RATE-INVARIANT AUTOENCODING OF TIME-SERIES

Kaushik Koneripalli[1,3] *Suhas Lohit*[1,4] *Rushil Anirudh*[2] *Pavan Turaga*[1]

[1] Geometric Media Lab, Arizona State University, Tempe, AZ, USA
[2] Lawrence Livermore National Laboratory, Livermore, CA, USA
[3] Siemens Corporate Technology, Princeton, NJ, USA
[4] Mitsubishi Electric Research Laboratories, Cambridge, MA, USA

ABSTRACT

For time-series classification and retrieval applications, an important requirement is to develop representations/metrics that are robust to re-parametrization of the time-axis. Temporal re-parametrization as a model can account for variability in the underlying generative process, sampling rate variations, or plain temporal mis-alignment. In this paper, we extend prior work in disentangling latent spaces of autoencoding models, to design a novel architecture to learn rate-invariant latent codes in a completely unsupervised fashion. Unlike conventional neural network architectures, this method allows to explicitly disentangle temporal parameters in the form of order-preserving diffeomorphisms with respect to a learnable template. This makes the latent space more easily interpretable. We show the efficacy of our approach on a synthetic dataset and a real dataset for hand action-recognition.

Index Terms— Rate-invariance, time-series, deep learning, neural networks, autoencoder

1. INTRODUCTION

A classic challenge in the modeling of time-series data is the need to account for temporal rate variation, or mis-alignment. Previous studies [1, 2, 3] have focused on developing metrics that are *invariant* to these nuisance factors, such that the similarity between two time-series remains unchanged even when each of them are subject to different rate variations. However, computing these metrics involves the computationally expensive step of solving for the right correspondence or alignment across both the time-series. They also rely on a template or a canonical time-series, that is either provided or estimated, against which a given time-series can be compared. As a result, such techniques do not scale well either to long time-series, or to large datasets. While recent deep learning based time-series modeling techniques such as LSTMs, 1D CNNs account for rate variations mostly in a supervised manner [4, 5], where they exploit label information to learn class-discriminative representations.

KK and SL were students at ASU during this work.

In this paper, we employ ideas from recent advances in unsupervised learning [6] to build rate-invariant autoencoders (see Fig.1) that are trained end-to-end to separate attributes related to elastic rate variation of a time-series, from all of its other information. Such a strategy scales well to large datasets or time-series, and at test time, can obtain rate-invariant transformations by just a feed-forward operation in the network, rather than solving an expensive iterative optimization problem. By design, the proposed autoencoder does not require an explicit template, and a rate-invariant metric can be obtained using a combination of the Euclidean norm in the rate-factored part of the latent space.

We achieve rate invariance by using a structured and more interpretable latent space, where we allocate a fixed set of dimensions to explicitly model rate variations using warping functions and a differentiable warping layer. Using such structured latent codes and layers enables us to perform rate-invariant autoencoding in a completely unsupervised way unlike comparable earlier works [7].

Contributions:

- We propose an unsupervised data-driven autoencoder (AE) framework to learn rate-invariant representations of time-series. At test time, factoring out rate information is achieved with a simple feed-forward operation.
- The proposed structured latent space explicitly accounts for rate variations, and as a result disentangles it from the core content of the signal.
- A by-product of the autoencoder is its ability to obtain *class discriminative* representations of time-series leading to improved predictive performance in many downstream machine learning applications.

2. RELATED WORK

Disentangled latent representations: Achieving disentangled latent representations is of great interest in unsupervised representation learning, by which we mean each variable/chunks of variables in the latent space have a semantic meaning associated with them. Kulkarni et. al [7] propose

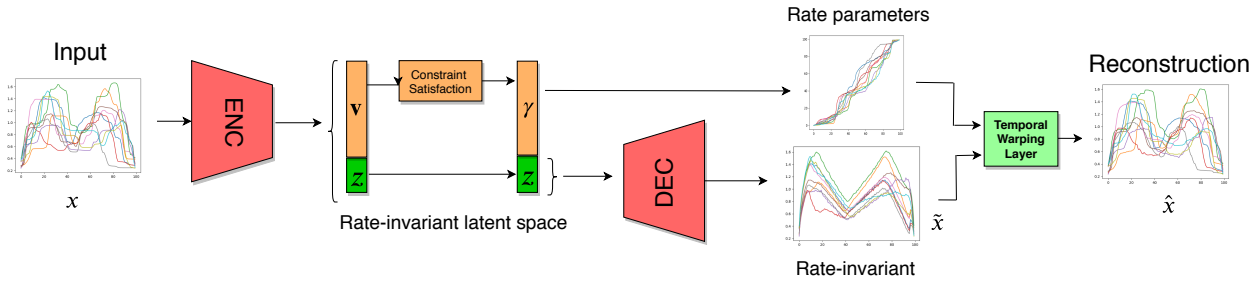

Fig. 1: **Rate-invariant autoencoder.** The figure shows an autoencoder architecture to disentangle rate variations from remaining information in the signal. The encoder, along with the constraint satisfaction layers, outputs two chunks of the latent space γ and $\mathbf{z}$. The decoder takes in only $\mathbf{z}$ and outputs a canonical representation of the signal, which is then warped with a differentiable layer with γ to produce the reconstruction.

a supervised method to achieve interpretable latent codes, by tuning one latent dimension at a time using supervised variations at the input. β-VAE [8] and Mixing autoencoders [9] propose successful disentanglement strategies, however they are not designed to make the latent codes interpretable. Shukla et al. [10] propose decomposing the latent space into a product of orthogonal spheres and demonstrate improved disentanglement. Shu et al. [6] propose Deforming Autoencoders to achieve disentangling spatial transforms in images in an interpretable manner. Our work here for rate-invariant features for time-series is inspired by this paper. Also related is a recent work by Gu et al. [11] introduce an algorithm to learn a latent space which is a product of variables belonging to flat, spherical and hyperbolic spaces, better suited to represent different types of nuisance factors.

Temporal alignment and rate-invariance: For time-series, temporal alignment of sequences has been traditionally addressed using DTW [1], and more recently using methods like SRVF [12] and Soft-DTW [2]. However, these alignment methods require a template to perform the alignment, and cannot naturally handle multiple templates for different classes. In data-driven methods, Tallec and Ollivier [4] show that LSTMs can learn to model rate variations. Jaderberg et. al [13] and Lohit et al. [5] propose frameworks to perform template-free alignment of images and time-series respectively, utilizing class labels. In contrast, our method produces template-free alignment of time-series in an unsupervised approach using an autoencoder framework.

3. UNSUPERVISED RATE DISENTANGLEMENT

In this section, first we describe the basic mathematical notation needed for modeling rate variations of time-series. Let $\alpha(t) \in \mathbb{R}^n$ be a single parameter curve that denotes a time-series signal. Let $\beta(t)$ denote a resampling of $\alpha(t)$ given by $\beta = \alpha \circ \gamma$. $\gamma \in \Gamma$ is called the warping function which is used to mathematically model rate variations between time-series i.e., we consider α and β to be the same signal, only differing in the execution rate. For a 1-differentiable function γ, defined on $[0, 1]$, to be a warping function, it needs to be in the set of order-preserving diffeomorphisms Γ [12]: $\forall \gamma \in \Gamma, \gamma(0) = 0, \gamma(1) = 1$ and $\gamma(t_1) < \gamma(t_2)$ if $t_1 < t_2$. These properties ensure that γ is a true time warping function. The order preserving property is important for applications like action recognition where order of frames/poses contains information. We denote the first derivative of γ by $\dot{\gamma}$ and we can easily see that $\gamma(t) = \int_0^t \dot{\gamma}(t)dt, \int_0^1 \dot{\gamma}(t)dt = \gamma(1) - \gamma(0) = 1$.

The above property along with the monotonicity of γ implies that $\dot{\gamma}$ has the same properties as a probability density function (PDF) and γ, that of a cumulative density function (CDF). For discretized signals, these properties become:

$$\gamma(t) = \sum_{i=0}^{t} \dot{\gamma}(i) \text{ and } \frac{1}{T} \sum_{i=0}^{T} \dot{\gamma}(i) = 1. \quad (1)$$

Now we describe the proposed method to disentangle rate variations from the time-series in an unsupervised fashion using neural networks, which we call a rate-invariant autoencoder. Fig. 1 represents the rate-invariant AE architecture. The signal is first fed into an encoder whose output is the latent space representation which consists of two parts: rate parameters which we enforce to be represented in the form of a warping function γ, and the remaining information in the signal encoded into $\mathbf{z}$. Next, only $\mathbf{z}$ is fed into the decoder. The output of the decoder $\tilde{\mathbf{x}}$ is ideally equal to the input signal with the rate variations completely factored out. Finally, a temporal warping layer [5] is used to warp the rate-invariant decoded representation using the estimated warping function to reconstruct the original signal. The network is trained end-to-end using the mean squared error (MSE) as the loss function. More formally, once the network is trained, for two input signals $\mathbf{x}_1$ and $\mathbf{x}_2$, if $\exists \gamma^* \in \Gamma$ s.t. $\mathbf{x}_2 = \mathbf{x}_1 \circ \gamma^*$, then ideally, the respective rate parameters generated are γ_1 and $\gamma_2 \in \Gamma$, and the outputs of the decoder are $\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2$ such that $\tilde{\mathbf{x}}_1 = \tilde{\mathbf{x}}_2$, and $\tilde{\mathbf{x}}_1 \circ \gamma_1 = \mathbf{x}_1$ and $\tilde{\mathbf{x}}_2 \circ \gamma_2 = \mathbf{x}_2$.

The temporal warping layer takes as input both $\tilde{\mathbf{x}}$ and the rate parameters γ in order to reconstruct the original signal. However, the raw output of the encoder, denoted by $\mathbf{v}$, does not automatically satisfy the constraints of a warping function, and hence, we employ the following constraint satisfaction layers which convert an arbitrary unconstrained vector $\mathbf{v}$ into a warping function γ.

Constraint satisfaction: The output of the encoder $\mathbf{v}$ is first converted into a density function: $\dot{\gamma} = \frac{\mathbf{v}}{||\mathbf{v}||} \odot \frac{\mathbf{v}}{||\mathbf{v}||}$, where $\odot$ refers to element-wise multiplication. Then, γ is computed from $\dot{\gamma}$ using summation as in (1). These transformations ensure that the output of the encoder is a warping function. Note that these transformations are differentiable, making efficient backpropagation feasible.

Temporal warping layer: We use the temporal warping layer proposed by Lohit et al. [5] to apply the rate parameters predicted by the encoder to the predicted rate-factored representation. The warping layer *warps* the time axis by performing a simple linear interpolation which is a differentiable operation enabling standard backpropagation to be used through the warping layer. The expression for gradients of the warping layer are derived from equations (5) & (6) of [5].

4. EXPERIMENTAL RESULTS

We now describe experimental results on two challenging datasets demonstrating disentanglement of rate parameters in a completely unsupervised manner.

4.1. Synthetic dataset for class-selective rate-invariance

In this experiment, we synthetically generate a dataset with two classes of functions: Gaussian functions and sine waves with 100 samples in time, with randomly varying amplitudes and further distorted with random time-warps. Each class has 4000 training examples and 1000 test samples. Fig. 2(a) & (b) show the entire training set before time-warping, and the classes are aligned. To illustrate rate-disentanglement and time-series alignment, we introduce random rate variations into both the classes of the dataset as in Figs. 2(c) and (d).

Network architecture and training: The encoder and decoder are both comprised of temporal convolutional (TC) layers for feature extraction and fully connected layers (FC) to map to and from the latent space. Both consist of three TC layers with a filter size of 16 and 32 filters in each layer and 1 FC layer. $tanh$ non-linearity is employed. The first T parameters contain the disentangled warping function estimate γ, where T is equal to the length of the time-series. The remaining d dimensions encode the remaining signal information. Thus, the latent-space is of dimension $T + d$. We choose $T = 100, d = 5$.

Results: As shown in Figs. 2 (e) and (f), we observe unsupervised disentanglement of rate and semantic content of the signal. We can easily see that the decoder learns to undo the rate variations and align the two classes separately. It is interesting to note that *class-dependent* aligned representations are learned even though no label information is provided. The resulting warping function as predicted by the neural network is shown in 2 (g) and (h) for classes 1 and 2 respectively. As it can be seen, all the properties of a warping function imposed by the constraint satisfaction layer are satisfied.

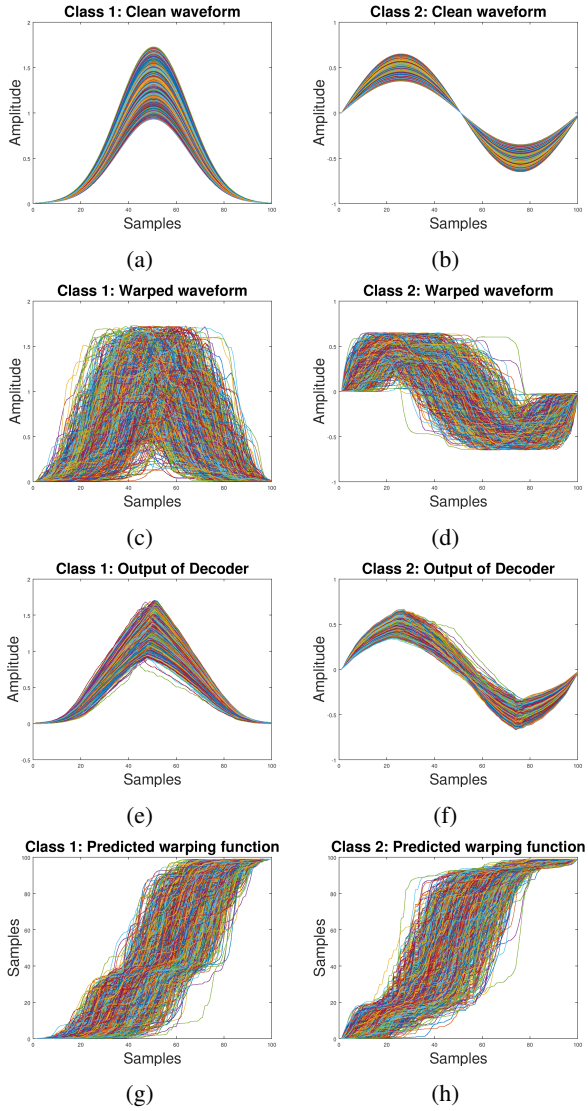

Fig. 2: (a) and (b) represent the clean version of the signal belonging to the two classes. (c) and (d) represent the same signals as (a) and (b) but with rate variations. (e) and (f) represent the canonical rate-invariant representations learned by the decoder for the two classes of signals in (a) and (b). (g) and (h) represent the warping functions generated by the encoder to transform (e) and (f) to (c) and (d) respectively.

4.2. ICL First-Person Hand Action Dataset

Dataset details: The ICL dataset [14] is a collection of 3D hand pose sequences and RGB-D videos of 6 subjects which capture common actions such as "pour milk" and "read paper". The 3D hand pose was captured using a Mocap system which accurately captures the position of each of the 21 joints of the hand. For our experiments, we only use the 3D hand-pose sequences. We use the train-test splits suggested by the authors i.e. data from subjects 1,3,4 for training and the rest for testing. The training set contains 600 sequences and the

test set contains 575 sequences. To make all the sequences of the same length, they are uniformly sampled such that they all contain 50 samples. Zero-padding is used for sequences smaller than 50 samples. The dataset is normalized such that the wrist position is at the origin.

Introducing rate variations: The ICL hand action dataset does not have sufficient rate variations for the purpose of this experiment. Therefore, we introduce rate variations into the dataset as follows. First, we set the sequence length to 100 such that the original sequence is centered between time-steps 25 and 75, and the remaining values are zero. Rate variations are induced into this data by introducing random affine warps which take the form $\gamma(t) = at + b$, t = 25 to 75. We use $a \in [0.75, 1.25]$ and $b \in \{0, 1, ..., 49\}$.

Network variants and training protocol: We train autoencoders on this dataset to learn latent codes that are useful in downstream tasks like classification. For the unsupervised settings, we first train two variants of autoencoders (a) vanilla AE with fixed latent space dimension ($d = 10$) (b) proposed rate-invariant AE with latent dimension $T + d$, where the T parameters encode the rate information in the form of the warping function γ and remaining dimensions encode the remaining information in the action sequence. In our case, $T = 100$, and we choose $d = 7$. Despite the apparent large difference in overall latent-space dimensions between the vanilla autoencoder and the proposed one (10 vs. 107), all our comparisons include only the d dimensional parts from the two (10 vs. 7). The encoder consists of 3 TC layers with filter size 16 and 64 feature maps, and 1 FC layer.

We use the task of action recognition to quantify the efficacy of the proposed method. We compute the latent representations for both vanilla AE and the rate-invariant AE (only the 7-dimensional rate-invariant part of the latent space) and train classifiers on top of them. The classifier is composed of 3 FC layers with 80 hidden units in each of them. As a baseline, we train a classifier end-to-end on the action sequences which is made of 1 TC layer with filter size of 8 and 64 filters and 1 FC layer. We use Adam optimizer [15] to train all networks.

Results: In Table 1, we show classification accuracies on the test dataset. We observe that rate-invariant latent space features from the proposed method outperform the latent representations from a vanilla AE by a large margin of 33 percentage points. More interestingly, we even see a boost of nearly 4 percentage points compared to a neural network classifier trained end-to-end with full supervision.

In order to determine the class-discriminative ability of the latent space, we also perform k-means clustering on the feature space for all the three cases. For vanilla AE, we use the entire latent space to perform k-means. For the baseline classifier, we perform k-means on the features of the penultimate layer. For the proposed method, we only use the rate-invariant part of the latent space to perform k-means. The clustering metrics corresponding to k-means are illustrated in Table 2. We use three metrics to evaluate clustering: purity [16], homogeneity and completeness [17]. We can see that for all three metrics, the proposed method outperforms the other two. These results clearly demonstrate that the proposed architecture to achieve rate-invariance is superior to the baselines considered.

Features for classification	Accuracy
End-to-end classification	72.86%
Latent codes from vanilla AE	43.66%
Latent codes from rate-invariant AE	**76.60%**

Table 1: Classification accuracies on the affine-warped ICL test set. It can be seen that the proposed rate-invariant AE method outperforms the baselines by a large margin.

Affine-warped ICL dataset	Purity	Homogeneity	Completeness
End-to-end classification	0.310	0.504	0.517
Vanilla AE	0.220	0.420	0.429
Rate-invariant AE	**0.435**	**0.590**	**0.611**

Table 2: Comparison of clusters obtained using different features of the affine-warped ICL test set. It can be seen that the clusters obtained by the proposed rate-invariant AE latent-space outperforms the baselines.

5. CONCLUSION AND FUTURE WORK

In this paper we present an autoencoder framework to disentangle rate parameters for time-series in a completely unsupervised fashion. We achieve rate-invariant representations using a structured latent space and a temporal warping layer. Furthermore, the decoder learns class-dependent canonical representations with respect to which the rate parameters in latent space are learned. As future work, we wish to extend these ideas to learn deformation-invariant latent spaces in other generative modeling techniques such as variational autoencoders and generative adversarial networks. We have shown that, compared to either purely model-based or purely data-driven methods, the combination of theoretically well-motivated mathematical models for nuisance factors and powerful data-driven models in the form of neural networks can lead to more efficient techniques, and more discriminative and interpretable representations.

6. ACKNOWLEDGEMENTS

This work was supported by NSF grant 1617999. The authors would like to thank Ankita Shukla for useful discussions, and NVIDIA Corporation for donating a Titan Xp GPU which was used for some experiments in this paper.

7. REFERENCES

[1] Donald J Berndt and James Clifford, "Using dynamic time warping to find patterns in time series.," in *KDD workshop*, 1994, vol. 10, pp. 359–370.

[2] Marco Cuturi and Mathieu Blondel, "Soft-DTW: a differentiable loss function for time-series," in *Proceedings of the 34th International Conference on Machine Learning*, 2017, pp. 894–903.

[3] Rushil Anirudh, Pavan Turaga, Jingyong Su, and Anuj Srivastava, "Elastic functional coding of Riemannian trajectories," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 5, pp. 922–936, 2016.

[4] Corentin Tallec and Yann Ollivier, "Can recurrent neural networks warp time?," *International Conference on Learning Representations*, 2018.

[5] Suhas Lohit, Qiao Wang, and Pavan Turaga, "Temporal transformer networks: Joint learning of invariant and discriminative time warping," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12426–12435.

[6] Zhixin Shu, Mihir Sahasrabudhe, Riza Alp Guler, Dimitris Samaras, Nikos Paragios, and Iasonas Kokkinos, "Deforming autoencoders: Unsupervised disentangling of shape and appearance," in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 650–665.

[7] Tejas D Kulkarni, William F Whitney, Pushmeet Kohli, and Josh Tenenbaum, "Deep convolutional inverse graphics network," in *Advances in Neural Information Processing Systems*, 2015, pp. 2539–2547.

[8] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner, "beta-VAE: Learning basic visual concepts with a constrained variational framework.," *International Conference on Learning Representations*, vol. 2, no. 5, pp. 6, 2017.

[9] Qiyang Hu, Attila Szabó, Tiziano Portenier, Paolo Favaro, and Matthias Zwicker, "Disentangling factors of variation by mixing them," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3399–3407.

[10] Ankita Shukla, Sarthak Bhagat, Shagun Uppal, Saket Anand, and Pavan Turaga, "Product of orthogonal spheres parameterization for disentangled representation learning," *British Machine Vision Conference*, 2019.

[11] Albert Gu, Frederic Sala, Beliz Gunel, and Christopher Ré, "Learning mixed-curvature representations in product spaces," 2018.

[12] Anuj Srivastava and Eric P Klassen, *Functional and shape data analysis*, Springer, 2016.

[13] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al., "Spatial transformer networks," in *Advances in Neural Information Processing Systems*, 2015, pp. 2017–2025.

[14] Guillermo Garcia-Hernando, Shanxin Yuan, Seungryul Baek, and Tae-Kyun Kim, "First-person hand action benchmark with RGB-D videos and 3d hand pose annotations," in *Proceedings of Computer Vision and Pattern Recognition*, 2018.

[15] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[16] Ying Zhao and George Karypis, "Criterion functions for document clustering: Experiments and analysis," 2001.

[17] Andrew Rosenberg and Julia Hirschberg, "V-measure: A conditional entropy-based external cluster evaluation measure," in *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*, 2007, pp. 410–420.