrogram $\boldsymbol{V}$ as the combination of a set of objects (single spectral features) and the activations of the each of these features in time. Convolutive NMF models each object as a sequence/cascade of spectral features and the activation of this cascade in time. Instead of activating each time slice (spectrum) of an object individually, we activate an entire cascade of basis functions:

$$\boldsymbol{V} \approx \tilde{\mathbf{V}} = \sum_{t=0}^{T-1} \mathbf{W}_t \overset{t\rightarrow}{H} . \tag{2}$$

The matrix $\tilde{\mathbf{V}} \in \mathbb{R}^{M\times N}$ is NMF's magnitude spectrogram estimate and $\mathbf{W}_t \in \mathbb{R}^{M\times R}$ and $\mathbf{H} \in \mathbb{R}^{R\times N}$ are its 1) temporally extended features and 2) activations of these cascades of spectral features. T is the number of time slices in each spectral cascade. The vector $\mathbf{W}_t[:,r]$ describes the spectrum of the r-th object t time steps after the object has begun. The time-shift operator $\overset{t\rightarrow}{(\cdot)}$ plays a crucial role: it moves its argument t places to the right. It is not a circular shift operator: the left most-columns are filled with zeros. CNMF's updates, expressed below, monotonically decrease the objective. Here $\mathbf{1}_{M\times N}$ is an $M \times N$ matrix of ones.

$$\mathbf{H} = \mathbf{H} \otimes \frac{\mathbf{W}_t^T \overset{\leftarrow t}{\left[\frac{\mathbf{V}}{\tilde{\mathbf{V}}}\right]}}{\mathbf{W}_t^T \mathbf{1}_{M\times N}}, \quad \mathbf{W}_t = \mathbf{W}_t \otimes \frac{\frac{\mathbf{V}}{\tilde{\mathbf{V}}} \overset{t\rightarrow T}{\mathbf{H}}}{\mathbf{1}_{M\times N} \overset{t\rightarrow T}{\mathbf{H}}} \tag{3}$$

C. Learning objects in auditory spectrograms

The difference between NMF and CNMF is explained by contrasting Fig 2 and 3. These examples are available in [21]. They provide a benchmark from the literature for comparing SNMF with NMF and CNMF [21]. In Fig. 2, two band-limited noise bursts are present in the magnitude spectrogram (centered around 2kHz and 6kHz). NMF is applied with $R = 2$ and the basis functions learned are shown along with their activations in the spectrogram. NMF correctly reveals that the magnitude spectrum, $\mathbf{V}$ is composed of two hidden parts –the spectra of the noise bursts. In Fig. 3, instead of noise bursts, two frequency sweeps centered on 2kHz and 6kHz make up the magnitude spectrogram $\mathbf{V}$. The application of CNMF, which allows the factorization to capture temporal evolution of the spectra (with temporal extent $T = 32$ and rank $R = 2$) reveals the cascade of spectra which form the auditory features. The activation functions, in row 3, are parsimonious. Only the starting point of each auditory object is activated. Here, although it is not shown due to space constraints, NMF fails to learn the correct decomposition. CNMF learns the correct decomposition. In this paper we attempt to learn the *correct* decomposition in each case, but in a manner that ensures that the learned solution is more efficient for storage.

III. Storable NMF

Non-adaptive Quantization: Taking a nonlinear approximation of each element of $\boldsymbol{W}_t$ and $\boldsymbol{H}$ to improve the compressibility of the factors introduces error into the factorization. For example quantizing the factors produced by the updates (Eqn. 3) at convergence using one quantization step-size $\Delta^\star$ for all rows and columns of the factors, where $\lfloor\cdot\rfloor$ denotes rounding, produces the new factors

$$\hat{\mathbf{H}} \leftarrow \left\lfloor \frac{\mathbf{1}}{\Delta^\star}\mathbf{H} + \frac{1}{2} \right\rfloor, \quad \hat{\mathbf{W}}_t \leftarrow \left\lfloor \mathbf{W}_t \frac{\mathbf{1}}{\Delta^\star} + \frac{1}{2} \right\rfloor . \tag{4}$$

The matrices $\hat{\boldsymbol{W}}_t$ and $\hat{\boldsymbol{H}}$ are element-wise NN and drawn from a subset of the integers $\mathbb{Z}$, which we denote $\mathbb{I}$. This indiscriminate, fixed quantization is inadequate as differences in the dynamic range of the values used to capture each component of $\boldsymbol{V}$, the matrices $\boldsymbol{A}_r$, may vary significantly.

$$\boldsymbol{A}_r = \sum_{t=0}^{T-1} \mathbf{W}_t[:,r] \overset{t\rightarrow}{\boldsymbol{H}} [r,:] \tag{5}$$

This affects compression efficiency as the choice of elements of the set $\mathbb{I}$ is not considered by the updates (Eqn. 3); A good $\mathbb{I}$ from a compression perspective may not be selected if it affects the quality of the factorization –this need not be dichotomous. Because one quantization step-size may not be appropriate for all components, certain components of the factorization may be indiscriminately penalized over others. Recall that NMF is non-unique [6]. The learned factorization is composed of the sum of members of the set of all possible components (where

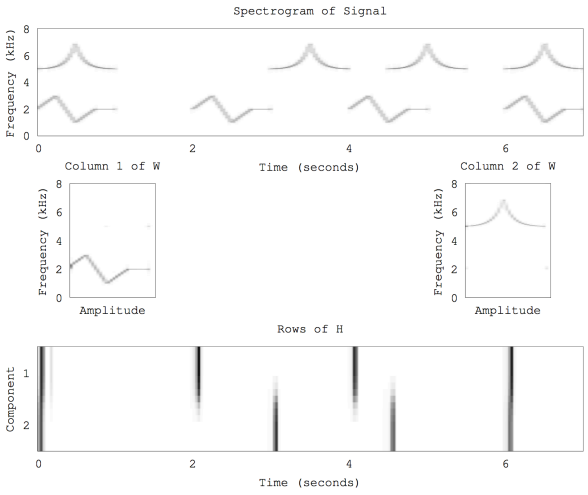

Fig. 3. Spectrogram of a signal composed of auditory objects that evolve with time (row 1). CNMF of this signal: Basis functions (that evolve with time) are illustrated in row two; Activations of these functions are in row three.

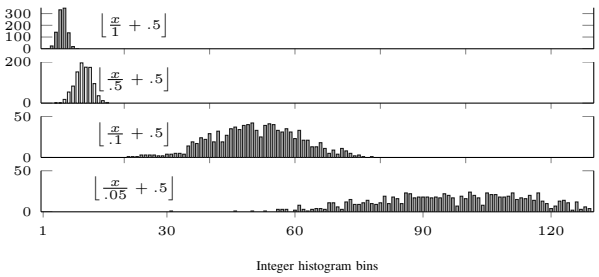

Fig. 4. Effect of tuning the step-size: Decreasing the step-size Δ, increases the resolution of the quantized vector and more bins are non-zero.

ϵ is an arbitrarily small accuracy target)

$$\Omega = \{\boldsymbol{A}_r | \boldsymbol{A}_r \text{ contributes to any approximation } \tilde{V} \text{ that satisfies } ||\boldsymbol{V} - \tilde{\boldsymbol{V}}||_2^2 \leq \epsilon\Big\}, \quad (6)$$

The dynamic range of the elements of the members of Ω may vary considerably. We desire a factorization that selects components from Ω that are compressible and accurate. The quantization functions in (Eqn. 4) may be adapted (one for each factor) by tuning the step-sizes and the non-linearities, $\lfloor \cdot \rfloor$, to achieve this. In Fig. 4, the values from a continuous rectified vector $\boldsymbol{x} \sim \mathcal{N}_+(4, 1)$ are quantized using a step-size from the set $\{1, .5, .1, .05\}$. Histograms of these new values are illustrated in rows 1–4. Decreasing the step-size, increases the resolution of the quantized vector (spreading it over more bins), which affects the cardinality of the symbol set $\mathbb{I}$. SNMF trades-off the resolution/cardinality of each of the different components of the factorization.

Adaptive Quantization: We start by adapting the quantization function's step-size for each column of $\boldsymbol{W}_t$ and each row $\boldsymbol{H}$. We interleave this adaptation with SNMF, as SNMF converges. This means the components chosen from the set Ω are quantized (by an adapted quantization function). They are element-wise integer, and generally more compressible, e.g.

$$\hat{\mathbf{W}}_t[:, r] \leftarrow \left\lfloor \frac{\mathbf{W}_t[:, r]}{\boldsymbol{\Delta}_w[r]} + \frac{1}{2} \right\rfloor, \forall r, t \quad (7)$$

$$\hat{\mathbf{H}}[r, :] \leftarrow \left\lfloor \frac{\mathbf{H}[r, :]}{\boldsymbol{\Delta}_h[r]} + \frac{1}{2} \right\rfloor, \forall r. \quad (8)$$

The cardinality of the set $\mathbb{I}$, denoted $|\mathbb{I}|$, is generally minimized as the set $\mathbb{I}$ is now indirectly learned with the factorization.

$$\mathbb{I} = \left\{ \bigcup_{m,r,t} \hat{\boldsymbol{W}}_t[m, r] \right\} \bigcup \left\{ \bigcup_{r,n} \hat{\boldsymbol{H}}[r, n] \right\}. \quad (9)$$

Initially $\mathbb{I}$ is a subset of the NN real numbers, $\mathbb{I} \subset \mathbb{R}_+$. At convergence, it is a subset of the positive integers $\mathbb{I} \subset \mathbb{Z}_+$. In the following problem statement diag($\boldsymbol{x}$) places the elements of its argument, $\mathbf{x} \in \mathbb{R}^{R \times 1}$, on the diagonal of an $R \times R$ matrix. The vector $\mathbf{1}_M$ denotes a $M \times 1$ vector of ones.

Definition 1: SNMF factorizes the matrix $\mathbf{V} \in \mathbb{R}^{M \times N}$ into the product of two element-wise integer matrices $\hat{\mathbf{W}}_t \in \mathbb{I}^{M \times R}$ and $\hat{\mathbf{H}} \in \mathbb{I}^{R \times N}$ and two diagonal matrices diag($\boldsymbol{\Delta}_w$) and diag($\boldsymbol{\Delta}_h$), where $\boldsymbol{\Delta}_w, \boldsymbol{\Delta}_h \in \mathbb{R}^{R \times 1}$ such that the KLD (Eqn. 1) is minimized. The SNMF mixing model is

$$\tilde{\mathbf{V}} = \sum_{t=0}^{T-1} \hat{\mathbf{W}}_t \text{diag}(\boldsymbol{\Delta}_w) \text{diag}(\boldsymbol{\Delta}_h) \overset{t\rightarrow}{\hat{\mathbf{H}}}. \quad (10)$$

In other words, the matrix $\boldsymbol{V}$ is approximated by the sum of quantized components $\tilde{\boldsymbol{V}} = \sum_r \boldsymbol{A}_r$, where

$$\boldsymbol{A}_r = \boldsymbol{\Delta}_w[r], \boldsymbol{\Delta}_h[r] \sum_{t=0}^{T-1} \hat{\mathbf{W}}_t[:, r] \overset{t\rightarrow}{\hat{\boldsymbol{H}}} [r, :]. \quad (11)$$

Appealingly the majority of the elements of each component in this model are integer, $\mathbf{W}_t[m, r], \overset{t\rightarrow}{\boldsymbol{H}} [r, n] \in \mathbb{I}$ and the quantization step-sizes $\boldsymbol{\Delta}_w[r], \boldsymbol{\Delta}_h[r] \in \mathbb{R}$, though real-valued, are few, and are adapted to minimize the cardinality of $\mathbb{I}$.

Quantization Step-size Updates: The quantization step-sizes $\boldsymbol{\Delta}_h[r]$ and $\boldsymbol{\Delta}_w[r]$, in the convolutional mixing model in (Eqn. 10) decrease the objective in (Eqn. 1) and produce NN updates if they are updated using

$$\boldsymbol{\Delta}_h[r] \leftarrow \boldsymbol{\Delta}_h[r] \frac{\mathbf{1}_M^T \left[\frac{\mathbf{V}}{\tilde{\mathbf{V}}} \otimes \left[\sum_{t=0}^{T-1} \hat{\mathbf{W}}_t[:, r] \boldsymbol{\Delta}_w[r] \overset{t\rightarrow}{\hat{\mathbf{H}}} [r, :] \right] \right] \mathbf{1}_N}{\mathbf{1}_M^T \left[\sum_{t=0}^{T-1} \hat{\mathbf{W}}_t[:, r] \boldsymbol{\Delta}_w[r] \overset{t\rightarrow}{\hat{\mathbf{H}}} [r, :] \right] \mathbf{1}_N} \quad (12)$$

$$\boldsymbol{\Delta}_w[r] \leftarrow \boldsymbol{\Delta}_w[r] \frac{\mathbf{1}_M^T \left[\frac{\mathbf{V}}{\tilde{\mathbf{V}}} \otimes \left[\sum_{t=0}^{T-1} \hat{\mathbf{W}}_t[:, r] \boldsymbol{\Delta}_h[r] \overset{t\rightarrow}{\hat{\mathbf{H}}} [r, :] \right] \right] \mathbf{1}_N}{\mathbf{1}_M^T \left[\sum_{t=0}^{T-1} \hat{\mathbf{W}}_t[:, r] \boldsymbol{\Delta}_h[r] \overset{t\rightarrow}{\hat{\mathbf{H}}} [r, :] \right] \mathbf{1}_N}. \quad (13)$$

Discussion: Alternating between 1) learning a quantization step-size for each row and column of $\boldsymbol{H}$ and $\boldsymbol{W}_t$, and 2) learning the factors $\boldsymbol{H}$ and $\boldsymbol{W}_t$ has the added advantage of allowing the factors to absorb the distortion introduced by quantization, and the quantization functions to absorb the distortion introduced by the most recent estimates of the factors. The precedent for alternating minimization of this form is the way that CNMF's $\boldsymbol{W}_t$ factor is updated based on the most recent $\boldsymbol{H}$ factor and vice versa. Interleaving adaptive quantization with CNMF tends to allow the quality of the factors and the quantization function to improve as the factors and quantization functions converge. IMF [18] describes this type of interleaved optimization for the first time. In this paper we also consider the effects of features vectors that have temporal extent in the hope that a more appropriate representation is learned for features that evolve with time, and greater compression achieved. We adapt a quantization step-size for each of the R CNMF components, $\boldsymbol{A}_r$, as the factors and quantizers converge so that SNMF is flexible enough to account for the dynamic range of the components of $\boldsymbol{V}$.

Randomized Quantization Functions: The quantization step-size updates in (Eqn. 12,13) decrease the objective. We must also consider the effect of the non-linearity $\lfloor \cdot \rfloor$ if we are to maintain the monotonic convergence property of SNMF. The non-linearity is applied after each CNMF $\hat{\boldsymbol{W}}_t$ estimate (Eqn. 3) is computed. It is applied to the average of the activation functions computed by CNMF (Eqn. 3). We describe the new SNMF extended updates in Table. I.

By quantizing the updated factors (using Eqn. 3) without considering the cost function, we may sacrifice CNMF's monotonic convergence property. We use the following heuristic, randomized quantization, to improve the convergence properties of SNMF. The success of our experiments supports the efficacy of approach. Each element of an updated factor, e.g. $\mathbf{W}_t[:, r]$ in (Eqn. 14) is quantized using a uniform rounding function and an adaptively tuned step-size $\boldsymbol{\Delta}_W[r]$. The output entropy of a uniform quantizer is asymptotically smaller than that of any other quantizer, independent of the density function of the error criteria according to Gish and Pierce [22]. Quantization is performed element-wise. Quantization of certain elements of $\hat{\boldsymbol{W}}_t$ may cause the KLD to increase. In general the step-size is much smaller than corresponding values of the factors, or at least its important (big) entries e.g. $\boldsymbol{\Delta}_W[r] \ll \hat{\boldsymbol{W}}_t[:, r]$. The only purpose of quantization in SNMF is to produce a mapping from each of the real-valued factor elements to the integer set

TABLE I. SNMF UPDATE RULES.

Definition 2: **SNMF $\hat{\boldsymbol{W}}_t$-update:** Update the factor

$$\mathbf{W}_t = \left(\hat{\mathbf{W}}_t \operatorname{diag}(\boldsymbol{\Delta}_W)\right) \otimes \frac{\frac{\mathbf{V}}{\hat{\mathbf{V}}} \left(\operatorname{diag}(\boldsymbol{\Delta}_H)\hat{\mathbf{H}}^{t\rightarrow}\right)^T}{\mathbf{1}_{M\times N} \left(\operatorname{diag}(\boldsymbol{\Delta}_H)\hat{\mathbf{H}}^{t\rightarrow}\right)^T}. \tag{14}$$

Compute the non-linearity to generate a matrix of integers

$$\hat{\mathbf{W}}_t[m,r] \leftarrow \left\lfloor \frac{\mathbf{W}_t[m,r]}{\boldsymbol{\Delta}_w[r]} \pm \frac{1}{2} \right\rfloor_{m,r,t}. \tag{15}$$

Update $\boldsymbol{\Delta}_W[r]$ for all r using (Eqn. 13). Re-scale $\boldsymbol{\Delta}_w$ so that $\mathbf{1}_R^T \boldsymbol{\Delta}_W = 1$.

Definition 3: **SNMF $\hat{\boldsymbol{H}}$-update:** Update $\boldsymbol{H}$ for all $\boldsymbol{W}_t$,

$$\mathbf{H}_t = \left(\operatorname{diag}(\boldsymbol{\Delta}_H)\hat{\mathbf{H}}\right) \otimes \frac{\left(\hat{\mathbf{W}}_t \operatorname{diag}(\boldsymbol{\Delta}_W)\right)^T \left[\frac{\mathbf{V}}{\hat{\mathbf{V}}}^{\leftarrow t}\right]}{\left(\hat{\mathbf{W}}_t \operatorname{diag}(\boldsymbol{\Delta}_W)\right)^T \mathbf{1}_{M\times N}}, \tag{16}$$

Take an element-wise average of $\boldsymbol{H}_t$ across t and call it $\boldsymbol{H}$. Compute the non-linearity to generate a matrix of integers

$$\hat{\mathbf{H}}[r,n] \leftarrow \left\lfloor \frac{\mathbf{H}[r,n]}{\boldsymbol{\Delta}_H[r]} \pm \frac{1}{2} \right\rfloor_{r,n}. \tag{17}$$

Update $\boldsymbol{\Delta}_H[r]$ for all r using (Eqn. 12). Re-scale $\boldsymbol{\Delta}_H$ so that $\mathbf{1}_R^T \boldsymbol{\Delta}_H = 1$.

$\mathbb{I}$. We have freedom in how we apply element-wise rounding, but desire monotonic convergence. If a given array of rounding functions $\left\lfloor \frac{\boldsymbol{W}_t[m,r]}{\boldsymbol{\Delta}_W[r]} + \frac{1}{2} \right\rfloor_{m,r,t}$ causes the KLD to increase, we randomly substitute in a number of new quantization functions $\left\lfloor \frac{\boldsymbol{W}_t[m,r]}{\boldsymbol{\Delta}_w[r]} - \frac{1}{2} \right\rfloor_{m,r,t}$, until the objective function is reduced by the new quantized factor. This randomization process is controlled by Bernoulli trials. The probability of success, p, controls the number of quantization functions that are switched to round-down functions. In our experiments initially a number of Bernoulli trials, over a range of probabilities p, are required to obtain a good first initial array of quantization functions, but once this good initial factorization is found, surprisingly few random quantization arrays are required after this to ensure monotonic convergence. This is not surprising because SNMF quickly learns good quantization step-sizes and factors, which in turn allows the non-linearity settle down. In our experiments, randomized quantization is always more efficient than selecting and testing all 2^{MRT} (or 2^{RN}) possible quantization functions.

NLA: Uniform mid-tread rounding quantization is chosen because it maps a NN real-valued argument to the NN integers, $f\colon \mathbb{R}_+ \rightarrow \mathbb{Z}_+$, which compliments SNMF's NN constraints.

$$\boldsymbol{W}_t[m,r] \rightarrow \left\lfloor \frac{\boldsymbol{W}_t[m,r]}{\Delta_w[r]} + \frac{1}{2} \right\rfloor_{m,r,t}. \tag{18}$$

Similar to the SNMF and step-size updates, any NN argument (to Eqn. 18) returns a positive value. Care must be taken when applying round-down quantization for real-valued inputs which are close to zero, as a round-down may produce a negative valued integer, and thus an unfeasible SNMF.

SNMF Algorithm: The KLD objective is minimized –such that $\hat{\mathbf{W}}_t \in \mathbb{I}^{M\times R\times T}$ and $\hat{\mathbf{H}} \in \mathbb{I}^{R\times N}$– by interlacing two new update rules for the parameters $\boldsymbol{\Delta}_w$ and $\boldsymbol{\Delta}_h$, with the updates for $\mathbf{W}$ and $\mathbf{H}$, presented above. The SNMF algorithm alternates these update rules in an iteration until some convergence condition is met (a number of iterations, or an accuracy criteria ϵ has been met). We note that the randomized quantization procedure does not guarantee the selected quantization function minimizes the objective; however, the success of the approach in the numerical evaluation section supports the claim that an *acceptably good* decomposition is learned. SNMF's factorization is within 1–3dB of CNMF's solution.

TABLE II. PERFORMANCE COMPARISON NMF VS SNMF ($T = 1$)

Algorithm	NMF	SNMF ($T = 1$)
Number of unique elements of $\mathbf{W}/\hat{\mathbf{W}}$	1026	138
Number of unique elements of $\mathbf{H}/\hat{\mathbf{H}}$	874	19
File-size (gzip / not gzipped) in bytes	17704/35839	1659/35890
Compression ratio		10:1
SNR (dB)	6.58dB	6.6dB

TABLE III. PERFORMANCE COMPARISON CNMF VS SNMF ($T = 32$)

Algorithm	CNMF	SNMF ($T = 32$)
Number of unique elements of $\mathbf{W}/\hat{\mathbf{W}}$	32832	155
Number of unique elements of $\mathbf{H}/\hat{\mathbf{H}}$	436	21
File-size (gzip/ not gzipped)	300239/625609	13841/674142
Compression ratio		20:1
SNR (dB)	12.5dB	11.56dB

IV. EXPERIMENTAL EVALUATION

The goal of these experiments is to compare the performance of NMF and CNMF with SNMF, on audio segments for an in-network learning task to establish what the potential benefits might be in a networked-sensor scenario. We evaluate the accuracy of the factorizations and the compression ratio achieved quantitatively and the similarity of the SNMF solution with the NMF/CNMF solution qualitatively. We use the tables (Tab. II and III) to demonstrate that: 1) SNMF learns a solution which is as accurate as NMF but more compressible than NMF's solution; and 2) SNMF learns a solution which is as accurate as CNMF but more compressible than CNMF's solution. In short we show that we can learn a factorization for instantaneous and convolutive mixing models, which are as useful as the traditional methods in the literature, e.g. NMF and CNMF, but they are also more amenable to efficient storage. Finally, we examine the cardinality of the symbol sets of the different factorizations to explain where the source of the performance gain achieved by SNMF originates.

A. Set-up for Synthetic Data Analysis and Compression

We run all variants of NMF for 100 iterations on the noise burst data-set and the evolving spectrum data-set in this section (cf. Fig. 2 and 3). This choice is determined by an inspection of the convergence plots (obtained from a number of randomly initialized trials for all techniques). The convolutive extensions (both compressive and traditional variants) learn basis functions of temporal extent $T = 32$ (≈ 1 second). The rank of each technique is set to $R = 2$. It should be noted that increasing R improves performance. This choice is justified by inspection of the data-set and to be consistent with [21], our performance baseline. These first results are determined for synthetic wav files. In the cases presented below, the average Signal-to-Noise-Ratio (SNR) of the decompositions is poor as a rank-2 decomposition is learned; however, these synthetic data-sets have the advantage of providing an intuitive demonstration of how (and why) NMF works. These data-sets have been used to demonstrate the operation of CNMF in the paper [21]. Learning a rank-2 decomposition also makes it easier to show the factors learned by traditional NMF and its compressive extension side-by-side.

Evaluation: Tabulated performance metrics are averages obtained from Monte Carlo trials (We truncate accuracy of the metrics) The Signal-to-Noise-Ratio (SNR) of both the traditional and compressive matrix factorizations is approximately

equal in Table II and III. There is no accuracy penalty incurred by learning SNMF instead of NMF/CNMF. The SNR for NMF is 6.58dB versus 6.6dB for SNMF ($T = 1$). CNMF and SNMF ($T = 32$) learn signal representations of SNR 12.5dB and 11.57dB respectively. When the rank of the factorization increases, the SNR of both SNMF and NMF/CNMF increases. From the perspective of compression, the real valued factor $\mathbf{W}$ of NMF is composed of elements from a set which has a significantly larger cardinality than the integer-valued set which is used to represent the integer-valued factor $\hat{\mathbf{W}}$, e.g. 1026 versus 138 elements, for SNMF ($T = 1$). Similarly, the cardinality of the symbol set used to represent the real-valued factor $\mathbf{H}$ for NMF and its integer-valued counterpart $\hat{\mathbf{H}}$ for SNMF is 874 versus 19 elements. The efficiency of the SNMF ($T = 1$) representation is captured by running gzip on the SNMF ($T = 1$) and NMF data-structures presented below.

Definition 4: **SNMF-data-structure:** SNMF is rearranged as the sequence:

$$\hat{S} = M|R|N|T|\hat{\boldsymbol{W}}_1[:,1]|\hat{\boldsymbol{W}}_1[:,2]|\cdots|\hat{\boldsymbol{W}}_2[:,1]|\cdots| \hat{\boldsymbol{H}}[1,:]|\hat{\boldsymbol{H}}[2,:]|\boldsymbol{\Delta}_w|\boldsymbol{\Delta}_h. \quad (19)$$

Definition 5: **NMF/CNMF-data-structure:** A similar data-structure is used to represent NMF and CNMF.

$$S = M|R|N|T|\boldsymbol{W}_1[:,1]|\boldsymbol{W}_1[:,2]|\cdots|\boldsymbol{W}_2[:,1]|\cdots| \boldsymbol{H}[1,:]|\boldsymbol{H}[2,:]. \quad (20)$$

The size of the file required to store NMF and SNMF ($T = 1$) before gzip is run is 35839 and 35890 bytes respectively –this file size is approximately the same for both types of factorization because double-precision floating points numbers are used to store the integer matrices in our Julia [23] implementation of SNMF ($T = 1$) and SNMF ($T = 32$). However, when gzip is run on NMF and SNMF ($T = 1$), the SNMF ($T = 1$) file size reduces dramatically to 1659 bytes, compared to the file size of NMF, which is 17704 bytes. A similar bandwidth saving (a factor of 10) is obtained when the data-structure $\hat{S}$ is packetized. It is reasonable to posit that this bandwidth saving is achieved because of the reduction in the cardinality of $\mathbb{I}$, which is the union of the symbol sets used to represent $\hat{\boldsymbol{W}}_t$ and $\hat{\boldsymbol{H}}$ above.

To give an initial qualitative demonstrate that SNMF factorizations are as useful as traditional NMFs, in Fig. 5 we overlay plots of: column one of NMF's $\mathbf{W}$ matrix with column one of SNMF's ($T = 1$) $\hat{\mathbf{W}}$ matrix; column two of NMF's $\mathbf{W}$ matrix with column two of SNMF's ($T = 1$) $\hat{\mathbf{W}}$ matrix; row one of NMF's $\mathbf{H}$ matrix with row one of SNMF's ($T = 1$) $\hat{\mathbf{H}}$ matrix; and finally, row two of NMF's $\mathbf{H}$ matrix with row two of SNMF's ($T = 1$) $\hat{\mathbf{H}}$ matrix. In each plot the vectors are normalized (to sum to one) and thus the amplitude of the spectra of $\mathbf{W}$ and $\hat{\mathbf{W}}$ and the assignment strength of the activations $\mathbf{H}$ and $\hat{\mathbf{H}}$ are not shown. Fig. 5 illustrates that both NMF and SNMF ($T = 1$) yield almost identical factorizations; and the objects are successfully separated. The only discernible difference is that SNMF ($T = 1$) tends to over-estimate small values (cf. the activation vector plots in Fig. 5). This overestimation is an artifact of the mid-tread method of quantization deployed, which avoids introducing negative entries into the factors. It explains the $<$ 1db difference between the CNMF and SNMF's factorization SNR.

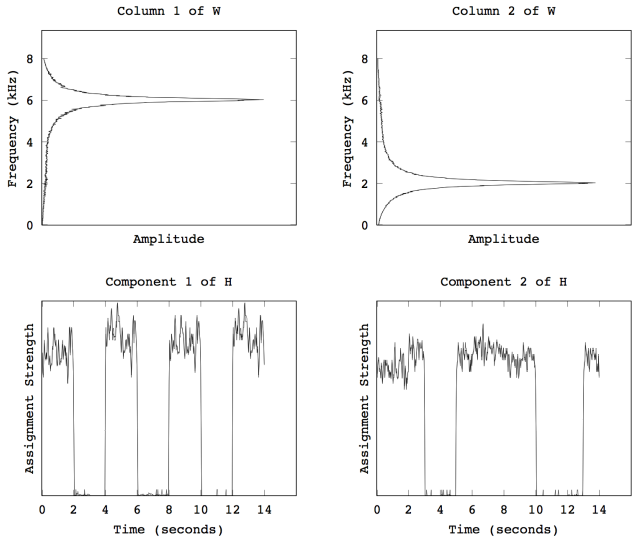

Fig. 5. Band-limited noise burst data: the two NMF basis functions are illustrated in row one. SNMF's ($T = 1$) basis functions are overlay-ed. NMF and SNMF ($T = 1$) yield very similar functions. The activations of these basis functions for SNMF ($T = 1$) and NMF are illustrated in row two. The factors are almost identical.

To underline why SNMF ($T = 1$) is more compressible than NMF, in Fig. 6 we plot histograms of the entries of SNMF's ($T = 1$) basis functions in row one, and histograms of the entries of NMF's basis functions in row two. SNMF ($T = 1$) produces an integer-valued matrix $\hat{\boldsymbol{W}}_t$ which has values clustered about one integer-valued histogram bin. The set of bins for NMF's factors is the real-line. Because the symbol set of $\mathbf{W}$ is unconstrained, typically each value of $\mathbf{W}$ is unique. Using an element-wise integer-valued mixing model affects the entropy of the learned factor, producing a more compressible vector in general. The basis functions in Fig. 5 are almost identical for SNMF and NMF. The histograms of symbol sets used to represent them using SNMF and NMF are completely different. In general most of the entries of SNMF's basis functions are close to zero, which explains the peak in the SNMF ($T = 1$) histogram (close to zero). A selected few histogram bins (corresponding to higher magnitudes) are activated by SNMF. NMF activates a bin per magnitude value of the basis functions. SNMF removes this redundancy without incurring significant error into the learned basis function (cf. Fig. 5). Note that SNMF was within 1dB of the NMF solution in $>$ 90 % of our trials. A similar quantitative analysis is summarized for CNMF and SNMF ($T = 32$) on the evolving spectrum data-set in Table III. The main points of this description are summarized as follows. The gain in compression is twice that of the gain achieved by SNMF ($T = 1$) over NMF. This increase can be attributed to the significant reduction in the number of elements required to represent $\mathbf{W}$ versus $\hat{\mathbf{W}}$, e.g. 32832 versus 155 elements. It is reasonable to assume that this improvement is due to the introduction of basis functions with temporal extent $T = 32$.

The Data Sets the Compressibility: in the next section we show that the actual improvement is due to the structure of the data. Crucially, SNMF learns the appropriate symbol-set for the data. It does not explicitly optimize compressibility, but removes needless redundancy from the CNMF factorization, but no more. The improvement in SNR for both CNMF and SNMF ($T = 32$) over SNMF ($T = 1$) and NMF above is due

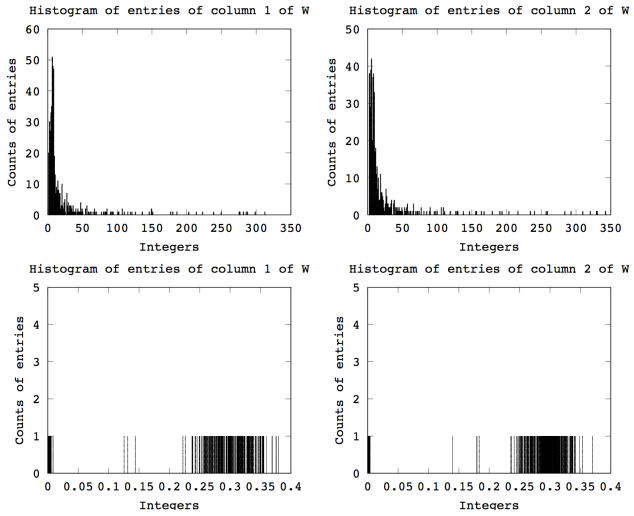

Fig. 6. Histograms of the elements of SNMF's ($T = 1$) basis functions $\hat{\mathbf{W}}$ and NMF's basis function $\mathbf{W}$. The number of bins of the histograms associated with SNMF ($T = 1$) is smaller than for NMF. NMF's basis function elements only have one element per bin.

TABLE IV. EFFECT OF T ON CNMF VERSUS SNMF ($1 \leq T \leq 7$)

	CNMF	SNMF	CNMF	SNMF
Algorithm	($T = 1$)	($T = 1$)	($T = 5$)	($T = 5$)
No. unique elements $\mathbf{W}$/$\hat{\mathbf{W}}$	10260	1133	51299	374
No. unique elements $\mathbf{H}$/$\hat{\mathbf{H}}$	1820	518	1818	54
File-size gzip bytes	112278	11442	489705	46389
File-size not gzipped bytes	232444	226119	1016321	1035544
Compression ratio		≈9.8:1		≈10.6:1
SNR (dB)	11.61	11.19	25.4	24.14
Algorithm	($T = 3$)	($T = 3$)	($T = 7$)	($T = 7$)
No. unique elements $\mathbf{W}$/$\hat{\mathbf{W}}$	30780	341	71820	421
No. unique elements $\mathbf{H}$/$\hat{\mathbf{H}}$	1820	86	1801	47
File-size gzip bytes	305111	30243	682557	63990
File-size not gzipped bytes	632236	633672	1417852	1438136
Compression ratio		≈10.1:1		≈10.67:1
SNR (dB)	15.63	15.83	29.28	26.5

to: 1) the temporal extent of the features and 2) the fact that the data in the convolutive example assumes a small number of amplitudes, relative to the dimensions of the factorization.

B. Speech: Data Analysis and Compression

Consider the performance of SNMF and CNMF on sample data from the TIMIT database [24] in order to establish whether or not the bandwidth savings achieved above are possible for real-world signals captured in a system of networked sensors. We run all techniques for 500 iterations. We perform Monte Carlo trials and average the performance statistics. The TIMIT corpus of read speech is a widely used and accepted speech corpus in the speech analysis community. We tabulate the results obtained by running a comparison of CNMF and SNMF, for one of the TIMIT speakers saying the sentence: "She had your dark suit in greasy wash water all year" in Table IV. To generate the tabulated results: 1) we use the same spectrogram parametrization as in the previous set of experiments; 2) we learn a decomposition of rank $R = 20$; 3) we vary the temporal extent parameter $T = \{1, 3, 5, 7\}$ for a fixed rank parameter ($R = 20$). This has the effect of increasing the SNR of the decompositions for both the CNMF and SNMF algorithms, to an acceptable level, as the parameter T is increased. It also allows us to determine the affect of the temporal extent parameter on the symbol set cardinality of both factors for this data-set, in isolation from the rank, R. It allows us to consider the question: is the compression gain due to the parametrization of the factorization, e.g. R and T, or due to SNMF learning the appropriate symbol-set for the data.

Symbol Set Cardinality: The number of unique elements of $\mathbf{W}$ for NMF and $\hat{\mathbf{W}}$ for SNMF increases as T increases. The growth in the symbol set cardinality of $\mathbf{W}$, for CNMF, is a linear function of T. We conclude that the characteristics of the data-set do not play a large role in determining the symbol set cardinality for CNMF. We believe that this is a disadvantage when this type of learning algorithm is required in a system of bandwidth and storage restricted networked sensors. The cardinality is approximately $M \times R \times T$ irrespective of the data-set. The growth of the cardinality of the symbol set of $\hat{\mathbf{W}}$ is not a linear function of T. Instead, we posit that it is a function of the complexity of the features learned, and that the data plays a major role here. Samples of the symbol set cardinality as a function of T are listed as follows (T, cardinality of $\hat{\boldsymbol{W}}_t$ for all t): $(1, 1133)$, $(3, 341)$, $(5, 374)$ and $(7, 421)$. We conclude that certain parametrizations achieve a better symbol set cardinality for this particularly data-set; however, this does not necessarily determine the level of compression achieved. A similar analysis follows for the CNMF factor $\mathbf{H}$ and the SNMF factor $\hat{\mathbf{H}}$. The symbol set cardinality of the factor $\mathbf{H}$ is approximately 1820 irrespective of the temporal extent T for CNMF, which is problematic. Secondly, the symbol set cardinality of $\mathbf{H}$ is approximately fixed even-though the symbol set cardinality of $\mathbf{W}$ is linearly increasing with T. This leads us to infer that it is not the complexity of the data that sets the symbol set cardinalities of both factors, but the parametrizations of CNMF. We conclude that CNMF learns factorization with significant symbol-set redundancy. In comparison, there is a decreasing trend in the symbol set cardinality, of SNMF's $\hat{\mathbf{H}}$, e.g. $(1, 518)$, $(3, 86)$, $(54, 5)$ and $(47, 7)$, which leads us to believe that it is a combination of the data and the parametrization that sets the symbol set cardinality of SNMF. SNMF is the first algorithm that attempts to learn the appropriate symbol set for convolutive mixing models.

V. BANDWIDTH, LEARNING & LATENCY

In this paper we demonstrated the performance of SNMF using a single channel acoustic sensor. Stereo recordings on everyday sensing devices typically perform mono recordings and produce identical copies of the single channel recording on both of the stereo channels. As this is generally the case we cannot appeal to additional information from a second channel.

Processing time vs Bandwidth: In practice modern computers are fast and frameworks such as MapReduce [25] improve computation speeds; however many IoT sensing devices may have orders of magnitude less processing power and may require that processing is performed on a local gateway or dedicated learning server. We consider the case where sensing is performed using a 1-channel microphone on a (standard) laptop computer. A modern laptop might have 8 cores operating at 3GHz, each core capable of 4 floating point operations per clock cycle, providing a peak performance of 96GFlops. Our parametrization of the discrete Fourier transform yields N windows of 513 frequency bins, where N is determined by the window overlap and the recording period. This parametrization may be optimized for different sensing platforms. The complexity of the SNMF update is critically dependent on R, T, the number of iterations and also on the number of random trials required for randomized

quantization. Typically, randomized quantization only affects the first 5 iterations, and pre-computed template quantization functions may be used instead. Convergence in tens of seconds is not unreasonable. We argue that floating point performance analysis is insufficient to capture the constraints imposed by real systems: the limiting factor is not computation, but rather network limits on bandwidth and latency (cf. [26]).

Congestion and Storage Efficiency: Given a set of features is learned at sensor A, congestion in the core of the network may reduce the available bandwidth for transmission of the features to sensor B. SNMF features and activation patterns may be stored at an intermediate point and transmitted when the channel is improved, which introduces considerable latency as freshly learned features are unavailable at the point-of-interest in the network. Reducing the size of the feature set by a factor of 10-20 here is useful as we are transmitting less data on a congested link. Even-though DSL connections are asymmetric, the down-link connection from a local gateway to sensor B may be congested due to the connection being shared, simultaneously, with prioritized video or VoIP sessions. Note that VoIP sessions are typically point-2-point, and that IPTV may be multicasted or broadcasted. A SNMF feature sharing scheme is likely to be a many-2-many sensor scheme, and unlike IPTV, to be transmitted on an shared network. IPTV providers do multicast content; however, a dedicated network infrastructure is used to ensure low latency and packet drop rates. The sensor networking application for SNMF we have used to motivate SNMF is opportunistic; we cannot rely on a guaranteed channel quality which implies that optimizing the feature representation to minimize bandwidth is crucial. Bear in mind that applications such as VoIP an IPTV have real-time constraints, and the packet-flows associated with them may receive higher priority routing paths through the network during route-discovery, than SNMF packets. The up-link connection from sensor A to the rest of the network may also be congested due to the sharing of resources, inevitably introducing latency. It may also be useful to cache features locally (to a sensor) to improve higher level learning processes, for example BSS and response times of services. Reducing the data-size (obtained from a local cache) by a factor of 10–20 (at the point where it is generated in the network) in a monaural BSS scenario is appealing as typically < 10 speakers are present. A 10–20-times increase in the number of features per speaker available locally, when there are few speakers present, is likely to have a great impact on separation. In future work we will evaluate whether SNMF's ability to allow us to cache 10-20-times more features locally, improves separation performance, and to what extent. We will also perform an empirical investigation of the reduction in network load.

VI. Conclusion

SNMF addresses the problem of how to learn storable and transmittable (convolutive) matrix factorizations. We show that composing the factors from a smaller symbol set produces compressible matrix factorizations. A factor of 10-20 storage space (bandwidth) is saved. We make the following experimentally supported claims: the symbol set of the factors learned by SNMF is smaller than for CNMF; the compressibility of SNMF is greater than a comparably dimension-ed CNMF; the compressibility is not affected by an increase in the temporal extent of the features; the error introduced into the factorization by using SNMF instead of CNMF is relatively insignificant; and finally, for a number of examples we demonstrated that the factors learned by CNMF and SNMF are generally indistinguishable. The storage and bandwidth savings achieved by SNMF are significant and may lead to improved classification performance in networked systems.

References

[1] P. Comon, "Independent Component Analysis: A New Concept," *Sig. Proc.*, vol. 36, pp. 287–314, 1994.

[2] A.J. Bell and T.J. Sejnowski, "An Information Maximization Approach to Blind Separation and Blind Deconvolution ," *Neur. Comp.*, vol. 7, no. 6, pp. 1129–59, 1995.

[3] A. Hyvärinen and E. Oja, "A fast fixed-point algorithm for independent component analysis," *Neur. Comp.*, vol. 9, no. 7, pp. 1483–92, 1997.

[4] D.D. Lee and H.S. Seung, "Algorithms for Nonnegative Matrix Factorization," *NIPS*, vol. 13, pp. 556–62, 2001.

[5] D.D. Lee and H.S. Seung, "Learning the parts of objects with Nonnegative Matrix Factorization," *Nature*, vol. 401, pp. 788–91, 1999.

[6] D. Donoho and V. Stodden, "When does Nonnegative Matrix Factorization give a correct decomposition into parts?," *NIPS*, vol. 16, 2004.

[7] S.A. Vavasis, "On the complexity of nonnegative matrix factorization," *SIAM J. on Optim.*, vol. 20, no. 3, pp. 1364–77, 2009.

[8] J. J.-Y. Wang, X. Wang, and X. Gao, "Non-negative matrix factorization by maximizing correntropy for cancer clustering," *BMC Bioinformatics*, vol. 107, no. 14, 2013.

[9] H. Liu, Z. Wu, D. Cai, and T.S. Huang, "Constrained non-negative matrix factorization for image representation," *IEEE Trans. Pat. Ana. Mach. Intel.*, vol. 34, no. 7, pp. 1299–1311, 2012.

[10] W.B. Pennebacker and J.L. Mitchell, "JPEG still image data compression standard," *Springer, New York*, 1993.

[11] D.S. Taubman and M. Marcellin, "JPEG2000: image compression fundamentals, standards and practice," *Kluwer Academic Publishers Norwell, MA*, 2001.

[12] J. Ziv and A. Lempel, "A universal algorithm for sequential data compression," *Inf. Th., IEEE Trans.*, vol. 23, no. 3, pp. 337–43, 1977.

[13] R.K. Ganti, Fan Ye, and Hui Lei, "Mobile crowdsensing: current state and future challenges," *Comm. Mag., IEEE*, vol. 49, no. 11, pp. 32–9, 2011.

[14] R.K. Potter, G.A. Kopp, and H.C. Green, "Visible speech," 1947.

[15] T. Virtanen, "Sound source separation using sparse coding with temporal continuity objective," *Proc. Int. Comp. Music Conf.*, 2003.

[16] P. Smaragdis, "Nonnegative matrix factor deconvolution; extraction of multiple sound sources from monophonic inputs," *International Conference Analysis and Blind Signal Separation*, pp. 494–9, 2004.

[17] Ruairí de Fréin, "Quantized Nonnegative Matrix Factorization," in *IEEE Digital Sig. Proc. 19th Int. Conf.*, 2014, pp. 377–82.

[18] Ruairí de Fréin, "Learning and Storing the Parts of Objects: IMF," pp. 1–6, 2014, Sep. 2014.

[19] Ruairí de Fréin, K. Drakakis, and S. Rickard, "Portfolio diversification using subspace factorisations," in *IEEE Inf. Sc. Sys. 42nd Conf.*, 2008, pp. 1075–80.

[20] R.H. Lambert, "Multichannel Blind Deconvolution: FIR Matrix Algebra and Separation of Multipath Mixtures," *PhD thesis*, 1996.

[21] P. D. O'Grady and B. A. Pearlmutter, "Discovering speech phones using convolutive non-negative matrix factorisation with a sparseness constraint," pp. 88–101, 2008, 2008.

[22] H. Gish and J. Pierce, "Asymptotically efficient quantizing," *Inf. Th., IEEE Trans.*, vol. 14, no. 5, pp. 676–83, Sep. 1968.

[23] Jeff Bezanson, Stefan Karpinski, Viral B. Shah, and Alan Edelman, "Julia: A fast dynamic language for technical computing," *CoRR*, vol. abs/1209.5145, 2012.

[24] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, and N. L. Dahlgren, "DARPA TIMIT acoustic phonetic continuous speech corpus CDROM," 1993.

[25] B. Xu, Ruairí de Fréin, E. Robson, and M. Ó Foghlú, "Distributed Formal Concept Analysis Algorithms Based on an Iterative MapReduce Framework," in *Formal Concept Analysis*, LNCS 7278, pp. 292–308. Springer Berlin Heidelberg, 2012.

[26] D. Hsu, N. Karampatziakis, J. Langford, and A. J. Smola, "Parallel online learning," *CoRR*, vol. abs/1103.4204, 2011.