DYNAMICAL CLUSTERING OF TIME SERIES DATA USING MULTI-DECODER RNN AUTOENCODER

Anonymous authors
Paper under double-blind review

ABSTRACT

Clustering algorithms have wide applications and play an important role in data analysis fields including time series data analysis. The performance of a clustering algorithm depends on the features extracted from the data. However, in time series analysis, there has been a problem that the conventional methods based on the signal shape are unstable for phase shift, amplitude and signal length variations. In this paper, we propose a new clustering algorithm focused on the dynamical system aspect of the signal using recurrent neural network and variational Bayes method. Our experiments show that our proposed algorithm has a robustness against above variations and boost the classification performance.

1 INTRODUCTION

The rapid progress of IoT technology has brought huge data in wide fields such as traffic, industries, medical research and so on. Most of these data are gathered continuously and accumulated as time series data, and the extraction of features from a time series have been studied intensively in recent years. The difficulty of time series analysis is the variation of the signal in time which gives rise to phase shift, compress/stretch and length variation. Many methods have been proposed to solve these problems. Dynamic Time Warping (DTW) was designed to measure the distance between warping signals (Rabiner & Juang, 1993). This method solved the compress/stretch problem by applying a dynamic planning method. Fourier transfer or wavelet transfer can extract the features based on the frequency components of signals. The phase shift independent features are obtained by calculating the power spectrum of the transform result. In recent years, the recurrent neural network (RNN), which has recursive neural network structure, has been widely used in time series analysis (Elman, 1990; 1991). This recursive network structure makes it possible to retain the past information of time series. Furthermore, this architecture enables us to apply this algorithm to signals with different lengths. Although the methods mentioned above are effective solutions for the compress/stretch, phase shift and signal length variation issues respectively, little has been studied about these problems comprehensively.

Let us turn our attention to feature extraction again. Unsupervised learning using a neural network architecture autoencoder (AE) has been studied as a feature extraction method (Hinton & Salakhutdinov, 2006; Vincent et al., 2008; Rifai et al., 2011). AE using RNN structure (RNN-AE) has also been proposed (Srivastava et al., 2015) and it has been applied to real data such as driving data (Dong et al., 2017) and others. RNN-AE can be also interpreted as the discrete dynamical system: chaotic behavior and the deterrent method have been studied from this point of view (Zerroug et al., 2013; Laurent & von Brecht, 2016).

In this paper, we propose a new clustering algorithm for feature extraction focused on the dynamical system aspect of RNN-AE. In order to achieve this, we employed a multi-decoder autoencoder with multiple decoders to describe different dynamical systems. We also applied the variational Bayes method (Attias, 1999; Ghahramani & Beal, 2001; Kaji & Watanabe, 2011) as the clustering algorithm.

This paper is composed as follows: in Section 4, we explain AE from a dynamical system view, then we define our model and from this, derive its learning algorithm. In Section 5, we describe the application of our algorithm to an actual time series to show its robustness, including running two experiments using periodic data and driving data. Finally we summarize our study and describe our future work in Section 7.

2 RELATED WORK

A lot of excellent clustering/representation algorithms of data using AE have been studied so far (Tschannen et al., 2018). Song et al. (2013) integrated the distance between data and centroids into an objective function to obtain a cluster structure in the encoded data space. Pineau & Lelarge (2018) proposed a generative model based on the variational autoencoder (VAE) (Kingma & Welling, 2014) with a clustering structure as a prior distribution. Wang et al. (2019) achieved a high separability clustering result by adding a regularization term for the orthogonality and balanced clusters of the encoded data. These, however, are regularization methods of the objective function, and focused on only the distribution of the encoded data.
They did not give the clustering policy based on the decoder structure, namely, the reconstruction process of the data. From dynamical system point of view, one decoder of RNN-AE corresponds to a single dynamics in the space of latent representation. Hence, it is natural to equip RNN-AE with multiple decoders to implement multiple dynamics. Such an extension of RNN-AE, however, has yet to be proposed in related works to the best of our knowledge.

3 RECURRENT NEURAL NETWORK AND DYNAMICAL SYSTEM

3.1 RECURRENT NEURAL NETWORK USING UNITARY MATRIX

RNN is a neural network designed for time series data. The architecture of the main unit is called cell, and mathematical expressions are shown in Fig. 1 and Eq. (1).

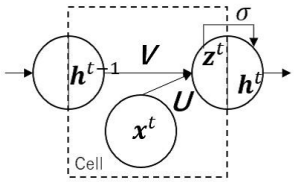

Figure 1: RNN Cell

Suppose we are given a time series,

$$\mathbf{X} = (\boldsymbol{X}^1, \cdots, \boldsymbol{X}^n, \cdots, \boldsymbol{X}^N),\ \boldsymbol{X}^n = (\boldsymbol{x}_n^1, \cdots, \boldsymbol{x}_n^t, \cdots, \boldsymbol{x}_n^T),$$
$$\boldsymbol{x}_n^t \in \mathbb{R}^D,\ n = 1, \cdots, N,\ t = 1, \cdots, T,$$

where D denotes data dimension. RNN, unlike the usual feed-forward neural network, operates the same transform matrix to the hidden valuable recursively,

$$\boldsymbol{z}^t = \boldsymbol{V}\boldsymbol{h}^{t-1} + \boldsymbol{U}\boldsymbol{x}^t + \boldsymbol{b},\ \ \boldsymbol{h}^t = \sigma(\boldsymbol{z}^t), \tag{1}$$

where $\sigma(\cdot)$ is an activation function and $\boldsymbol{z}^t,\ \boldsymbol{h}^t,\ \boldsymbol{b} \in \mathbb{R}^L$. This recursive architecture makes it possible to handle signals with different lengths, although it is vulnerable to the vanishing gradient problem as with the deep neural network (DNN) (Elman, 1990; 1991). Long short-term memory (LSTM) and gated recurrent unit (GRU) are widely known solutions to this problem (Gers et al., 2000; Hochreiter & Schmidhuber, 1997; Cho et al., 2014). These methods have the extra mechanism called a gate structure to control output scaling and retaining/forgetting of the signal information. Though this mechanism works effectively in many application fields (Malhotra et al., 2015; Rana, 2016), the architecture of network is relatively complicated. As an alternative simpler method to solve this problem, the algorithm using a unitary matrix as the transfer matrix $\boldsymbol{V}$ was proposed in recent years. Since the unitary matrix does not change the norm of the variable vector, we avoid the vanishing gradient problem. In addition, the network architecture remains unchanged from the original RNN.
In this paper, we focus on the dynamical system aspect of the original RNN. We employ the unitary matrix type RNN to take advantage of this dynamical system structure. However, to implement the above method, we need to find the transform matrix $\boldsymbol{V}$ in the space of unitary matrices $\mathbb{U} = \{\boldsymbol{U}(L) \in \mathrm{GL}(L) | \boldsymbol{U}(L)^* \boldsymbol{U}(L) = \boldsymbol{I}\}$, where $\mathrm{GL}(L)$ is the set of complex-valued general linear matrices with size $L \times L$ and $*$ means the adjoint matrix. Several methods to find the transform

matrix from the $\mathbb{U}$ has been reported so far (Pascanu et al., 2013; Jing et al., 2017; Wisdom et al., 2016; Arjovsky et al., 2016; Jing et al., 2019). Here, we adopt the method proposed by Jing et al. (2017).

3.2 RNN AUTOENCODER AND DYNAMICAL SYSTEM

The architecture of AE using RNN is shown in Fig. 2. AE is composed of an encoder unit and a decoder unit. The parameters $(\boldsymbol{V}_{enc}, \boldsymbol{U}_{enc}, \boldsymbol{V}_{dec}, \boldsymbol{U}_{dec})$ are trained by minimizing $\|\boldsymbol{X} - \boldsymbol{X}_{dec}\|_F^2 = \sum_{t=1}^{T} \|\boldsymbol{x}^t - \boldsymbol{x}_{dec}^t\|^2$, where $\boldsymbol{X}$ is the input data and $\boldsymbol{X}_{dec}$ is the decoded data.

The input data is recovered from only the encoded signal $\boldsymbol{h}$ using the matrix $(\boldsymbol{V}_{dec}, \boldsymbol{U}_{dec})$, therefore $\boldsymbol{h}$ is considered as the essential information of the input signal. When focusing on the transformation

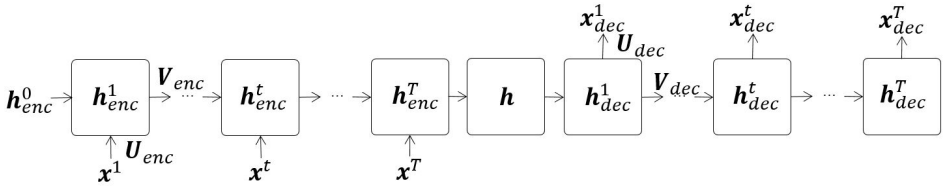

Figure 2: Architecture of RNN Autoencoder

of the hidden variable, this recursive operation has the same structure of a discrete dynamical system expression as described in the following equation:

$$\boldsymbol{h}^t = f(\boldsymbol{h}^{t-1}),$$

where f is given by Eq. (1). From this point of view, we can understand that RNN describes the universal dynamical system structure which is common to the all input signals.

4 DERIVATION OF MULTI-DECODER RNN AE ALGORITHM

In this section, we will give the architecture of the Multi-Decoder RNN AE (MDRA) and its learning algorithm. As we discussed in the previous section, RNN can extract the dynamical system characteristics of the time series. In the case of the original RNN, the model expresses just one dynamical system, hence all input data are recovered from the encoded result $\boldsymbol{h}$ by the same recovery rule. Therefore $\boldsymbol{h}$ is usually used as the feature value of the input data. In contrast, in this paper, we focus on the transformation rule itself. For this purpose, we propose MDRA which has multiple decoders to extract various dynamical system features. The architecture of MDRA is shown in Fig. 3. Let us put $\mathbf{W}_{dec}^k = (\boldsymbol{V}_{dec}^k, \boldsymbol{U}_{dec}^k)$ for $k = 1, \cdots, K$, $\mathbf{W}_{enc} = (\boldsymbol{V}_{enc}, \boldsymbol{U}_{enc})$, and $\mathbf{W} = (\mathbf{W}_{enc}, \mathbf{W}_{dec}^1, \cdots, \mathbf{W}_{dec}^K)$. We will derive the learning algorithm to optimize the whole set of parameters $\mathbf{W}$ in the following section.

4.1 DECOMPOSITION OF FREE ENERGY

We applied a clustering method to derive the learning algorithm of MDRA. Many clustering algorithms have been proposed: here we employ the variational Bayes (VB) method, because the VB method enabled us to adjust the number of clusters by tuning the hyperparameters of a prior distribution. We first define free energy, which is negative marginal log-likelihood, by the following equation,

$$F(\mathbf{X}|\mathbf{W}) = -\log \int \int \left\{ \prod_{n=1}^{N} \sum_{\boldsymbol{y}_n} p_{\mathbf{W}}(\boldsymbol{X}^n|\boldsymbol{y}_n, \boldsymbol{h}_n, \beta) p(\boldsymbol{y}_n|\boldsymbol{\alpha}) \right\} p(\boldsymbol{\alpha}) p(\beta) d\boldsymbol{\alpha} d\beta, \tag{2}$$

where $\mathbf{X}$ is data tensor defined in Section 3 and $\mathbf{W}$ is parameter tensor of MDRA defined above. $\boldsymbol{Y} = (\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_N)$ is the set of latent variables each of which means an allocation for a decoder. That is, $\boldsymbol{y}_n = (y_{n1}, \cdots, y_{nK})^{\mathrm{T}} \in \mathbb{R}^K$, where $y_{nk} = 1$ if $\boldsymbol{X}^n$ is allocated to the k-th decoder

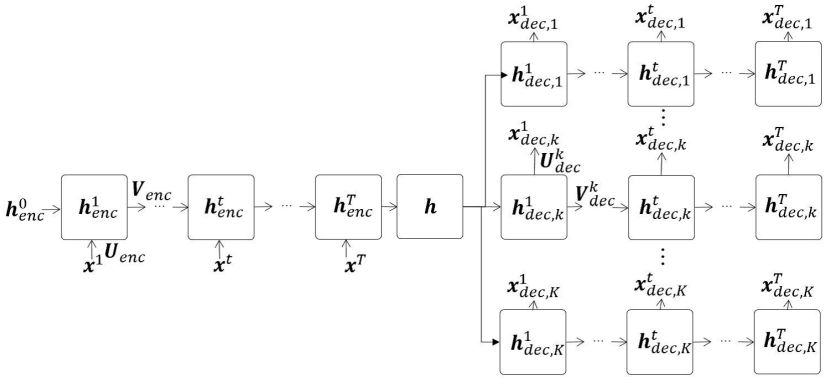

Figure 3: Architecture of MDRA

and otherwise $y_{nk} = 0$. $p_{\mathbf{W}}(\boldsymbol{X}^n|\boldsymbol{y}_n, \boldsymbol{h}_n, \beta)$ is the probability density function representation of MDRA parametrized by tensor $\mathbf{W}$, $p(\boldsymbol{\alpha})$ and $p(\beta)$ are its prior distributions for a probability vector $\boldsymbol{\alpha} = (\alpha_1, \cdots, \alpha_K)$ and a precision parameter $\beta > 0$. We applied the Gaussian mixture model as our probabilistic model. Hence $p(\boldsymbol{\alpha})$ and $p(\beta)$ were given by Dirichlet and gamma distributions respectively which are the conjugate prior distributions of multinomial and Gaussian distributions. These specific distributions are given as follows:

$$p_{\mathbf{W}}(\mathbf{X}, \boldsymbol{Y}, \boldsymbol{\alpha}, \beta|\boldsymbol{H}) = p_{\mathbf{W}}(\mathbf{X}|\boldsymbol{Y}, \boldsymbol{H}, \beta)p(\boldsymbol{Y}|\boldsymbol{\alpha})p(\boldsymbol{\alpha})p(\beta),$$

$$p(\boldsymbol{\alpha}) = \frac{\Gamma(\theta_0 K)}{\Gamma(\theta_0)^K} \prod_{k=1}^{K} \alpha_k^{\theta_0 - 1}, \quad p(\beta) = \frac{\lambda_0^{\nu_0}}{\Gamma(\nu_0)} \beta^{\nu_0 - 1} \exp(-\lambda_0 \beta),$$

$$p_{\mathbf{W}}(\mathbf{X}|\boldsymbol{Y}, \boldsymbol{H}, \beta) = \prod_{n=1}^{N} p_{\mathbf{W}}(\boldsymbol{X}^n|\boldsymbol{y}_n, \boldsymbol{h}_n, \beta), \quad p(\boldsymbol{Y}|\boldsymbol{\alpha}) = \prod_{n=1}^{N} p(\boldsymbol{y}_n|\boldsymbol{\alpha}),$$

$$p_{\mathbf{W}}(\boldsymbol{X}^n|\boldsymbol{y}_n, \boldsymbol{h}_n, \beta) = \prod_{k=1}^{K} \left\{ \left(\frac{\beta}{\pi}\right)^{\frac{T_n D}{2}} \exp(-\beta \|\boldsymbol{X}^n - g(\boldsymbol{h}_n|\mathbf{W}_{dec}^k)\|_F^2) \right\}^{y_{nk}}, p(\boldsymbol{y}_n|\boldsymbol{\alpha}) = \prod_{k=1}^{K} \alpha_k^{y_{nk}}.$$

Here, $\theta_0 > 0, \nu_0 > 0$ and $\lambda_0 > 0$ are hyperparameters and $g(\boldsymbol{h}_n|\mathbf{W}_{dec}^k) = \boldsymbol{X}_{dec,k}^n$ denotes decoder mapping of RNN from the encoded n-th data $\boldsymbol{h}_n$, $\boldsymbol{H} = (\boldsymbol{h}_1, \cdots, \boldsymbol{h}_N)$ and $T_n D$ is the total signal dimension of input signal $\boldsymbol{X}^n$ including dimension of input data. To apply the variational Bayes algorithm, we the derive the upper bound of the free energy by applying Jensen's inequality,

$$\begin{aligned}
F(\mathbf{X}|\mathbf{W}) &= -\log \int \int \left\{ \prod_{n=1}^{N} \sum_{\boldsymbol{y}_n} p_{\mathbf{W}}(\boldsymbol{X}^n|\boldsymbol{y}_n, \boldsymbol{h}_n, \beta) p(\boldsymbol{y}_n|\boldsymbol{\alpha}) \right\} p(\boldsymbol{\alpha}) p(\beta) d\boldsymbol{\alpha} d\beta \\
&= -\log \mathbb{E}_{q(\boldsymbol{Y})q(\boldsymbol{\alpha})q(\beta)} \left[\frac{p_{\mathbf{W}}(\mathbf{X}|\boldsymbol{Y}, \boldsymbol{H}, \beta) p(\boldsymbol{Y}|\boldsymbol{\alpha}) p(\boldsymbol{\alpha}) p(\beta)}{q(\boldsymbol{Y}) q(\boldsymbol{\alpha}) q(\beta)} \right] \\
&\leq D_{\mathrm{KL}}(q(\boldsymbol{Y})q(\boldsymbol{\alpha})q(\beta) \| p(\boldsymbol{Y}, \boldsymbol{\alpha}, \beta|\mathbf{X})) + F(\mathbf{X}|\mathbf{W}) \\
&= D_{\mathrm{KL}}(q(\boldsymbol{Y})q(\boldsymbol{\alpha})q(\beta) \| p(\boldsymbol{Y}|\boldsymbol{\alpha}) p(\boldsymbol{\alpha}) p(\beta)) - \sum_{n=1}^{N} \mathbb{E}_{q(\boldsymbol{y}_n)q(\beta)} \left[\log p_{\mathbf{W}}(\boldsymbol{X}^n|\boldsymbol{y}_n, \boldsymbol{h}_n, \beta) \right] \\
&\equiv \bar{F}(\mathbf{X}|\mathbf{W}),
\end{aligned}$$

where $D_{\mathrm{KL}}(\cdot\|\cdot)$ is the Kullback−Leibler divergence. The upper bound $\bar{F}(\mathbf{X}|\mathbf{W})$ is called the variational free energy or (negated) evidence lower bound (ELBO). The variational free energy is minimized using the variational Bayes method under the fixed parameters $\mathbf{W}$. Furthermore, it is also minimized with respect to the parameters $\mathbf{W}$ by applying the RNN learning algorithm to the second term of $\bar{F}(\mathbf{X}|\mathbf{W})$,

$$-\sum_{n=1}^{N} \mathbb{E}_{q(\boldsymbol{y}_n)q(\beta)} \left[\log p_{\mathbf{W}}(\boldsymbol{X}^n|\boldsymbol{y}_n, \boldsymbol{h}_n, \beta) \right] \propto \sum_{n=1}^{N} \mathbb{E}_{q(\boldsymbol{y}_n)q(\beta)} \left[\sum_{k=1}^{K} y_{nk} \beta \|\boldsymbol{X}^n - g(\boldsymbol{h}_n|\mathbf{W}_{dec}^k)\|_F^2 \right].$$

4.2 MINIMIZATION OF THE VARIATIONAL FREE ENERGY

In this section, we derive the variational Bayes algorithm for MDRA to minimize the variational free energy. We show the outline of the derivation below (for a detailed derivation, see Appendix A). The general formula of the variational Bayes algorithm is given by

$$\log q(\boldsymbol{Y}) = \mathbb{E}_{q(\boldsymbol{\alpha},\beta)}[\log p_{\mathbf{W}}(\mathbf{X}, \boldsymbol{Y}, \boldsymbol{H}, \boldsymbol{\alpha}, \beta)] + const.,$$
$$\log q(\boldsymbol{\alpha}, \beta) = \mathbb{E}_{q(\boldsymbol{Y})}[\log p_{\mathbf{W}}(\mathbf{X}, \boldsymbol{Y}, \boldsymbol{H}, \boldsymbol{\alpha}, \beta)] + const..$$

By applying the above equations to the above probabilistic models, we obtained the specific algorithm shown in Appendix A.1. Then we minimize the following term using RNN algorithm:

$$\sum_{n=1}^{N}\sum_{k=1}^{K}\left\{r_{nk}\|\boldsymbol{X}^n - g(\boldsymbol{h}_n|\mathbf{W}_{dec}^k)\|_F^2\right\},$$

where $r_{nk} = \mathbb{E}_{q(\boldsymbol{y}_n)}[y_{nk}]$ as detailed in Appendix A.2. We denote $\boldsymbol{R} = (\boldsymbol{r}_1, \cdots, \boldsymbol{r}_N)$, where $\boldsymbol{r}_n = (r_{n1}, \cdots, r_{nK})^{\mathrm{T}} \in \mathbb{R}^K$. From the above discussion, we finally obtained the following MDRA algorithm. Fig. 4 describes the relation of the VB and RNN steps of MDRA algorithm.

Algorithm 1 MDRA

Input: X: set of input signals
Output: W: weight tensors, $\boldsymbol{R}$: allocation weights, $\boldsymbol{H}$: encoded signals
Set hyperparameters $\theta_0, \nu_0, \lambda_0$ and the initial value of $\mathbf{W}$ randomly.
repeat
Calculate $\mathbf{W}$ that minimizes the following value by RNN algorithm:

$$\sum_{n=1}^{N}\sum_{k=1}^{K}\left\{r_{nk}\|\boldsymbol{X}^n - g(\boldsymbol{h}_n|\mathbf{W}_{dec}^k)\|_F^2\right\}.$$

Calculate $\boldsymbol{R} = (r_{nk})$ by the algorithm *VB part of MDRA* (Algorithm 2).
until the difference of variational free energy $\bar{F}(\mathbf{X}|\mathbf{W})$ $<$ Threshold

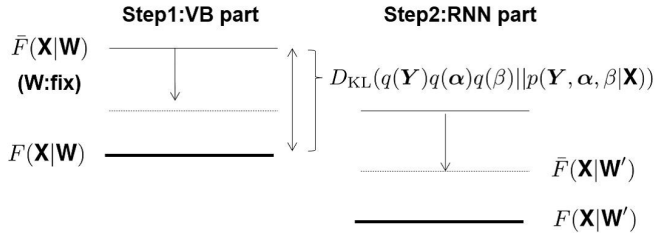

Figure 4: MDRA algorithm

5 EXPERIMENTS

5.1 PERIODIC SIGNALS

We first examined the basic performance of our algorithm using periodic signals. Periodic signals are typical time series signals expressed by dynamical systems. Input signals have 2, 4, and 8 periods respectively in 64 steps. Each signal is added a phase shift (maximum one period), amplitude variation (from 50% to 100% of the maximum amplitude), additional noise (maximum 2% of maximum amplitude) and signal length variation (maximum 80% of the maximum signal length). Examples of input data are illustrated in Fig. 5.

We compared RNN-AE to MDRA on its feature extraction performance using the above periodic signals. Fig. 6 and Fig. 7 show the results of RNN-AE and MDRA respectively. The parameter

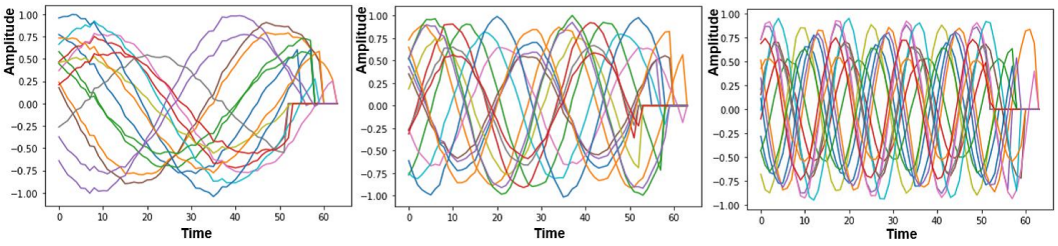

Figure 5: Examples of periodic signals

setting is listed in Table 3 in Appendix B. We used multi-dimensional scaling (MDS) as the dimension reduction method to visualize the distributions of features in Fig. 6 and Fig. 7. Fig. 6 shows the distribution of the encoded data $\boldsymbol{h}_n$ which is the initial value of the decoder unit in Fig. 2. We found that RNN-AE can separate the input data into three regions corresponding to each frequency. However each frequency region is distributed widely, therefore some part of the region overlap each other.

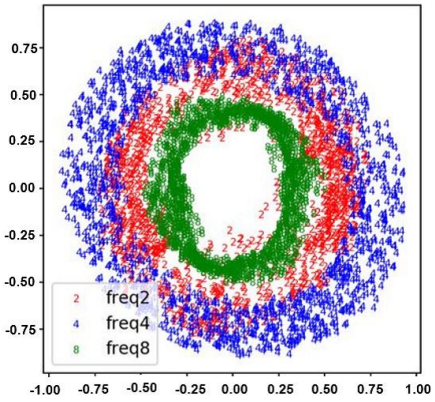

Figure 6: Visualization of features extracted by RNN-AE (periodic signals)

Fig.7 shows the distributions of the encoded data $\boldsymbol{h}_n$ and the clustering allocation weight $\boldsymbol{r}_n$ extracted by MDRA. The distribution of $\boldsymbol{r}_n$, shown in the right figure of Fig. 7, is completely separated into each frequency component without overlap. This result shows that the distribution of $\boldsymbol{r}_n$ as the feature extraction has robustness for a phase shift, amplitude and signal length variation. We also show that MDRA can boost the classification accuracy using an actual driving data in the next section.

5.2 Experiment of real driving data

We applied our algorithm to a driver identification problem. We use the data consisting of 3 drivers signals when driving, including speed, acceleration, braking and steering angle signals.[1] The input signal was about 10 seconds differential data (128 steps), which was cut out from the original data by a sliding window.[2] The detailed information of the input data is shown in Table 1. We also show samples of data (difference of acceleration) in Fig. 10 in Appendix C. The feature extraction results by RNN-AE and MDRA are shown in Fig. 8. The parameter setting of this experiment is listed in Table 4 in Appendix B. The left figure and middle figure are the distributions of $\boldsymbol{h}_n$ of RNN-AE

[1]This data was created by HQL (Research Institute of Human Engineering for Quality Life: https:// www.hql.jp/ howhql/ spirit.html).

[2]We use only the data of which the maximum acceleration difference is more than a certain threshold.

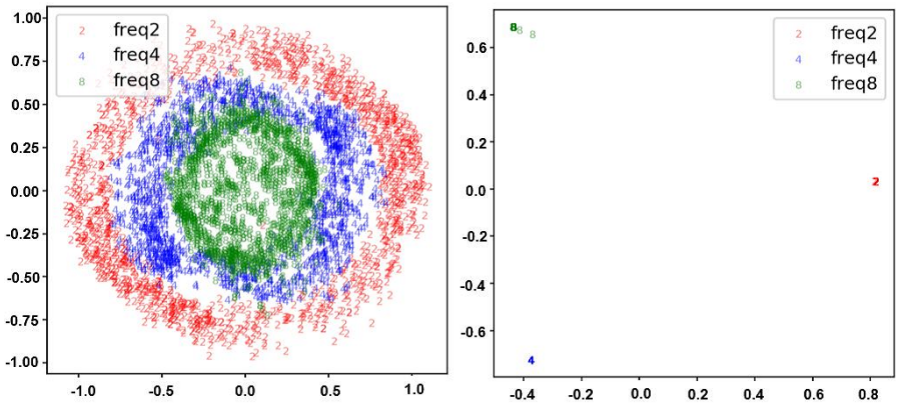

Figure 7: Visualization of features extracted by MDRA (periodic signals) left: $\boldsymbol{h}_n$, right: $\boldsymbol{r}_n$

Table 1: Driving data specification

Training	Test	Driver	Signal length	Sampling pitch	Slide
44851	11213	3	128	0.1 sec.	4

and MDRA respectively, the right figure is the distribution of $\boldsymbol{r}_n$ of MDRA. We can find different trends in the distributions of the latent variable $\boldsymbol{h}_n$ and $\boldsymbol{r}_n$ of MDRA. The distribution of $\boldsymbol{r}_n$ spreads wider than that of $\boldsymbol{h}_n$. Table 2 shows the accuracy of the driver identification results using the above

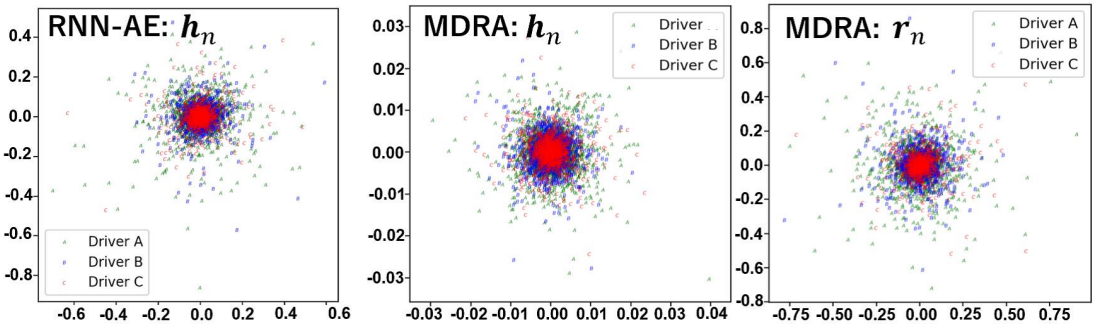

Figure 8: Extraction result of driving features

features as the input data. We adopt 10 folds cross variation to calculate the performance. The neural network for the classification is composed of 4 layers and illustrated in Fig. 11 in Appendix D. From this result, we consider that the features extracted by MDRA are quite effective for the classification of time series data.

6 Discussion

We verified the feature extraction peformance of the MDRA using actual time series data. In Section 5.1, we saw that the periodic signals are completely classified by the frequency using clustering weight $\boldsymbol{r}_n$. In this experiment, the average clustering weights, the elements of $\frac{1}{N}\sum_{n=1}^{N}\boldsymbol{r}_n \in \mathbb{R}^K$, are (3.31e-01, 8.31e-47, 8.31e-47, 3.46e-01, 8.31e-47, 3.19e-01, 8.31e-47), with only three components having effective weights. This weight narrowing-down is one of the advantages of VB learning.

The left of Fig. 9 shows an enlarged view of around "freq 4" in Fig. 7 (right). We found that the distribution of "freq 4" is in fact spread linearly. The right of Fig. 9 is the result

Table 2: Driving identification results

RNN-AE	MDRA
49.22%(+/- 0.74%)	53.80%(+/- 0.46%)
49.30%(+/- 0.52%)	53.77%(+/- 0.55%)
49.38%(+/- 0.71%)	54.01%(+/- 0.50%)
49.10%(+/- 0.47%)	52.78%(+/- 0.54%)
49.46%(+/- 0.55%)	53.51%(+/- 0.63%)
Ave. 49.29	Ave. 53.59

of the dimension reduction of features by t-Distributed Stochastic Neighbor Embedding (t-SNE) (van der Maaten & Hinton, 2008). We found that each frequency data formed several spreading clusters without overlapping.

As we saw earlier, the distribution of $\boldsymbol{r}_n$ has a spreading distribution compared to that of $\boldsymbol{h}_n$. We inferred that the spreading of the distribution $\boldsymbol{r}_n$ was caused by extracting the diversity on the driving scenes. In addition, the identification result shows that the combination of the features given by $\boldsymbol{r}_n$ and $\boldsymbol{h}_n$ can improve the performance. Dong et al. (2017), which studied a driver identify algorithm using the AE, proposed the minimization of the error integrating the reconstruction error of AE and the classification error of deep neural network. This algorithm can avoid the over-fitting by using unlabeled data whose data collection cost is smaller than labeled data. From these results, we can expect that the MDRA can contribute to not only boost identification performance but also restrain the over-fitting.

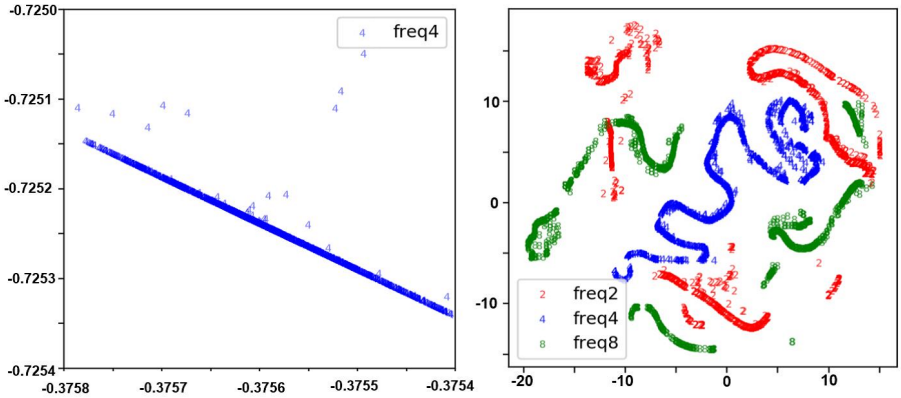

Figure 9: Left: Enlarged view of "freq 4" in Fig. 7 (right) Right: t-SNE result of $\boldsymbol{r}_n$ distribution

7 Conclusion

In this paper, we proposed a new algorithm MDRA that can extract dynamical system features of a time series data. We conducted experiments using periodic signals and actual driving data to verify the advantages of MDRA. The results show that our algorithm not only has robustness for the phase shift, amplitude and signal length variation, but also can boost classification performance.

8 Future work

The phase transition phenomenon of the variational Bayes learning method, depending on the hyperparameters, has been reported in (Watanabe & Watanabe, 2006). The hyperparameter setting of the prior distribution has a great effect on the clustering result and classification performance. We intend to undertake a detailed study of the relation between the feature extraction performance and hyperparameter setting of the prior distributions in the future.

REFERENCES

Martin Arjovsky, Amar Shah, and Yoshua Bengio. Unitary evolution recurrent neural networks. In *Proceedings of The 33rd International Conference on Machine Learning (ICML)*, pp. 1120–1128, 2016.

Hagai Attias. Inferring parameters and structure of latent variable models by variational Bayes. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence (UAI)*, pp. 21–30, 1999.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1724–1734, 2014.

Weishan Dong, Ting Yuan, Kai Yang, Changsheng Li, and Shilei Zhang. Autoencoder regularized network for driving style representation learning. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 1603–1609, 2017.

Jeffrey L. Elman. Finding structure in time. *Cognitive Science*, 14(2):179–211, 1990.

Jeffrey L. Elman. Distributed representations, simple recurrent networks, and grammatical structure. *Machine Learning*, 7(2):195–225, 1991.

Felix A. Gers, Jürgen Schmidhuber, and Fred Cummins. Learning to forget: Continual prediction with LSTM. *Neural Computation*, 12(10):2451–2471, 2000.

Zoubin Ghahramani and Matthew J. Beal. Graphical models and variational methods. In *Advanced Mean Field Methods Theory and Practice*, pp. 161–177. MIT Press, 2001.

Geoffrey E. Hinton and Ruslan R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8): 1735–1780, 1997.

Li Jing, Yichen Shen, Tena Dubcek, John Peurifoy, Scott Skirlo, Yann LeCun, Max Tegmark, and Marin Soljačić. Tunable efficient unitary neural networks (EUNN) and their application to RNNs. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pp. 1733–1741, 2017.

Li Jing, Caglar Gulcehre, John Peurifoy, Yichen Shen, Max Tegmark, Marin Soljacic, and Yoshua Bengio. Gated orthogonal recurrent units: On learning to forget. *Neural Computation*, 31(4): 765–783, 2019.

Daisuke Kaji and Sumio Watanabe. Two design methods of hyperparameters in variational Bayes learning for Bernoulli mixtures. *Neurocomputing*, 74(11):2002–2007, 2011.

Diederik P. Kingma and Max Welling. Auto-encoding variational Bayes. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2014.

Thomas Laurent and James H. von Brecht. A recurrent neural network without chaos. *CoRR*, abs/1612.06212, 2016.

Pankaj Malhotra, Lovekesh Vig, Gautam Shroff, and Puneet Agarwal. Long short term memory networks for anomaly detection in time series. In *23rd European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*, pp. 89–94, 2015.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *Proceedings of the 30th International Conference on Machine Learning (ICML)*, pp. 1310–1318, 2013.

Edouard Pineau and Marc Lelarge. Infocatvae: Representation learning with categorical variational autoencoders. *CoRR*, abs/1806.08240, 2018.

Lawrence Rabiner and Biing-Hwang Juang. *Fundamentals of Speech Recognition*. Prentice Hall, Upper Saddle River, NJ, USA, 1993.

Rajib Rana. Gated recurrent unit (GRU) for emotion classification from noisy speech. *CoRR*, abs/1612.07778, 2016.

Salah Rifai, Pascal Vincent, Xavier Muller, Xavier Glorot, and Yoshua Bengio. Contractive auto-encoders: Explicit invariance during feature extraction. In *Proceedings of the 28th International Conference on Machine Learning (ICML)*, pp. 833–840, 2011.

Chunfeng Song, Feng Liu, Yongzhen Huang, Liang Wang, and Tieniu Tan. Auto-encoder based data clustering. In *Iberoamerican Congress on Pattern Recognition (CIARP)*, pp. 117–124, 2013.

Nitish Srivastava, Elman Mansimov, and Ruslan Salakhudinov. Unsupervised learning of video representations using lstms. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, pp. 843–852, 2015.

Michael Tschannen, Mario Lucic, and Olivier Bachem. Recent advances in autoencoder-based representation learning. In *Proceedings of Workshop on Bayesian Deep Learning (NeurIPS)*, 2018.

Laurens van der Maaten and Geoffrey E. Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(Nov):2759–2605, 2008.

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th International Conference on Machine Learning (ICML)*, pp. 1096–1103, 2008.

Wei Wang, Dan Yang, Feiyu Chen, Yunsheng Pang, Sheng Huang, and Yongxin Ge. Clustering with orthogonal autoencoder. *IEEE Access*, 7:62421–62432, 2019.

Kazuho Watanabe and Sumio Watanabe. Stochastic complexities of Gaussian mixtures in variational Bayesian approximation. *Journal of Machine Learning Research*, 7(Apr):625–645, 2006.

Scott Wisdom, Thomas Powers, John Hershey, Jonathan Le Roux, and Les Atlas. Full-capacity unitary recurrent neural networks. In *Advances in Neural Information Processing Systems 29 (NIPS)*, pp. 4880–4888. 2016.

Abdelhamid Zerroug, Labib S. Terrissa, and Alain Faure. Chaotic dynamical behavior of recurrent neural network. *Annual Review of Chaos Theory, Bifurcations and Dynamical Systems*, 4:55–56, 2013.

A DERIVATION OF ALGORITHM

A.1 MINIMIZATION ALGORITHM OF THE VARIATIONAL FREE ENERGY

Initially, we suppose that the posterior is expressed by $q(\boldsymbol{Y}, \boldsymbol{\alpha}, \beta) = q(\boldsymbol{Y})q(\boldsymbol{\alpha}, \beta)$. Then

$$\begin{aligned}\log q(\boldsymbol{Y}) &= \mathbb{E}_{q(\boldsymbol{\alpha},\beta)}[\log p_{\mathbf{W}}(\mathbf{X}, \boldsymbol{Y}, \boldsymbol{H}, \boldsymbol{\alpha}, \beta)] + const. \\ &= \mathbb{E}_{q(\boldsymbol{\alpha},\beta)}[\log p_{\mathbf{W}}(\mathbf{X}|\boldsymbol{Y}, \boldsymbol{H}, \beta)] + \mathbb{E}_{q(\boldsymbol{\alpha},\beta)}[\log p(\boldsymbol{Y}|\boldsymbol{\alpha})] \\ &\quad + \mathbb{E}_{q(\boldsymbol{\alpha},\beta)}[\log p(\boldsymbol{\alpha})] + \mathbb{E}_{q(\boldsymbol{\alpha},\beta)}[\log p(\beta)] + const. \\ &= \mathbb{E}_{q(\boldsymbol{\alpha})}[\log p(\boldsymbol{Y}|\boldsymbol{\alpha})] + \mathbb{E}_{q(\boldsymbol{\alpha})}[\log p(\boldsymbol{\alpha})] \\ &\quad + \mathbb{E}_{q(\beta)}[\log p_{\mathbf{W}}(\mathbf{X}|\boldsymbol{Y}, \boldsymbol{H}, \beta)] + \mathbb{E}_{q(\beta)}[\log p(\beta)] + const..\end{aligned}$$

In addition,

$$\mathbb{E}_{q(\boldsymbol{\alpha})}[\log p(\boldsymbol{Y}|\boldsymbol{\alpha})] = \mathbb{E}_{q(\boldsymbol{\alpha})}\left[\log \prod_{n=1}^{N}\prod_{k=1}^{K} \alpha_k^{y_{nk}}\right] = \sum_{n=1}^{N}\sum_{k=1}^{K} y_{nk}\mathbb{E}_{q(\boldsymbol{\alpha})}\left[\log \alpha_k\right],$$

$$\begin{aligned}&\mathbb{E}_{q(\beta)}\left[\log p_{\mathbf{W}}(\mathbf{X}|\boldsymbol{Y}, \boldsymbol{H}, \beta)\right] \\ &= \sum_{n=1}^{N}\sum_{k=1}^{K} y_{nk}\mathbb{E}_{q(\beta)}\left[-\beta\|\boldsymbol{X}^n - g(\boldsymbol{h}_n|\mathbf{W}_{dec}^k)\|_F^2 + \frac{T_n D}{2}(\log\beta - \log\pi)\right],\end{aligned}$$

where $T_n D$ means total signal dimension. Therefore, we obtain

$$\log q(\boldsymbol{Y}) = \sum_{n=1}^{N}\sum_{k=1}^{K} y_{nk}\log\rho_{nk} + const..$$

We here put

$$\log\rho_{nk} = \mathbb{E}_{q(\boldsymbol{\alpha})}\left[\log\alpha_k\right] + \mathbb{E}_{q(\beta)}\left[-\beta\|\boldsymbol{X}^n - g(\boldsymbol{h}_n|\mathbf{W}_{dec}^k)\|_F^2 + \frac{T_n D}{2}(\log\beta - \log\pi)\right]. \tag{3}$$

Hence $q(\boldsymbol{Y}) \propto \prod_{n=1}^{N}\prod_{k=1}^{K}\rho_{nk}^{y_{nk}}$, by putting $r_{nk} = \frac{\rho_{nk}}{\sum_{k=1}^{K}\rho_{nk}}$, we obtain

$$q(\boldsymbol{Y}) = \prod_{n=1}^{N}\prod_{k=1}^{K} r_{nk}^{y_{nk}}.$$

Next we calculate $\log q(\boldsymbol{\alpha}, \beta)$,

$$\begin{aligned}\log q(\boldsymbol{\alpha}, \beta) &= \mathbb{E}_{q(\boldsymbol{Y})}[\log p_{\mathbf{W}}(\mathbf{X}, \boldsymbol{Y}, \boldsymbol{H}, \boldsymbol{\alpha}, \beta)] + const. \\ &= \mathbb{E}_{q(\boldsymbol{Y})}[\log p(\boldsymbol{Y}|\boldsymbol{\alpha})] + \log p(\boldsymbol{\alpha}) \\ &\quad + \mathbb{E}_{q(\boldsymbol{Y})}[\log p_{\mathbf{W}}(\mathbf{X}|\boldsymbol{Y}, \boldsymbol{H}, \beta)] + \log p(\beta) + const..\end{aligned}$$

Above equation can be divided into the two terms including $\boldsymbol{\alpha}$ and β respectively,

$$\begin{aligned}\log q(\boldsymbol{\alpha}) &\propto \mathbb{E}_{q(\boldsymbol{Y})}[\log p(\boldsymbol{Y}|\boldsymbol{\alpha})] + \log p(\boldsymbol{\alpha}) + const. \\ &= \sum_{n=1}^{N}\sum_{k=1}^{K}\log\alpha_k\mathbb{E}_{q(\boldsymbol{y}_n)}\left[y_{nk}\right] + (\theta_0 - 1)\sum_{k=1}^{K}\log\alpha_k + const..\end{aligned}$$

Substituting $\mathbb{E}_{q(\boldsymbol{y}_n)}[y_{nk}] = 1\cdot q(y_{nk}=1) + 0\cdot q(y_{nk}=0) = q(y_{nk}=1) = r_{nk}$ to the above equation, we obtain

$$\log q(\boldsymbol{\alpha}) = \sum_{n=1}^{N}\sum_{k=1}^{K}\log\alpha_k r_{nk} + (\theta_0 - 1)\sum_{k=1}^{K}\log\alpha_k + const..$$

On the other hand,

$$\begin{aligned}\log q(\beta) &= \mathbb{E}_{q(\boldsymbol{Y})}[\log p_{\mathbf{W}}(\mathbf{X}|\boldsymbol{Y},\boldsymbol{H},\beta)] + \log p(\beta) + const.\\ &= \sum_{n=1}^{N}\sum_{k=1}^{K}\left[E_{q(\boldsymbol{y}_n)}\left[y_{nk}\right]\left\{-\beta\|\boldsymbol{X}^n - g(\boldsymbol{h}_n|\mathbf{W}_{dec}^k)\|_F^2 + \frac{T_n D}{2}(\log\beta - \log\pi)\right\}\right]\\ &+ (\nu_0 - 1)\log\beta + \lambda\beta + const..\end{aligned}$$

Similarly, by applying $\mathbb{E}_{q(\boldsymbol{y}_n)}[y_{nk}] = r_{nk}$, we obtain

$$\begin{aligned}&\log q(\beta)\\ &= \sum_{n=1}^{N}\sum_{k=1}^{K}\left[r_{nk}\left\{-\beta\|\boldsymbol{X}^n - g(\boldsymbol{h}_n|\mathbf{W}_{dec}^k)\|_F^2 + \frac{T_n D}{2}(\log\beta - \log\pi)\right\}\right]\\ &+ (\nu_0 - 1)\log\beta + \lambda\beta + const..\end{aligned}$$

We finally calculate $\log\rho_{nk}$ in Eq. (3).We first calculate $\log q(\boldsymbol{\alpha})$ and $\log q(\beta)$,

$$\begin{aligned}\log q(\beta) &= \sum_{n=1}^{N}\sum_{k=1}^{K}\left[-r_{nk}\left\{\beta\|\boldsymbol{X}^n - g(\boldsymbol{h}_n|\mathbf{W}_{dec}^k)\|_F^2 + \frac{T_n D}{2}(\log\beta - \log\pi)\right\}\right]\\ &+ (\nu_0 - 1)\log\beta - \lambda_0\beta + const.\\ &= \beta f + \left(\nu_0 + \frac{1}{2}\sum_{n=1}^{N} T_n D - 1\right)\log\beta - \lambda\beta + const.,\end{aligned}$$

where we put $f = \sum_{k=1}^{K}\sum_{n=1}^{N} -r_{nk}\|\boldsymbol{X}^n - g(\boldsymbol{h}_n|\mathbf{W}_{dec}^k)\|_F^2$. In addition, putting $\bar{\lambda} = \lambda_0 - f$, $\bar{\nu} = \nu_0 + \frac{1}{2}\sum_{n=1}^{N} T_n D$,

$$\begin{aligned}q(\beta) &= e^{\beta f}\beta^{\nu_0 + \frac{1}{2}\sum_{n=1}^{N} T_n D - 1} e^{-\lambda_0\beta}\cdot const.\\ &= e^{-\bar{\lambda}\beta}\beta^{\bar{\nu}-1}\cdot const. = \frac{\bar{\lambda}^{\bar{\nu}}}{\Gamma(\bar{\nu})}\beta^{\bar{\nu}-1}e^{-\bar{\lambda}\beta} = Gamma(\beta|\bar{\nu},\bar{\lambda}).\end{aligned}$$

By using the expectations of β and $\log\beta$ by gamma distribution $\mathbb{E}_{q(\beta)}[\beta] = \nu\lambda^{-1}, \mathbb{E}_{q(\beta)}[\log\beta] = \psi(\nu) - \log\lambda$ (Appendix A.3), we obtain

$$\begin{aligned}&\mathbb{E}_{q(\beta)}\left[-\beta\|\boldsymbol{X}^n - g(\boldsymbol{h}_n|\mathbf{W}_{dec}^k)\|_F^2 + \frac{T_n D}{2}(\log\beta - \log\pi)\right]\\ &= -\|\boldsymbol{X}^n - g(\boldsymbol{h}_n|\mathbf{W}_{dec}^k)\|_F^2\bar{\nu}\bar{\lambda}^{-1} + \frac{T_n D}{2}(\psi(\bar{\nu}) - \log\bar{\lambda}) - \frac{T_n D}{2}\log\pi.\end{aligned}$$

Similarly, $q(\boldsymbol{\alpha})$ turns out to be the Dirichlet distribution with parameters $(\bar{\theta}_1,\cdots,\bar{\theta}_K)$, and $\mathbb{E}_{q(\boldsymbol{\alpha})}[\log\alpha_k] = \psi(\bar{\theta}_k) - \psi\left(\sum_{k=1}^{K}\bar{\theta}_k\right)$ is calculated by the same way in the general mixture model. Therefore we finally obtain

$$\begin{aligned}\log\rho_{nk} &= \psi(\bar{\theta}_k) - \psi\left(\sum_{k=1}^{K}\bar{\theta}_k\right) - \|\boldsymbol{X}^n - g(\boldsymbol{h}_n|\mathbf{W}_{dec}^k)\|_F^2\bar{\nu}\bar{\lambda}^{-1}\\ &+ \frac{T_n D}{2}(\psi(\bar{\nu}) - \log\bar{\lambda}) - \frac{T_n D}{2}\log\pi.\end{aligned}$$

From the above results, the following variational Bayes algorithm is derived.

A.2 Minimization algorithm of the second term of the RNN term (The second term of variational free energy)

In this section, we minimize $-\mathbb{E}_{q(\boldsymbol{Y})q(\beta)}\left[\sum_{n=1}^{N}\log p_{\mathbf{W}}(\boldsymbol{X}^n|\boldsymbol{y}_n,\boldsymbol{h}_n,\beta)\right]$ to minimize the free energy Eq. (2) with respect to $\mathbf{W}$. More specifically, we minimize

Algorithm 2 VB part of MDRA

Input: X: set of input signals
Output: $\boldsymbol{R}$**:** allocation weight
Set hyperparameters $\theta_0, \nu_0, \lambda_0$ and the number of iterations I
for $i \leftarrow 0$ to I **do**
VB E-step:

$$\log \rho_{nk} = \psi(\bar{\theta}_k) - \psi\left(\sum_{k=1}^{K} \bar{\theta}_k\right) - \|\boldsymbol{X}^n - g(\boldsymbol{h}_n|\mathbf{W}_{dec}^k)\|_F^2 \bar{\nu}\bar{\lambda}^{-1} + \frac{T_n D}{2}(\psi(\bar{\nu}) - \log\bar{\lambda}) - \frac{T_n D}{2}\log\pi$$

$$r_{nk} = \frac{\rho_{nk}}{\sum_{k=1}^{K}\rho_{nk}}$$

VB M-step:

$$N_k = \sum_{n=1}^{N} r_{nk}, \quad \bar{\theta}_k = \theta_0 + N_k$$

$$\bar{\lambda} = \lambda_0 + \sum_{k=1}^{K}\sum_{n=1}^{N} r_{nk}\|\boldsymbol{X}^n - g(\boldsymbol{h}_n|\mathbf{W}_{dec}^k)\|_F^2, \quad \bar{\nu} = \nu_0 + \frac{1}{2}\sum_{n=1}^{N} T_n D$$

end for

$$
\begin{aligned}
&-\sum_{n=1}^{N}\mathbb{E}_{q(\boldsymbol{y}_n)q(\beta)}\left[\log\left\{\prod_{k=1}^{K}\left\{\left(\frac{\beta}{\pi}\right)^{\frac{T_n D}{2}} e^{-\beta\|\boldsymbol{X}^n - g(\boldsymbol{h}_n|\mathbf{W}_{dec}^k)\|_F^2}\right\}^{y_{nk}}\right\}\right] \\
&= -\sum_{n=1}^{N}\mathbb{E}_{q(\boldsymbol{y}_n)q(\beta)}\left[\sum_{k=1}^{K} y_{nk}\left\{\frac{T_n D}{2}(\log\beta - \log\pi) - \beta\|\boldsymbol{X}^n - g(\boldsymbol{h}_n|\mathbf{W}_{dec}^k)\|_F^2\right\}\right] \\
&= -\sum_{n=1}^{N}\sum_{k=1}^{K}\mathbb{E}_{q(\boldsymbol{y}_n)}[y_{nk}]\left\{\frac{T_n D}{2}(\mathbb{E}_{q(\beta)}[\log\beta] - \log\pi) - \mathbb{E}_{q(\beta)}[\beta]\|\boldsymbol{X}^n - g(\boldsymbol{h}_n|\mathbf{W}_{dec}^k)\|_F^2\right\} \\
&= \mathbb{E}_{q(\beta)}[\beta]\sum_{n=1}^{N}\sum_{k=1}^{K} r_{nk}\|\boldsymbol{X}^n - g(\boldsymbol{h}_n|\mathbf{W}_{dec}^k)\|_F^2 - \sum_{n=1}^{N}\sum_{k=1}^{K} r_{nk}\frac{T_n D}{2}(\mathbb{E}_{q(\beta)}[\log\beta] - \log\pi) \\
&\propto \sum_{n=1}^{N}\sum_{k=1}^{K} r_{nk}\|\boldsymbol{X}^n - g(\boldsymbol{h}_n|\mathbf{W}_{dec}^k)\|_F^2 + const.
\end{aligned}
$$

where we used $r_{nk} = \mathbb{E}_{q(\boldsymbol{y}_n)}[y_{nk}]$. We achieve this by applying RNN algorithm. From the above discussion including Appendix A.1, we obtain the MDRA algorithm.

A.3 Derivation of $\mathbb{E}_{Gamma(\beta|\nu,\lambda)}[\log\beta]$

By putting $\beta = e^x$, we obtain $x = \log\beta, d\beta = e^x dx$,

$$
\begin{aligned}
\mathbb{E}_{Gamma(\beta|\nu,\lambda)}[\log\beta] &= \int_0^\infty \log\beta\frac{\lambda^\nu}{\Gamma(\nu)}\beta^{\nu-1}e^{-\lambda x}d\beta = \int x\frac{\lambda^\nu}{\Gamma(\nu)}(e^x)^{\nu-1}e^{-\lambda e^x}e^x dx \\
&= \int x\frac{\lambda^\nu}{\Gamma(\nu)}e^{x(\nu-1)}e^{-\lambda e^x}e^x dx = \int x\frac{\lambda^\nu}{\Gamma(\nu)}e^{x\nu-\lambda e^x}dx.
\end{aligned}
$$

We here use

$$\frac{d}{d\nu}e^{x\nu-\lambda e^x} = xe^{x\nu-\lambda e^x},$$

then the above equation is

$$\mathbb{E}_{Gamma(\beta|\nu,\lambda)}[\log\beta] = \int \frac{\lambda^\nu}{\Gamma(\nu)} \frac{d}{d\nu} e^{x\nu-\lambda e^x} dx = \frac{\lambda^\nu}{\Gamma(\nu)} \frac{d}{d\nu} \int e^{x\nu-\lambda e^x} dx.$$

In addition, $\int_0^\infty x^{\nu-1} e^{-\lambda e^x} dx$ is the normalization constant of gamma distribution, therefore it equals to $\Gamma(\nu)/\lambda^\nu$. Hence we finally obtain

$$\begin{aligned}\mathbb{E}_{Gamma(\beta|\nu,\lambda)}[\log\beta] &= \frac{\lambda^\nu}{\Gamma(\nu)} \frac{d}{d\nu} \frac{\Gamma(\nu)}{\lambda^\nu} = \frac{\lambda^\nu}{\Gamma(\nu)} \frac{\Gamma'(\nu)\lambda^\nu - \Gamma(\nu)\lambda^\nu \log\lambda}{\lambda^{2\nu}} \\ &= \frac{\Gamma'(\nu)}{\Gamma(\nu)} - \log\lambda = \frac{d}{d\nu}\log\Gamma(\nu) - \log\lambda = \psi(\nu) - \log\lambda.\end{aligned}$$

B PARAMETER SETTING

In this section, we show the parameter setting of the experiments in Section 5.

Table 3: Parameter setting (periodic signals)

	L	EUNN			VB			
		cap.	fft	cpx	K	θ_0	ν_0	λ_0
RNN-AE	16	4	T	F	-	-	-	-
MDRA	16				7	5.0	1.0	0.01

Table 4: Parameter setting of Driving identification

	L	EUNN			VB			
		cap.	fft	cpx	K	θ_0	ν_0	λ_0
RNN-AE	32	8	T	F	-	-	-	-
MDRA	32				7	0.83	100.0	0.01

Here L is the dimension of hidden variable $\boldsymbol{h}$, capacity, fft and cpx are parameters of EUNN (Jing et al., 2017), K is the number of the decoders, $\theta_0, \nu_0, \lambda_0$ are hyperparameters of prior distributions.

C EXAMPLE OF DRIVING DATA

We applied our algorithm to the driving data in the Section 5.2. We used the differential signals of speed, acceleration, braking and steering angle signals in the experiment. Fig. 10 shows examples of acceleration signals.

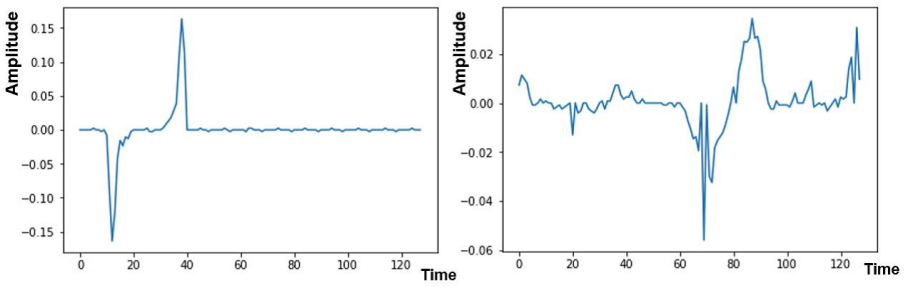

Figure 10: Examples of driving data

D NEURAL NETWORK FOR DRIVER IDENTIFICATION

We estimated the feature extraction performance of MDRA in the Section 5.2. We employed the quite simple neural network in Fig. 11.

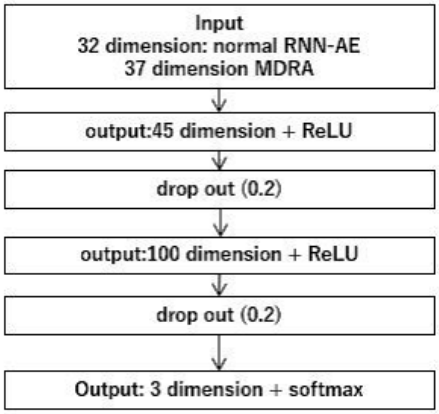

Figure 11: Fully connected neural network for classification