Unsupervised Interpretable Pattern Discovery in Time Series Using Autoencoders

Kevin Bascol[1], Rémi Emonet[1], Elisa Fromont[1], and Jean-Marc Odobez[2]

[1] Univ Lyon, UJM-Saint-Etienne, CNRS, Institut d'Optique Graduate School, Laboratoire Hubert Curien UMR 5516, F-42023, SAINT-ETIENNE, France
[2] Idiap Research Institute, CH-1920 Martigny Switzerland

Abstract. We study the use of feed-forward convolutional neural networks for the unsupervised problem of mining recurrent temporal patterns mixed in multivariate time series. Traditional convolutional autoencoders lack interpretability for two main reasons: the number of patterns corresponds to the manually-fixed number of convolution filters, and the patterns are often redundant and correlated. To recover clean patterns, we introduce different elements in the architecture, including an adaptive rectified linear unit function that improves patterns interpretability, and a group-lasso regularizer that helps automatically finding the relevant number of patterns. We illustrate the necessity of these elements on synthetic data and real data in the context of activity mining in videos.

1 Introduction

Unsupervised discovery of patterns in temporal data is an important data mining topic due to numerous application domains like finance, biology or video analysis. In some applications, the patterns are solely used as features for classification and thus the classification accuracy is the only criterion. This paper considers different applications where the patterns can also be used for data analysis, data understanding, and novelty or anomaly detection [5,6,4,18].

Not all time series are of the same nature. In this work, we consider the difficult case of multivariate time series whose observations are the result of a combination of different recurring phenomena that can overlap. Examples include traffic videos where the activity of multiple cars causes the observed sequence of images [6], or aggregate power consumption where the observed consumption is due to a mixture of appliances [10]. Unlike many techniques from the data mining community, our aim is not to list *all* recurrent patterns in the data with their frequency but to reconstruct the entire temporal documents by means of a *limited and unknown* number of recurring patterns together with their occurrence times in the data. In this view, we want to un-mix multivariate time series to recover how they can be decomposed in terms of recurrent temporally-structured patterns. Following the conventions used in [6], we will call a temporal pattern a *motif*, and an input multivariate time series a *temporal document*.

Artificial neural networks (or deep learning architectures) have (re)become tremendously popular in the last decade due to their impressive, and so far

not beaten, results in image classification, speech recognition and natural language processing. In particular, autoencoders are artificial neural networks used to learn a compressed, distributed representation (encoding) for a set of data, typically for the purpose of dimensionality reduction. It is thus an unsupervised learning method whose (hidden) layers contain representations of the input data sufficiently powerful for compressing (and decompressing) the data while loosing as few information as possible. Given the temporal nature of your data, our pattern discovery task is fundamentally convolutional (the same network is applied at any instant and is thus time-shift invariant) since it needs to identify motifs whatever their time(s) of occurrence To tackle this task, we will thus focus on a particular type of autoencoders, the convolutional ones. However, while well adapted for discriminative tasks like classification [1], the patterns captured by (convolutional) autoencoders are not fully interpretable and often correlated.

In this paper, we address the discovery of interpretable motifs using convolutional auto-encoders and make the following contributions:

- we show that the interpretability of standard convolutional autoencoders is limited;
- we introduce an adaptive rectified linear unit (AdaReLU) which allows hidden layers to capture clear occurrences of motifs,
- we propose a regularization inspired by group-lasso to automatically select the number of filters in a convolutional neural net,
- we show, through experiments on synthetic and real data, how these elements (and others) allow to recover interpretable motifs[3].

It is important to note that some previous *generative models* [21,6] have obtained very good results on this task. However, their extensions to semi-supervised settings (i.e. with partially labelled data) or hierarchical schemes are cumbursome to achieve. In contrast, in this paper, to solve the same modeling problem we present a radically different method which will lend itself to more flexible and systematic end-to-end training frameworks and extensions.

The paper is organized as follows. In Section 2, we clarify the link between our data mining technique and previous work. Section 3 gives the details of our method while Section 4 shows experiments both on synthetic and real data. We conclude and draw future directions in Section 5.

2 Related Work

Our paper shows how to use a popular method (autoencoders) to tackle a task (pattern discovery in time series) that has seldom been considered for this type of method. We thus briefly review other methods used in this context and then, other works that use neural networks for unsupervised time series modeling.

Unsupervised pattern discovery in time series. Traditional unsupervised approaches that deal with time series do not aim at modeling series but rather at extracting interesting pieces of the series that can be used as high level descriptions for direct analysis or as input features for other algorithms. In this

[3] The complete source code will be made available online

category fall all the event-based (e.g. [23,22,7]), sequence [15] and trajectory mining methods [25]. On the contrary of the previously cited methods, we do not know in advance the occurrence time, type, length or number of (possibly) overlapping patterns that can be used to describe the entire multivariate time series. These methods cannot be directly used in our application context.

The *generative methods* for modeling time series assume an apriori model and estimate its parameters. In the precursor work of [16], the unsupervised problem of finding patterns was decomposed into two steps, a supervised step involving an oracle who identifies patterns and series containing such patterns and an EM-step where a model of the series is generated according to those patterns. In [13], the authors propose a functional independent component analysis method for finding linearly varying patterns of activation in the data. They assume the availability of pre-segmented data where the occurrence time of each possible pattern is known in advance. Authors of [10] address the discovery of overlapping patterns to disaggregate the energy level of electric consumption. They propose to use additive factorial hidden Markov models, assuming that the electrical signal is univariate and that the known devices (each one represented by one HMM) have a finite known number of states. This also imposes that the motif occurrences of one particular device can not overlap. The work of [6] proposes to extract an apriori unknown number of patterns and their possibly overlapping occurrences in documents using Dirichlet processes. The model automatically finds the number of patterns, their length and occurrence times by fitting infinite mixtures of categorical distributions to the data. This approach achieved very good results, but its extensions to semi-supervised settings [19] or hierarchical schemes [2] were either not so effective [19] or more cumbersome [2]. In contrast, the neural network approach of this paper will lend itself to more flexible and systematic end-to-end training frameworks and extensions.

Networks for time series mining. A recent survey [11] reviews the network-based unsupervised feature learning methods for time series modeling. As explained in Sec. 1, autoencoders [17] and also Restricted Boltzmann Machines (RBM) [8] are neural networks designed to be trained from unsupervised data. The two types of networks can achieve similar goals but differ in the objective function and related optimization algorithms. Both methods were extended to handle time series [14,1], but the goal was to minimize a reconstruction error without taking care of the interpretability or of finding the relevant number of patterns. In this paper, we show that convolutional autoencoders can indeed capture the spatio-temporal structure in temporal documents. We build on the above works and propose a model to discover the right number of meaningful patterns in the convolution filters, and to generate sparse activations.

3 Motif Mining with Convolutional Autoencoders (AE)

Convolutional AEs [12] are particular AEs whose connection weights are constrained to be convolution kernels. In practice, this means that most of the

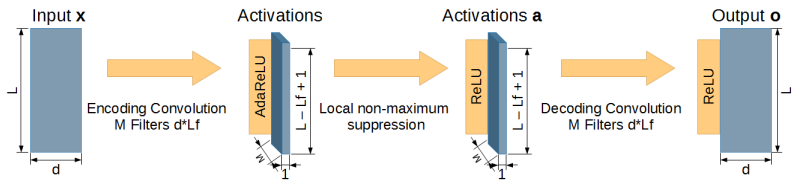

Fig. 1: Autoencoder architecture. Temporal documents of L time steps of d dimensional observations are encoded (here using M convolutional filters of size $d \times L_f$ forming the ${}^{\mathbf{e}}\mathbf{W}$ weights) to produce an activation layer. A decoding process (symmetric to encoding; parameterized by the weights ${}^{\mathbf{d}}\mathbf{W}$ of M decoding convolutional filters of size $d \times L_f$) regenerates the data.

learned parameters are shared within the network and that the weight matrices which store the convolution filters can be directly interpreted and visualized. Below, we first present the traditional AE model and then introduce our contributions to enforce at the same time a good interpretability of the convolutional filters and a clean and sparse activation of these filters.

3.1 Classical Convolutional Autoencoders

A main difference between an AE and a standard neural network is the loss function used to train the network. In an AE, the loss does not depend on labels, it is the reconstruction error between the input data and the network output. Fig. 1 illustrates the main network modeling components of our model. In our case, a training example is a multivariate time series $\mathbf{x}$ whose L time steps are described by a vector $\mathbf{x}_{(:,t)} \in \mathcal{R}^d$, and the network is parameterized by the set of weights $\mathbf{W} = \{{}^{\mathbf{e}}\mathbf{W}, {}^{\mathbf{d}}\mathbf{W}\}$ involved in the coding and decoding processes. If we denote by $\mathbf{X} = \{\mathbf{x}^b \in \mathcal{R}^{L \times d},\ b = 1 \ldots N\}$ the set of all training elements, the estimation of these weights is classically conducted by optimizing the cost function $C(\mathbf{W}, \mathbf{X}) = MSE(\mathbf{W}, \mathbf{X}) + R_{reg}(\mathbf{W}, \mathbf{X})$ where the Mean Squared Error (MSE) reconstruction loss can be written as:

$$MSE(\mathbf{W}, \mathbf{X}) = \frac{1}{N} \sum_{b=1}^{N} \sum_{i=1}^{d} \sum_{t=1}^{L} \left(\mathbf{x}^b_{(i,t)} - \mathbf{o}^b_{(i,t)} \right)^2 \tag{1}$$

where $\mathbf{o}^b$ (which depends on parameters $\mathbf{W}$) is the AE output of the bth input document. To avoid learning trivial and unstable mappings, a regularization term R_{reg} is often added to the MSE and usually comprises two terms. The first one, known as weight decay as it avoids unnecessary high weight values, is a ℓ_2 norm on the matrix weights. The second one (used with binary activations) consists of a Kullback-Leibler divergence $\sum_{j=1}^{M} KL(\rho || \hat{\rho}_j)$ encouraging *all* hidden activation units to have their probability of activation $\hat{\rho}_j$ estimated across samples to be close to a chosen parameter ρ, thus enforcing some activation sparsity when ρ is small. The parameters are typically learned using a stochastic gradient descent algorithm (SGD) with momentum using an appropriate rate scheduling [3].

3.2 Interpretable Pattern Discovery with Autoencoders

In our application, the learned convolution filters should not only minimize the reconstruction error but also be directly interpretable. Ideally, we would like to only extract filters which capture and represent interesting data patterns, as illustrated in Fig. 2-c-d. To achieve this, we add a number of elements in the network architecture and in our optimization cost function to constrain our network appropriately.

Enforcing non-negative decoding filters. As the AE output is somehow defined as a linear combination of the *decoding* filters, then these filters can represent the patterns we are looking for and we can interpret the hidden layers activations $\mathbf{a}$ (see Fig. 1) as the occurrences of these patterns. Thus, as our input is non-negative (a temporal document), we constraint the decoding filters weights to be non-negative by thresholding them at every SGD iteration. The assumption that the input is non-negative holds in our case and it will also hold in deeper AEs provided that we use ReLU-like activation functions. Note that for encoding, we do not constrain filters so they can have negative values to compensate for the pattern auto-correlation (see below).

Sparsifying the filters. The traditional ℓ_2 regularization allows many small but non-zero values. To force these values to zero and thus get sparser filters, we replaced the ℓ_2 norm by the sparsity-promoting norm ℓ_1 known as lasso:

$$R_{las}(\mathbf{W}) = \sum_{f=1}^{M}\sum_{i=1}^{d}\sum_{k=1}^{L_f}\left|{}^{\mathbf{e}}\mathbf{W}^{f}_{(i,k)}\right| + \sum_{f=1}^{M}\sum_{i=1}^{d}\sum_{k=1}^{L_f}\left|{}^{\mathbf{d}}\mathbf{W}^{f}_{(i,k)}\right| \tag{2}$$

Encouraging sparse activations. The traditional KL divergence aims at making *all* hidden units equally useful *on average*, whereas our goal is to have the activation layer to be as sparse as possible for each given input document. We achieve this by encouraging peaky activations, i.e. of low entropy when seen as a document-level probability distribution, as was proposed in [20] when dealing on topic models for motif discovery. This results in an entropy-based regularization expressed on the set $\mathbf{A} = \{\mathbf{a}^b\}$ of document-level activations:

$$R_{ent}(\mathbf{A}) = -\frac{1}{N}\sum_{b=1}^{N}\left(\sum_{f=1}^{M}\sum_{t=1}^{L-L_f+1}\hat{\mathbf{a}}^b_{f,t}\log\left(\hat{\mathbf{a}}^b_{f,t}\right)\right) \text{ with } \hat{\mathbf{a}}^b_{f,t} = \mathbf{a}^b_{f,t}\Big/\sum_{f=1}^{M}\sum_{t=1}^{L-L_f+1}\mathbf{a}^b_{f,t} \tag{3}$$

Local non-maximum activation removal. The previous entropy regularizer encourages peaked activations. However, as the encoding layer remains a convolutional layer, if a filter is correlated in time with itself or another filter, then the activations cannot be sparse. This phenomenon is due to the feed forward nature of the network, where activations depend on the input, not on each others: hence, no activation can inhibit its neighboring activations. To handle this issue we add a local non-maximum suppression layer which, from a network perspective, is obtained by convolving activations with a temporal Gaussian filter, subtracting from the result the activation intensities, and applying a ReLU, focusing in this way spread activations into central peaks.

Handling distant filter correlations with AdaReLU. The Gaussian layer cannot handle non local (in time) correlations. To handle this, we propose to replace the traditional ReLU activation function by a novel one called adaptive ReLU. AdaReLU works on groups of units and sets to 0 all the values that are below a percentage (e.g., 60%) of the maximal value in the group. In our architecture, AdaReLU is applied separately on each filter activation sequence.

Finding the true number of patterns. One main advantage and contribution of our AE-based method compared to methods presented in Section 2 is the possibility to discover the "true" number of patterns in the data. One solution to achieve this is to introduce in the network a large set of filters and "hope" that the learning leads to only a few non null filters capturing the interesting patterns. However, in practice, standard regularization terms and optimizations tend to produce networks "using" all or many more filters than the number of true patterns which results in partial and less interpretable patterns. To overcome this problem, we propose to use a group lasso regularization term called $\ell_{2,1}$ norm [24] that constrains the network to "use" as few filters as possible. It can be formulated for our weight matrix as:

$$R_{grp}(\mathbf{W}) = \sum_{f=1}^{M} \sqrt{\sum_{i=1}^{d} \sum_{k=1}^{L_f} \left({}^{\mathbf{e}}\mathbf{W}^{f}_{(i,k)} \right)^2} + \sum_{f=1}^{M} \sqrt{\sum_{i=1}^{d} \sum_{k=1}^{L_f} \left({}^{\mathbf{d}}\mathbf{W}^{f}_{(i,k)} \right)^2} \tag{4}$$

Overall objective function. Combining equations (1), (2), (4) and (3), we obtain the objective function that is optimized by our network:

$$C(\mathbf{W}, \mathbf{X}) = MSE(\mathbf{W}, \mathbf{X}) + \lambda_{las} R_{las}(\mathbf{W}) + \lambda_{grp} R_{grp}(\mathbf{W}) + \lambda_{ent} R_{ent}(\mathbf{A}(\mathbf{W}, \mathbf{X})) \tag{5}$$

4 Experiments

4.1 Experimental Setting

Datasets. To study the behavior of our approach, we experimented with both synthetic and real video datasets. The synthetic data were obtained using a known generation process: temporal documents were produced by sampling random observations of random linear combinations of motifs along with salt-and-pepper noise whose amount was defined as a percentage of the total document intensities (noise levels: 0%, 33%, 66%). Six motifs (defined as letter sequences for ease of visualization) were used. A document example is shown in Fig. 2-a, where the the feature dimension ($d = 25$) is represented vertically, and time horizontally ($L = 300$). For each experiments, 100 documents were generated using this process and used to train the autoencoders. This controlled environment allowed us to evaluate the importance of modeling elements. In particular, we are interested in i) the number of patterns discovered (defined as the non empty decoding filters[4]; ii) the "sharpness" of the activations; and iii) the robustness of

[4] We consider a filter *empty* if the sum of its weights is lower or equal to $\frac{1}{2}$ (the average sum value after initialization).

our method according to parameters like $\lambda_{lasso}, \lambda_{grp}, \lambda_{ent}$, the number of filters M, and the noise level.

We also applied our approach on videos recorded from fixed cameras. We used videos from the QMUL [9] and the far-field datasets [21]. The data pre-processing steps from the companion code of [6] were applied. Optical flow features were obtained by estimating, quantifying, and locally collecting optical flow over 1 second periods. Then, temporal documents were obtained by reducing the dimensionality of these to $d = 100$, and by cutting videos into temporal documents of size $L = 300$ time steps.

Architecture details and parameter setting. The proposed architecture is given in Fig. 1. As stated earlier, the goal of this paper is to make the most of a convolutional AE with a *single* layer (corresponding to the activation layer)[5]. Weights are initialized according to a uniform distribution between 0 and $\frac{1}{d*L_f}$.

In general, the filter length L_f should be large enough to capture the longest expected recurring pattern of interest in the data. The filter length has been set to $L_f = 45$ in synthetic experiments, which is beyond the longer motif of the ground-truth. In the video examples, we used $L_f = 11$, corresponding to 10 seconds, and which allows to capture the different traffic activities and phases of our data [21].

4.2 Results on the Synthetic Dataset

Since we know the "true" number of patterns and their expected visualization, we first validate our approach by showing (see Fig. 2-c) that we can find a set of parameters such that our filters exactly capture our given motifs and the number of non empty filters is exactly the "true" number of motifs in the dataset even when this dataset is noisy (this is also true for a clean dataset). In this case (see Fig.2-e) the activations for the complete document are, as expected, sparse and "peaky". The output document (see Fig.2-b) is a good un-noisy reconstruction of the input document shown in Fig.2-a.

In Fig. 3, we evaluate the influence of the given number of filters M and the noise level on both the number of recovered motifs an the MSE while fixing the parameters as in Fig. 2. We can see that with this set of parameters, the AE is able to recover the true number of filters for the large majority of noise levels and values of M. For all noise levels, we see from the low MSE that the AEs is able to well reconstruct the original document as long as the number of given filters is at least equal to the number of "true" patterns in the document.

Model selection: influence of $\boldsymbol{\lambda_{grp}}$. Fig.4 shows the number of non zero filters in function of λ_{grp} and of the noise level for the synthetic dataset with 6 known motifs when using 12 filters (left) and 16 filters (right). The light blue area is the area in which the AEs was able to discover the true number of patterns.

[5] Note however that the method can be generalized to hierarchical motifs using more layers, but then the interpretation of results would slightly differ.

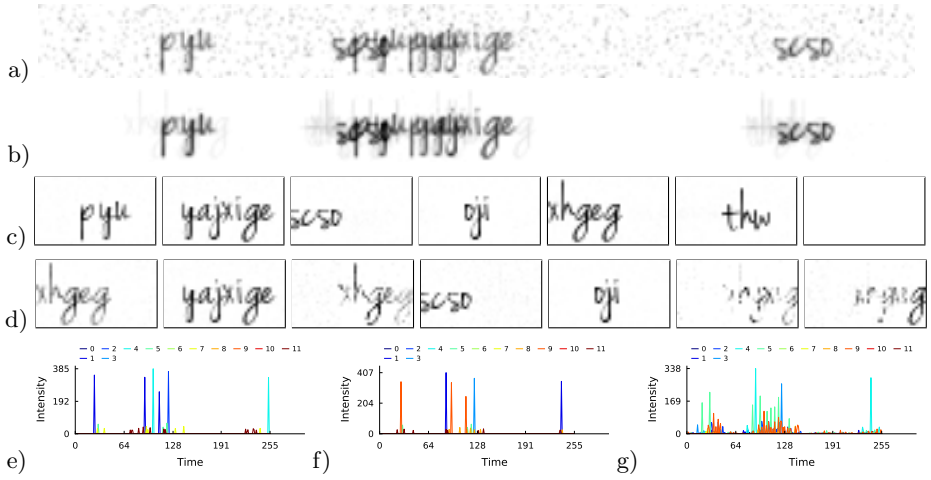

Fig. 2: Results on the synthetic data built from 6 patterns, with 66% of noise, $M = 12$ filters, and unless stated otherwise $\lambda_{las} = 0.004$, $\lambda_{grp} = 2$, $\lambda_{ent} = 0.2$. a) Sample document; b) Obtained output reconstructed document; c) Weights of seven out of the 12 obtained filters (the 5 remaining filters are empty); d) Weights of seven filters when not using group lasso, i.e. with $\lambda_{grp} = 0$ (note that the 5 remaining filters are non empty); (e,f,g) Examples of activation intensities (colors correspond to a given filter) with default parameters (e); without the entropy sparsifying term ($\lambda_{ent} = 0$) (f); with ReLU instead of AdaReLU (g).

With no group lasso regularization ($\lambda_{grp} = 0$), the AE systematically uses all the available filters capturing the original patterns (see $2^{nd}, 4^{th}$ or 5th filters in Fig. 2-d), redundant variants of the same pattern (filters 1st and 3rd in Fig. 2-d) or a more difficult to interpret mix of the patterns (filters 6th and 7th in Fig. 2-d). On the contrary, with too high values of λ_{grp}, the AE does not find any patterns (resulting in a high MSE). A good heuristic to set the value of λ_{grp} could thus be to increase it as much as possible until the resulting MSE starts increasing. In the rest of the experiments, λ_{grp} is set equal to 2.

Influence of λ_{ent}, λ_{lasso}, AdaReLU, and Non-Local Maxima suppression. We have conducted the same experiments as in Fig. 2 on clean and noisy datasets (up to 66% of noise) with M =3, M =6 M =12 to assess the behavior of our system when canceling the parameters: 1) λ_{ent} that controls the entropy of the activation layer, 2) λ_{las}, the lasso regularizer 3) the AdaReLU function (we used a simple ReLU in the encoding layer instead) and 4) the Non-Local Maxima activation suppression layer. In all cases, all parameters but one were fixed according to the best set of values given in Fig.2. For lack of space, we do not give all the corresponding figures but we comment the main results.

The λ_{ent} is particularly important in the presence of noise. Without noise and when this parameter is set to 0, the patterns are less sharp and smooth and the activations are more spread along time with much smaller intensities.

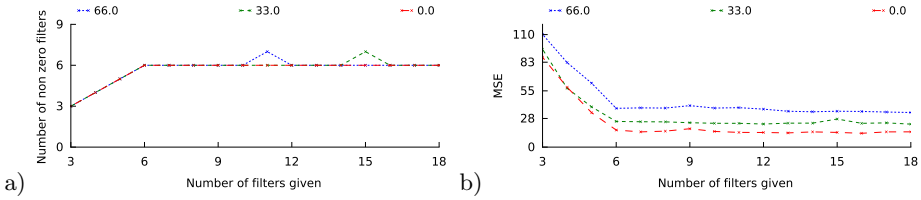

Fig. 3: Influence of the given number of filters M and the noise level (0%, 33% and 66%) on: a) the number of recovered motifs and b) the Mean Squared Error. Experiments on the synthetic dataset with $\lambda_{las} = 0.004$, $\lambda_{grp} = 2$, $\lambda_{ent} = 0.2$.

Fig. 4: Evolution of the number of non-zero filters (sparsity) with respect to the noise level when we vary the parameter λ_{grp} (λ_{glasso} in the figure) that controls the group lasso regularization for the synthetic dataset with 6 known motifs when using 12 filters (right) and 16 filters (left).

However, the MSE is as low as for the default parameters. In the presence of noise (see Fig.2-f), the AE is more likely to miss the recovery of some patterns even when the optimal number of filters is given (e.g. in some experiments only 5 out of the 6 filters were not empty) and the MSE increases a lot compared to experiments on clean data. This shows again that the MSE can be a good heuristic to tune the parameters on real data. The λ_{las} has similar effects with and without noise: it helps removing all the small activation values resulting in much sharper (and thus interpretable) patterns.

The non-local maximum suppression layer (comprising the *Gaussian* filter) is compulsory in our proposed architecture. Indeed, without it, the system was not able to recover any patterns when M =3 (and only one blurry "false" pattern in the presence of noise). When M =6, it only captured 4 patterns (out of 6) in the clean dataset and did not find any in the noisy ones. When M =12, it was able to recover the 6 original true patterns in the clean dataset but only one blurry "false" pattern in the noisy ones.

The AdaReLU function also plays an important role to recover interpretable patterns. Without it (using ReLU instead) the patterns recognized are not the "true" patterns, they have a very low intensity and are highly auto-correlated (as illustrated by the activations in Fig.2-g).

4.3 Results on the Real Video Dataset

Due to space limitations, we only show in Fig. 5 some of the obtained results. The parameters were selected using grid search by minimizing the MSE on the

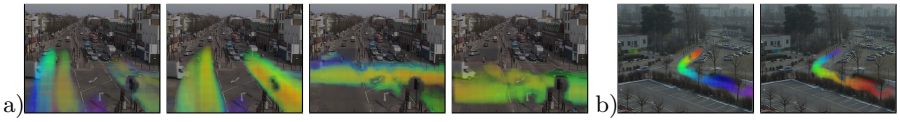

Fig. 5: Traffic patterns. M =10 filters. a) The four motifs recovered on the Junction 1 dataset, (6 empty ones are not shown). b) Two filters (out of the five recovered) on the far-field dataset.

targeted dataset. For instance, on the Junction 1 dataset, the final parameters used are $\lambda_{las} = 0.2$, $\lambda_{grp} = 50$, $\lambda_{ent} = 5$. Note that this is larger than in the synthetic case but the observation size is also much larger (100 vs 25) and the filters are thus sparser in general. In the Junction 1 dataset, the autoencoder recovers 4 non-empty and meaningful filters capturing the car activities related to the different traffic signal cycles, whereas in the far-field case, the main trajectories of cars were recovered as also reported in [21].

5 Conclusion

We have shown that convolutional AEs are good candidate unsupervised data mining tools to discover interpretable patterns in time series. We have introduced a number of layers and regularization terms to the standard convolutional AEs to enforce the interpretability of both the convolutional filters and the activations in the hidden layers of the network. The filters are directly interpretable as spatio-temporal patterns while the activations give the occurrence times of each patterns in the temporal document. This allow us to un-mix multivariate time series. A direct perspective of this work is the use of multi-layer AEs to capture combination of motifs. If this was not the aim of this article, it may help to reduce the number of parameters needed to obtain truly interpretable patterns and capture more complex patterns in data.

Acknowledgement

This work has been supported by the ANR project SoLStiCe (ANR-13-BS02-0002-01).

References

1. M. Baccouche, F. Mamalet, C. Wolf, C. Garcia, and A. Baskurt. Spatio-Temporal Convolutional Sparse Auto-Encoder for Sequence Classification. In *British Machine Vision Conference (BMVC)*, 2012.
2. T. Chockalingam, R. Emonet, and J-M. Odobez. Localized anomaly detection via hierarchical integrated activity discovery. In *AVSS*, 2013.
3. C. Darken and J. E. Moody. Note on learning rate schedules for stochastic optimization. In *NIPS*, pages 832–838, 1990.
4. X. Du, R. Jin, l. Ding, V.E. Lee, and J.H. Thornton. Migration motif: a spatial - temporal pattern mining approach for financial markets. In *KDD*, pages 1135–1144. ACM, 2009.

5. R. Emonet, J. Varadarajan, and J.-M. Odobez. Multi-camera open space human activity discovery for anomaly detection. In *IEEE Int. Conf. on Advanced Video and Signal-Based Surveillance (AVSS), Klagenfurt, Austria*, aug 2011.
6. R. Emonet, J. Varadarajan, and J-M. Odobez. Temporal analysis of motif mixtures using dirichlet processes. *IEEE PAMI*, 2014.
7. M. Marwah H. Shao and N. Ramakrishnan. A temporal motif mining approach to unsupervised energy disaggregation: Applications to residential and commercial buildings. In *Proceedings of the 27th AAAI conference*, 2013.
8. G E Hinton and R R Salakhutdinov. *Science*, 313(5786):504–507, July 2006.
9. T. Hospedales, S. Gong, and T. Xiang. A markov clustering topic model for mining behavior in video. In *ICCV*, 2009.
10. J. Z. Kolter and T. Jaakkola. Approximate inference in additive factorial hmms with application to energy disaggregation. In *Proc. of AISTATS Conf.*, 2012.
11. L. Karlsson M. Längkvist and A. Loutfi. A review of unsupervised feature learning and deep learning for time-series modeling. *Pattern Recognition Letters*, 2014.
12. J. Masci, U. Meier, D. Cireşan, and J. Schmidhuber. Stacked convolutional auto-encoders for hierarchical feature extraction. In *Proc. of ICANN*, 2011.
13. N. A. Mehta and A. G. Gray. Funcica for time series pattern discovery. In *Proceedings of the SIAM International Conference on Data Mining*, pages 73–84, 2009.
14. R. Memisevic and G. E. Hinton. Unsupervised learning of image transformations. In *Computer Vision and PatternRecognition (CVPR)*, 2007.
15. C. H. Mooney and J. F. Roddick. Sequential pattern mining – approaches and algorithms. *ACM Comput. Surv.*, 45(2):19:1–19:39, March 2013.
16. Tim Oates. Peruse: An unsupervised algorithm for finding recurring patterns in time series. In *ICDM*, 2002.
17. M. Ranzato, C. Poultney, S. Chopra, and Y. LeCun. Efficient learning of sparse representations with an energy-based model. In *NIPS*. MIT Press, 2006.
18. A. Sallaberry, N. Pecheur, S. Bringay, M. Roche, and M. Teisseire. Sequential patterns mining and gene sequence visualization to discover novelty from microarray data. *Journal of Biomedical Informatics*, 44(5):760 – 774, 2011.
19. R. Tavenard, R. Emonet, and J-M. Odobez. Time-sensitive topic models for action recognition in videos. In *Int. Conf. on Image Processing (ICIP)*, Mebourne, 2013.
20. J. Varadarajan, R. Emonet, and J.-M. Odobez. A sparsity constraint for topic models - application to temporal activity mining. In *NIPS Workshop on Practical Applications of Sparse Modeling: Open Issues and New Directions*, 2010.
21. J. Varadarajan, R. Emonet, and J-M. Odobez. A sequential topic model for mining recurrent activities from long term video logs. *Int. jl of computer vision*, 2013.
22. F. Zhou W.-S. Chu and F. De la Torre. Unsupervised temporal commonality discovery. In *Computer Vision – ECCV 2012*, 2012.
23. W.-C. Peng Y.-C. Chen and S.-Y. Lee. Mining temporal patterns in time interval-based data. *IEEE Transactions on Knowledge and Data Engineering*, 2015.
24. M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society (B)*, 68(1):49–67, 2006.
25. Y. Zheng. Trajectory data mining: An overview. *ACM Trans. Intell. Syst. Technol.*, 6(3):29:1–29:41, May 2015.