ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2018, i6doc.com publ., ISBN 978-287587047-6. Available from http://www.i6doc.com/en/.

Learning compressed representations of blood samples time series with missing data

Filippo Maria Bianchi, Karl Øyvind Mikalsen and Robert Jenssen *

Machine Learning Group – UiT the Arctic University of Norway

Abstract. Clinical measurements collected over time are naturally represented as *multivariate time series* (MTS), which often contain *missing data*. An *autoencoder* can learn low dimensional vectorial representations of MTS that preserve important data characteristics, but cannot deal explicitly with missing data. In this work, we propose a new framework that combines an autoencoder with the *Time series Cluster Kernel* (TCK), a kernel that accounts for missingness patterns in MTS. Via kernel alignment, we incorporate TCK in the autoencoder to improve the learned representations in presence of missing data. We consider a classification problem of MTS with missing values, representing blood samples of patients with surgical site infection. With our approach, rather than with a standard autoencoder, we learn representations in low dimensions that can be classified better.

1 Introduction

The application of machine learning and deep learning in healthcare industry is improving diagnosis outcomes and may change the way of providing care to patients [1]. The main challenge that machine learning is asked to solve is to discover relevant structural patterns in clinical data, usually concealed and difficult to detect manually.

An important fraction of electronic health records are clinical measurements collected from patients over time, which are represented as multivariate time series (MTS) [2]. Several efforts have been devoted to learn informative and compact representations of MTS [3], not only to improve the quality of the analysis, but also to manage the large amounts of data necessary to train deep learning models [4]. Furthermore, MTS are characterized by complex relationships across the variables and time that must be accounted in the analysis. However, most methods are designed to treat vectorial data and they cannot be trivially extended to capture such relationships.

The autoencoder (AE) is a type of neural network originally conceived as a non-linear dimensionality reduction algorithm [5], which has been further exploited to learn data representations in deep architectures [6]. AEs have been adopted to map time series data into *codes*, which are real-typed vectors lying in a lower dimensional space [7].

Clinical measurements are often recorded at irregular frequencies that change for different patients, across variables, and over time. Hence, after discretizing

*This work was funded by the Norwegian Research Council FRIPRO grant no. 239844 *Next Generation Learning Machines* and IKTPLUSS grant no. 270738 *Deep Learning for Health.*

ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2018, i6doc.com publ., ISBN 978-287587047-6.
Available from http://www.i6doc.com/en/.

time, the resulting MTS end up containing missing values [8]. Missing values follow patterns that reflect medical conditions of the patients or decisions of the doctors and, therefore, are important to be included in the analysis. Since AE cannot process data containing missing values, those are usually replaced with imputation techniques that, however, cannot capture those patterns as they only fill blanks trying to introduce as less bias as possible. On the other hand, a recently proposed method, called Time series Cluster Kernel (TCK) [9], computes an unsupervised kernel similarity between MTS with missing data. TCK leverages on the configurations of missingness patterns to improve the evaluation of the similarity.

In this work, we propose a completely unsupervised approach for learning compressed representations of MTS in presence of missing data. Towards that end, we utilize the *deep kernelized autoencoder* (dkAE) [10], a recently proposed architecture that embeds the properties of a given prior kernel in the code representation of an AE through kernel alignment. By introducing TCK as prior kernel, we extend the dkAE framework to time series. Moreover, due TCK's properties, the relationships among the learned codes accounts for the presence of missing data, yielding a more discriminative representation of the data.

We apply our method to classify MTS of blood samples, relative to patients with site infections contracted after surgery and with a high percentage of missing data. We compare the classification results obtained on the representations learned by a standard AE with the ones of a dkAE implementing the alignment to TCK. Results indicate that the learned codes not only provide a compact vectorial representation, but the same classifier achieves better results when operates in our code space rather than in the input space.

2 Methods

2.1 Time series Cluster Kernel

The *Time series Cluster Kernel* [9] exploits the missing patterns in MTS to compute their similarities, rather than relying on imputation methods that may introduce strong biases. TCK implements an ensemble learning approach wherein the robustness to hyperparameters is ensured by joining the clustering results of many Gaussian mixture models (GMM) to form the final kernel. Hence, no critical hyperparameters must be tuned by the user.

To deal with missing data, the GMMs are extended using informative prior distributions [11]. The TCK matrix is built by fitting GMMs to the set of time series for a range of numbers of mixture components, to provide partitions with different resolutions that capture both local and global structures in the data. To enhance diversity in the ensemble, each partition is evaluated on a random subset of attributes and segments, using random initializations and randomly chosen hyperparameters. This also provides robustness in the hyperparameters selection. TCK is then built by summing (for each partition) the inner products between pairs of posterior distributions corresponding to different MTS.

ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2018, i6doc.com publ., ISBN 978-287587047-6. Available from http://www.i6doc.com/en/.

2.2 Autoencoder

AEs simultaneously learn two functions. The first one, *encoder*, provides a mapping from an input domain, $\mathcal{X}$, to a code domain, $\mathcal{C}$, i.e., the hidden representation. The second function, *decoder*, maps from $\mathcal{C}$ back to $\mathcal{X}$. In AEs with a single hidden layer, the encoding and decoding function are $\mathbf{c} = \phi(\mathbf{W}_E\mathbf{x} + \mathbf{b}_E)$ and $\tilde{\mathbf{x}} = \psi(\mathbf{W}_D\mathbf{c} + \mathbf{b}_D)$, where $\mathbf{x}$, $\mathbf{c}$, and $\tilde{\mathbf{x}}$ denote, respectively, a sample from the input space, its hidden representation (the *code*), and its reconstruction. While $\phi(\cdot)$ is usually implemented as a sigmoid, in the case inputs are real-valued vectors, the squashing nonlinearity in $\psi(\cdot)$ can be replaced by a linear activation. Finally, $\mathbf{W}_E$ and $\mathbf{W}_D$ are the weights and $\mathbf{b}_E$ and $\mathbf{b}_D$ the bias of the encoder and decoder, respectively.

To minimize the discrepancy between the input and its reconstruction, model parameters are learned by minimizing a reconstruction loss

$$L_r(\mathbf{x}, \tilde{\mathbf{x}}) = \mathbb{E}\left\{\|\mathbf{x} - \tilde{\mathbf{x}}\|^2\right\} . \tag{1}$$

By stacking more hidden layers an AE is capable of learning more complex representations by transforming inputs through multiple nonlinear transformations. In its native formulation, an AE can process vectorial data and, therefore, a MTS is flattened into a uni-dimensional vector when fed to the AE. Since an AE processes inputs of same lengths, missing are filled with numeric values.

2.3 Deep Kernelized Autoencoder

A dkAE is trained by minimizing the loss function

$$L = (1 - \lambda)L_r(\mathbf{x}, \tilde{\mathbf{x}}) + \lambda L_c(\mathbf{C}, \mathbf{K}), \tag{2}$$

where $L_r(\cdot, \cdot)$ is the reconstruction loss in Eq. 1 and λ is a hyperparameter that balances the contribution of the two cost terms. If $\lambda = 0$, L becomes the traditional AE loss in Eq. 1. $L_c(\cdot, \cdot)$ is the *code loss* that enforces similarity between two matrices: $\mathbf{K} \in \mathbb{R}^{N \times N}$, the kernel matrix given as prior, and $\mathbf{C} \in \mathbb{R}^{N \times N}$, the inner product matrix of codes associated to input data (N is the number of samples in the dataset). A depiction of the training procedure is reported in Fig. 1.

$L_c(\cdot, \cdot)$ can be implemented as the normalized Frobenius distance between $\mathbf{C}$ and $\mathbf{K}$. Each matrix element C_{ij} in $\mathbf{C}$ is given by $C_{ij} = \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j)$ and the code loss reads

$$L_c(\mathbf{C}, \mathbf{K}) = \left\| \frac{\mathbf{C}}{\|\mathbf{C}\|_F} - \frac{\mathbf{K}}{\|\mathbf{K}\|_F} \right\|_F . \tag{3}$$

By minimizing the normalized Frobenius distance from TCK, we indirectly include in the codes the information it captures about the missingness patterns and we improve the quality of the learned codes in presence of missing data.

The dkAE model is trained using mini-batches. Therefore, a training matrix $\mathbf{C}_m$ is generated from the codes associated to the elements in the mth mini-batch and distance L_c is computed on the submatrix of $\mathbf{K}$ related to the entries in the mini-batch m.

ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2018, i6doc.com publ., ISBN 978-287587047-6.
Available from http://www.i6doc.com/en/.

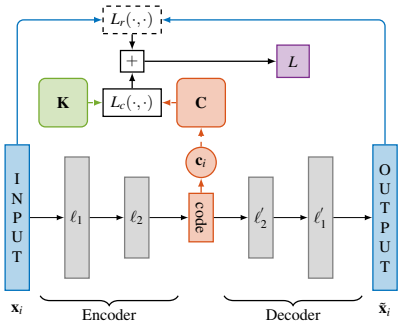

Fig. 1. Schematic illustration of dkAE architecture. The total loss function L depends on two terms. First, $L_r(\cdot,\cdot)$, which computes the reconstruction error between true input $\mathbf{x}_i$ and output of dkAE, $\tilde{\mathbf{x}}_i$. The second term, $L_c(\cdot,\cdot)$, is the distance measure between the matrices $\mathbf{C}$ (computed as inner products of codes $\{\mathbf{c}_i\}_{i=1}^N$) and the target prior kernel matrix $\mathbf{K}$.

3 Experiments

We analyze blood measurements collected from patients undergoing a gastrointestinal surgery at University Hospital of North Norway in the years 2004–2012. Each patient in the dataset is represented by a MTS of blood samples extracted within 20 days after surgery. The MTS contain measurements of 10 variables, which are alanine aminotransferase, albumin and alkaline phosphatase, creatinine, CRP, hemoglobine, leukocytes, potassium, sodium and thrombocytes. We focus on a cohort of two classes of patients: the ones with and without surgical site infections. Dataset labels are assigned according to International Classification of Diseases and NOMESCO Classification of Surgical Procedures, relative to patients with severe postoperative complications.

Missing data in MTS correspond to measurements that are not collected for a given patient in one day of the observation period. Patients with less than two measurements are excluded from the cohort. We ended up with 883 MTS, of which 232 are patients with infections. The first 80% of the datasets is used as training set and the rest as test set.

The dataset, the code implementing all the methods described in this paper, and a detailed description of experimental setup are publicly available[1].

3.1 Results

To evaluate the effect of the alignment with TCK kernel, we compare the classification results obtained on the codes learned by standard AE and dkAE. Missing values are filled with three different imputation techniques: zero imputation (AE-z and dkAE-z), mean imputation (AE-m and dkAE-m) and last-value-carried-forward imputation (AE-l and dkAE-l). The codes are classified by a k-NN with $k = 3$ equipped with Euclidean distance. We also consider the results yielded in the input space by a kNN with TCK similarity (TCK-i).

In Tab. 1 we report the mean and standard deviation of F1 score and area under the ROC curve (AUC) of the test set in 10 independent runs. For AE and dkAE we also report the mean squared error (MSE) between the encoder input and the decoder output. A low MSE of the reconstruction does not only

[1] `https://github.com/FilippoMB/TCK_AE`

ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2018, i6doc.com publ., ISBN 978-287587047-6. Available from http://www.i6doc.com/en/.

guarantee to learn a better representation of the input, but it implies an accurate back-mapping from code to input space. In both AE without kernel alignment and dkAE with zero imputation and last-value-carried-forward we obtain the best and worst classification performance, respectively.

For each imputation method, codes learned by dkAE are classified more accurately and the reconstruction error does not increase even if the codes are aligned with the prior kernel. This demonstrate the importance of embedding into the codes the similarity information yielded by TCK, which captures missingness patterns. Indeed, those patterns are ignored if one relies solely on imputation, whose purpose is to fill missing entries introducing as less bias as possible. It is interesting to notice that the classification in the input space based on TCK similarity is slightly less accurate than the classification in the code space of dkAE. Therefore, dkAE not only yields codes of reduced dimensionality that can be handled more easily and processed faster, but they are discriminated easier than the inputs themselves from a simple classifier.

Method	MSE	F1	AUC
`AE-z`	0.103±0.002	0.654±0.028	0.751±0.018
`dkAE-z`	0.096±0.001	**0.748**±0.017	**0.813**±0.011
`AE-m`	0.094±0.003	0.569±0.035	0.7034±0.02
`dkAE-m`	**0.091**±0.001	0.690±0.029	0.773±0.018
`AE-l`	0.136±0.002	0.662±0.010	0.764±0.006
`dkAE-l`	0.128±0.000	0.678±0.026	0.763±0.016
`TCK-i`	–	0.698±0.021	0.776±0.012

Table 1. Reconstruction MSE and classification results of the codes learned by AE and dkAE. We also report the classification results in the input space using TCK as similarity. In AE and dkAE we apply three different imputations: zero imputation (z), mean imputation (m) and last value carried forward (l). Best results are highlighted in bold.

In Fig. 2 we visualize the first two PCA components of the test set, both in input and in the code spaces. We compute a linear PCA on the codes and on the TCK kernel matrix (this corresponds to compute kernel PCA in the input space using TCK as kernel). Coloring depends on the ground truth label and we observe the two classes to be better separated in the code space of dkAE. Interestingly, in dkAE we notice the same structure yield by kPCA in the input space with TCK as kernel. This demonstrate how the kernel alignment procedure successfully embed in the codes the properties of TCK, without compromising the precision of the decoder reconstruction. We underline that by using an AE rather than kPCA we avoid performing a costly eigendecomposition and we also learn the inverse mapping from the code to the input space, provided by the decoder.

4 Conclusions

In this paper, we proposed a novel approach for learning compressed vectorial representations of MTS with missing values, which are common in clinical records. This is achieved by combining a deep kernelized Autoencoder with TCK, a similarity measure for MTS that accounts for missing values. We tackled the classification of blood samples from patients with postoperative infections,

ESANN 2018 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 25-27 April 2018, i6doc.com publ., ISBN 978-287587047-6.
Available from http://www.i6doc.com/en/.

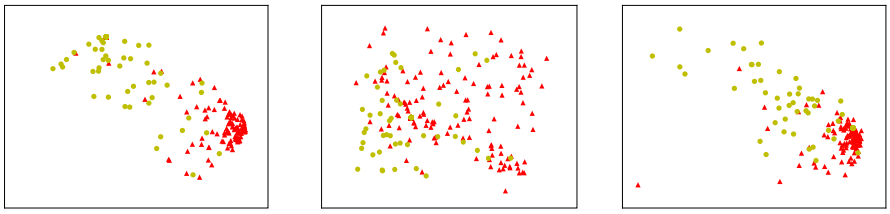

Fig. 2. Projection of test set on the first two PCA components using (i) kPCA on the input space, (ii) PCA on AE code space, and (iii) PCA on dkAE code space. Yellow dots and red triangles represent infected and non-infected patients respectively.

where data are MTS with a high percentage of missing data. Our results showed that by aligning the codes in the AE to TCK kernel matrix, we embed into the representation important information relative to patterns of missingness in the data and improve the classification outcome.

References

[1] Y. Cheng, F. Wang, P. Zhang, and J. Hu. Risk prediction with electronic health records: A deep learning approach. In *Proceedings of the 2016 SIAM International Conference on Data Mining*. SIAM, 2016.

[2] Z. Che, S. Purushotham, K. Cho, D. Sontag, and Y. Liu. Recurrent neural networks for multivariate time series with missing values. *arXiv preprint arXiv:1606.01865*, 2016.

[3] Zhengping Che, Sanjay Purushotham, David Kale, Wenzhe Li, Mohammad Taha Bahadori, Robinder Khemani, and Yan Liu. Time series feature learning with applications to health care. In *Mobile Health*. Springer, 2017.

[4] R. Miotto, L. Li, B. A Kidd, and J. T Dudley. Deep patient: An unsupervised representation to predict the future of patients from the electronic health records. *Scientific reports*, 2016.

[5] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 2006.

[6] Y. Bengio. Learning deep architectures for AI. *Foundations and trends in Machine Learning*, 2009.

[7] M. Längkvist, L. Karlsson, and A. Loutfi. A review of unsupervised feature learning and deep learning for time-series modeling. *Pattern Recognition Letters*, 2014.

[8] K. Ø. Mikalsen, F. M. Bianchi, Cristina Soguero-Ruiz, Stein Olav Skrøvseth, Rolv-Ole Lindsetmo, Arthur Revhaug, and Robert Jenssen. Learning similarities between irregularly sampled short multivariate time series from ehrs. *International Conference on Pattern Recognition*, 2016.

[9] K. Ø. Mikalsen, F. M. Bianchi, C. Soguero-Ruiz, and R. Jenssen. Time series cluster kernel for learning similarities between multivariate time series with missing data. *Pattern Recognition*, 2017.

[10] M. Kampffmeyer, S. Løkse, F. M. Bianchi, R. Jenssen, and L. Livi. Deep kernelized autoencoders. In *Scandinavian Conference on Image Analysis*. Springer, 2017.

[11] B. M. Marlin, D. C. Kale, R. G. Khemani, and R. C. Wetzel. Unsupervised pattern discovery in electronic health care data using probabilistic clustering models. In *Proc. of 2nd ACM SIGHIT Int. Health Informatics Symposium*, 2012.