Routine Modeling with Time Series Metric Learning

Paul Compagnon, Grégoire Lefebvre, Stefan Duffner, Christophe Garcia

Routine Modeling with Time Series Metric Learning

Paul Compagnon[1,2], Grégoire Lefebvre[1],
Stefan Duffner[2] and Christophe Garcia[2]

[1] Orange Labs, Grenoble, France
{paul.compagnon, gregoire.lefebvre}@orange.com
[2] LIRIS, UMR 5205 CNRS INSA-Lyon, France
{stefan.duffner, christophe.garcia}@liris.cnrs.fr

Abstract. Traditionally, the automatic recognition of human activities is performed with supervised learning algorithms on limited sets of specific activities. This work proposes to recognize recurrent activity patterns, called routines, instead of precisely defined activities. The modeling of routines is defined as a metric learning problem, and an architecture, called SS2S, based on sequence-to-sequence models is proposed to learn a distance between time series. This approach only relies on inertial data and is thus non intrusive and preserves privacy. Experimental results show that a clustering algorithm provided with the learned distance is able to recover daily routines.

Keywords: Metric Learning, Sequence-to-Sequence Model, Activity Recognition, Time Series, Inertial data

1 Introduction

Human Activity Recognition (HAR) is a key part of several intelligent systems interacting with humans: smart home services [10], actigraphy and telemedecine, sport applications [3], etc. It is particularly useful for developing eHealth services and monitoring a person in its everyday life. It has been so far mainly performed in supervised contexts with data annotated by experts or with the help of video recordings [8]. Not only is this approach time consuming, but it also restricts the number of activities that can be recognized. It is associated with scripted datasets where subjects are asked to perform sequences of predefined tasks. This approach is thus unrealistic and difficult to set up for real environments where people do a vast variety of specific activities everyday and can diverge from a pre-established behavior in many different ways (e.g., falls, accidents, contingencies of life, etc.). Besides, most people present some kind of habitual behavior, called *routines* in this paper: the time they go to sleep, morning ritual before going to work, meal times, etc. Results from behavioral psychology show that habits are hard and long to form but also hard to break when well installed [20]. From a data-driven perspective, Gonzalez et al. [14] observed the high regularity of human trajectories thanks to localization data and show that "humans follow

simple reproducible patterns". Routines produce distinguishable patterns in the data which, if not identifiable semantically, could be retrieved over time and so produce a relevant signature of the daily life of a person. In this paper, we advocate for the modeling of such routines instead of activity recognition, and we propose a machine learning model able to identify routines in the daily life of a person. We want this system to be unintrusive and to respect people's privacy and therefore to rely only on inertial data that can be gathered by a mobile phone or a smart watch. Moreover, routines do not need to be semantically characterized, and the model does not have to use any activity labels. The daily routines of a person may present characteristics of almost-periodic functions, periodic similarity, regarding a certain metric which we propose to learn. To do so, we adapted the siamese neural network architecture proposed by Bromley et al. [7] to learn a distance from pairs of sequences and propose experiments to evaluate the quality of the learned metric on the problem of routine modeling. The contributions of this paper are threefold:

1. a formulation of routine modeling as a metric learning problem by defining routines as almost-periodic functions,
2. an architecture to jointly learn a representation and a metric for time series using siamese sequence-to-sequence models and an improvement of the loss functions to minimize,
3. results showing that the proposed architecture is effectively able to recover human routines from inertial data without using any activity labels.

The remainder of the paper is organized as follows. Section 2 is dedicated to routine modeling definition. Section 3 gives an overview of time series metrics. The proposed approach to recognize routines is presented in Section 4 and Section 5 presents experimental protocols and results. Finally, conclusions and perspectives are drawn in the last section.

2 Routine Modeling

A routine can be seen as a recurrent behavior of an individual's daily life. For example, a person roughly does the same thing in the same order when waking up or going to work. These sequences of activities should produce distinguishable patterns in the data and can thus be used to monitor the life of an individual without knowing what he or she is doing exactly. The purpose of this work is to design an intelligent system which is able to recognize routines. To tackle routines with machine learning, we propose a starting principle similar to the one used in natural language processing: *similar words appear in similar contexts*. The context surrounding a word designates the previous and following words of the sentence, for example. The context of a routine corresponds here to the moment of the day or the week, etc. it generally happens.

Principle 1. *Similar routines occur at similar moments, almost periodically.*

From this principle, we seek now to propose a mathematical formulation of routines which would include the notions of periodicity and similarity. The almost periodic functions defined by Bohr [6] show similar properties:

Definition 1. *Let $f : \mathbb{R} \to \mathbb{C}$ be a continuous function. f is an almost-periodic function with respect to the uniform norm if $\forall \epsilon > 0$, $\exists T > 0$ called an ϵ-almost period of f such as:*

$$sup\ |f(t+T) - f(t)| \leqslant \epsilon. \tag{1}$$

Obviously, the practical issue of routine modeling presents several divergences from this canonical definition: data are discrete time series and the periodicity of activities cannot be evaluated point-wise. Nevertheless, it is possible to adapt it to our problem. Let $S : \mathbb{N} \to \mathbb{R}^n$ be an ordered discrete sequence of vectors of dimension n. If the frequency of S is sufficiently high, it is possible to get a continuous approximation of it, by interpolation for example. We now consider a function f_S of the following form with a fixed interval length l:

$$\begin{aligned} f_S :\ \mathbb{R}_+ &\to \mathbb{R}^{n \times l} \\ t &\mapsto [S(t) : S(t+l)[, \end{aligned} \tag{2}$$

where $[S(t) : S(t+l)[$ is the set of vectors between $S(t)$ and $S(t+l)$ sampled at a certain frequency from the continuous approximation. l is typically one or several hours: a sufficiently long period of time to absorb the little changes from one day to another (e.g., waking up a little earlier or later, etc.). The objective is to define almost-periodicity with respect to a distance d between sequences, such that $\forall \epsilon > 0$, $\exists T > 0$:

$$d(f_S(t), f_S(t+T)) \leqslant \epsilon. \tag{3}$$

The parameter T can be a day, a week or a sufficiently long period of time to observe repetitions of behavior. The metric d must be sufficiently flexible to handle the high variability of activities which can be similar but somewhat different in their execution while exhibiting a similar pattern. We therefore postulate that d may be learned for a specific user from its data and we will now show that f_S respects the condition established in Eq. (3) with respect to d. To learn d if pairs of similar and dissimilar sequences are known, a Recurrent Neural Network (RNN) encoder parametrized by W, called G_W, can encode the sequences into vector representations and the contrastive loss [15] can be used to learn the metric from pairs of sequence encodings:

$$L(W, Y_1, Y_2, y) = (1-y)\frac{1}{2}d(Y_1, Y_2)^2 + y\frac{1}{2}\max(0, m - d(Y_1, Y_2))^2, \tag{4}$$

where y is equal to zero or one depending if the sequences are respectively similar or not, Y_1 and Y_2 are the last output of the RNN for both sequences and $m > 0$ a margin that defines the minimal distance between dissimilar samples. Several justifications arise for the use of a margin in metric learning. It is necessary to prevent flat energy surface, according to energy-based learning theory [21], a

situation where the energy is low for every input/output associations, not only those in the training set. It also insures that metric learning models are robust to noise [29]. As the learning process aims to minimize the distances between similar sequences which are, by definition, shifted by a period T, we get, for a fixed $T > 0$ and $\forall t \in \mathbb{R}_+$:

$$d(G_W(f_S(t)), G_W(f_S(t+T))) \leqslant m. \tag{5}$$

The margin m can be chosen as close to zero as possible and thus Eq. (5) identifies itself with Eq. (3). In practice, this optimization is only possible up to some point, depending on the model and the data. This argumentation suggests the interest of modeling routines with metric learning as, in this case, the main property of almost-periodic functions is fulfilled.

3 Related Work

The traditional approach to compute distances between sequences (or time series, or trajectories) is to perform Dynamic Time Warping (DTW) [25] which was introduced in 1978. Since then, several improvements of the algorithm have been published, notably a fast version by Salvador et al. [26]. DTW is considered one of the best metric to use for sequence classification [31] combined with k-nearest neighbors. Recently, Abid et al. [1] proposed a neural network architecture to learn the parameters of a warping distance accordingly to the euclidean distances in a projection space. However, DTW, as other shaped-based distances [11], is only able to retrieve local similarities when time series have a relatively small length and are just shifted or not well aligned.

Similar routines could present different data profiles which would necessitate a more complex and global notion of similarity. This justifies the extraction of high-level features to produce a vector representation of the structure and the semantics of the data [22]. Traditional metrics can be used to compare vector representations: Euclidean, cosine or Mahalanobis. These vectors can be build with features extracted by various methods such as discrete Fourier and Wavelet transforms, signal processing, singular value decomposition or Hidden Markov Models (HMM) [2]. HMM belong to a category of approaches which suppose the existence of an underlying model which has produced the data; other examples include AutoRegressive-Moving-Average (ARMA) or multivariate extensions (VARIMA), Markov chains, etc. In this case, similarity can be assess by comparing model parameters. More theoretical approaches based on the study of the spectral properties of these models have also been proposed in [18, 23]. The problem with these approaches is that it is difficult to select relevant features and/or to chose an accurate model and parameters for a given task. It would be better if an appropriate representation of the data could automatically be extracted accordingly to the problem, by a Neural Network (NN) for example.

Besides, Bromley et al. [7] proposed a Siamese Neural Network (SNN) architecture to learn a metric. They have since then been used for many applications

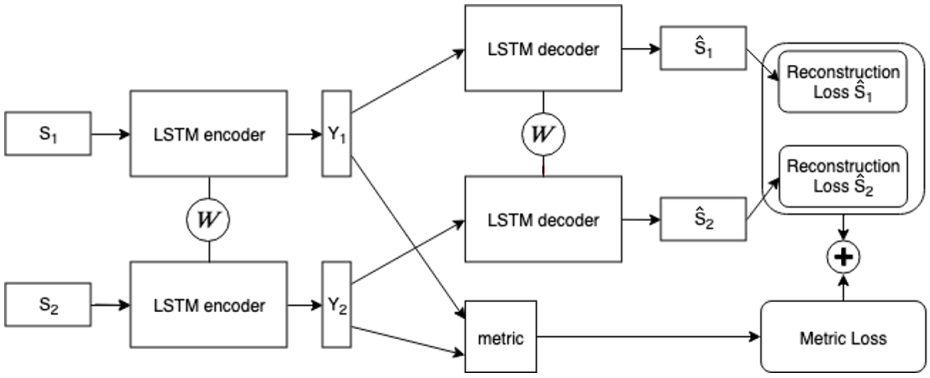


Fig. 1: Proposed SS2S architecture.

with feedforward or convolutionnal NN such as person reidentification [32], gesture recognition [4], object tracking [5], etc. RNN and particularly Long-Short Term Memory (LSTM) NN [16] are well-adapted to work with long sequential data as they are able to deal with long-term dependencies. Müller et al. [24] used a siamese recurrent architecture to learn sentence similarity by encoding sequence of word vectors previously extracted belonging to the same sentence.

In the following section, we propose a novel Siamese Sequence to Sequence (SS2S) neural network architecture to learn to model routines without label supervision. The model effectively combines automatic feature extraction and a similarity metric by jointly learning a robust projection of time series in a metric space. This approach is able to deal with long sequences by using LSTM networks and do not necessitate to choose a model to fit or features to extract.

4 Siamese Sequence to Sequence Model

4.1 Feature Extraction Approach

The time series data obtained from inertial sensors may be very noisy and certainly vary for the same general activity (e.g., cooking). Robust feature representations of time series should therefore be learned before learning a metric. We thus propose (Fig. 1) to map each sequence to a vector using a Sequence to Sequence model [1, 9, 27]. The sequence is given as input to the first LSTM network (the encoder) to produce an output sequence, the last output vector is considered as the learned representation. This representation is then given to the second LSTM (the decoder) which tries to reconstruct the input sequence. Typically, an autoencoder is trained to reconstruct the original sequence with the Mean Squared Error (MSE):

$$\mathrm{MSE}(S, \hat{S}) = \frac{1}{l} \sum_{t=0}^{l-1} (S(t) - \hat{S}(t))^2, \tag{6}$$

where S is the sequence and $\hat{S}$ the output sequence produced by the autoencoder from the vector. Similarly, we propose a new Reconstruction Loss (RL) based on cosine similarity, the Cosine Reconstruction Loss (CRL):

$$\mathrm{CRL}(S, \hat{S}) = l - \sum_{t=0}^{l-1} \cos(S(t), \hat{S}(t)). \tag{7}$$

CRL is close to 0 if the cosine similarity between each pair of vectors is close to one when the vectors are collinear.

4.2 Metric Learning

Our architecture is a siamese network [7], that is to say it is constituted of two subnetworks sharing the same parameters W (see Fig. 1). It takes pairs of similar or dissimilar sequences as input constituted with what is called *equivalence constraints*. The objective of our architecture is therefore to learn a metric which makes close similar elements and separates the dissimilar ones in the projection space. Three metric forms can generally be used: Euclidean, cosine or Mahalanobis [15, 32, 12]. The first two are not parametric and only a projection is learned. Learning a Mahalanobis-like metric implies not only learning the projection but also the matrix which will be used to compute the metric. One different Metric Loss (MeL) is proposed to learn each metric form. Y_1 and Y_2 are the representations learned by the autoencoder from the inputs of the siamese network. The first is the contrastive loss [15] (see Eq. (4)) to learn an euclidean distance. The second is a cosine loss to learn a cosine distance:

$$L(W, Y_1, Y_2, y) = \begin{cases} 1 - \cos(Y_1, Y_2), & \text{if } y \ \ = 1 \\ \max(0, \cos(Y_1, Y_2) - m), & \text{if } y = -1. \end{cases} \tag{8}$$

Finally, Mahalanobis metric learning can be performed with the KISSME algorithm [19] which can be integrated into a NN [12]. This algorithm aims to maximize the dissimilarity log-likelihood of dissimilar pairs and conversely for similar pairs. The model learns a mapping under the form of a matrix W and an associated metric matrix M of the dimension of the projection space. W is integrated into the network as a linear layer (just after the recurrent encoding layers in SS2S) trained with backpropagation while M is learned in a closed-form manner and updated after a fix number of epochs with the following formula:

$$M = \mathrm{Proj}((W^T \Sigma_S W)^{-1} - (W^T \Sigma_D W)^{-1}). \tag{9}$$

Σ_S and Σ_D are the covariance matrices of similar and dissimilar elements in the projection space and Proj is the projection onto the positive semi definite cone. We propose a modified version of the KISSME loss proposed in [12] which we found was easier to train based on the contrastive loss (Eq. (4)):

$$\begin{aligned} L(W, Y_1, Y_2, y) =& (1 - y)\frac{1}{2}(Y_1 - Y_2)M(Y_1 - Y_2)^T \\ &+ y\frac{1}{2}\max(0, m - (Y_1 - Y_2)M(Y_1 - Y_2)^T). \end{aligned} \tag{10}$$

4.3 Training Process

Two training processes can be considered for this architecture. Train the autoencoder and then "freeze" the network parameters to learn the metric if it is parametric. Or, add the metric loss to the reconstruction loss and learn jointly both tasks. In this case, several difficulties could appear. Both losses must have similar magnitudes to have similar influences on the training process. The interaction between the two must also be considered. Both tasks could have eventually divergent or not completely compatible objectives. Indeed, we proposed the CRL with the *a priori* that it should better interact with the learning of a cosine metric than MSE due to the similar form between the two. This leads to our first hypothesis (H1):

Hypothesis 1. *Learning a cosine distance along a representation with CRL gives better results than with MSE.*

Despite the possible issues, we hope that learning both tasks jointly should lead to the learning of more appropriate representations and thus to better results. This leads to our second hypothesis (H2):

Hypothesis 2. *Jointly learning a metric and a representation with a sequence to sequence model gives better results than learning both separately.*

5 Experiments

5.1 Experimental Setup

Dataset Presentation. Long-term unscripted data from wearable sensors are difficult to gather. The only dataset we found that could fit our requirements has been obtained by Weiss et al. [30] and is called Long Term Movement Monitoring dataset (LTMM)[1]. This dataset contains recordings of 71 elderly people which have worn an accelerometer and a gyroscope during three days with no instructions. This dataset contains no labels. Fig 2.a presents two days of data coming from one axis of the accelerometer: similar profiles can be observed at similar moment. Fig 2.b presents the autocorrelation of the accelerometer signal: the maximum of 0.4 is reached for a phase of 24h. These figures show the interest of this dataset as the data show periodic nature while presenting major visual differences. That said, the definition of periodicity that our algorithm is made to achieve is stronger as it is based on a metric between extracted feature vectors, not just correlations of signal measurements.
To constitute our dataset, we selected in the original dataset a user who did not remove the sensor during the three days to avoid missing values. We set up a data augmentation process to artificially increase the quantity of data while preserving its characteristic structure. The dataset is sampled at 100 Hz and thus, to multiply the number of days by ten, each vector measurement at the same index modulo 10 will be affected to a new day (the order is respected). This

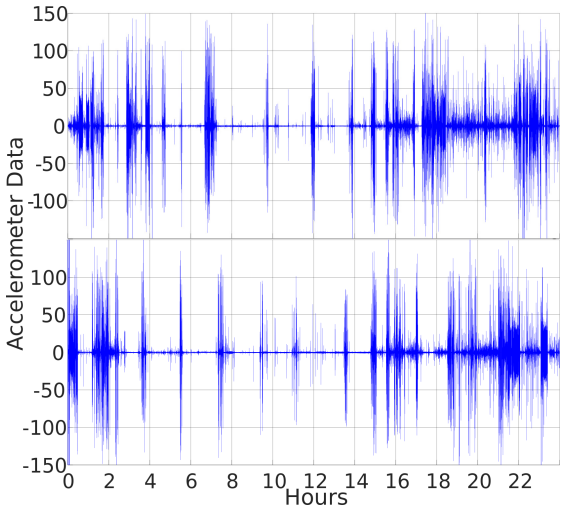
[1] `https://www.physionet.org/physiobank/database/ltmm`

new dataset has a sampling rate of 10 Hz which means that one hour of data is a sequence of size 36000, we consider only non overlapping sequences. Thus, to make the computation more tractable, polyphase filtering is applied to resample each sequence of one hour to a size of 100. Finally, equivalence constraints need to be defined in order to make similar and dissimilar pairs: two sequences of one hour, not from the same day but recorded at the same time are considered similar, all other combinations are considered dissimilar. This approach does not therefore require semantic labels.

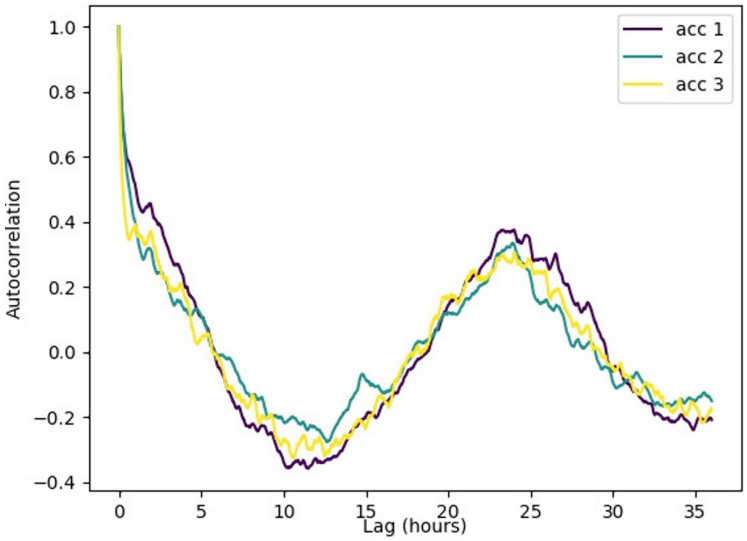
Model Parameters and Training Details. We describe here the hyperparameters used to train the models. The autoencoders are constituted of one layer of 100 LSTM neurons for the encoder and the decoder. For the KISSME version, the encodings are then projected into a 50-dimensional space, and the distance matrix, which thus has also dimension 50, was updated with the closed-form every 30 epochs. These parameters were determined after preliminary tests where deeper architectures and higher dimensional spaces were tested. Models are trained with 20 similar pairs for each time slot and the same total number of dissimilar pairs for a total of 960 training pairs coming from 12 different days of data. The training was stopped based on the loss computed on the validation set which contains three days of data i.e., 72 sequences. The testing set is composed of 15 days or 360 sequences. The data in the training set were rescaled between -1 and 1 and the same parameters were applied on the validation and testing sets. A learning rate of 0.001 was used and divided by 10 if the loss did not decrease anymore during 10 epochs. A batch size of 50, a margin of 1 for the contrastive loss and of 0.5 for the cosine loss were chosen. We also observed that changing to zero 30% of the values of the training sequences sliglty improved the results as suggested in [28].

5.2 Experimental Results and Discussion

Since the only available labels are time indications and to keep minimal supervision, the evaluation metrics rely on clustering. We report average values on 20 tests for 4 clustering evaluation metrics. *Completeness* assesses if sequences produced at the same hour are in the same clusters. *Silhouette* describes the cluster shapes, if they are dense and well-separated. *Normalized Mutual Information* (NMI) is a classical metric for clustering and measures how two clustering assigments concur, the second being the time slots. *Adjusted Mutual Information* (AMI) is the adjusted against chance version of NMI. A spectral clustering into 5 clusters is performed with the goal not to find the precise number of clusters maximizing the metrics but to choose a number which will make appear coherent and interpretable routines of the day, namely sleep moments, meals and other daily activities performed every day. Finally, to make our distances usable by the spectral clustering, they are converted to kernel functions. The following transformation was applied to the Euclidean, Mahalanobis and DTW distances: $\exp(-\text{dist} \cdot \gamma)$ where γ is the inverse of the length of an encoding vector (respec-



(a) 2 days of accelerometer data.



(b) Input signal autocorrelation for accelerometer data.

Fig. 2: LTMM dataset used to evaluate routine modeling procedure.

tively number of features time length of sequence for DTW). 1 was added to the cosine similarity so it becomes a kernel.

Evaluation of Cosine Reconstruction Loss. The performance of the CRL on LTMM is first evaluated. An experiment was performed by jointly training models for Euclidean or cosine distances with CRL or MSE. The results are re-

ported in Table 1. An asterisk means the average results are significantly higher according to a Welch's test. The results demonstrate a significant improvement of the proposed CRL over MSE when trained with the cosine similarity for Completeness, NMI and AMI. For the Silhouette score, better results are obtained with the MSE. However, the standard deviations are large, and this improvement is thus not significant. With the Euclidean distance, the same improvement is not realized with a slight advantage of MSE over CRL. These results confirm our hypothesis H1 that it is more appropriate to learn a cosine distance with CRL. They also suggest a positive interaction between the two as the same effect could not be observed with the Euclidean distance. We then use CRL in the remaining of the paper.

MeL \ RL	**CRL**	**MSE**
Cosine	**0.714*** ± 0.048	0.666 ± 0.066
Euclidean	0.609 ± 0.042	0.635 ± 0.064

(a) Completeness

MeL \ RL	**CRL**	**MSE**
Cosine	0.618 ± 0.105	**0.667** ± 0.144
Euclidean	0.402 ± 0.05	0.408 ± 0.042

(b) Silhouette

MeL \ RL	**CRL**	**MSE**
Cosine	**0.449*** ± 0.032	0.397 ± 0.040
Euclidean	0.419 ± 0.033	0.434 ± 0.047

(c) NMI

MeL \ RL	**CRL**	**MSE**
Cosine	**0.253*** ± 0.03	0.205 ± 0.033
Euclidean	0.255 ± 0.027	0.264 ± 0.038

(d) AMI

Table 1: Evaluations of CRL and MSE on LTMM dataset.

Evaluation of the SS2S Architecture. Next, we investigated the benefit of the SS2S architecture over DTW and Siamese LSTM (SLSTM) [24] as well as the interest of jointly learning the encoder-decoder and the metric on the LTMM dataset. Results are presented in Table 2. To test the DTW, the better radius was selected on the validation set and the spectral clustering was performed using DTW as kernel. Although Completeness, NMI and AMI are higher than every SS2S architectures except one, we observe a negative silhouette value which indicates a poor quality of the clustering and seems to confirm than indeed shaped based distances are not suitable for this type of data. Concerning the encoding architecture, SS2S gives overall better results than SLSTM and the best results are achieved by using the disjoint version of KISSME with a completeness of 0.983 and an NMI of 0.619. These results are not surprising as KISSME uses a parametric distance which can therefore be more adapted to the data. For the silhouette score, cosine distances performed best, i.e., they learned more compact and well-defined clusters. We also note that disjoint versions of the architectures performed better than the joint versions, thus invalidating our

hypothesis H2.

To investigate the reasons of this difference which could be due to the autoencoder not being learned properly, Table 3 reports average best Reconstruction Errors on Validation set (REV). The lowest errors are systematically achieved when the encoder is learned alone before the metric therefore supporting the hypothesis that learning the metric prevents the autoencoder from being trained at its full potential. It explains why the joint learning does not perform best. For the CRL, results are closer than for MSE suggesting why this reconstruction loss is easier to learn jointly.

Finally, Fig. 3 shows clustering representations for two approaches: DTW and disjoint KISSME. The clusterings reflect the sequences of one hour that were found similar across the days on the testing set. If these sequences are at the same hour or cover the same time slots, we can argue it is a recurrent activity (or succession of activities) and therefore a routine. The disjoint KISSME version exhibits more coherent discrimination of routines, which, according to the 4 evaluation metrics reported was predictable. Several misclassified situations seem to appear for DTW which is coherent with the negative silhouette score. High regularities can be observed, and it is actually possible to make interpretations: yellow probably corresponds to sleeping moments and nights, and purple to activities during the day. Other clusters seem to correspond to activities at the evening or during meal time. Consequently, the SS2S architecture is able to learn a metric which cluster and produce a modeling of the daily routines of the person without labels. In this example, the clusters are coarse, the granularity of this analysis could be improved simply by working with sequences of half an hour or even shorter and produce more clusters.

Metric	**Model**	**Joint**	**Completeness**	**Silhouette**	**NMI**	**AMI**
DTW [26]	x	x	0.804	-0.93	0.528	0.32
Euclidean	SLSTM	x	0.616 ± 0.032	0.427 ± 0.053	0.414 ± 0.022	0.246 ± 0.019
Cosine	SLSTM	x	0.617 ± 0.06	0.572 ± 0.143	0.372 ± 0.052	0.192 ± 0.046
Euclidean	SS2S	no	0.674 ± 0.04	0.528 ± 0.07	0.458 ± 0.03	0.28 ± 0.027
Euclidean	SS2S	yes	0.635 ± 0.064	0.408 ± 0.042	0.434 ± 0.047	0.264 ± 0.038
Cosine	SS2S	no	0.71 ± 0.05	**0.756*** ± 0.089	0.467 ± 0.028	0.275 ± 0.024
Cosine	SS2S	yes	0.714 ± 0.048	0.618 ± 0.105	0.449 ± 0.032	0.253 ± 0.03
KISSME	SS2S	no	**0.983*** ± 0.016	0.439 ± 0.077	**0.619*** ± 0.035	**0.363*** ± 0.046
KISSME	SS2S	yes	0.667 ± 0.021	0.316 ± 0.039	0.446 ± 0.012	0.266 ± 0.012

Table 2: Evaluations on LTMM dataset of the SS2S architecture (x means non applicable).

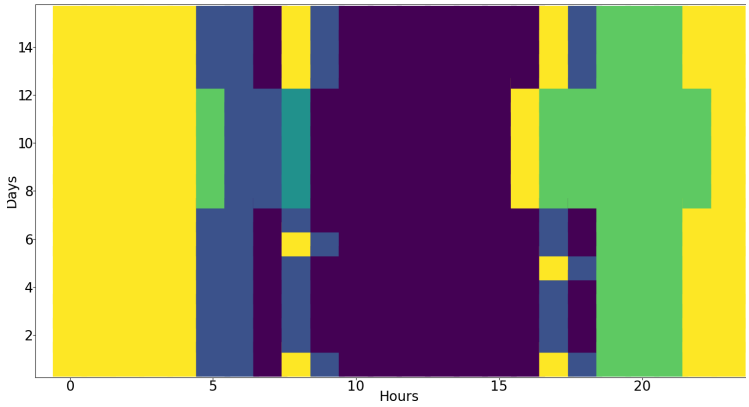
Metric	REV
Euclidean	0.707 ± 0.112
KISSME	0.736 ± 0.099
Disjoint	**0.55*** ± 0.083

(a) MSE

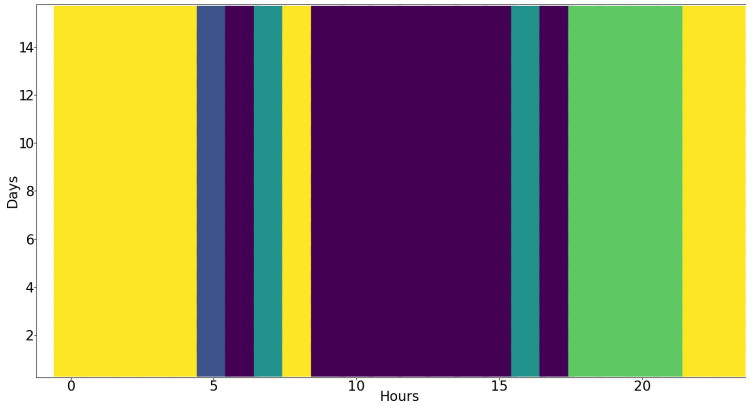
Metric	REV
Cosine	0.339 ± 0.036
Disjoint	**0.298*** ± 0.03

(b) CRL

Table 3: Average reconstruction errors on the validation set of LTMM.



(a) DTW [26].



(b) SS2S and KISSME, disjoint learning.

Fig. 3: Examples of clustering obtained with our model on LTMM.

6 Conclusions and perspectives

We presented a metric learning model to cluster routines in the daily behavior of individuals. By defining routines as almost-periodic functions, we have been able to study them in a metric learning framework. We thus proposed an approach which combines metric learning and representation learning of sequences. Our

proposed architecture relies on no labels and is learned only from time slots. A new reconstruction loss was also proposed to be learned jointly with a cosine metric and it showed better results than MSE in this case. Our SS2S architecture with KISSME and disjoint learning process achieved stimulating results with 0.983 of completeness and 0.619 of NMI. A visual evaluation analysis allows to interpret the recurrent behaviors discovered by the architecture. However, these results invalidate in this case our second hypothesis that combining metric learning and sequence to sequence learning would give better results.
In the future, we will investigate more deeply joint learning of representations and metrics. Several architecture improvements could also be made, for examples: work with triplets instead of pairs, replace the LSTM with a convolutionnal neural network [13] or an echo states network [17]. This last approach works quite differently from a normal neural network and would require subsequent modifications of the architecture. Finally, we will study in further details the link between almost-periodic functions and metric learning.

References

1. Abid, A., Zou, J.: Autowarp: Learning a warping distance from unlabeled time series using sequence autoencoders. arXiv preprint arXiv:1810.10107 (2018)
2. Aghabozorgi, S., Shirkhorshidi, A.S., Wah, T.Y.: Time-series clustering–a decade review. Information Systems 53, 16–38 (2015)
3. Avci, A., Bosch, S., Marin-Perianu, M., Marin-Perianu, R., Havinga, P.: Activity recognition using inertial sensing for healthcare, wellbeing and sports applications: A survey. In: ARCS. pp. 1–10. VDE (2010)
4. Berlemont, S., Lefebvre, G., Duffner, S., Garcia, C.: Class-balanced siamese neural networks. Neurocomputing 273, 47–56 (2018)
5. Bertinetto, L., Valmadre, J., Henriques, J.F., Vedaldi, A., Torr, P.H.: Fully-convolutional siamese networks for object tracking. In: ECCV. pp. 850–865. Springer (2016)
6. Bohr, H.: Zur theorie der fastperiodischen funktionen. Acta Mathematica 46(1-2), 101–214 (1925)
7. Bromley, J., Guyon, I., LeCun, Y., Säckinger, E., Shah, R.: Signature verification using a "siamese" time delay neural network. In: NIPS. pp. 737–744 (1994)
8. Chatzaki, C., Pediaditis, M., Vavoulas, G., Tsiknakis, M.: Human daily activity and fall recognition using a smartphone's acceleration sensor. In: ICT4AWE. pp. 100–118. Springer (2016)
9. Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using rnn encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078 (2014)
10. Cumin, J., Lefebvre, G., Ramparany, F., Crowley, J.L.: Human activity recognition using place-based decision fusion in smart homes. In: CONTEXT. pp. 137–150. Springer (2017)
11. Esling, P., Agon, C.: Time-series data mining. ACM CSUR 45(1), 12 (2012)
12. Faraki, M., Harandi, M.T., Porikli, F.: Large-scale metric learning: A voyage from shallow to deep. IEEE TNNLS 29(9), 4339–4346 (2018)

13. Gehring, J., Auli, M., Grangier, D., Yarats, D., Dauphin, Y.N.: Convolutional sequence to sequence learning. In: ICML. pp. 1243–1252 (2017)
14. Gonzalez, M.C., Hidalgo, C.A., Barabasi, A.L.: Understanding individual human mobility patterns. nature 453(7196), 779 (2008)
15. Hadsell, R., Chopra, S., LeCun, Y.: Dimensionality reduction by learning an invariant mapping. In: CVPR. vol. 2, pp. 1735–1742. IEEE (2006)
16. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation 9(8), 1735–1780 (1997)
17. Jaeger, H.: The "echo state" approach to analysing and training recurrent neural networks-with an erratum note. Bonn, Germany: German National Research Center for Information Technology GMD Technical Report 148(34), 13 (2001)
18. Kalpakis, K., Gada, D., Puttagunta, V.: Distance measures for effective clustering of arima time-series. In: IEEE ICDM. pp. 273–280. IEEE (2001)
19. Koestinger, M., Hirzer, M., Wohlhart, P., Roth, P.M., Bischof, H.: Large scale metric learning from equivalence constraints. In: CVPR. pp. 2288–2295. IEEE (2012)
20. Lally, P., Van Jaarsveld, C.H., Potts, H.W., Wardle, J.: How are habits formed: Modelling habit formation in the real world. European journal of social psychology 40(6), 998–1009 (2010)
21. LeCun, Y., Chopra, S., Hadsell, R., Ranzato, M., Huang, F.: A tutorial on energy-based learning. Predicting structured data 1(0) (2006)
22. Lin, J., Li, Y.: Finding structural similarity in time series data using bag-of-patterns representation. In: SSDBM. pp. 461–477. Springer (2009)
23. Martin, R.J.: A metric for ARMA processes. IEEE Trans. Signal Process. 48(4), 1164–1170 (2000)
24. Müller, J., Thyagarajan, A.: Siamese recurrent architectures for learning sentence similarity. In: AAAI. pp. 2786–2792 (2016)
25. Sakoe, H., Chiba, S.: Dynamic programming algorithm optimization for spoken word recognition. IEEE Trans. Acou. Speech Signal Process. 26(1), 43–49 (1978)
26. Salvador, S., Chan, P.: Toward accurate dynamic time warping in linear time and space. Intelligent Data Analysis 11(5), 561–580 (2007)
27. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: NIPS. pp. 3104–3112 (2014)
28. Vincent, P., Larochelle, H., Bengio, Y., Manzagol, P.A.: Extracting and composing robust features with denoising autoencoders. In: Proceedings of the 25th international conference on Machine learning. pp. 1096–1103. ACM (2008)
29. Weinberger, K.Q., Saul, L.K.: Distance metric learning for large margin nearest neighbor classification. JMLR 10(Feb), 207–244 (2009)
30. Weiss, A., Brozgol, M., Dorfman, M., Herman, T., Shema, S., Giladi, N., Hausdorff, J.M.: Does the evaluation of gait quality during daily life provide insight into fall risk? a novel approach using 3-day accelerometer recordings. Neurorehabilitation and neural repair 27(8), 742–752 (2013)
31. Xi, X., Keogh, E., Shelton, C., Wei, L., Ratanamahatana, C.A.: Fast time series classification using numerosity reduction. In: Proceedings of the 23rd international conference on Machine learning. pp. 1033–1040. ACM (2006)
32. Yi, D., Lei, Z., Liao, S., Li, S.Z.: Deep metric learning for person re-identification. In: ICPR. pp. 34–39. IEEE (2014)