ESANN 2016 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2016, i6doc.com publ., ISBN 978-287587027-8.
Available from http://www.i6doc.com/en/.

Learning Embeddings for Completion and Prediction of Relationnal Multivariate Time-Series

Ali Ziat[1,2], Gabriella Contardo[2], Nicolas Baskiotis[2], Ludovic Denoyer[2]

1- Institut VEDECOM, Versailles, France

2- Sorbonne Universités, UPMC Univ Paris 06,
UMR 7606, LIP6, F-75005, Paris, France

Abstract. We focus on learning over multivariate and relational time-series where relations are modeled by a graph. We propose a model that is able to simultaneously fill in missing values and predict future ones. This approach is based on representation learning techniques, where temporal data are represented in a latent vector space so as to capture the dynamicity of the process and also the relations between the different sources. Information completion (missing values) and prediction are performed simultaneously using a unique formalism, whereas most often they are addressed separately using different methods.

1 Introduction

Temporal data correspond to a wide variety of phenomena from stock market to internet traffic forecasting. Different kind of temporal data can be produced: monovariate and multivariate time series where the produced values are real values. The recent emergence of sensors everywhere - e.g mobile phones which typically produce temporal sequences of complex data (GPS, events, ...) - is an example that illustrates the need of new machine learning models for temporal data processing. Indeed, the produced information has particular characteristics that can't be handled by classical sequential and temporal models: they contain multiple missing values and one has to consider simultaneously multiple sources that can be somehow related, by spatial proximity for example. (i)On one side, several models have been proposed for multivariate time-series or sequences prediction. The most popular are probably neural networks [1] - an overview of related methods is given in Section 5. However, these models have not been conceived for dealing with missing data. On the other side, some models have been also proposed with the goal of automatically completing missing information (data imputation) [2], like matrix factorization techniques [3]. But data imputation and data prediction are usually seen as two different problems. Moreover, two time series can be related to each other. At the best of our knowledge, there exist no model able to deal with all these characteristics and tasks conjointly. We propose a novel method that aims at integrating all the aspects of complex temporal data in one single model. The proposed approach is based on representation learning techniques aiming at projecting the observations in a continuous latent space, each sequence being modeled at each time-step by

ESANN 2016 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2016, i6doc.com publ., ISBN 978-287587027-8.
Available from http://www.i6doc.com/en/.

a point in this space. It has many advantages w.r.t existing techniques: (i) it is able to simultaneously learn how to fill missing values and to predict the future of the observed temporal data, avoiding to use two different models, (ii) it naturally allows one to deal with information sources that are organized among a graph structure (iii) the model is based on continuous optimization schemes, allowing a fast optimization over large scale datasets.

1.1 Related Work

The representation learning (and deep learning) is a very active field where different recent works target some of the aspects studied here. For example, the problem of learning representation over sequential data has been used with different approaches like Recurrent Neural Networks [4]. The main difference w.r.t our model is that the RNN-based methods are inductive (the representation is induced from observations) while our model is a transductive model (the observations are induced from learned representations) which makes it more suitable for the completion problem: missing values are built from the representations. Our model is also related to multi-view deep neural networks like [5] . Here also, our transductive approach is more suitable for data where some of the views are missing. As far as we know, there is no existing model in this community that mix relational information, temporal information and heterogeneity.
At last, the traffic prediction problem is an old topic. In particular, different techniques have been evaluated like ARIMA (Autoregressive Integrated Moving Average) which have been applied in the context of multivariate time-series [6]. For a large overview on this techniques in the field of traffic forecasting, you can refer to [7]. In practice, neural networks are predominant and are at the center of a large number of publications which propose different architectures [8]. Neural Networks are also the baseline competitor which is often used [9].

1.2 Notations

We consider a set of n temporal sequences $\mathbf{x_1}, ..\mathbf{x_n}$ such that $x_i^{(t)} \in \mathcal{X}$ is the value of the i-th sequence at time t defined by $\mathbf{x_i} = (x_i^{(1)}, .., x_i^{(T)})$. The sequences contain missing values so we also define a mask $m_i^{(t)}$ such that $m_i^{(t)} = 1$ if value $x_i^{(t)}$ is observed - and thus available for training the system - and $m_i^{(t)} = 0$ if $x_i^{(t)}$ is missing - and thus has to be predicted by the model. In addition, we consider that there exists a set of relations between the sequences which correspond to an external information, like spatial proximity for example when $\mathcal{X}$ is discrete. The sequences are thus organized in a graph $\mathcal{G} = \{e_{i,j}\}$ such that $e_{i,j} = 1$ means that $\mathbf{x_i}$ and $\mathbf{x_j}$ are related, and $e_{i,j} = 0$ elsewhere. The two tasks that we want to (conjointly) solve are the following: (i) The problem of prediction consists in predicting *what happens next* given the observed temporal data. (ii) Data completion consists in missing values inference in the sequences.

ESANN 2016 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2016, i6doc.com publ., ISBN 978-287587027-8.
Available from http://www.i6doc.com/en/.

2 Representation-baSed temporal relational model

The goal of our model is to take into account the different types of information available in the dataset which are: (i) the observed values, (ii) the relations between the information sources and (iii) the dynamicity of the system. We propose to capture this information in a large dimensional latent space $\mathcal{Z}$ in which each observation will correspond to a particular point (or embedding) at each time step, denoted $z_i^{(t)} \in \mathcal{Z}$. The **RepresentAtIoN-baSed TempOral Relational Model** (RAINSTORM) is a loss-based model which is described through a continuous derivable loss function that will be optimized using classical optimization techniques. The approach proposed is close to the ones of the deep learning community but the proposed model is different from classical deep neural networks techniques since an explicit representation $z_i^{(t)}$ is learned for each time-step and each source. The main interest of this approach is to be able to deal easily with missing values while classical NN-based techniques are less suitable for this case.

Let us define $\mathcal{L}(\theta, \gamma, \mathbf{z})$ the loss function to minimize where $\mathbf{z}$ is the set of all the vectors $z_i^{(t)}$ for $i \in [1..n]$ and $t \in [1..T]$, T being the size of the observed time windows i.e. the history of the time series. We define $\mathcal{L}$ as:

$$\mathcal{L}(\theta, \gamma, \mathbf{z}) = \underbrace{\frac{1}{O} \sum_{i=1}^{n} \sum_{t=1}^{T} m_i^{(t)} \Delta(f_\theta(z_i^{(t)}), x_i^{(t)})}_{\text{(term 1)}} + \lambda_{dyn} \underbrace{\sum_{i=1}^{n} \sum_{t=1}^{T-1} ||z_i^{(t+1)} - h_\gamma(z_i^{(t)})||^2}_{\text{(term 2)}}$$
$$+ \lambda_{struct} \sum_{i,j \in [1..N]^2} \sum_{t=1}^{T} e_{i,j} ||z_i^{(t)} - z_j^{(t)}||^2 \text{ (term 3)} \tag{1}$$

where O is the number of observed values i.e. values such that $m_i^{(t)} = 1$.

This loss function contains three terms, each one associated with one of the constraints that have been presented previously: Term 1 simultaneously learn $\mathbf{z}$ and a function f_θ - called **decoding function** - such that, from $z_i^{(t)}$, f_θ can be used to predict the value $x_i^{(t)}$. The function $f_\theta(z_i^{(t)})$ is defined as $f_\theta : \mathbb{R}^N \to \mathcal{X}$. Δ is used to measure the error between predicting $f_\theta(z_i^{(t)})$ instead of $x_i^{(t)}$, $m_i^{(t)}$ playing the role of a mask restricting to compute this function only on the observed values. Term 2 aims at finding values $z_i^{(.)}$ and a dynamic model h_γ such that, when applied to $z_i^{(t)}$, h_γ allows us to predict the representation of the next state of time series i i.e. $z_i^{(t+1)}$. h_γ is the **dynamic function** which models the dynamicity of each series directly in the latent space: $h_\gamma : \mathbb{R}^N \to \mathbb{R}^N$. At last, term 3 corresponds to a **structural regularity** over the graph structure that encourages the model to learn closer representations for time series that are related. This will force the model to learn embeddings that reflect the structure of the considered graph. λ_{dyn} and λ_{struct} are manually defined coefficients that weight the importance of the different elements in the loss function. For f_θ and

ESANN 2016 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2016, i6doc.com publ., ISBN 978-287587027-8.
Available from http://www.i6doc.com/en/.

h_γ, different architectures can be chosen; we propose one in the experimental section.

2.1 Learning and Inference

The learning problem aims at minimizing the loss function $\mathcal{L}(\theta, \gamma, \mathbf{z})$ simultaneously on θ, γ and $\mathbf{z}$. By restricting the f_θ and h_γ to be continuous derivable functions, we can use classical stochastic gradient-descent (SGD) based optimization approaches. Let us now describe how inference is made.

Completion of missing values: For all missing values $x_i^{(t)}$ such that $m_i^{(t)} = 0$, the proposed learning algorithm has learned a z-value $z_i^{(t)}$ in the latent space. This learned value has been mainly 'chosen' based on term 2 and 3 of the loss function (Equation 1) while the decoding term has not been tuned on $z_i^{(t)}$ since $x_i^{(t)}$ is unknown. In order to predict this missing value, our approach simply computes the value $f_\theta(z_i^{(t)})$ which produces a plausible output value.

Predicting the future: For all $t > T$, the model does not compute z-values and these $z_i^{(t)}$ are unknown. But our model learns a dynamic function h_γ which goal is to allow the prediction of $z_i^{(t+1)}$ given $z_i^{(t)}$. So, for any i, $z_i^{(T+1)}$ can be computed by $h_\gamma(z_i^{(T)})$. The future value $x_i^{(T+s)}$ can thus be predicted by simply computing $f_\theta(h_\gamma(h_\gamma(h_\gamma(.....h_\gamma(z_i^{(T)})..))))$ where h_γ is applied s times, the obtained vector being then transformed to prediction by using f_θ.

3 Experiments

We consider a road network as a graph where each node corresponds to a road, and edges are connections between roads: two roads are connected if they belong to the same crossroads. Experiences have been made on two datasets corresponding to two different cities: Warsaw and Beijing which are composed of about 20,000 roads (see [10] and [11] for complete statistics). The sampling frequency is about 10 minutes, and at each timestep some sensors return measures over a subset of roads (i.e. the roads such that $m_i^{(t)} = 1$). This measure is typically the average speed of cars on the road (for Warsaw dataset), or a measure of the volume of cars (for Beijing). To evaluate our method, we consider a set of training values and a set of testing values. Testing values are of two types: part of the testing values are sampled uniformly in the set of observed values for $t \in [1..T]$, T being the size of the observed data. These testing values will be used for evaluating the quality of the completion model. All values for $t > T$ are considered as testing values and will be used for evaluating our model in prediction. Moreover, a sub-part of the testing values will be used as validation data for tuning the hyperparameters.

3.1 Models

Different architectures for RAINSTORM have been tested and we have kept the following by validation: it uses a one hidden layer neural network for h_γ with

ESANN 2016 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2016, i6doc.com publ., ISBN 978-287587027-8. Available from http://www.i6doc.com/en/.

N	Model/Data	Beijing	Warsaw
	RoadMean	5.51	11.02
	NeuralNetwork	4.77	8.05
	SAE	4.75	7.85
5	RAINSTORM	4.82	7.74
10	RAINSTORM	4.78	7.21
20	RAINSTORM	**4.54**	**7.19**
50	RAINSTORM	4.66	7.60

Table 1: Prediction performance at $T+1$ for different size of latent space N using a root mean square error (RMSE)

Model/Data	Beijing	Warsaw
RoadMean	5.55	11.10
MF	3.58	6.80
MF-Geo	3.24	6.49
RAINSTORM	2.99	6.49
RAINSTORM	3.03	6.24
RAINSTORM	3.22	**6.23**
RAINSTORM	**2.97**	6.70

Table 2: Completion performance for 50% missing data for different sizes N of the latent space using a root mean square error (RMSE)

200 hidden neurons and a hyperbolic tangent as activation function. A Simple Linear function is used for the decoding function. It has been compared with the following baselines.

Completion: **MF**: This correspond to the classical matrix factorization framework, described for instance for the task of traffic forecasting in [12]. **MF-with geographic context**: This method is the one named TSE (traffic speed estimation) in [12]. It consists on minimizing a reconstruction cost on a traffic matrix for which external information such as geographic position in a city is incorporated.

Prediction: **NeuralNetwork**: This is a classical baseline method used in traffic forecasting based on a neural network architecture, described for instance in [13]. **SAE**: This is the method described in [14] ; it consists on a deep architecture of stacked auto-encoders trained on the traffic history. **RoadMean** predicts and fills missing values with the mean of observed values on the whole sequence.

3.2 Experiments and Results

We first focus on filling missing values in the two datasets. We have used 50% of the observations for training, 40% for testing and 10% for validation. Each performance has been computed by averaging the results obtained on 20 different runs. Table 2 illustrates the results obtained by the baselines and by our models, considering different sizes N of the latent space. One can see that the three RAINSTORM models outperforms the baselines for almost all the tested dimensions N.

For the second set of experiments, we focus on the prediction problem. Here, the values at time $t > T$ are removed from the training set; 20% of the removed values are used as validation set and the 80% remaining are the test set. Table 1 shows the performance of the different models for the prediction task using a RMSE evaluation at $t = T + 1$ on the volume or average speed of the cars on

ESANN 2016 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2016, i6doc.com publ., ISBN 978-287587027-8. Available from http://www.i6doc.com/en/.

each road. It also shows that, used as a prediction model, RAINSTORM obtains better results than baseline techniques.

4 Conclusion

We have presented a new way to learn over incomplete multiple sources of temporal relational data sources. Our approach is based on representation learning techniques and aims at integrating in a latent space the observed information, the dynamicity of the sequences of data, and their relations. In comparison to baselines models that have been developed for prediction only or completion only, our approach shows interesting performance and is able to simultaneously complete missing values and predict the future evolution of the data.

References

[1] Kanad Chakraborty, Kishan Mehrotra, Chilukuri K Mohan, and Sanjay Ranka. Forecasting the behavior of multivariate time series using neural networks. *Neural networks*, 5(6), 1992.

[2] James Honaker and Gary King. What to do about missing values in time-series cross-section data. *American Journal of Political Science*, 54(2):561–581, 2010.

[3] Jingbo Shang, Yu Zheng, Wenzhu Tong, Eric Chang, and Yong Yu. Inferring gas consumption and pollution emission of vehicles throughout a city. In *KDD 2014*, 2014.

[4] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376. ACM, 2006.

[5] Yoonseop Kang and Seungjin Choi. Restricted deep belief networks for multi-view learning. In *Machine Learning and Knowledge Discovery in Databases*. Springer, 2011.

[6] Anthony Stathopoulos and Matthew G Karlaftis. A multivariate state space approach for urban traffic flow modeling and prediction. *Transportation Research Part C: Emerging Technologies*, 11(2):121–135, 2003.

[7] Eleni I Vlahogianni, John C Golias, and Matthew G Karlaftis. Short-term traffic forecasting: Overview of objectives and methods. *Transport reviews*, 24(5):533–557, 2004.

[8] Corrado De Fabritiis, Roberto Ragona, and Gaetano Valenti. Traffic estimation and prediction based on real time floating car data. In *Intelligent Transportation Systems, 2008. ITSC 2008. 11th International IEEE Conference on*, pages 197–203. IEEE, 2008.

[9] Marco Lippi, Matteo Bertini, and Paolo Frasconi. Short-term traffic flow forecasting: An experimental comparison of time-series analysis and supervised learning. *Intelligent Transportation Systems, IEEE Transactions on*, 14(2):871–882, 2013.

[10] I.e.e.e icdm contest: Tomtom traffic prediction for intelligent gps navigation.

[11] Yu Zheng. T-drive trajectory data sample, August 2011.

[12] Jingbo Shang, Yu Zheng, Wenzhu Tong, Eric Chang, and Yong Yu. Inferring gas consumption and pollution emission of vehicles throughout a city. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2014.

[13] Mark S Dougherty and Mark R Cobbett. Short-term inter-urban traffic forecasts using neural networks. *International journal of forecasting*, 13(1):21–31, 1997.

[14] Yisheng Lv, Yanjie Duan, Wenwen Kang, Zhengxi Li, and F-Y Wang. Traffic flow prediction with big data: A deep learning approach.