Regime searching in time series data using Variational Autoencoder

Artem Prosvetov\textsuperscript{1,2}

\textsuperscript{1} LANIT, Moscow
\textsuperscript{2} Space Research Institute Russian Academy of Science

e-mail: prosvetov@gmail.com

Abstract. In the current work we propose a method to extract regimes from time series using unsupervised learning. The proposed method is based on neural network with architecture of variational autoencoder and clusterization in latent space. The method has been proven by extracting regimes from a steam turbine telemetry data set and from human activity recognition data, which suggests that the proposed approach can extract regimes from time series data obtained for different areas.

1. Introduction

During time series processing the problem of regime extraction regularly appears. By the regime we understand the stable state, which is characterized by their own probability distributions, and the transition of one regime to another is governed by another process or variable. While supervised methods of machine learning can help in time series classification or regression, one has to use unsupervised methods in order to extract regimes in multicomponent time series data.

One of the approaches to extract regimes can be based on the Change Point Detection methods like Pruned Exact Linear Time \cite{1}, Optimal detection method \cite{2}, or Binary segmentation method \cite{3}. However Optimal detection method require the number of change points, and some of CPD methods is hard to use on large multi component data sets with a high number of components due to it’s computationally complexity.

One can use the statistical properties of components, searching the stationary periods in time series and combine them in regimes, however one has to assume that all components should pass one of stationary tests, while in regimes from real world data only part of time series components meet mentioned properties and only domain experts can help in the process of component selection.

Nowadays neural networks successively solve many problems in time series analysis and we suggest to use them in regime extraction.

2. Data set

We applied our approach on several data sets from different domains. First data set is provided from Cogeneration steam turbine PT-75 / 80-8.8-1.26M by our partners. Data set contains 16 components, corresponding to temperature, pressure, output power and control parameters. Our purpose was to recognize the working regimes of the steam turbine.

Second data set was obtained from the human activity monitoring domain. The workers used a fitness tracker to record the movement of hands. The data from the accelerometer and gyroscope in
three coordinate axes with a frequency of 50 Hz, corresponding to a period of 0.02 s was used in order to recognize base actions, like walking, relaxing, smoking, etc. The probabilities of each action type were used in chronological order in the same way, as it was proposed in our previous work [4]. On fig. 1 one can find the example of prepared time series. Our purpose was to recognize the general regimes of the worker during the day.

![Figure 1](image_url). The encoded times series for human activity data set: 6 time intervals are presented horizontally in chronological order, each interval has 14 encoded types of low level action during 5 seconds, like movement, computer work, relaxation etc.

### 3. Model details

In order to obtain generalization of time series we used a neural network with architecture of variational autoencoder [5]. The input layer was applied on the tensor of parameter evolution during one time frame: the period of the time frame was 30 seconds for human activity and 180 seconds for steam turbine.

Encoder had 4 convolution layers [6] (2d) with 32 filters on the first layer and 64 filters on the 3 subsequent layers. Kernel size was equal to 3, and “ReLu” activation function. A fully connected layer was applied after convolution layers with 32 neurons and ReLu activation function.

The latent dimension layer had a size equal to 2, that allowed us to visualize the results.

Decoder had an input layer of 2 neurons, a fully connected layer with ReLu activation function. The next decoder layer was a fully connected layer with the size equal to M*N/2, where M and N are the shape of input tensor. Next decoder layer was the Transposed convolution layer also known as Deconvolution with 16 filters and kernel size equals 3. The need for transposed convolutions arises from the desire to use a transformation going in the opposite direction of a normal convolution. Finally the decoder had a Convolution layer to obtain the tensor size equal to the input tensor size.

For the loss function we used a reconstruction term and a regularisation term in form of Kullback-Leibler divergence. The reconstruction term in loss function was based on binary cross entropy loss:

\[
H_1 = \sum_{i=0}^{N} y_i \log(p(y_i)) + (1 - y_i) \log(1 - p(y_i))
\]

(1)

The regularization term in form of Kullback-Leibler divergence was rewritten in the following form [5]:

\[
H_1 = \sum_{i=0}^{N} y_i \log(p(y_i)) + (1 - y_i) \log(1 - p(y_i))
\]
\[ H_2 = -\sum 1 + \mu^2 - e^\sigma \]

Where \( \mu \) and \( \sigma \) are the individual means and standard deviations.

The total loss function was a mean of H1 and H2 loss:

\[ H = \frac{H_1 + H_2}{2} \]

Also we used a reparametrisation trick [7] to make the backpropagation possible through the network.

4. Results

We trained neural networks during 2–3 epochs in each case, while loss decayed on validation dataset.

After model training we were able to use the encoder to transform multidimensional time series into a 2-dimensional latent space. The 2D projection of time series were grouped by clusterization algorithms (Agglomerative Clustering [8]) that allowed us to obtain regimes.

In case of steam turbine we found the following distribution of time series in latent space (fig. 2):

**Figure 2.** The 2D projection on latent space of time series data from the steam turbine.

The distribution of points in latent space allows us to obtain clusters.

If we consider the evolution of the turbine output power over time in the distribution over clusters, we get the figure 3. As one can see on the picture, adjacent clusters in figure 2 are adjacent in time. Interestingly, the green and purple clusters are characterized by approximately the same power, but are separated quite well by the algorithm. The difference between the green and purple clusters can be found in the distribution of other parameters. The mentioned difference between green and purple regimes was approved by domain experts.
Figure 3. The distribution of turbine output power over time, the color corresponds to the cluster in the latent space.

In case of fitness tracker data we found the following distribution of time series in latent space (fig. 4):

Figure 4. The 2D projection on latent space of time series data from the fitness-tracker.

The distribution of points in latent space allows us to obtain clusters either

In case of fitness tracker data the Gaussian Mixture model was applied to group points in latent space into clusters. The Gaussian Mixture model implements the expectation-maximization algorithm [9] for fitting the data with mixture-of-Gaussian models. The labels for worker actions in high level actions were obtained by assessor’s markup. The obtained label allied us to understand if the cluster has common properties and group clusters with the same dominant label. The resulting distribution of point in latent space by the clusters can be found on fig. 5.
Figure 5. The 2D projection on latent space of time series data from the fitness-tracker. The distribution of points in latent space allows us to obtain clusters either. The dominant labels in clusters, obtained from assessor's markup, are presented in legend.

As one can see, in both cases the clusterization algorithm applied on the projection in the latent space can obtain regimes in time series data, but post processing can be useful in one of the cases.

5. Conclusion
In the current work we applied neural networks in order to find regimes in time series data. Two sources of data were used: steam turbine telemetry and fitness tracker recognition data. The architecture of neural networks was a variational autoencoder with convolution layers. After model training we were able to obtain a projection of each time series frame on latent space. The clusters of point distribution in latent dimension were used in order to find regimes in time series. The found regimes corresponded to the requirements of the problem and were characterized by the common type of distribution of the time series components. The proposed method can be applied on time series from different domains to obtain regimes in time series.

References
[1] R Killick, P Fearnhead and I A Eckley 2012 Journal of the American Statistical Association 107(500) pp 1590–1598
[2] S M Kay and A V Oppenheim 1993 Fundamentals of Statistical Signal Processing Volume II: Detection Theory. Prentice Hall
[3] A Sen and M S Srivastava 1975 The Annals of Statistics 3(1) pp 98–108
[4] A Prosvetov2018 J. Phys.: Conf. Ser. 1117 012006
[5] An Jinwon and S Cho 2015 Special Lecture on IE
[6] Zhang Wei 1988 Proceedings of Annual Conference of the Japan Society of Applied Physics.
[7] Diederik P Kingma and Max Welling 2013 arXiv:1312.6114 [stat.ML]
[8] Rokach Lior and Oded Maimon 2005 Data mining and knowledge discovery handbook (Springer US) pp 321–352
[9] Dempster A P, Laird N M and Rubin D B 1977 Journal of the Royal Statistical Society, Series B 39 (1) pp 1–38