Online Non-Negative Convolutive Pattern Learning for Speech Signals

Dong Wang, *Member, IEEE,* Ravichander Vipperla, *Member, IEEE,* Nicholas Evans, *Member, IEEE,* and Thomas Fang Zheng, *Senior Member, IEEE*

***Abstract*—The unsupervised learning of spectro-temporal patterns within speech signals is of interest in a broad range of applications. Where patterns are non-negative and convolutive in nature, relevant learning algorithms include convolutive non-negative matrix factorization (CNMF) and its sparse alternative, convolutive non-negative sparse coding (CNSC). Both algorithms, however, place unrealistic demands on computing power and memory which prohibit their application in large scale tasks. This paper proposes a new online implementation of CNMF and CNSC which processes input data piece-by-piece and updates learned patterns gradually with accumulated statistics. The proposed approach facilitates pattern learning with huge volumes of training data that are beyond the capability of existing alternatives. We show that, with unlimited data and computing resources, the new online learning algorithm almost surely converges to a local minimum of the objective cost function. In more realistic situations, where the amount of data is large and computing power is limited, online learning tends to obtain lower empirical cost than conventional batch learning.**

***Index Terms*—Non-negative matrix factorization, convolutive NMF, online pattern learning, sparse coding, speech processing, speech recognition**

I. Introduction

Many signals exhibit clear spectro-temporal patterns; the discovery and learning of such patterns with automatic approaches is often needed for signal interpretation and for the design of suitable algorithms in practical applications. In speech signals, for instance, patterns of interest might be related to the speaker identity or the phonetic content. Whilst some of these patterns might be readily defined and learned with supervised approaches, e.g. neural networks, more complex patterns are difficult to pre-define and annotate, particularly when they involve large datasets, hence the need for unsupervised approaches.

Various unsupervised learning techniques have been developed for automatic pattern discovery. The general idea behind such learning approaches involves the search for a number of patterns which can be used to reconstruct a set of training signals according to a certain cost function, e.g. minimum reconstruction loss, and an appropriate set of constraints. This can be written formally as:

$$\tilde{W} = \arg\min_{W}\{\min_{H} \ell(X, \tilde{X}(W,H))\} \quad s.t. \quad \{g_i(W,H)\} \quad (1)$$

where X represents a set of training signals and $\tilde{X}$ their reconstructed approximations. $\ell(\cdot,\cdot)$ represents the objective function and $\{g_i(W,H)\}$ represents the set of constraints. The reconstruction usually takes a linear form:

$$\tilde{X}(W,H) = W \times H$$

where H represents the projection of $\tilde{X}$ onto a set of patterns W. Pattern learning is thus closely related to matrix factorization, a field that has been studied extensively in mathematics and statistics. In signal processing and pattern learning, W is referred to as a *dictionary* whereas in statistics, W is referred to as a *basis*. The coefficient matrix H is known as a *factor matrix* or a *code matrix* in some literature. In this paper we refer to W and H as 'patterns' and 'coefficients' respectively.

Different cost functions and constraints lead to different learning techniques. An l-2 reconstruction loss or Kullback-Leibler divergence cost function and a non-negative constraint applied to both patterns and coefficients leads to non-negative matrix factorization (NMF) [1]–[6]. In contrast to other pattern learning approaches NMF is capable of learning partial patterns and has thus proved to be popular in applications such as data analysis, speech processing, image processing and pattern recognition [7]–[11].

A number of extensions have been introduced to improve the basic NMF approach, e.g. [12]–[23]. Convolutive NMF (CNMF) [24], [25] and sparse NMF [26]–[28] are among the most significant. Patterns learned with convolutive NMF span a number of consecutive frames and thus capture spectro-temporal features. With sparse NMF, sparsity constraints imposed on both patterns and coefficients generally lead to improved representation and noise robustness. The two extensions can be combined, resulting in a more powerful learning approach referred to as convolutive non-negative sparse coding (CNSC) [29]–[32].

While promising results have been demonstrated in some tasks, such as speech enhancement [33] and source separation [34], NMF and its variants such as CNMF and CNSC place high demands on both computing resources and memory when the training database is large. The original form of the multiplicative update procedure [4] requires all the signals to be read into memory and processed in each iteration; this is prohibitive

This work was conducted when Dong Wang was at EURECOM as a post-doctoral research fellow and was completed when he was a visiting researcher at Tsinghua University and a senior research engineer at Nuance. It was partially supported by the French Ministry of Industry (Innovative Web call) under contract 09.2.93.0966, collaborative Annotation for Video Accessibility (ACAV) and by the Adaptable Ambient Living Assistant (ALIAS) project funded through the joint national Ambient Assisted Living (AAL) programme.

Dong Wang and Thomas Fang Zheng are with Tsinghua University, Ravichander Vipperla and Nicholas Evans are with EURECOM.

in large scale applications such as large vocabulary speech recognition which usually involves thousands of gigabytes of training data. This problem is more pronounced for both CNMF, where patterns cover a greater number of signal frames and so are usually large, and CNSC, which not only involves larger patterns but also a greater number of patterns. In some cases their number might even be larger than the signal dimension (sometimes referred to as 'over-complete patterns'). Most related publications in speech processing accordingly focus on small databases, e.g. TIDIGITS or TIMIT and learning is often based on even smaller subsets or random samples [35], [36]. Such ad-hoc learning schemes are clearly unacceptable for complex tasks.

To address this problem, we propose in this article a novel on-line learning approach for CNMF and CNSC, which processes input signals piece-by-piece and updates learned patterns gradually using accumulated statistics. With this approach, only a limited segment of the input signal is processed at a time. This approach resolves the problem of memory usage and computing cost from which conventional NMF suffers, thereby facilitating learning from large databases. As is the case for batch learning, we prove that the proposed online approach *almost surely* converges to a local minimum of the objective cost function when the amount of data and computational resources are unlimited. We furthermore demonstrate that the new online approach tends to obtain lower empirical cost than batch learning in practical applications.

In the following section, we first formulate the learning task and present the online CNSC algorithm (CNMF can be regarded as a special case of CNSC with zero sparsity). Section III presents a complexity analysis and convergence study. Experimental results are reported in Section IV. Our conclusions are presented in Section V with ideas for further work.

II. Online convolutive pattern learning

A. Problem formulation

CNSC can be formulated according to different cost functions [31], [37]. We adopt the formulation in [31] which defines the learning problem as the minimization of the following cost function:

$$f(W,H) = \ell(X, \tilde{X}(W,H)) \quad s.t. \quad W_{i,j,k}, H_{i,j} \geq 0 \qquad (2)$$

where $X \in \mathbb{R}_{0,+}^{M \times N}$ represents the original signal of length N in M-dimensional space[1] and $\tilde{X}$ is its reconstructed approximation. It is obtained from a pattern matrix $W \in \mathbb{R}_{0,+}^{M \times R \times P}$ with R patterns of convolution range P and a coefficient matrix $H \in \mathbb{R}_{0,+}^{R \times N}$ according to:

$$\tilde{X}(W,H) = \sum_{p=0}^{P-1} W(p) \overset{p\rightarrow}{H} \quad s.t. \quad H_{i,j} \geq 0 \qquad (3)$$

[1]The term "signal" here denotes any sequential data which may be non-negative in its natural form. In general, however, they are alternative or transformed non-negative representations of the original signal, such as power spectra.

where $\overset{p\rightarrow}{H}$ shifts H by p columns to the right and where $W(p) \in \mathbb{R}_{0,+}^{M \times R}$ is the pattern matrix corresponding to $\overset{p\rightarrow}{H}$. Finally, the cost function is the sparse-regularized least square distance given by:

$$\ell(X, \tilde{X}) \quad = \quad ||X - \tilde{X}||_2^2 + \lambda ||H||_l \qquad (4)$$

where $||\ .\ ||_l$ denotes the element-wise l-norm, which is equivalent to the sum of squares of the matrix elements when $l = 2$ or the sum of their absolute values when $l = 1$. The factor λ is introduced to control the sparsity of H. To avoid a nullified H, patterns in the pattern matrix W are forced to be unity, i.e., $||W(p)||_2 = 1$ for any p.

Note that in the optimization problem (2), both patterns W and coefficients H are free variables and need to be optimized simultaneously, even if patterns W are the primary target. This co-optimization problem is not convex and it is difficult to find a globally optimal solution. A multiplicative update approach is presented in [31] to search for a local minimum solution by extending the procedure presented in the seminal NMF paper [4]. This is formulated as follows:

$$H \leftarrow H \odot \frac{[W(p)]^T \overset{\leftarrow p}{X}}{[W(p)]^T \overset{\leftarrow p}{\tilde{X}} + \lambda \Xi} \qquad (5)$$

$$W(p) \leftarrow W(p) \odot \frac{X \overset{p\rightarrow T}{H}}{\tilde{X} \overset{p\rightarrow T}{H}} \qquad (6)$$

where $\odot$ is the element-wise product and where the division is also element-wise. Ξ is a matrix whose elements are all equal to 1. Note that the update of H is different for different p and so, in practice, H is averaged over all p to obtain the updated coefficients.

The above equations show that most of the computation is involved in calculating the reconstruction $\tilde{X}$, which has a complexity of $O(M \times N \times R \times P)$. This is highly demanding for large pattern sets (large R) and large databases (large N). More importantly, since all signals must be loaded into memory and processed together, both the memory and computational requirements become prohibitive when the training corpus is large.

B. Online CNSC

In order to extend the application of CNSC to large scale tasks which involve large volumes of training data and complex patterns, we present an online learning approach which reads in and processes only a part of the training data at a time and updates patterns gradually until the whole training corpus is processed. The online approach has been presented previously to train probabilistic models in machine learning (e.g., [38], [39]), however it has seldom been studied in a multiplicative update setting such as in CNSC. A recent contribution presented by Mairal *et al.* is online dictionary learning (ODL) [40]. ODL reads in and decomposes signals frame-by-frame and updates patterns as each frame is processed. The authors show that such 'partial learning' almost surely converges to a stationary point of the objective function, given

unlimited training data and a few reasonable assumptions. Similar research can be found in [41], [42].

In this paper we present an alternative online learning approach which is based on the simple NMF-style multiplicative update rule, and employ this approach to learn temporal patterns based on CNSC. Note that, while ODL supports NMF or sparse NMF by enforcing the non-negativity constraint on both patterns and coefficients, our work is the first to couple online and convolutive learning.

We start by designing a partial learning formulation which retains the temporal information within training data. We define a signal *piece* as a number of neighboring frames within which the signal is correlated, while different pieces are assumed to be independent. Temporal patterns can be learned by processing pieces sequentially and separately. Through a simple re-arrangement, the pattern update rule (6) can be re-written as follows:

$$W(p) \leftarrow W(p) \odot \frac{\sum_u \dot{B}(p,u)}{\sum_q W(q) \sum_u \dot{A}(q,p,u)} \tag{7}$$

where u is the piece index and

$$\dot{A}(q,p,u) = \overset{q\rightarrow}{H_u} \overset{p\rightarrow T}{H_u}$$

$$\dot{B}(p,u) = X_u \overset{p\rightarrow T}{H_u}$$

are the statistics contributed by piece u. The contribution of the first u pieces can then be 'memorized' in two auxiliary variables defined as follows:

$$A(q,p;u) = \sum_{t=1}^{u} \dot{A}(q,p,t)$$

and

$$B(p;u) = \sum_{t=1}^{u} \dot{B}(p,t).$$

The most significant difference between rules (6) and (7) is that the training data are broken into small pieces and processed sequentially. The application of rule (7) is thus more suitable in applications involving live, streamed data and the adaptive learning of new patterns in time-variant data. Second, through rule (7) the contribution of processed signals is stored in two auxiliary variables. Their size is independent of training data quantities which thus reduces memory and computational demands and hence enables the learning of complex patterns from large corpora. Finally, piece-wise learning allows the updating of patterns with each new signal piece. This leads to 'early learning' which significantly increases convergence speed as presented in Sections III and IV.

This leads to online CNSC which is presented in Algorithm 1. The flag variable $activeW$ defines two different learning schemes: if $activeW = True$, both patterns and coefficients are updated K times iteratively when processing each piece; if $activeW = False$, only coefficients are iteratively updated. The former approach is referred to as *active learning* whereas the second approach is referred to as *inertial learning*. Note that, in both cases, patterns are nonetheless learned actively with the first piece to ensure reasonable initialization of the pattern matrix. In general, active learning converges with fewer iterations than inertial learning but places a greater demand on computing resources. We address this point further in Sections III and IV. Algorithm 2 illustrates the pattern update process (7). Matlab code for these algorithms is available online[2].

We note that the online CNSC algorithm emphasizes pattern learning and thus coefficients obtained for each signal piece might be sub-optimal as a result of early learning. This is in contrast to batch learning where patterns and coefficients are optimized simultaneously with respect to the objective cost function. With the learned patterns, however, the coefficients can be optimized easily either by iteratively applying (5) or by more efficient techniques such as quadratic optimization; both are amenable to parallel computation. Finally we note

[2]http://audio.eurecom.fr/software

Algorithm 1 Online CNSC learning

U: number of pieces
K: iteration
$A(i,j) \in \mathbb{R}^{R\times R}, \quad 0 < i,j < P$
$B(i) \in \mathbb{R}^{M\times R}, \quad 0 < i < P$
$A(i,j) \leftarrow 0, \quad \forall i,j$
$B(i) \leftarrow 0, \quad \forall i$
for u := 0 to U-1 **do**
  randomize(H)
  for k := 0 to K-1 **do**
    if (activeW=$true$) or $(k = 0)$ **then**
      $W = updateW(A, B, X_u, W, H)$
    end if
    $H = updateH(X, W, H)(Eq.5)$
  end for
  $[\dot{A}, \dot{B}, W] = updateW(A, B, X_u, W, H)$
  $A(i,j) \leftarrow A(i,j) + \dot{A}(i,j)$
  $B(i) \leftarrow B(i) + \dot{B}(i)$
end for

Algorithm 2 CNSC pattern update

Require: A, B, X, W, H
$\dot{A} \in \mathbb{R}^{R\times R}, \quad 0 < i,j < P$
$\dot{B} \in \mathbb{R}^{M\times R}, \quad 0 < i < P$
$\dot{A}(i,j) = \overset{i\rightarrow j\rightarrow}{H} \overset{T}{H} \qquad \forall i,j$
$\dot{B}(i) = X \overset{i\rightarrow T}{H} \qquad \forall i$
$A = A + \dot{A}$
$B = B + \dot{B}$
for p :=0 to P-1 **do**
  $F \leftarrow 0$
  for q :=0 to P-1 **do**
    $F = F + W_q A(q,p)$
  end for
  $\dot{W}_p = W_p \odot \frac{B(p)}{F}$
end for
$W_p = \frac{\dot{W}_p}{||\dot{W}_p||_2^2} \quad \forall p \;\; s.t. \;\; W_p \in \mathbb{R}_{0,+}^{M\times R}$
return $[A, B, W]$

that the choice of segmenting signals into pieces is a trade-off between intra-piece correlation and inter-piece independence. For speech signals, a segmentation according to sentence boundaries avoids the splitting of voiced patterns and is thus a natural choice.

III. Complexity and convergence analysis

In this section we analyze the computational complexity and convergence of the online CNSC algorithm. We show that online learning requires comparable (for active learning) or less (for inertial learning) computation than batch learning but still converges to a local minimum of the objective cost function with probability 1, or *almost surely*. In practice, with the same computational load, online learning tends to obtain lower empirical cost than batch learning.

A. Computational complexity

The computing demand for both conventional batch learning and online learning consists of updating the coefficient matrix H and the pattern matrix W. We start the analysis with the computation required for one iteration.

Firstly, for the coefficient update rule (5) which is shared by both batch and online learning, one update requires $(2M + PM + 2)RN$ multiplications and RN divisions. Considering updates for various shifts p, the computational complexity is in the order of $O(M \times N \times R \times P)$ for one iteration and is identical for both batch and online learning. In addition, rule (6) shows that one pattern update iteration for batch learning requires $(3N+1)MRP$ multiplications and MRP divisions, while one online pattern update iteration (7) requires

$$(N + M)R^2P^2 + (N + 1)MRP$$

multiplications and MRP divisions. If the signals are segmented into U pieces and the length of the u^{th} piece is N_u, the computing demand to process the entire signals amounts to

$$\sum_{u=0}^{U-1}(N_u + M)R^2P^2 + (N_u + 1)MRP$$

or

$$NR^2P^2 + NMRP + UMR^2P^2 + UMRP$$

multiplications and $UMRP$ divisions. In the case where N is dominant, batch learning and online learning require approximately $3MRPN$ and $(RP+M)RPN$ multiplications respectively.

A simple calculation shows that online learning is more efficient if $2M > RP$. This implies that, with high dimensional features and when learning a small number of patterns with a small convolution range, online learning update rule (7) is more efficient than batch learning update rule (6). For example, in speech processing we usually choose $M = 128$ for power spectra, and the convolution range is often chosen to be small, e.g., $P = 4$. If we learn a modest number of patterns, i.e., $R < 64$, then the online algorithm is more efficient than the batch algorithm. For sparse coding where the pattern matrix is over-complete e.g., $R > M$, then online learning is slower than batch learning. The compensation, however, is that a greater volume of training data can be handled.

We consider computing complexity for multiple iterations. To simplify the comparison, we assume that both online and batch learning use update rule (7) and therefore have the same complexity for a single update. For batch learning, K iterations require K times the computation required for one iteration. For online learning, without considering the special treatment for the first piece, active learning invokes K iterations for coefficient update and $K+1$ iterations for pattern update, while inertial learning invokes K iterations for coefficient update and 1 iteration for pattern update. Thus active learning always requires more computation than batch learning, while inertial learning is always more efficient than batch learning. As we discuss in the next section, both active and inertial learning converge almost surely and the greater computational demand for active learning is compensated for by faster convergence.

B. Convergence analysis

We here study the convergence behavior of the online CNSC algorithm. As in ODL, we define the learning task as an optimization problem which aims to minimize an *objective cost function* $f_u(W)$ with respect to the pattern matrix W, where $f_u(W)$ is defined as follows:

$$f_u(W) \equiv \frac{1}{u}\sum_{t=1}^{u}\ell_{X_t}(W)$$

where

$$\ell_X(W) = \min_H \frac{1}{|X|}\ell(X, \tilde{X}(W, H))$$

is the cost of signal X and $|X|$ denotes the number of frames of X. The limit of the objective cost function is defined as the *expected cost function*, denoted by $f(W)$ given as follows:

$$\begin{aligned} f(W) &\equiv \mathbb{E}_X[\ell_X(W)] \\ &= \lim_{u\to\infty} f_u(W) \end{aligned}$$

where $\mathbb{E}_X$ represents expectation over X.

Note the definitions of $f_u(W)$ and $f(W)$ are independent of the specific learning process. In order to study the convergence of the proposed CNSC online learning algorithm, we define W_t as the pattern matrix learned at the t^{th} step, and H_t as the coefficient matrix of the t^{th} signal obtained in learning. An *empirical cost function* is defined as follows to evaluate quality of the learning process:

$$\dot{f}_u(W) = \frac{1}{u}\sum_{t=1}^{u}\ell_{X_t}(W, H_t) \tag{8}$$

where

$$\ell_X(W, H) = \frac{1}{|X|}\ell(X, \tilde{X}(W, H))$$

is the empirical cost of signal X. Note that the coefficients H_t are 'imperfect' in general, which means the multiplicative update does not converge for each piece of signal, usually due to limitations on computing resources. We therefore refer to $\dot{f}_u(W)$ as the *imperfect empirical cost function*, and $\ell_{X_t}(W, H_t)$ as the *imperfect cost* of X_t.

If the computing resources are unlimited and the learning is 'perfect', the coefficient matrix H_t is optimized and explicitly denoted by:

$$\hat{H}_t = \arg\min_H \ell(X_t, \tilde{X}(W_t, H)$$

or

$$\hat{H}_t = \arg\min_H \ell(X_t, \tilde{X}(W_{t-1}, H)$$

with active and inertial learning respectively. The corresponding empirical cost function is referred to as the *perfect empirical cost function* and is explicitly denoted by $\hat{f}$:

$$\hat{f}_u(W) = \frac{1}{u}\sum_{t=1}^{u} \ell_{X_t}(W, \hat{H}_t) \tag{9}$$

where $\ell_{X_t}(W, \hat{H}_t)$ is referred to as the *perfect cost* of X_t.

With a few straightforward assumptions it can be proved that the empirical cost $\hat{f}_u(W_u)$ of perfect learning converges to a local minimum of the expected cost function $f(W)$ when u approaches infinity. With imperfect learning, the empirical cost $\hat{f}(W_u)$ converges to a stationary point of a cost function in the form $g(W) = f(W) + \epsilon(W)$ where $\epsilon(W)$ is an error function related to the learning 'imperfection'. The proof can be found in Appendix A.

C. Batch learning and online learning

The convergence analysis in the previous section shows that both batch and online learning converge to a stationary point of the expected cost function $f(W)$ with unlimited data and unlimited computing resources. This situation is only valid in theory. For small scale tasks where data are limited, but computing resources are unlimited, batch learning converges to a stationary point of the cost function $f_u(W)$ while online learning fails to converge, resulting in suboptimal patterns. For large scale tasks, the more common situation is where training data are abundant but computing resources are limited. In this situation, due to its early learning property, online learning tends to obtain lower empirical cost than batch learning, as demonstrated by the experimental results presented in the next section.

IV. Experiments

We present two experiments which demonstrate the characteristics and benefits of the proposed online CNSC approach. The first experiment is a small-scale speech separation task which aims to compare the behavior of the two online learning approaches and batch learning; the second experiment involves a noise cancelation task for large-scale speech recognition as defined by the CHiME challenge[3]. It aims to demonstrate the power of online learning in real applications.

[3]http://spandh.dcs.shef.ac.uk/projects/chime/challenge.html

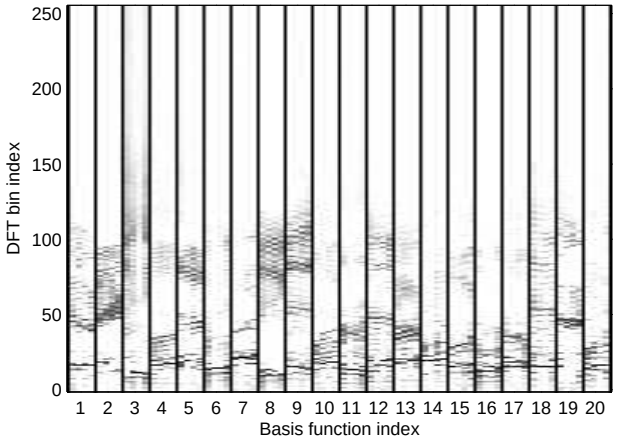


Fig. 1: An example of patterns learned with active learning.

A. Speech separation

In this experiment, we study the behavior of the proposed online pattern learning algorithm using a toy experiment proposed by Smaragdis[4] in a study of CNMF [24]. The task is to learn two sets of patterns from individual speech signals of a male and female speaker respectively, and then to use the corresponding patterns to separate the two voices from a segment of mixed speech. Smaragdis showed that speaker specific patterns can be learned using CNMF and then employed to separate the speech signal according to the constituent speakers. It has also been shown in [34] that sparse coding can deliver improved performance in a similar signal separation task.

The two individual speech segments used for pattern learning are in the order of 30 seconds in length, are sampled at 16kHz and are mixed together by simple addition with appropriate zero-padding being applied to the shorter speech recording. Signals are windowed into frames of 32ms with a frame shift of 16ms, thereby resulting in a frame rate of 62.5 frames per second. The discrete Fourier transform is applied to each frame and the magnitude spectrum is used as a non-negative representation which is suitable for processing with NMF and CNSC. All experiments reported here are based on fixed parameters of $R = 20$, $P = 4$ and $\lambda = 0.01$ which are all chosen heuristically. Finally, all experiments were conducted on a desktop machine with two dual-core 2.60GHz CPUs and memory of 4GB.

Before presenting the separation task, we investigate several factors which impact on the convergence behavior of online learning. The recording of male speech is used to conduct pattern learning and signal reconstruction; learning quality is measured using the value of the cost function (4). Fig. 1 illustrates example patterns learned for the male speaker.

1) Convergence and computing resources:: The first factor which impacts on convergence is the number of multiplicative update iterations, which is directly related to computing resources. The male speech signals are divided into 10 pieces

[4]http://www.cs.illinois.edu/~paris/demos/

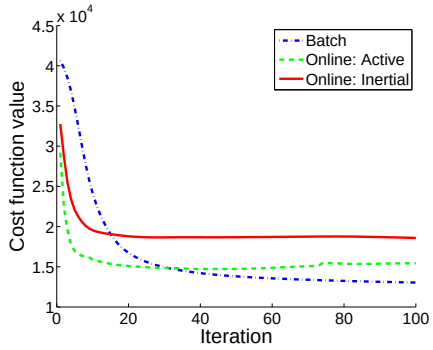


Fig. 2: Value of the cost function for the first 100 iterations with online and batch learning.

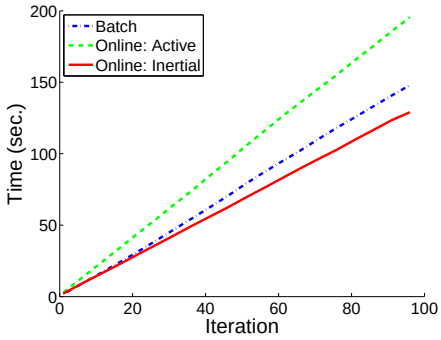


Fig. 3: Average run-time for the first 100 iterations with online and batch learning.

to conduct online learning. Batch learning is implemented as active online learning with the number of pieces set to 1.

Fig. 2 presents the cost values obtained with various learning approaches for the first 100 iterations. We observe that active online learning converges in the first 5 iterations while inertial online learning requires 10 iterations to converge. An interesting observation is that the cost obtained with the two online learning approaches increases with a higher number of iterations, indicating some over-fitting to the first few signal pieces. Upon comparison of the two online learning approaches, we see that active learning leads to lower cost. This is expected considering the more aggressive early learning.

Batch learning converges much more slowly than online learning: it requires 15 and 30 iterations to reach the same cost obtained with inertial and active learning respectively, and requires more than 80 iterations to converge itself. In spite of slow convergence, batch learning delivers lower cost if the number of iterations is sufficiently large, thereby demonstrating its advantage in small scale tasks. These observations are consistent with the convergence analysis presented in Section III-B.

Fig. 3 shows the corresponding average run-time for the first 100 iterations of the three learning approaches. We see that inertial online learning is the most efficient while active learning is the most expensive. This observation is consistent with the computational complexity analysis presented in Section III-A.

2) Convergence and piece length:: The second factor which impacts on the convergence of online learning is the manner

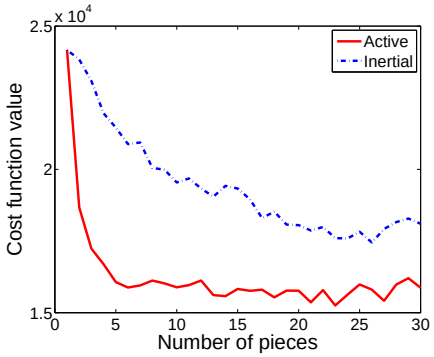


Fig. 4: Value of the cost function for $U = 1$ to 10 pieces and after 10 iterations for active and inertial online learning.

in which signals are split into pieces. Generally speaking, the splitting of signals into more pieces leads to more frequent pattern updates and hence more aggressive early learning; we therefore expect lower cost with smaller pieces for online learning, subject to the assumption of independence among pieces being held.

To test this conjecture, the signals of the male speaker are split into U pieces, and then the patterns are learned by setting the number of multiplicative iterations to 10. The cost of the two online learning approaches is shown in Fig. 4 for $U = 1$ to 10. Note that active learning with $U = 1$ is equivalent to batch learning. As expected, we observe that the two online learning approaches obtain substantially lower cost than batch learning ($U = 1$) and that smaller pieces lead to lower cost. Note that, with increasing number of pieces, the cost function exhibits some variation. This can be attributed to the boundary effect stemming from signal segmentation.

The corresponding average run-time is shown in Fig. 5. We first observe that active learning requires more computational resources as the training data are split into more pieces, due to the increased number of pattern updates. Inertial learning exhibits different behavior: the computational demand first decreases when the data are split into a small number of pieces; with increasing number of pieces, the computational requirements increase steadily by a small factor. This is because the initial pattern update for the first piece (Algorithm 1) is less costly when the data are split into smaller pieces. As the number of pieces increases, the computational saving with the initial pattern update becomes marginal while the cost associated with pattern update for each piece increases.

3) Convergence and data volume:: In the third experiment, we study the impact of the amount of training data, for which the male speech signals are duplicated and concatenated to simulate increasing data volume. This simulation certainly cannot fully represent practical scenarios with large amounts of data, however it does approximate a stationary compact distribution.

We first study the case of perfect learning. From Fig. 2 we see that, with 100 iterations, both online and batch learning can be regarded as converged. We therefore set up an experiment where the number of iterations is fixed to 100 and the amount of training data is increased by data duplication. Results are

Fig. 5: Average run-time for between $U = 1$ to 10 pieces and after 10 iterations for active and inertial online learning.

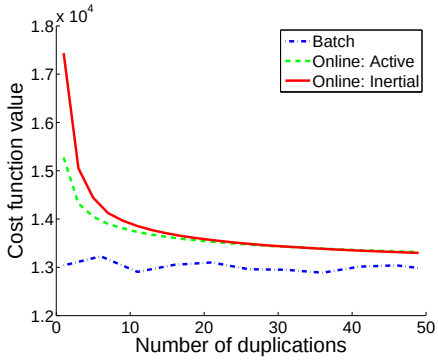


Fig. 6: Value of the cost function with the multiplicative update fixed to 100 iterations and the speech data duplicated up to 50 times.

shown in Fig. 6 where the x-axis denotes the number of duplications and the y-axis is the cost. We see that, with limited data, batch learning obtains significantly lower cost than the two online learning algorithms; with more and more data, however, the cost obtained with online learning approaches that obtained with batch learning. Although not reported in the figure, when the duplication number increases to over 500, the three learning approaches obtain very similar cost, thus supporting the convergence theory presented in Section III-B. Note that for batch learning, the cost profile exhibits some fluctuation which can be attributed to the boundary effect between duplications.

In another experiment we study the case of imperfect learning. To simulate this situation, the number of multiplicative updates is set to 10 and the training data are again increasingly duplicated. Results are shown in Fig. 7. We observe that online learning obtains significantly lower cost than batch learning and that the two online learning approaches converge to the same cost. When compared to the results in the case of perfect learning (Fig. 6), we find that the costs obtained with perfect and imperfect learning are comparable when empirical convergence is reached. This suggests that a few iterations might be sufficient for online learning on large scale tasks, as we will see in the noise cancelation experiment presented in Section IV-B.

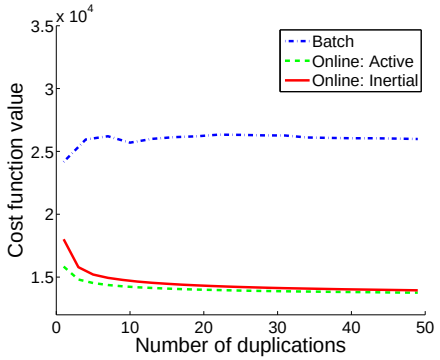


Fig. 7: Value of the cost function with the multiplicative update fixed to 10 iterations and the speech data duplicated 50 times.

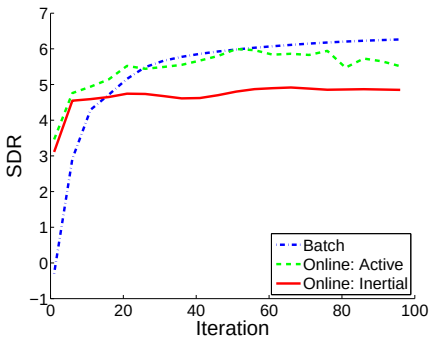


Fig. 8: SDR of speech separation.

4) Speech separation:: In the speech separation task, both the male and female speech utterances are split into 30 pieces, which has been shown to be effective for online learning. The male and female patterns are learned using corresponding training speech by applying either online or batch learning approach. The spectrum of the mixed speech signal is then projected independently onto the two sets of patterns and the reconstruction cost of the resulting magnitude spectrum is computed for the individual male and female speech signals respectively[5].

The separation performance is evaluated in terms of the signal-to-distortion ratio (SDR), defined as follows:

$$SDR = \frac{1}{2} \sum_{i \in \{male, female\}} 10 log_{10} \frac{||s^i_{dist}||^2}{||e^i_{interf} + e^i_{noise} + e_{artif}||^2}$$

where i denotes channels (female or male), s_{dist} is the original speech signal, e_{interf}, e_{noise} and e_{artif} denote interference among channels, noise and artifacts introduced by separation [43]. The BSS Eval tool was used to conduct the evaluation[6]. Results are shown in Fig. 8 where the x-axis represents the number of multiplicative iterations and the y-axis represents SDR. It can be observed that, with a small number

[5]The original and reconstructed speech waveforms in this experiment are available at http://audio.eurecom.fr/software/ol

[6]http://bass-db.gforge.inria.fr/bss_eval.

of iterations, the two online learning algorithms result in better separation than batch learning. With an increasing number of iterations, inertial learning converges to an approximate SDR of 4.5 while batch and active learning give an approximate SDR of 6.0, though batch learning ultimately outperforms active learning. Note that this does not mean online learning is less important: with a large amount of data, batch learning simply runs out of memory whereas online learning requires far less resources. In this case, online learning is thus the only viable choice.

B. Denoising for speech recognition

In the second experiment we apply the online learning approach to suppress multi-source noise from speech signals for improved automatic speech recognition (ASR). The basic idea is to learn the patterns of clean speech and background noise. Clean speech representations are then obtained by distributing the signal energy among the speech and noise patterns and by discarding that attributed to noise. The procedure is described in [24].

1) Experimental setup: Our experiments are set up within the framework of the CHiME challenge [44], where the task is to recognize the speech utterances in a home environment with various kinds of background noise under six different signal-to-noise ratio (SNR) conditions. The background noise comprises voices, television sounds, music, noise from home appliances and a host of other ambient noises typically observed in a home environment. All audio signals were recorded with a binaural microphone array. The location of the target speaker is specified to be 2 meters directly in-front of the microphone array, while the type of noise sources and their locations are unknown and variable.

The database contains recordings from 34 speakers and a set of 84 recordings of ambient noise, each of which is 5 minutes in duration. The training set comprises 500 utterances per speaker amounting to approximately 15.3 minutes of audio per speaker. The test set comprises 600 utterances under each of the following SNR conditions: -6dB, -3dB, 0dB, 3dB, 6dB and 9dB. All utterances follow a simple grammar that involves digits and letters; only the hypotheses for the letter and digit are scored to evaluate recognition performance.

As the first step, the two channels were mixed with zero delay to obtain a mono-channel audio signal for further processing. We used the standard ASR setup provided for the CHiME challenge. The setup uses speaker dependent acoustic models trained on Mel frequency cepstral coefficients (MFCC) with energy plus the first and second order derivatives. Cepstral mean normalization (CMN) is applied to improve robustness to additive noise. The language model is a simple lattice that covers all possible sequences in the grammar mentioned above and the utterances are decoded using HTK [45].

Spectral representations are extracted using a window size of 25ms and an overlap of 10ms. Speaker patterns were learnt from the training set available for each speaker using a convolutional span of 4 frames. This is equivalent to capturing prominent spectro-temporal patterns that span about 70ms, i.e. subphone patterns. For each speaker, a pattern matrix is learnt using batch learning [46]. Its dimension was empirically set to 100.

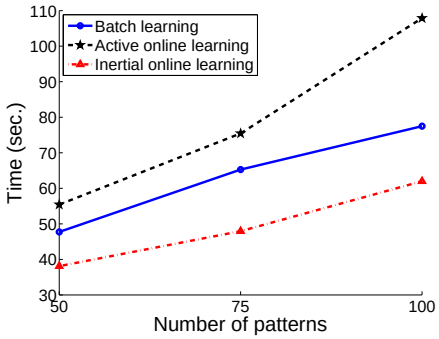


Fig. 9: Average run-time for 10 iterations with online and batch learning.

The learning of noise patterns is more complex. Since the noise is highly diverse and variable, we would ideally like to learn as many patterns as possible from the 7 hours of noise recordings provided in the development set. However, the memory and computational requirements to store and process such large amounts of data are very demanding and the classical batch learning approach simply fails. This problem can be avoided through incomplete training [35], [36], or through online learning as proposed in this work.

2) Incomplete noise pattern learning:: In this experiment, we choose a subset of the background training data to learn the noise patterns. On the one hand this avoids the prohibitive computing and memory demands with batch learning and, on the other hand, provides an opportunity to compare batch learning and online learning in a real application. In our experiment, a single 10 second audio segment from each of the 5 minute waveforms are randomly sampled to obtain 840 seconds of background noise.

With this data, patterns of size 50, 75 and 100 were learnt using the batch, active online and inertial online learning approaches. Each of the 10 seconds of audio segments acts as a piece in online learning. Similar to the experiments in Section IV-A, experiments were conducted with a single dual core 2.6GHz processor with 4GB memory.

The time taken for pattern learning with 10 iterations using the three learning methods is shown in Fig. 9. As in the experiments of Section IV-A, the values presented in the figure are averaged over 100 runs to avoid computational fluctuations. Results confirm that active online learning takes longer than batch learning while inertial online learning outperforms batch mode in terms of computational time, as discussed in Section III-A. ASR performance on the evaluation data is shown in Fig. 10. We observe considerable improvement in accuracies for low SNR conditions and a marginal improvement over the baseline for high SNR conditions with all the three learning approaches. An interesting trend that can be observed in Fig. 10 is that, at lower SNR conditions, active online learning outperforms the other two approaches, while at high SNR conditions and in particular, with higher number of noise patterns, active learning tends to give the worst performance.

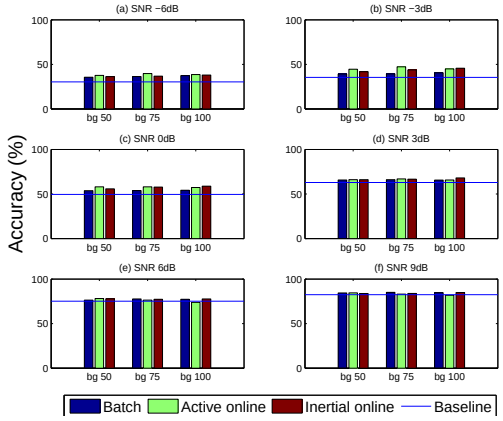


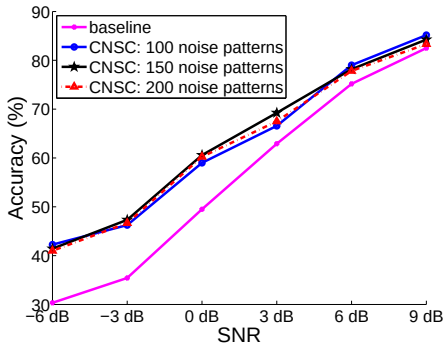
Fig. 10: ASR accuracies with CNSC-based noise cancelation where the background noise patterns are learnt on a set of randomly sampled background audio segments with batch, active online and inertial online learning.

This might be explained by the quick convergence to partial patterns with active learning as a result of which the speech energy may be incorrectly attributed to the noise patterns. This problem is more severe in high SNR conditions where the noise is low while the number of noise patterns is large. This leads to the incorrect attribution of speech energy.

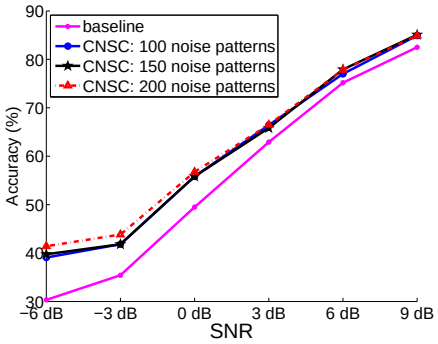
3) Complete noise pattern learning:: Methods employing random sampling techniques are not desirable in practice since a large proportion of the training data remains unused. In order to learn noise patterns from the entire training data, we apply the proposed online pattern learning algorithm. The background training data are provided in segments of 5 minutes each. We use the same partition structure as pieces in the online training algorithm. We learn background bases with 100, 150 and 200 dimensions respectively, all with a convolutional span of 4 frames.

ASR accuracies on the evaluation data with noise cancelation using background patterns learnt with active and inertial online learning algorithms are presented in Fig. 11. We observe that pattern-based denoising significantly improves ASR performance with both active and inertial online learning. Again, active learning is more effective in low SNR conditions than inertial learning; it however does not show much advantage at high SNRs. Simply increasing the number of patterns does not result in significant gains.

Fig. 12 presents a comparison between incomplete and complete learning. We see that the patterns learned from the entire background noise data provide improved accuracies over those learned with random sampling in the case of active learning; for inertial learning, the advantage of using the entire data is not evident, indicating that the training data does not fit a stationary distribution and thus slow inertial learning cannot reach empirical convergence with the use of additional data. Nevertheless, these results clearly demonstrate the capability of online learning in real, large-scale applications.



(a) Active Online learning



(b) Inertial Online learning

Fig. 11: Accuracies with CNSC-based noise cancelation where the background noise patterns are learnt using active and inertial online CNSC.

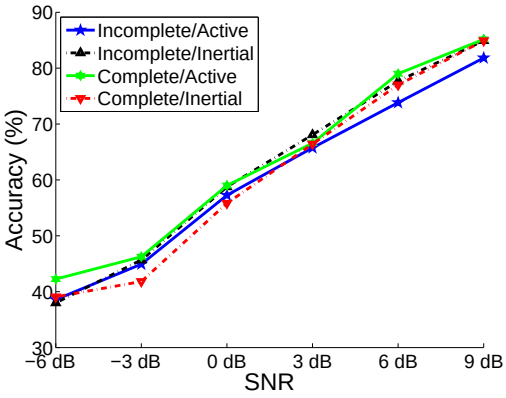


Fig. 12: Accuracy with online CNSC-based noise cancelation where the 100 background noise patterns are learnt based on random or entire data respectively. Results using both active and inertial learning are presented.

V. Conclusion

This paper presents a new online CNSC algorithm to learn convolutive non-negative patterns with sparse coding. Compared to conventional batch learning, the proposed approach is able to learn complex patterns from large volumes of training data and is thus suited to large-scale applications. The theoretical analysis shows that the online algorithm almost surely converges to a stationary point of the cost function

with unlimited computational resources and training data. In real applications where the computational resources are limited and the training data volume is large, online approach tends to gain lower empirical cost than batch learning. This analysis is confirmed by the results we obtained with a toy experiment in speech separation. A noise cancelation task for a large-scale speech recognition system we constructed within the CHiME challenge framework demonstrates that the online learning approach is efficient in learning complex patterns from large corpora in real applications.

Future work includes the study of incremental patterns with online learning. Another direction is to extend the online CNSC approach to other unsupervised learning techniques such as sparse PCA.

APPENDIX A
CONVERGENCE PROOF

A. Convergence of perfect learning

This section proves convergence of the perfect empirical cost $\hat{f}_u(W_u)$. The proof is inspired by the convergence proof for ODL [40] and is adapted to the CNSC algorithm. Before presenting the proof, the following assumptions are made:

(A) **The training signals follow a time-invariant distribution supported by a compact set.** This assumption is reasonable for many signals such as audio and video due to the acquisition process.

(B) **The update rules (5) and (6) are well-defined so that the multiplicative update converges to local minima with unlimited iterations.** Theoretically the convergence with this simple rule is not guaranteed, but it has little impact in practice. In addition, with a simple modification, the convergence can be enforced [6], [10]. We do not consider such complexity in this work, and just assume convergence. Reasonable initialization for the update process and a bounded denominator matrix in the update rules help to respect this assumption.

(C) **The empirical cost function $\hat{f}_u$ is strictly convex with lower-bounded Hessians.** This assumption can be guaranteed with a threshold on the smallest eigenvalue of the accumulated statistics $\frac{1}{u}A(u)$ [40][7].

(D) **The existence of a unique solution for the coefficient matrix is satisfied for signal pieces.** For CNMF, this means the rank of the pattern matrix is not larger than the feature dimension; for CNSC, this means a unique sparse code exists and can be found by l-1 optimization [47].

First notice that assumption (B) implies batch learning converges to a local minimum of the cost function (4). Convergence proof for online learning is not straightforward and involves several lemmas regarding the convergence of variation and the Lipschitz property of the objective and empirical cost functions. Due to space limitations, we simply cite the results; readers can find the proof in [48].

[7]Precisely, $A(u) = \sum_{t=1}^{u} \frac{\dot{A}(t)}{|X_t|}$ following the utterance-averaged cost function (9). This is different from the statistics in Algorithm 1 where the cost function is frame-averaged. The use of utterance-averaged costs in the proof simplifies notation; the proof presented here can be applied similarly to the frame-averaged cost function.

Lemma A.1: Given assumptions (A)-(D), the following convergence properties hold for the objective cost:

(1) $|f_{u+1}(W) - f_u(W)| = O(\frac{1}{u})$

(2) $|f_{u+1}(W_{u+1}) - f_u(W_u)| = O(\frac{1}{u})$

Lemma A.2: Given assumptions (A)-(D), if the update process converges for each signal piece, then $\hat{f}_{u+1} - \hat{f}_u$ is Lipschitz with a factor $k = O(\frac{1}{u})$

With these results, the following property holds for online learning (again, the full proof is given in [48] and we simply cite the results).

Lemma A.3: Given assumptions (A)-(D), if the update process converges for each signal piece, then inertial online learning ensures that:

$$\hat{f}_u(W_{u+1}) - \hat{f}_u(W_u) = O(\frac{1}{u})$$

and active learning ensures that:

$$\hat{f}_u(W_{u+1}) - \hat{f}_u(W_u) = O(\frac{1}{u^2}).$$

Another useful proposition states that the empirical cost variation is $O(\frac{1}{u})$ with both active and inertial learning [48]:

Proposition A.4: Given assumptions (A)-(D), if the multiplicative update converges for each signal piece, the following property holds with both active and inertial online learning:

$$\hat{f}_{u+1}(W_{u+1}) - \hat{f}_u(W_u) = O(\frac{1}{u}).$$

The convergence of online learning can be proved with these results. As Proposition (3) in [40], the following proof applies Theorem B.1 from [49], which states that if the sum of the positive variations of a sequence v_u is bounded, then v_u is quasi-martingale, which converges with probability one (see Theorem B.1). The following proposition states the convergence of active learning.

Proposition A.5: Given assumptions (A)-(D), if the multiplicative update converges for each signal piece, the empirical cost converges to the objective cost almost surely with active online learning, i.e.,

$$\lim_{u\to\infty} \hat{f}_u(W_u) = \lim_{u\to\infty} f_u(W_u) \quad a.s.$$

Proof:

Defining $v_u \equiv \hat{f}_u(W_u)$, we have:

$$\begin{aligned} v_{u+1} - v_u &= \hat{f}_{u+1}(W_{u+1}) - \hat{f}_u(W_u) \\ &\leq \frac{u(\hat{f}_u(W_{u+1}) - \hat{f}_u(W_u))}{u+1} \\ &\quad + \frac{\ell_{X_{u+1}}(W_u) - f_u(W_u)}{u+1} \\ &\quad + \frac{f_u(W_u) - \hat{f}_u(W_u)}{u+1} \end{aligned} \tag{10}$$

where we have applied

$$\ell_{X_{u+1}}(W_{u+1}) < \ell_{X_{u+1}}(W_u).$$

To use Theorem B.1, define the filter of the past information as $\mathcal{F}_u$. Considering the causal nature of $\mathcal{F}_u$ and applying the

fact that $f_u - \hat{f}_u \leq 0$, we have

$$\begin{aligned}\mathbb{E}[v_{u+1} - v_u|\mathcal{F}_u] &\leq \frac{u(\hat{f}_u(W_{u+1}) - \hat{f}_u(W_u))}{u+1} \\ &\quad + \frac{\mathbb{E}[\ell_{X_{u+1}}(W_u)] - f_u(u)}{u+1} \\ &= \frac{u(\hat{f}_u(W_{u+1}) - \hat{f}_u(W_u))}{u+1} \\ &\quad + \frac{f(W_u) - f_u(W_u)}{u+1} \\ &\leq \frac{u(\hat{f}_u(W_u) - \hat{f}_u(W_u))}{u+1} \\ &\quad + \frac{||f - f_u||_\infty}{u+1}\end{aligned}$$

where

$$||f - f_u||_\infty = \sup_W |f(W) - f_u(W)|.$$

According to Lemma A.3,

$$\hat{f}_u(W_{u+1}) - \hat{f}_u(W_u) = O(\frac{1}{u^2})$$

and according to Lemma B.2 from [50] (see in Appendix),

$$\mathbb{E}[||f - f_u||_\infty] = O(\frac{1}{\sqrt{u}}).$$

We therefore have:

$$\mathbb{E}[\mathbb{E}[v_{u+1} - v_u|\mathcal{F}_u]^+] = O(\frac{1}{u^{\frac{3}{2}}}).$$

Thus the expected positive variance of the process v is bounded, so Theorem B.1 can be applied to prove $v = \hat{f}_u(W_u)$ converges with probability one, and that

$$\sum_{u=1}^{\infty} |\mathbb{E}[v_{u+1} - v_u|\mathcal{F}_u]| < +\infty \quad a.s.$$

This further implies that:

$$\sum_{u=1}^{\infty} \mathbb{E}[v_{u+1} - v_u|\mathcal{F}_u] > -\infty \quad a.s.$$

Returning to (10) and by summing over all variations on the two sides, we can prove that:

$$\sum_{u=1}^{\infty} \frac{\hat{f}_u(W_u) - f_u(W_u)}{u+1} < +\infty. \quad a.s.$$

Let $a_u = \hat{f}_u(W_u) - f_u(W_u)$ and $b_u = \frac{1}{u+1}$, we have:

$$\begin{aligned}|a_{u+1} - a_u| &\leq (\hat{f}_{u+1}(W_{u+1}) - \hat{f}_u(W_u)) + \\ &\quad |f_{u+1}(W_{u+1}) - f_u(W_u)|.\end{aligned}$$

According to Proposition A.4 and Lemma A.1, the two items on the right side are $O(\frac{1}{u})$, and so $|a_{u+1} - a_u| = O(\frac{1}{u})$. Applying Lemma B.3, we obtain

$$\lim_{u\to\infty} a_u = \lim_{u\to\infty} \hat{f}_u(W_u) - \lim_{u\to\infty} f_u(W_u) = 0 \quad a.s.$$

This proves that, with unlimited data, the empirical cost with active online learning converges to the objective cost almost surely.

■

It can be further proved that the objective cost function f_u converges to the expected objective cost function, i.e.,

$$||f_u - f||_\infty \to_{u\to\infty} 0$$

and therefore

$$\lim_{u\to\infty} \hat{f}_u(W_u) - \lim_{u\to\infty} f(W_u) = 0 \quad a.s.$$

This result shows that with unlimited data, the empirical cost $\hat{f}(W_u)$ converges to the expected cost $f(W_u)$ almost surely. Since W_u is a stationary point of $\hat{f}_u$ with active learning, it can be proved that the distance of W_u and the set of stationary points of the expected cost function f converges to 0. The proof for this stronger result is similar to Proposition 4 in ODL [40]. Finally, the same approach can be applied with minor modification to prove convergence of the inertial learning. The reader can find the full proof in [48].

B. Convergence of imperfect learning

Here we prove the convergence of the imperfect empirical cost function $\dot{f}_u(W_u)$. This corresponds to 'imperfect learning' where the multiplicative update does not converge for each signal piece. This is generally the case in practice due to limited computing resources.

First define $\delta_u(W)$ as the bias of the imperfect empirical cost shifted away from the perfect empirical cost of the u^{th} piece of signal, i.e.,

$$\delta_u(W) = \ell_{X_u}(W, H_u) - \ell_{X_u}(W, \hat{H}_u).$$

The following proposition states that online learning converges with imperfect coefficients.

Proposition A.6: Suppose online learning does not converge for each signal piece due to limited computational resources, and that the bias $\delta_u(W)$ has the same expectation in spite of u, i.e.,

$$\mathbb{E}_{X,H}[\delta_u(W)] = \epsilon(W) \quad \forall \quad u \tag{11}$$

where the expectation is taken on both signals X and coefficients H. Given assumptions (A)-(D), online learning converges almost surely and that:

$$\lim_{u\to\infty} \dot{f}_u(W_u) = \lim_{u\to\infty} \hat{f}_u(W_u) + \epsilon(W_u) \quad a.s.$$

The proof again can be found in [48]. Theoretically, the perfect empirical cost $\hat{f}_u(W_u)$ does not necessarily converge, but if we assume that the update on W_u in each step improves $\hat{f}_u(W_u)$, then it can be verified that $\hat{f}_u(W_u)$ converges indeed and

$$\lim_{u\to\infty} \dot{f}_u(W_u) = \lim_{u\to\infty} f(W_u) + \epsilon(W_u) \quad a.s. \tag{12}$$

This means that online learning converges to a stationary point of a new function $g(W) = f(W) + \epsilon(W)$, which is obviously suboptimal for our task which intends to minimize the expected cost $f(W)$. If the multiplicative update approaches to convergence, $\epsilon(W)$ approaches to 0 and the learning process converges to $f(W)$, which is just the form of a perfect learning.

Note that the above analysis relies on strong assumptions. First the identical expected bias assumption (11) is not always

respected, although some support can be found from the random coefficient initialization and identical number of multiplicative iterations. Second, the assumption of improvement on $\hat{f}_u(W_u)$ can be simply false if the coefficients of each piece of signal are highly imperfect. This suggests that the convergence property with imperfect learning is not guaranteed in practice, and may highly task-dependent.

Some interesting results can be obtained from the empirical convergence proposition. First, notice that under the proposed assumptions, both active learning and inertial learning converge to local minima of cost functions in the same form $g(w) = f(W) + \epsilon(W)$, which means the two learning strategies may obtain similar empirical cost. Second, since ℓ is an utterance-based average cost, the convergence behavior is not impacted by the length of signal pieces, and involving pieces of variable lengths does not impact the convergence property.

APPENDIX B
THEOREMS USED IN THE PAPER

In this section, we cite some theorems that are used for the convergence proof in this paper. These theorem are mostly reproduced from [40] for convenience of readers.

Theorem B.1: **[Sufficient condition of convergence for a stochastic process. See [49], [51], [52]]**

Let (Q, F, P) be a measurable probability space, u_t, for $t \geq 0$, be the realization of a stochastic process and F_t be the filtration determined by the past information at time t. Let

$$\delta_t = \begin{cases} 1 & \text{if } \mathbb{E}[u_{t+1} - u_t | F_t] > 0, \\ 0 & \text{otherwise.} \end{cases}$$

If for all t, $u_t \geq 0$ and $\sum_{t=1}^{\infty} \mathbb{E}[\delta_t(u_{t+1} - u_t) < \infty$, then u_t is a quasi-martingale and converges almost surely. Moreover,

$$\sum_{t=1}^{\infty} |\mathbb{E}[u_{t+1} - u_t | F_t]| < +\infty \quad a.s.$$

Lemma B.2: **[A corollary of Donsker theorem for $O(\frac{1}{\sqrt{n}})$ of $|f_n - f|$. See [50], chap. 19.]**

Let $F = \{f_\theta : \chi \to \mathbb{R}, \theta \in \Theta\}$ be a set of measurable functions indexed by a bounded subset Θ of $\mathbb{R}^d$. Suppose that there exists a constant K such that

$$|f_{\theta_1} - f_{\theta_2}| \leq K||\theta_1 - \theta_2||_2$$

for every θ_1 and θ_2 in Θ and x in χ. Then F is P-Donsker. For any f in F, define $\mathbb{P}_n f$, $\mathbb{P} f$ and $\mathbb{G}_n f$ as

$$\mathbb{P}_n f = \frac{1}{n}\sum_{i=1}^{n} f(X_i), \mathbb{P}_f = \mathbb{E}_X[f(X)], \mathbb{G}_n f = \sqrt{n}(\mathbb{P}_n f - \mathbb{P} f).$$

Further suppose for all f, $\mathbb{P} f^2 < \delta^2$ and $||f||_\infty < M$ and that the random elements X_i are Borel-measurable. Then we have

$$\mathbb{E}_P||\mathbb{G}_n||_F = O(1),$$

where $||\mathbb{G}_n||_F = sup_{f \in F}|G_n f|$.

Lemma B.3: **[A lemma on positive converging sums. See [53], prop 1.2.4.]**

Let a_n, b_n be two real sequences such that for all n, $a_n \geq 0$ and $b_n \geq 0$, $\sum_{n=1}^{\infty} a_n = \infty$, $\sum_{n=1}^{\infty} a_n b_n < \infty$, $\exists K > 0 \;\; s.t. \;\; |b_{n+1} - b_n| < K a_n$. Then $\lim_{n \to +\infty} b_n = 0$.

REFERENCES

[1] P. Paatero and U. Tapper, "Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values," *Environmetrics*, vol. 5, no. 12, pp. 111–126, 1994.

[2] P. Paatero, "Least squares formulation of robust non-negative factor analysis," *Chemometrics and Intelligent Laboratory Systems*, vol. 37, pp. 23–35, 1997.

[3] D. D. Lee and H. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, October 1999.

[4] ——, "Algorithms for non-negative matrix factorization," in *Proc. NIPS*, 2000, pp. 556–562.

[5] D. L. Donoho and V. Stodden, "When does non-negative matrix factorization give a correct decomposition into parts?" in *Proc. NIPS 2003*, 2003.

[6] C. Lin, "On the convergence of multiplicative update algorithms for non-negative matrix factorization," *IEEE Transactions on Neural Networks*, vol. 18, no. 6, pp. 1589–1596, Nov. 2007.

[7] A. Shashua and T. Hazan, "Non-negative tensor factorization with applications to statistics and computer vision," in *Proc. International Conference on Machine Learning (ICML)*, 2005, pp. 792–799.

[8] P. Smaragdis and J. C. Brown, "Non-negative matrix factorization for polyphonic music transcription," in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2003, pp. 177–180.

[9] D. Guillamet, J. Vitrià, and B. Schiele, "Introducing a weighted non-negative matrix factorization for image classification," *Pattern Recognition Letters*, vol. 24, no. 14, pp. 2447–2454, 2003.

[10] M. W. Berry, M. Browne, A. N. Langville, V. P. Pauca, and R. J. Plemmons, "Algorithms and applications for approximate non-negative matrix factorization," *Computational Statistics and Data Analysis*, vol. 52, no. 1, pp. 155–173, 2006.

[11] W. Xu, X. Liu, and Y. Gong, "Document clustering based on non-negative matrix factorization." in *Proc. SIGIR*, 2003, pp. 267–273.

[12] M. Welling and M. Weber, "Positive tensor factorization," *Pattern Recognition Letters*, vol. 22, no. 12, pp. 1255–1261, 2001.

[13] I. S. Dhillon and S. Sra, "Generalized non-negative matrix approximations with Bregman divergences," in *Proc. Neural Information Proc. Systems*, 2005, pp. 283–290.

[14] R. Zdunek and A. Cichocki, "Non-negative matrix factorization with quasi-newton optimization," in *Proc. Eighth International Conference on Artificial Intelligence and Soft Computing, ICAISC*. Springer, 2006, pp. 870–879.

[15] A. Cichocki, R. Zdunek, and S. ichi Amari, "Csiszár's divergences for non-negative matrix factorization: Family of new algorithms," in *Lecture Notes in Computer Science (LNCS)*. Springer, 2006, pp. 32–39.

[16] C. Lin, "Projected gradient methods for non-negative matrix factorization," *Neural Computation*, vol. 19, no. 10, pp. 2756–2779, 2004.

[17] R. Zdunek and A. Cichocki, "Nonnegative matrix factorization with constrained second-order optimization," *Signal Processing*, vol. 87, no. 8, pp. 1904–1916, 2007.

[18] C. Févotte, N. Bertin, and J.-L. Durrieu, "Nonnegative matrix factorization with the Itakura-Saito divergence: With application to music analysis," *Neural Computation*, pp. 793–830, 2009.

[19] R. Kompass, "A generalized divergence measure for non-negative matrix factorization," *Neural Computation*, vol. 19, no. 3, pp. 780–791, 2007.

[20] H. Kim and H. Park, "Nonnegative matrix factorization based on alternating nonnegativity constrained least squares and active set method," *SIAM J. Matrix Anal. Appl.*, vol. 30, pp. 713–730, July 2008.

[21] A. T. Cemgil, "Bayesian inference for non-negative matrix factorisation models," *Intell. Neuroscience*, vol. 2009, pp. 4:1–4:17, January 2009.

[22] M. N. Schmidt, O. Winther, and L. K. Hansen, "Bayesian non-negative matrix factorization," in *Independent Component Analysis and Signal Separation, International Conference on*, ser. Lecture Notes in Computer Science (LNCS), vol. 5441. Springer, 2009, pp. 540–547.

[23] Z. Yang and E. Oja, "Linear and non-linear projective non-negative matrix factorization," *IEEE Transactions on Neural Network*, vol. 21, no. 5, pp. 734–749, 2010.

[24] P. Smaragdis, "Convolutive speech bases and their application to supervised speech separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 1, pp. 1–12, January 2007.

[25] D. FitzGerald and E. Coyle, "Shifted non-negative matrix factorisation for sound source separation," in *Proc. IEEE conference on Statistics in Signal Processing*, 2005.

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

[27] M. Heiler and C. Schnorr, "Learning sparse representations by non-negative matrix factorization and sequential cone programming," *Journal of Machine Learning Research*, vol. 7, pp. 1385–1407, 2006.
[28] H. Kim and H. Park, "Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis," *Bioinformatics*, vol. 23, no. 12, pp. 1495–1502, 2007.
[29] P. D. O'Grady and B. A. Pearlmutter, "Convolutive non-negative matrix factorisation with a sparseness constraint," in *Proc. IEEE International Workshop on Machine Learning for Signal Processing (MLSP 2006)*, Maynooth, Ireland, Sep. 2006, pp. 427–432.
[30] T. Virtanen, "Monaural sound source separation by nonnegative matrix factorization with temporal continuity and sparseness criteria," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 3, pp. 1066–1074, 2007.
[31] W. Wang, "Convolutive non-negative sparse coding," in *Proc. IJCNN'08*, 2008, pp. 3681–3684.
[32] W. Wang, A. Cichocki, and J. A. Chamber, "A multiplicative algorithm for convolutive non-negative matrix factorization based on squared Euclidean distance," *IEEE Transactions on Signal Processing*, vol. 57, no. 5, pp. 2858–2864, 2009.
[33] C. D. Sigg, T. Dikk, and J. M. Buhmann, "Speech enhancement with sparse coding in learned dictionaries," in *Proc. ICASSP'10*, 2010.
[34] T. Virtanen, "Separation of sound sources by convolutive sparse coding," in *Proc. SAPA'2004*, 2004.
[35] W. Smit and E. Barnard, "Continuous speech recognition with sparse coding," *Computer Speech and Language*, vol. 23, no. 2, 2009.
[36] G. Sivaram, S. Nemala, M. Elhilali, T. Tran, and H. Hermansky, "Sparse coding for speech recognition," in *Proc. ICASSP'10*, 2010.
[37] P. O'Grady and B. Pearlmutter, "Convolutive non-negative matrix factorisation with a sparseness constraint," in *Proc. IEEE workshop on Machine Learning for Signal Processing*, September 2006, pp. 427–432.
[38] H. Robbins and S. Monro, "A stochastic approximation method," *Annals of Mathematical Statistics*, vol. 22, pp. 400–407, 1951.
[39] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. Fifth Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1, 1967, pp. 281–297.
[40] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *Journal of Machine Learning Research*, vol. 2010, no. 11, pp. 19–60, January 2010.
[41] S. S. Bucak and B. Gunsel, "Incremental subspace learning via non-negative matrix factorization," *Pattern Recognition*, vol. 42, no. 5, pp. 788–797, 2009.
[42] A. Lefevre, F. Bach, and C. Fevotte, "Online algorithms for nonnegative matrix factorization with the itakura-saito divergence," in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, Mohonk, NY, October 2011.
[43] E. Vincent, C. Fevotte, and R. Gribonval, "Performance measurement in blind audio source separation," *IEEE Trans. Audio, Speech and Language Processing*, vol. 14, no. 4, pp. 1462–1469, 2006.
[44] H. Christensen, J. Barker, N. Ma, and P. Green, "The CHiME corpus: A resource and a challenge for computational hearing in multisource environments," in *Interspeech*, 2010.
[45] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. Liu, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. Woodland, *The HTK Book (for Hidden Markov Model Toolkit Version 3.4)*, 2006.
[46] R. Vipperla, S. Bozonnet, D. Wang, and N. Evans, "Robust speech recognition in multi-source noise environments using convolutive non-negative matrix factorization," in *CHiME: Workshop on Machine Listening in Multisource Environments*, 2011, pp. 74–79.
[47] J. jacques Fuchs, "On sparse representations in arbitrary redundant bases," *IEEE Transactions on Information Theory*, vol. 50, no. 6, pp. 134–1344, 2004.
[48] D. Wang, R. Vipperla, N. Evans, and T. F. Zheng, "Online non-negative convolutive pattern learning for speech signals," EURECOM, Tech. Rep. RR-11-261, 2011.
[49] D. Fisk, "Quasi-martingales," *Transactions of the American Mathematical Society*, vol. 120, no. 3, pp. 359–388, 1965.
[50] A. W. Van der Vaart, *Asymptotic Statistics*. Cambridge University Press, 1998.
[51] L. Bottou, "Online algorithms and stochastic approximations," in *Online Learning and Neural Networks*, D. Saad, Ed., 1998.
[52] M. Métivier, *Semi-martigales*. Walter de Gruyter, 1983.
[53] D. Bertsekas, *Nonlinear Programming*. Athena Scientific Belmont, 1999.

Dong Wang (M'09) received the B.Sc. and M.Sc. in computer science at Tsinghua Univ. in 1999 and 2002, and then worked for Oracle China in 2002-2004 and IBM China in 2004-2006. He joined CSTR, University of Edinburgh in 2006 as a research fellow and PhD student supported by a Marie Curie fellowship, from where he received his Ph.D. in 2010. From 2010 to 2011 he worked in EURECOM as a post doctoral fellow, and from 2011 to 2012 he was a senior research scientist in Nuance. He is now an assistant professor of Tsinghua University.

Ravichander Vipperla is a post doctoral researcher in the department of Multimedia Communications at Eurecom, France. He received his PhD degree in Informatics from University of Edinburgh in 2011. Prior to that, he served as a research engineer at Reliance Communications, India from 2004-2006 where he was actively involved in the development of automatic speech recognition technology for Indian languages. He received his M.Tech degree in ICT from DA-IICT, India and B.Engg degree in Electronics and Telecommunications from Mumbai University in 2004 and 2002 respectively.

Nicholas Evans was awarded M.Eng. and Ph.D. degrees from the University of Wales Swansea (UWS), UK in 1999 and 2003 respectively and was appointed as was appointed as a Lecturer in Communications in 2002. After one year at the Laboratoire Informatique d'Avignon (LIA) he joined EURECOM as an Assistant Professor in 2007 where he now heads the Speech and Audio Processing Research Group. His current research interests include speaker diarization, speaker recognition, biometrics, speech enhancement, noise compensation and echo cancellation. He is a member of ISCA, EURASIP, the IEEE and its Signal Processing Society and currently serves as an associate editor for the EURASIP Journal on Audio, Speech and Music Processing.

Thomas Fang Zheng (M'99-SM'06) received his PhD degree in computer science and technology at Tsinghua University, Beijing, China in 1997. He is now a research professor and director of Center for Speech and Language Technologies at Tsinghua University. His research focuses on speech and language processing and has published more than 210 papers. He also plays active roles in a number of communities, including the Chinese Corpus Consortium, the Standing Committee of Chinas National Conference on Man-Machine Speech Communication, the Oriental COCOSDA, the Asia-Pacific Signal and Information Processing Association (APSIPA), Chinese Information Processing Society of China, the Acoustical Society of China, and the Phonetic Association of China. He is in the editorial board of IEEE Transactions on Audio, Speech and Language Processing, Speech Communication, APSIPA Transaction on Signal and Information Processing, Journal of Signal and Information Processing, and the Journal of Chinese Information Processing, and so on.