Online Pattern Learning for Non-Negative Convolutive Sparse Coding

Dong Wang, Ravichander Vipperla and Nicholas Evans

Multimedia Communications Department, EURECOM
F-06560 Sophia-Antipolis, France
dong.wang@eurecom.fr, vipperla@eurecom.fr, evans@eurecom.fr

Abstract

The unsupervised learning of spectro-temporal speech patterns is relevant in a broad range of tasks. Convolutive non-negative matrix factorization (CNMF) and its sparse version, convolutive non-negative sparse coding (CNSC), are powerful, related tools. A particular difficulty of CNMF/CNSC, however, is the high demand on computing power and memory, which can prohibit their application to large scale tasks. In this paper, we propose an online algorithm for CNMF and CNSC, which processes input data piece-by-piece and updates the learned patterns after the processing of each piece by using accumulated sufficient statistics. The online CNSC algorithm remarkably increases converge speed of the CNMF/CNSC pattern learning, thereby enabling its application to large scale tasks.

Index Terms: non-negative convolutive sparse coding, online pattern learning

1. Introduction

Many signals exhibit clear spectro-temporal patterns; the automatic discovery of such patterns is highly important in order to understand the signals and for the design of suitable approaches in related, practical applications. For instance, patterns in speech signals are highly correlated to speaker characteristics and speech contents. Some of the patterns can be defined by humans and learned using supervised approaches such as neural networks, whereas complex patterns are difficult to pre-define and annotate, hence the need to resort to unsupervised approaches.

Various unsupervised learning techniques have been developed need for automatic pattern discovery and are generally based on the idea of searching for a number of patterns that can be used to reconstruct training signals most precisely with respect to a certain objective function and subject to a set of constraints. The constraints of orthogonal patterns and the minimisation of the l-2 reconstruction loss leads to principle component analysis (PCA); those of non-Gaussian patterns and the minimisation of the mutual information (or non-Gaussianity) leads to independent component analysis (ICA); non-negative constraints applied to both patterns and decomposition coefficients leads to non-negative matrix factorisation (NMF) [1].

Such approaches can all be extended by imposing further constraints. The l-1 minimisation, e.g., Lasso [2], is particularly appealing since it leads to sparse decomposition and allows over-complete patterns, i.e. the number of patterns (usually referred to as 'bases' in the case of NMF) is larger than the signal dimension. For example, a large number of bases that lead to indefinite decomposition in NMF can be applied without difficulties in sparse NMF (SNMF).

All these unsupervised approaches assume independent signals and therefore they cannot be applied to discover temporal patterns. A simple solution involves the reconstruction of original signals through the concatenation of a window of neighbouring signals and then the retrieval of patterns in the resulting temporal signals. A more sophisticated approach involves the sharing of decompositions among a set of bases with a time shift. The latter leads to a convolutive pattern learning approach, e.g. convolutive NMF (CNMF) [3] or convolutive non-negative sparse coding (CNSC) [4, 5].

While promising results have been demonstrated in some tasks such as speech enhancement [6] and source separation [7], sparse approaches such as CNMF and CNSC place high demands on both processing power and memory. It is further far from straight forward to apply parallel computing due to the iterative nature of related algorithms. This problem is more serious for CNSC where the number of bases is usually relatively larger than it is for CNMF. For this reason most related publications focus on small databases, e.g. TIDIGITS or TIMIT and learning is often based on even smaller subsets or random samples [8, 9]. Such partial-learning schemes are clearly unacceptable for complex tasks. For instance, in speech recognition, a large amount of data are necessary to ensure the learned patterns are speaker independent. To address this problem we propose in this paper a new, on-line learning approach for CNMF and CNSC, which processes input signals piece-by-piece and updates the set of bases using accumulated, sufficient statistics. With very few iterations for each piece of the signal, learned patterns quickly converge to local minima of the objective function thereby facilitating its application to large scale tasks.

In the following sections, we first present the online CNSC algorithm (CNMF can be regarded as a special case of CNSC with zero sparsity). In Section 3 the proposed approach is compared with the conventional batch model approach using a toy experiment. Conclusions and ideas for further work are given in Section 4.

2. Online convolutive pattern learning

CNSC can be formulated slightly differently depending on the objective function [4, 5]. We follow the formulation in [5] and minimise the following:

$$L(W,H) \quad = \quad |X - \hat{X}(W)|_2^2 + \lambda |H|_1 \tag{1}$$

where λ is a factor controlling the sparsity of H, and $|.|_l$ denotes the Frobenius l-norm, which is equivalent to the sum of squares when $l = 2$ or to the sum of absolute values when $l = 1$. $X \in R_{0,+}^{M \times N}$ represents the original signal of length N in the M-dimensional space, $W \in R_{0,+}^{M \times R \times P}$ represents R bases with convolution range P, and $H \in R_{0,+}^{R \times N}$ are the coefficients. $\hat{X}$ is the approximate reconstruction of X and has the form:

$$\hat{X} = \sum_{p=0}^{P-1} W_p \overrightarrow{{}^{p}H} \tag{2}$$

where $\overrightarrow{{}^{p}H}$ shifts H by p columns to the right and where $W_p \in R_{0,+}^{M \times R}$ are the corresponding bases of $\overrightarrow{{}^{p}H}$.

As presented in [5] the update equations for W and H can be derived by slightly modifying the procedure presented in the seminal NMF paper [1], leading to:

$$H \leftarrow H \odot \frac{W_p^T \overleftarrow{X}^p}{W_p^T \overleftarrow{\hat{X}}^p + \lambda \Xi} \quad (3)$$

$$W_p \leftarrow W_p \odot \frac{X \overrightarrow{H}^{pT}}{\hat{X} \overrightarrow{H}^{pT}}$$

where $\odot$ is the element-wise product and where the division is also element-wise. Ξ is a matrix with all the elements equal to 1, and $\overleftarrow{H}^p$ is the coefficient matrix shifted p columns to the left. Note that, for different p, the update for H is different and is usually averaged over p. The above equations show that most of the computation is involved in calculating the reconstruction $\hat{X}$, which is of complexity $O(M \times N \times R \times P)$. This is unrealistic for complex patterns (large R) and large databases (large N). All signals must furthermore be loaded into the memory for computation and thus memory requirements are prohibitive.

An online dictionary learning (ODL) approach has been presented recently to address the computational requirements in the case of independent signals [10]. This approach reads in and decomposes signals one-by-one and updates the bases each time a signal is processed, according to accumulated statistics. The authors show that learning can 'almost' converge to a global optimum with unlimited training data, subject to some reasonable assumptions. Although it cannot be applied to convolutive approaches, this work directly inspired our investigation into an online approach for learning convolutive patterns, particularly for CNSC.

Compared to the work addressed in [10] which assumes independent signals we need to deal with the convolution in CNSC. We solve this by assuming that signals are convolutively generated by patterns within a piece of neighbouring signal, while signals in different pieces are independent. By applying this assumption and substituting (2) we obtain a modified learning approach:

$$W_p \leftarrow W_p \odot \frac{\sum_u B(p;u)}{\sum_q W_q \sum_u A(q,p;u)} \quad (4)$$

where

$$A(q,p;u) = \overrightarrow{H}^q(u) \overrightarrow{H}^{pT}(u)$$

and

$$B(p;u) = X(u) \overrightarrow{H}^{pT}(u)$$

where u is the piece index. The segmentation of the signal into pieces is arbitrary. For speech signals, a segmentation according to sentence boundaries avoids the splitting of voiced patterns and is thus a natural choice.

Note that the sizes of $A(q,p;u)$ and $B(p;u)$ are independent of the data size or number of pieces, and can be regarded as statistics of the piece u. The computation can now be applied to signals piece-by-piece, and for each piece, the bases are updated with a number of iterations using (3) and (4), thereby resulting in optimal patterns for the data processed *so far*. An important aspect of the piece-wise iteration is that the computation only relates to the present piece, i.e. $A(q,p;u)$ and $B(p;u)$, and that the contribution of pieces processed previously can be 'memorised' using two auxiliary variables

$$A(q,p) = \sum_u A(q,p;u)$$

and

$$B(p) = \sum_u B(p;u)$$

and hence being used for 'warm startup'.

Algorithm 1 Online CNCS pattern learning

U: number of pieces
K: iteration
$A(i,j) \in R^{R \times R}, \quad 0 < i,j < P$
$B(i) \in R^{M \times R}, \quad 0 < i < P$
$A(i,j) \leftarrow 0, \quad \forall i,j$
$B(i) \leftarrow 0, \quad \forall i$
for u := 0 to U-1 **do**
  randomize(H)
  for k := 0 to K-1 **do**
    if activeW **then**
      $W = updateW(A, B, X(u), W, H)$
    end if
    $H = updateH(X, W, H)(Eq.3)$
  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
$\dot{A} \in R^{R \times R}, \quad 0 < i,j < P$
$\dot{B} \in R^{M \times R}, \quad 0 < i < P$
$\dot{A}(i,j) = \overrightarrow{H}^{ij} \overrightarrow{H}^{T}$
$\dot{B}(i) = X \overrightarrow{H}^{iT}$
$A = A + \dot{A}$
$B = B + \dot{B}$
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
$W_p = \frac{\dot{W}_p}{|\dot{W}_p|_2^2} \quad \forall p \ s.f. \ W_p \in R_{0,+}^{M \times R}$
return $[\dot{A}, \dot{B}, W]$

This leads to the online pattern learning approach for CNSC, as shown in Algorithm 1, where the flag $activeW$ indicates if the bases should be updated when updating the coefficients (see Section 3). Algorithm 2 illustrates the pattern update process (4). Matlab code for these algorithms is online available[1].

Besides the handling of convolution, the main difference between our online pattern learning approach and ODL [10] is that we do not pursue an optimal H for each signal piece; instead, we apply a small number of iterations to obtain a sub-optimal H and assume it is sufficient for accumulating statistics. This may lead to a sub-optimal solution for a particular dataset but can remarkably speed up the computation. This form is largely inherited from the conventional CNSC update. As with ODL, learned patterns tend to be more and more accurate as the quantity of data increases; with increasing I, resulting patterns approach the optimal. The CNSC-style pattern update in Algorithm 2 can be replaced by the quadratic approach as in ODL. While we do not have space here for an extensive discussion, the conclusion is that the same accumulated statistics A and B are necessary and sufficient.

Finally we note that the proposed online learning approach aims to deliver efficient pattern discovery instead of optimising coefficients. With the discovered patterns the optimal CNSC de-

[1] `http://audio.eurecom.fr/software`

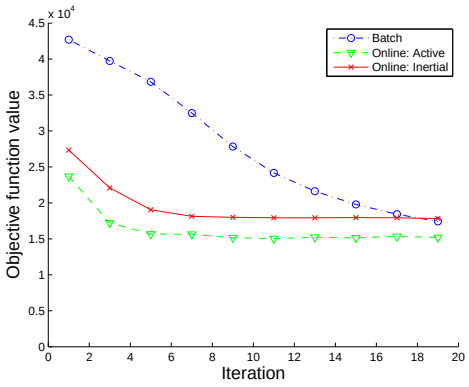

Figure 1: Value of the objective function (Equation 1) for the first 20 iterations with online and batch learning.

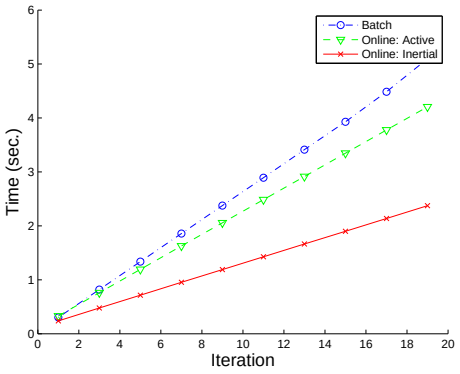

Figure 2: Average run-time for the first 20 iterations with online and batch learning.

composition can be conducted on pieces either by applying (3) or by more efficient techniques such as quadratic optimisation, and this is amenable to parallel computation.

3. Experiments

In this section, we study the behavior of the proposed online pattern learning algorithm and compare it to conventional batch-mode CNSC learning. We use a toy experiment proposed by P. Smaragdis[2] in a study of CNMF [3]. 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 bases to separate the two voices from a segment of mixed speech. The two individual speech segments that are used to learn patterns 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. Smaragdis showed that CNMF is able to find patterns that are specific to the individual speakers and that it can thus be employed to separate the speech signal into to its two individual components, i.e. speakers. Sparse coding has been shown to deliver improved performance [7].

The two speech 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 Fourier transform is applied to each frame and the magnitude spectrum is used as a non-negative representation which is suitable for NMF and NCSC. All results reported here relate to fixed parameters of $R = 20$, $P = 4$ and $\lambda = 0.01$. They have not been optimised for the task in hand and were chosen according to good results obtained on a small data subset. All experiments were conducted on a desktop machine with two dual-core 2.60GHz CPUs and memory of 4GB.

3.1. Online learning and batch learning

In the first experiment we compare the convergence speed of the online and batch learning approaches. These experiments were conducted using the recording of male speech. For the online approach, each speech segment is divided into 10 pieces, and two configurations are assessed: active online learning, which allows the simultaneously updating of both patterns and coefficients ($activeW = true$ in Algorithm 1), and inertial online learning, which fixes the patterns while optimising coefficients ($activeW = false$ in Algorithm 1). To avoid unrelated fluctuations in computing capacity we ran the experiments 100 times and computed the average run-time.

Figure 1 shows the value of the objective function, in Equation 1, for the first 20 iterations. It is interesting to see that both variants of online learning converge in the first 4 or 5 iterations whereas batch learning requires about 20 iterations to reach the same level. With more iterations, although not shown here, batch learning achieves better results, according to reasons we discuss shortly. The comparison of the two online learning approaches shows that active learning gives better results. This is to be expected given its facility to update patterns in addition to coefficient optimisation.

Figure 2 shows the corresponding average run-time for the first 20 iterations of all three algorithms. We see that, as expected, inertial online learning is most efficient. Interestingly, active online learning proves to be more efficient than batch learning even though an extra pattern update is performed for each piece. This rather unexpected improvement in efficiency may be due to the avoidance of large matrix operations in the case of active online learning.

3.2. Piece length

In the second experiment we study the impact of piece segmentation on the rate of convergence for online approaches. For that we split the recording of male speech (2085 frames) into U pieces, and conduct pattern learning as described above. Note that, if the entire speech signal is treated as a single piece ($U = 1$), active online learning is equivalent to batch learning.

The value of the objective function for the two online learning approaches is shown in Figure 3 for $U = 1$ to 10 and after 10 iterations of the algorithm in all cases. We first observe that the two online learning approaches substantially outperform batch learning ($U = 1$) and that better patterns can be obtained by learning on smaller pieces.

The corresponding average run-time is shown in Figure 4. Inertial online learning tends to be more efficient when the speech signal is segmented into more pieces as the computation for accompanying pattern updates is saved. Active learning is more efficient than batch learning ($U = 1$) with a small number of pieces; however with a larger number of pieces, it is slightly less efficient due to the final pattern update which is required for each piece.

A better understanding of online learning is achieved by taking both the number of iterations and pieces into account. First, active online learning can be seen as an extension of batch learning where the signals are split into multiple pieces instead of a single piece. The splitting of signals into smaller pieces allows 'early learning' of patterns before all signals are processed. The simultaneous updating of patterns and coefficients leads to faster converge for new data; however, if the pieces are too small and there are too many iterations, there is a risk that learned patterns are over-fitted to the most recent piece, thereby

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

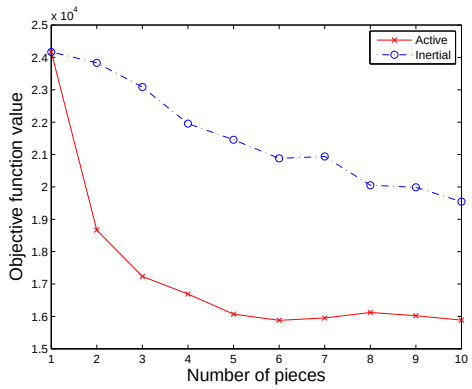

Figure 3: Value of the objective function for $U = 1$ and 10 pieces and after 10 iterations of either active or inertial online learning.

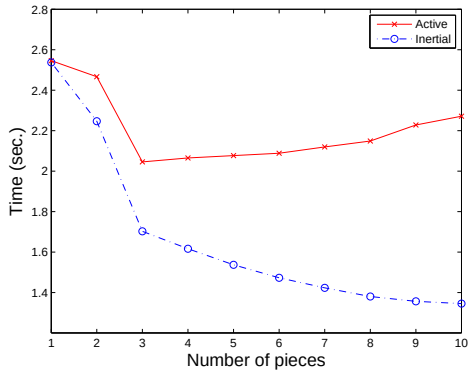

Figure 4: Average run-time for between $U = 1$ to 10 pieces and after 10 iterations of either active or inertial online learning.

leading to some bias. It is thus necessary that the piece length is configured according to the number of iterations, or vice versa. In the extreme case where the number of iterations is infinite, the speech signal should not be split so as to avoid overfitting, which is the case for batch learning. This means that active learning never performs worse than batch learning if a suitable piece size is selected.

Inertial online learning shares the same advantage of early learning but suffers less from over-fitting. In fact, smaller pieces are preferred in this case so that coefficient optimisation is more accurate with a limited number of iterations, in which case there is a higher chance of converging to global optimum.

3.3. Speech separation

We finally investigate the use of bases learned through online and batch approaches to separate the male/female mixed speech. Each of the two individual training utterances (1 male, 1 female) are split into 10 pieces and patterns are learned with 10 iterations of each algorithm. The spectrum of the mixed speech signal is then projected independently onto the two sets of bases and the reconstruction residual (square error) of the resulting magnitude spectrum is computed for the individual male and female speech signals respectively. The error is calculated for each frame and the accumulated error, computed for the entire utterance, is used as the evaluation metric.

Results shown in Table 1 are consistent with the above discussion in that the online approaches tends to give better performance than batch learning and that active online learning outperforms inertial online learning, but remains less efficient. The original and resulting speech files resulting from these experiments are available online[3].

Learning approach	Residual	Av. learning time (sec.)
Batch	78,5	5.5
Online (Active)	60,9	4.6
Online (Inertial)	70,4	2.7

Table 1: Speech separation performance with batch and online learning. The second column illustrates the reconstruction residual (error) and the third column illustrates the average runtime.

4. Conclusion

This paper presents a new online learning approach for unsupervised convolutive pattern learning. Compared to conventional batch learning and online dictionary learning approaches for independent signals, the proposed approach processes signals and updates learned patterns piece-by-piece. This approach retains the convolutive features in signal generation while speeding up convergence significantly. Future work involves employing quadratic coefficient optimisation and the application of the online approach to large-scale tasks.

5. Acknowledgements

This work was partially supported by the French Ministry of Industry (Innovative Web call) under contract 09.2.93.0966, "Collaborative Annotation for Video Accessibility" (ACAV).

6. References

[1] 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.

[2] R. Ribshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society*, vol. 58, pp. 267–288, 1996.

[3] 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.

[4] 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.

[5] W. Wang, "Convolutive non-negative sparse coding," in *Proc. IJCNN'08*, 2008, pp. 3681–3684.

[6] C. D. Sigg, T. Dikk, and J. M. Buhmann, "Speech enhancement with sparse coding in learned dictionaries," in *Proc. ICASSP'10*, 2010.

[7] T. Virtanen, "Separation of sound sources by convolutive sparse coding," in *Proc. SAPA'2004*, 2004.

[8] W. Smit and E. Barnard, "Continuous speech recognition with sparse coding," *Computer Speech and Language*, vol. 23, no. 2, 2009.

[9] G. Sivaram, S. Nemala, M. Elhilali, T. Tran, and H. Hermansky, "Sparse coding for speech recognition," in *Proc. ICASSP'10*, 2010.

[10] 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.

[3] `http://audio.eurecom.fr/public/OL`