Seq2VAR: multivariate time series representation with relational neural networks and linear autoregressive model

Edouard Pineau, Sebastien Razakarivony, Thomas Bonald

Seq2VAR: multivariate time series representation with relational neural networks and linear autoregressive model

Edouard Pineau[†⋆], Sébastien Razakarivony[†], Thomas Bonald[⋆]

Safran Tech, Signal and Information Technologies[†]
Telecom Paris, Institut Polytechnique de Paris[⋆]

Abstract. Finding understandable and meaningful feature representation of multivariate time series (MTS) is a difficult task, since information is entangled both in temporal and spatial dimensions. In particular, MTS can be seen as the observation of simultaneous causal interactions between dynamical variables. Standard way to model these interactions is the vector linear autoregression (VAR). The parameters of VAR models can be used as MTS feature representation. Yet, VAR cannot generalize on new samples, hence independent VAR models must be trained to represent different MTS. In this paper, we propose to use the inference capacity of neural networks to overpass this limit. We propose to associate a relational neural network to a VAR generative model to form an encoder-decoder of MTS. The model is denoted Seq2VAR for Sequence-to-VAR. We use recent advances in relational neural network to build our MTS encoder by explicitly modeling interactions between variables of MTS samples. We also propose to leverage reparametrization tricks for binomial sampling in neural networks in order to build a sparse version of Seq2VAR and find back the notion of Granger causality defined in sparse VAR models. We illustrate the interest of our approach through experiments on synthetic datasets.

Keywords: Multivariate time series, Vector linear autoregression, Relational neural networks, Granger causality

1 Introduction

Nowadays, more and more data come as multivariate time series (MTS) and finding understandable and meaningful feature representation of observed MTS samples is needed for information mining and downstream machine learning tasks. Among standard MTS representation and modeling [3], vector linear autoregression (VAR) [20] captures dynamical and causal information contained in the data. Yet, VAR does not have representation inference mechanism: one model fits all samples. Consequently, to represent different MTS, independent VAR models must be trained. This operation becomes very expensive when VAR parameters are constrained, for example with sparsity [1] or symmetry, since there are no closed-form nor simple regularization to solve it. In contrast, efficient and

powerful inference is a key advantage of neural networks based representation learning.

Many recent works on neural generative models have been used for representation learning, with particular attention on variational autoencoders (VAE) [7,21]. The VAE, as a latent variable model, finds the joint distribution between observed data and a set of latent variables. With appropriate assumptions on the variational distribution and regularization, one hopes to uncover and disentangle independent causal sources of variations in the data, with a certain form of interpretability and a good representation power. However, finding such disentangled representation cannot be successfully done without an appropriate inductive bias [11].

In this context, a specialized VAE architecture, called neural relational inference (NRI), has recently been developed for interaction inference in MTS [8]. NRI uses the concept of relational learning [17] to explicitly capture the variable interactions that explain the data dynamics. The inferred binary interaction graph is used as a variable selection preceding a nonlinear auto-regressive decoding scheme. The objective, with an appropriate setting, is both to help the generative decoder with sparsity and to uncover the real interactions in an unsupervised manner. Nevertheless, this approach suffers from its powerful decoder, that does not necessarily need meaningful latent representation to recover data. The general problem of information vanishing in the presence of a too expressive decoder is classical [13,19]: in extreme cases, the inference part becomes meaningless, transforming the VAE into a standard generative auto-regressive model [9]. In our experiments, we show that in the NRI case, this problem manifests when the interactions to discover are heterogeneous, i.e., the latent interaction graph is weighted. As a result, the unsupervised interaction discovery does not correspond anymore to true physical reality while the model still decodes properly. Finally, by construction, NRI only covers information contained in first previous time step for prediction part. Therefore, it needs lagged information as explicit input, which mechanically increases the number of parameters to train.

In this paper, we propose an alternative, simple MTS representation learning framework that exploits both VAR generative models and neural network inference capacities. As in [8], we assume that MTS are the observation of a system that can be represented with causal information. Following the classical framework of encoder-decoder, we build the Seq2VAR model, whose architecture presents three main advantages. First, we prevent information vanishing by representing our MTS samples as the parameters of a VAR instead of the input of a neural auto-regressive decoder. Second, we leverage recent advances in neural networks binary sampling [6,10,12] to build explicit sparse representations; this way, non-zero entries of the inferred representation can be interpreted as Granger causalities. Third, we help our model generalizing over the notion of interactions by using a relational inference neural network (RINN) [8] as MTS encoder. These three properties make Seq2VAR able to find meaningful MTS representations.

After presenting Seq2VAR model and assumptions, we perform some experiments to assess its different advantages. We then discuss (1) the interest of using RINN encoder instead of recurrent neural networks (RNN), that are explicitly made for time series modeling, and (2) the relations and differences between Seq2VAR and the NRI model.

2 Model description

2.1 Notations and assumptions

Let $\mathcal{X} \subseteq \mathbb{R}^{d \times T}$ be some finite set of n d-dimensional MTS defined over the time range $t = 1, \dots, T$. We assume that each MTS $X^{(s)}$, $s = 1, \dots, n$ is generated by some dynamical system. Specifically, we assume that each MTS follows a K-order linear auto-regressive model, i.e., there exists some tensor $A^{(s)} \in \mathbb{R}^{K \times d \times d}$ such that the MTS $X^{(s)} = (X_1^{(s)}, \dots, X_T^{(s)}) \in \mathcal{X}$ satisfies $X_t^{(s)} = \sum_{k=1}^{K} A_k^{(s)} X_{t-k}^{(s)}$ for all $t = K+1, \dots, T$. Both the tensor $A^{(s)}$ and the initial values $(X_1^{(s)}, \dots, X_K^{(s)})$ are unknown and can change from one sample to the other. Some additive observation noise can be added to the samples. In this paper, we assume that the set of tensors $\{A^{(s)}, s = 1, \dots, n\}$ belongs to $\mathcal{A}$, the set of acceptable dynamical systems. This set $\mathcal{A}$ is also unknown.

2.2 General problem setup

We place ourselves in the encoder-decoder framework. We use a relational inference neural network (RINN) as MTS encoder [8], which takes a sample $X^{(s)}$ as input and outputs a tensor $\hat{A}^{(s)}$. The decoder is a VAR model parametrized by $\hat{A}^{(s)}$. This tensor is therefore the latent representation of the MTS, with respect to the linear decoding assumption. We denote the encoder by F_ϕ, where ϕ are a parameter in some finite-dimensional space. Observe that F_ϕ is a mapping from $\mathbf{R}^{d \times T}$ to $\mathbb{R}^{K \times d \times d}$. The parameter ϕ is optimized for the objective function:

$$\sum_{s=1}^{n} \left\{ \sum_{t=K+1}^{T} \left\| X_t^{(s)} - \sum_{k=1}^{K} F_\phi(X^{(s)})_k X_{t-k}^{(s)} \right\|_2^2 + \lambda \Omega \left(F_\phi(X^{(s)}) \right) \right\}. \qquad (1)$$

Here Ω is a regularizing penalty function and λ a positive coefficient of penalization. These can be used to control the sparsity of the output (number of non-zero entries of the tensor $F_\phi(\cdot)$). We call our model Seq2VAR (sequence-to-VAR).

2.3 Sparsity in Seq2VAR

Seq2VAR offers the possibility to modify the encoding posterior distribution and the regularization depending on the task of interest. In particular, the output of the encoder F_ϕ may be made sparse [1,4,22]. For such constraint, we use the recent advances in sparse representation learning based on derivations of Gumbel-Softmax trick [5,14].

In theory, true sparse learning rely on L_0 regularization. In practice L_0 norm cannot be directly used as penalty in the objective function since it is non-differentiable. General approaches uses Ridge, LASSO or group-LASSO regularization [4] as proxy for L_0 regularization. This sparsity becomes implicit and is difficult to control. As an alternative and to obtain explicit sparsity, we propose to use stochastic gates determining for each samples $X^{(s)}$ which entries of $F_\phi(X^{(s)})$ are set to zero [12]. The probability of each gate is an additional output of the RINN encoder. Gates are sampled using a continuous relaxation of binomial sampling [5,14], to let the gradient flow. The encoder outputs the probability of opening the gates, denoted $P_\phi(X^{(s)})$. We denote $G_\phi(X^{(s)})$ the binary gates that are sampled with the following derivation of Gumbel-softmax trick proposed in [10]:

$$G_\phi(X^{(s)}) = \sigma\left(\frac{P_\phi(X^{(s)}) + \log U - \log(1-U)}{\tau}\right), \tag{2}$$

with σ the sigmoid function, τ the temperature of the relaxation (the higher, the smoother the approximated binary sampling) and $U \sim \mathcal{U}(0,1)$. The σ, log and $+$ are matrix element-wise operators, U has the same dimensions than $P_\phi(\cdot)$, with all entries sampled independently. Depending on the task, $G_\phi(\cdot)$ lives either in $\mathbb{R}^{K\times d\times d}$ or in $\mathbb{R}^{1\times d\times d}$. The latter corresponds to variable selection, while the former corresponds to variable and lag selection. The collection of gates forms a binary mask that is directly applied to the dense tensor $F_\phi(\cdot)$.

We can now control the sparsity of the model with an appropriate penalty function Ω applied to $G_\phi(\cdot)$ in the following sparse problem:

$$\sum_{s=1}^{n}\left\{\sum_{t=K+1}^{T}\left\|X_t^{(s)} - \sum_{k=1}^{K}\left(G_\phi(X^{(s)}) \odot F_\phi(X^{(s)})\right)_k X_{t-k}^{(s)}\right\|_2^2 + \lambda\Omega\left(G_\phi(X^{(s)})\right)\right\}, \tag{3}$$

where $\odot$ denotes component-wise multiplication.

In [12], the authors propose a method for creating sparse neural networks with a similar method, in a Bayesian deep learning setting. They regularize the sparsity with the norm of the probability of the gates. In (3) we generalize this method to representation inference. We use the number of positive gates, i.e., the sum of all the entries of the (almost) binary mask as a proxy for the norm L_0 on encoder output. Thanks to the continuous relaxation of the binary sampling (2), this sum is differentiable with respect to ϕ and can be used as a penalization in the objective function.

Remark 1. A particular case of sparse approach for Seq2VAR is the pure binary, i.e., by using only $G_\phi(\cdot)$ as a VAR parameters. This case is studied in the Experiment 3.2. In the experiments, we denote such Seq2VAR model as binary-Seq2VAR.

2.4 Symmetry in Seq2VAR

Another constraint that can be easily imposed to Seq2VAR is the symmetry of the tensor $\hat{A}^{(s)}$. Thanks to the inference mechanism, we can force $\hat{A}^{(s)}$ to be

symmetric by imposing $\hat{A}^{(s)} = 1/2(F_\phi(X^{(s)}) + F_\phi(X^{(s)})^{T(1,2)})$, where $T(1,2)$ is the transposition of dimensions 1 and 2 of the tensor. This setting can be particularly important in the case where the interactions between variables are physical. We experiment more specifically this constraint in section 3.3, where this model is referred to as symmetric-Seq2VAR. Note that symmetry can be easily coupled with sparsity.

3 Experiments

3.1 Methodology

As a preliminary experimenal work, we illustrate our approach on several synthetic datasets, each with several levels of difficulty to illustrate generalization capacity, causality discovery and representation power. First, in order to assess the generalization capacity of Seq2VAR, we use a test set with intrinsic parameters (causality graphs, interaction intensities) that differ from those of the train set in all experiments. Second, causality discovery is assessed through the F1-score between inferred and ground truth causality graphs. When $\hat{A}^{(s)}$ is dense, we keep the most significant entries, i.e., those above a certain percentile of all entries in absolute value. This percentile is chosen with a priori knowledge. Third, since the set $\mathcal{A}$ is finite, we can assess unsupervisely learned representation with a standard downstream task. We classify the systems with the inferred tensors $\hat{A}^{(s)}$. The classification is a 1-nearest-neighbor on the tensors, for the Frobenius norm. The classification is completed in test set representation. Test set is devided in train and test subset for the classification task.

For each experiment, we compare Seq2VAR to NRI [8] and VAR [20], from which we respectively inspire for encoder and decoder. We remind that NRI learning relies on variational inference, whose objective function contains a Kullback-Leibler divergence between variational posterior and a certain prior distribution [7]. In binary and sparse setting, the prior of each latent graph is the proportion of 1 wanted in the whole latent graph, since all latent entries are independently sampled, by the mean-field assumption of VAEs. We always give NRI the true proportion as prior. For Seq2VAR, eventual usage of sparsity or regularization is specified when existing.

To provide fair comparison benchmarks, the experiments are built such that both VAR and NRI model fit in. Technical details about neural network architectures and parameters are given in Appendix A. We used the *Statsmodel* python library [18] for the VAR. The code used for the experiments is available at: `https://github.com/edouardpineau/Seq2VAR`. Experiments on real datasets, in particular in the context of system health monitoring at Safran, will be the developed as a future work.

3.2 Assessing Seq2VAR in binary linear setting

For our first experiment, we aim at (1) showing that binary sampling approach is relevant for Seq2VAR model and (2) analysing how Seq2VAR behaves in the

presence of noise. We place ourselves in a favorable setting for VAR, NRI and Seq2VAR. We generate samples from a stationary 1-order linear autoregressive model, where the linear transition matrix is a permutation matrix. We add several level of additive Gaussian noise to the observations (see Figure 1 for an illustration).

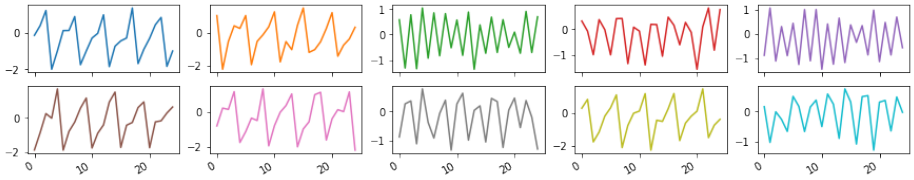


Fig. 1: 10-dimensional permutation MTS with observation noise $\mathcal{N}(0, 0.3)$.

We chose $d = 10$, $T = 50$, $K = 1$. Taking $\#\mathcal{A} = 20$, we generate 10 permutation matrices for the train set and 10 other permutation matrices for the test set. From each matrix we generate 100 samples with random $\mathcal{N}(0, 1)$ initial conditions. We train two versions of the Seq2VAR: dense with no penalty and binary. After training, Seq2VAR has generalized the notion of permutation. We note that the binary version is naturally the only one to find the real permutation matrices. Figure 2 shows the outputs of different versions of Seq2VAR encoder at test time, with different level of observation noise.

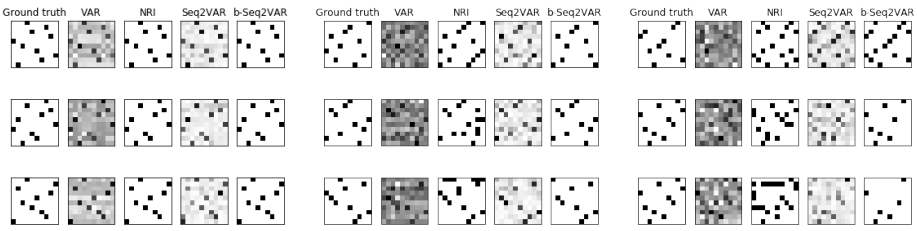


Fig. 2: Inferred transition matrices for different test samples, using VAR, NRI, Seq2VAR and binary Seq2VAR, compared to ground truth, depending on the level of observation noise: $\mathcal{N}(0, 0.1)$ (left), $\mathcal{N}(0, 0.3)$ (middle), $\mathcal{N}(0, 0.5)$ (right).

We see in Figure 2 an illustration of the problems created by an increase of observation noise variance. While signal-to-noise ratio decreases, disentanglement of the permutations from the noise becomes harder. VAR overestimate the density of the matrix. Optimal mean-square solution (closed form) of VAR decodes noisy signal by mixing noisy signals. On the contrary, Seq2VAR (without regularization) is parsimonious (but not sparse), i.e. it it gathers many entries of $\hat{A}^{(s)}$ around zero (see Figure 3).

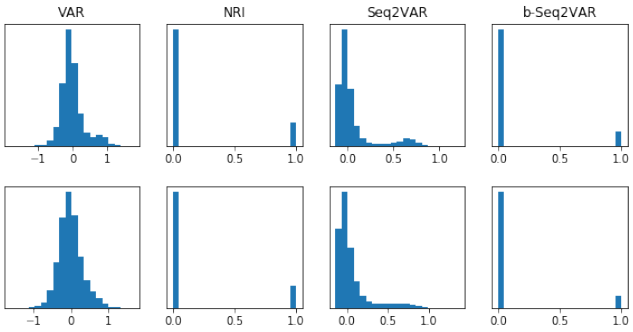


Fig. 3: Histogram of the distribution of the entries of inferred transition matrices over a test set, using VAR, NRI, Seq2VAR and binary Seq2VAR, compared to ground truth, depending on the level of observation noise: $\mathcal{N}(0, 0.1)$ (left), $\mathcal{N}(0, 0.5)$ (right).

We explain this behavior by two facts. First the inference mechanism of Seq2VAR is shared by all samples and has integrated the noise by seeing numerous noisy examples. Second, the low expressive decoder cannot integrate noise nor deal with complex mixture of noisy signals. This second point explains why Seq2VAR resists better to noise than NRI.

We present in Table 1 the classification accuracy and causality recovery in the presence of different level of observation noise. Seq2VAR approaches outperform VAR and NRI when noise gets stronger. binary-Seq2VAR give also better results, as its assumption fits the data better.

Dataset	Perm. + $\mathcal{N}(0, 0.1)$		Perm. + $\mathcal{N}(0, 0.3)$		Perm. + $\mathcal{N}(0, 0.5)$	
Tasks	Supervised classification	Causality detection	Supervised classification	Causality detection	Supervised classification	Causality detection
VAR [20]	**100** ± 0.0	72.2 ± 0.2	96.85 ± 3.8	61.8 ± 4.5	96.5 ± 1.6	52.6 ± 3.9
NRI [8]	**100** ± 0.0	95.63 ± 3.1	97.6 ± 3.4	84.3 ± 4.9	97.0 ± 2.1	68.3 ± 3.7
Seq2VAR	**100** ± 0.0	**97.3** ± 0.3	97.0 ± 4.3	92.5 ± 2.1	**97.8** ± 3.9	83.6 ± 2.3
B-Seq2VAR	**100** ± 0.0	97.2 ± 0.1	**100** ± 0.0	**94.6** ± 2.7	97.0 ± 4.3	**90.1** ± 2.9

Table 1: Test classification accuracy (%) and causality discovery (F1-score). The standard deviations correspond to the variation in results between different generated datasets (train and test). B-Seq2VAR stands for binary Seq2VAR.

In Figure 4, we see the boxplot representations of the distribution of the L_1 distance between ground truth and inferred causality graph within the test set, for several levels of observation noise. We first see that all methods suffer from noise. When noise is low, both NRI, Seq2VAR and binary-Seq2VAR offers almost perfect graph discover. However, the outliers (red crosses) of NRI spread

further than the one of Seq2VAR, which means that NRI fails to find the latent graph not by simply missing some entries but by finding a completely different graph that still fits the decoding requirements.

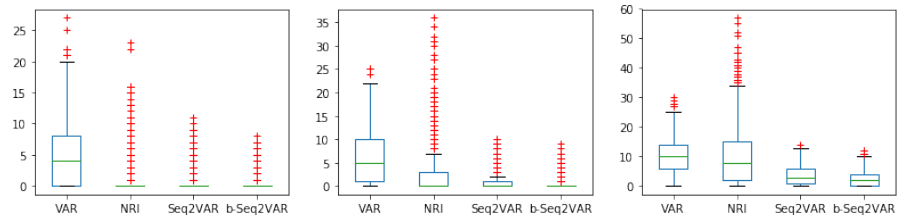


Fig. 4: Quartiles of the distribution of the L_1 distances between true and inferred causality graphs with VAR, NRI, Seq2VAR and binary Seq2VAR respectively with observation noise $\mathcal{N}(0, 0.1)$ (left), $\mathcal{N}(0, 0.3)$ (middle) and $\mathcal{N}(0, 0.5)$ (right). b-Seq2VAR stands for binary Seq2VAR.

3.3 Assessing Seq2VAR in physical setting

We now propose to assess the capacity of Seq2VAR to find Granger causality graph hidden in physical data. We use 10-ball-springs system data, consisting of the simultaneous trajectories of 10 identical balls in a 2D space, each ball being connected to others by springs with probability 0.5. The connection network is called interaction graph. System dynamics follows Newton's law of motion, sampled with $\Delta t = 0.001$ and then subsampled at frequency $1/100$. The system can be represented as a Granger causality graph [2] (see Figure 5). Each sample is 49 timesteps long. Note that this experiment is used in NRI paper [8].

For the experiments of this section, we sample 20 different balls-spring binary interaction graphs, 10 for the train set and 10 for the test set. Each of the 20 dynamical systems associated to the 20 graphs is built at random, with balls linked by a spring with probability 0.5. Each graph characterizes a class. We propose two different datasets: one with identical rigidity 1 for all springs (unweighted interaction graph) and one with variable rigidity (weighted interaction graph). For the later, the rigidity of each spring is uniformly sampled in $[0.5, 1]$. Each binary graph characterizes a class. We use 1000 samples per class. As for permutation MTS (see 3.2) we use different systems for the train set and the test set to challenge the generalization power of Seq2VAR.

We trained respectively a VAR, a dense Seq2VAR, a symmetric dense Seq2VAR and a sparse Seq2VAR. We chose $K = 2$ for all the experiments. For the later, we impose a slight regularization on the level of sparsity, i.e. on the number of null entries in the matrix. In Equation 3 we use $\Omega(G_\phi(X^{(s)})) = \left|\frac{1}{100}\|G_\phi(X^{(s)})\|_1 - 0.5\right|$ and λ is set at $1e^{-3}$, 0.5 being the sparsity prior. Without this regularization, the sparse Seq2VAR generally converges naturally towards

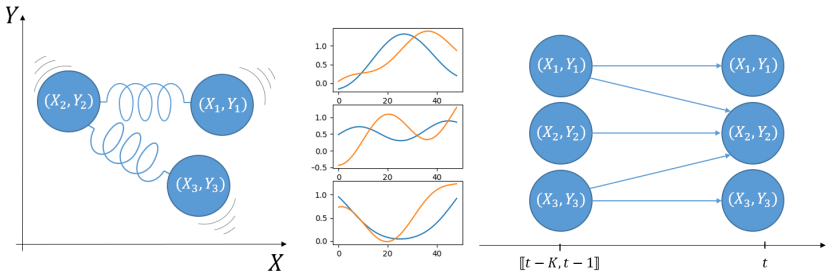


Fig. 5: Example of 3-ball-springs system (left) and its Granger causality graph (right). The causality graph represents the first-order dynamical dependencies between balls position at each time t and the positions of their direct neighbors at previous time steps $t-1 \dots t-K$. For all $t = K \dots T$ and $i \in \{1, 2, 3\}$, arrows indicates a dynamical dependency between $(X_t^{(i)}, Y_t^{(i)})$ and $\left\{(X_{t-1}^{(j)}, Y_{t-1}^{(j)}), \dots, (X_{t-K}^{(j)}, Y_{t-K}^{(j)})\right\}_{j \in pa(i)}$, with $pa(i)$ the set of parents of node i in the directed graph.

the right proportion of true zeros. The regularization is added for preventing an eventual trivial solution where the mask is a matrix of ones or local minima where only the diagonal paramaters are kept. The only occurrence of this problem happened in the heterogeneous problem.

Table 2 gathers the results. We see that Seq2VAR scores better on both quality measures than the usual VAR approach learned on the test set. If NRI gives very good results of causality detection on homogeneous springs, its causality discovery performance drops when dealing with heterogeneous springs. On the contrary, Seq2VAR gives good results for both homogeneous and heterogeneous rigidity graph.

Dataset	Homogeneous springs		Heterogeneous springs	
Tasks	Supervised classification	Causality detection	Supervised classification	Causality detection
VAR [20]	17	55.7	14.2	54.0
NRI [8]	**100**	**96.1**	**100**	78.5*
Seq2VAR	**100**	89.4	**100**	84.5
Symmetric Seq2VAR	99.9	91.4	**100**	**90.4**
Sparse Seq2VAR	99.7	88.2	99.2	81.4

Table 2: Test classification accuracy (%) and causality discovery (F1-score). *Hyperparameters different than the one used for homogeneous springs case (from the original paper [8]) to obtain better results. See Appendix A for more details.

It is interesting to notice that using an expressive inference network that explicitly model dependencies between variables capitalize on the global informa-

tion of the dataset and (1) generalize on new data and (2) overpass the statistical identifiability problem due to subsampling. It is also interesting to notice that sparsity is not optimal in Seq2VAR as it is in NRI. As variable spring rigidity makes the distribution wider and makes the causality less identifiable in both linear (Seq2VAR) and nonlinear (NRI) decoding process, with respect to our experimental setup.

We find in Figure 6 an illustration of the inference capacities of our model.

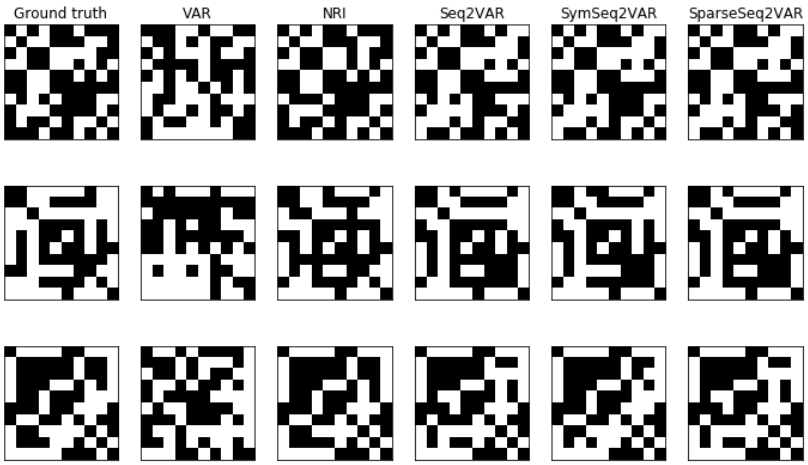


Fig. 6: Three examples of inferred transition matrices over a test set, using VAR, NRI, Seq2VAR, Symmetric Seq2VAR, Sparse Seq2VAR and Sparse Symmetric Seq2VAR, compared to ground truth.

4 Discussions

This section proposes discussions around our Seq2VAR approach for MTS representation with VAR matrices and causality graph.

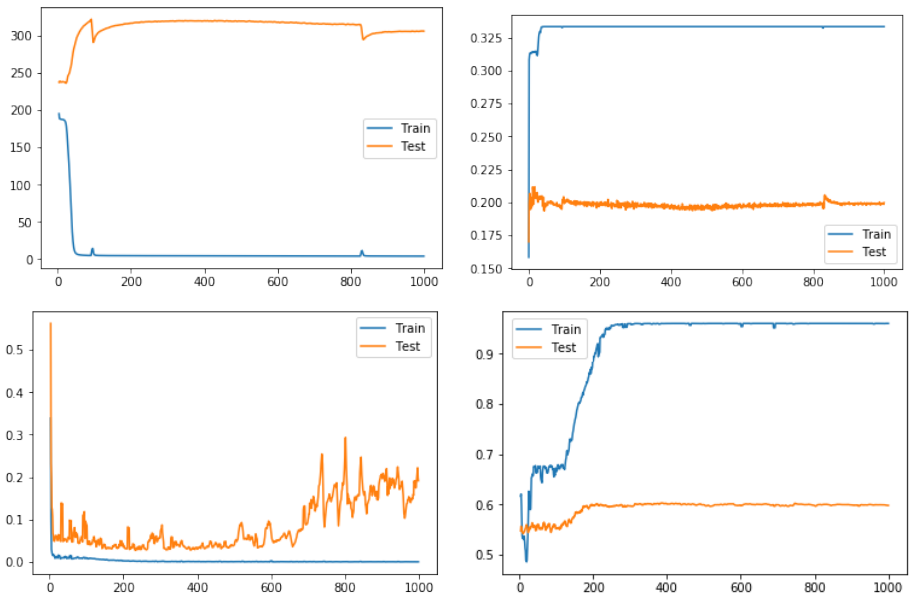
4.1 RINN vs. RNN as Seq2VAR encoder

The main assumption when modeling time series data is the autoregressive structure of the observed signal. A generic and expressive autoregressive model is the recurrent neural network (RNN). Hence, in practice we could use RNN to take the MTS as input and do the inference of the tensor A.

Our different attempts with RNN encoder were not able to generalize over the notion of causal interactions, no matter the regularization technique used to avoid overfitting (reduction of network memory capacities, increasing of depth [16], dropout [23], batch-norm [15], L_1-norm and L_2-norm penalty on weights).

We show in Figure 7 an example of train and test performances evolution during training, respectively for permutation and ball-springs experiments with GRU encoder instead of RINN encoder. We can see that a Seq2VAR using RNN as encoder overfits.

We explain this by the fact that RINN explicitly takes as input all the pairs of univariate time series constituting the MTS and outputs a tensor whose entries represent an embedding of each pair. Therefore, this inductive bias incites the network to model one-to-one interactions. We remind that the causality graphs of the test set are different from the causality graphs of the train set. Hence we ask our inference network to generalize over a discontinuous manifold. RINN inference networks and its explicit one-to-one interactions modeling learns well to generalize. Conversely, RNNs implicitly learn the notion of interactions during training, with a vector output that needs to be folded into the right shape. The generalization is more difficult.



(a) MSE prediction (b) Causality F1-score

Fig. 7: Train and test performances during training of Seq2VAR with GRU encoder instead of RINN encoder. Column (a): MSE of the 1-step VAR prediction. Column (b): F1-score of the inferred causality graph. Top row: permutation dataset. Bottom row: homogeneous 10-ball-springs dataset.

4.2 Relation with NRI

For the inference learning, the closer existing model is the Neural Relational Inference (NRI) [8], that uses a modified version of the interaction neural networks [17]. The NRI consists on a graph inference model, set as a VAE with binary latent space. The inferred graph is used as a variable selection procedure for a prediction model. This configuration specifically applies to physical interacting systems MTS (like ball-springs system). Three major differences appear between Seq2VAR and NRI.

First the form of the decoder. NRI decoding scheme is a non linear network that takes as input an embedding of the pairwise variable interactions at each time step and output the incremental change to predict next time step from current time step. On the contrary, we propose to leverage the simplicity of a linear autoregressive decoder that is potentially less expressive but do not require additional parameters.

Second, as a consequence of the form of the decoder: the latent representation. We can infer both binary and real latent representation to respectively represent existence and intensity of the causal interactions in the data. The real part is implicit in NRI. Experiment 3.3 shows that NRI does not disentangle existence and intensity of the interactions: when springs are not equally rigid, NRI is perturbed and finds a latent graph that does not correspond to physical reality. Our Seq2VAR, thanks to its continuous part, explicitly disentangles latent causal structure from other information and finds an acceptable causal graph.

Third difference, which is also as a consequence of the decoder: the minimal input information requirement. In fact, the notion of time lag is absent from NRI and lagged information needs to be furnished as input. For example, with the ball-springs systems data, Seq2VAR only needs measures of the location of each ball at each time step while NRI requires both location and velocity. Beyond the minimal input information requirement, the absence of lag in NRI modeling imposes that causality graph remains the same for all lags, like in physical structures.

Finally, these differences materialize with the results of the experiments presented in Section 3. Note that they also procure a significant advantage in term of complexity. We assess this complexity with the number of parameters and computing CPU time. These results are in Table 3.

5 Conclusions and future works

In this paper, we propose the Seq2VAR model, which learns to represent multivariate time series as Granger causal graph. Seq2VAR a encoder-decoder model consists of a relational neural network for inference and a linear autoregressive for generation. By construction, our model is immune to information vanishing in neural autoregressive framework. It is also capable of generalizing over the notion of interaction and Granger causality, i.e. estimating good VAR representation for unseen samples. The chosen representation is robust to continuous structural

Experiments	Permutations (batch size=128)		Ball-springs (batch size=64)	
	Number of parameters	CPU time per epoch (s)	Number of parameters	CPU time per epoch (s)
NRI [8]	65031	5.1	72966	38.8
Seq2VAR	47811	1.2	52550	4.7
Symmetric Seq2VAR	-	-	52550	4.7
Binary Seq2VAR	47811	1.4	-	-
Sparse Seq2VAR	-	-	61071	5.7

Table 3: Memory and computing time. Absence of results means that model is not used.

changes in data, and can easily interpretated. We demonstrated these properties with experiments on two different cases, and compared to state-of-the-art methods.

In further work, we intend to generalize the model to non-linear settings, as well as extending the way sparsity is controlled and constrained.

A Appendix: hyperparameters

For our Seq2VAR, we used a succession of 2-layers perceptrons for our relational encoder, as in NRI [8]. The parameters to chose concern latent dimension for all layers and the temperature parameter τ for the relaxed binary sampling (see Section 2.3). They are presented in Table 4.

Experiments	Permutations		Homogeneous springs		Heterogeneous springs	
Parameters	Latent dimension	τ	Latent dimension	τ	Latent dimension	τ
NRI [8]	64	0.5	64	0.5	64	0.1
Seq2VAR	64	0.1	64	0.5	64	0.5
Symmetric Seq2VAR	64	0.1	64	0.5	64	0.5
Sparse Seq2VAR	64	0.1	64	0.5	64	0.5

Table 4: Training parameters.

For NRI, all other parameters are the one of the original paper for the homogeneous springs rigidity except for the dimensionality of the latent space which we set to 64 instead of 256, since in the experimental setup of the original paper, it gives the same results while diminishing computing time and memory needs. For the heterogeneous rigidity, the parameter *prediction_steps* is set to 5 instead of 10 and τ is set to 0.1. These parameters gave the best average results. In fact, due to the highly expressive form of its decoder, NRI was able to build good predictor with not the good graph. We played with parameters to get more stable and better results. For the experiments, we also tried to change the *skip*

first parameter that is set to False or True in the original paper [8], depending on the dataset studied. It did not change the results of the experiments.

Acknowledgments This work is supported by the company Safran through the CIFRE convention 2017/1317.

References

1. Richard A Davis, Pengfei Zang, and Tian Zheng. Sparse vector autoregressive modeling. *Journal of Computational and Graphical Statistics*, 25(4):1077–1096, 2016.
2. Michael Eichler and Vanessa Didelez. Causal reasoning in graphical time series models. *arXiv preprint arXiv:1206.5246*, 2012.
3. Philippe Esling and Carlos Agon. Time-series data mining. *ACM Computing Surveys (CSUR)*, 45(1):12, 2012.
4. Stefan Haufe, Klaus-Robert Müller, Guido Nolte, and Nicole Krämer. Sparse causal discovery in multivariate time series. In *Causality: Objectives and Assessment*, pages 97–106, 2010.
5. Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.
6. Łukasz Kaiser and Samy Bengio. Discrete autoencoders for sequence models. *arXiv preprint arXiv:1801.09797*, 2018.
7. Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.
8. Thomas Kipf, Ethan Fetaya, Kuan-Chieh Wang, Max Welling, and Richard Zemel. Neural relational inference for interacting systems. *arXiv preprint arXiv:1802.04687*, 2018.
9. Hugo Larochelle and Iain Murray. The neural autoregressive distribution estimator. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 29–37, 2011.
10. Zhuohan Li, Di He, Fei Tian, Wei Chen, Tao Qin, Liwei Wang, and Tie-Yan Liu. Towards binary-valued gates for robust lstm training. *arXiv preprint arXiv:1806.02988*, 2018.
11. Francesco Locatello, Stefan Bauer, Mario Lucic, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. *arXiv preprint arXiv:1811.12359*, 2018.
12. Christos Louizos, Max Welling, and Diederik P Kingma. Learning sparse neural networks through l_0 regularization. *arXiv preprint arXiv:1712.01312*, 2017.
13. Xuezhe Ma, Chunting Zhou, and Eduard Hovy. Mae: Mutual posterior-divergence regularization for variational autoencoders. *arXiv preprint arXiv:1901.01498*, 2019.
14. Chris J Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*, 2016.
15. Stephen Merity, Nitish Shirish Keskar, and Richard Socher. Regularizing and optimizing lstm language models. *arXiv preprint arXiv:1708.02182*, 2017.
16. Haşim Sak, Andrew Senior, and Françoise Beaufays. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In *Fifteenth annual conference of the international speech communication association*, 2014.

17. Adam Santoro, David Raposo, David G Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Timothy Lillicrap. A simple neural network module for relational reasoning. In *Advances in neural information processing systems*, pages 4967–4976, 2017.
18. Skipper Seabold and Josef Perktold. Statsmodels: Econometric and statistical modeling with python. In *Proceedings of the 9th Python in Science Conference*, volume 57, page 61. Scipy, 2010.
19. Xiaoyu Shen, Hui Su, Shuzi Niu, and Vera Demberg. Improving variational encoder-decoders in dialogue generation. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.
20. Hiro Y Toda and Peter CB Phillips. Vector autoregression and causality: a theoretical overview and simulation study. *Econometric reviews*, 13(2):259–285, 1994.
21. Michael Tschannen, Olivier Bachem, and Mario Lucic. Recent advances in autoencoder-based representation learning. *arXiv preprint arXiv:1812.05069*, 2018.
22. Saehoon Yi and Vladimir Pavlovic. Sparse granger causality graphs for human action classification. In *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, pages 3374–3377. IEEE, 2012.
23. Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*, 2014.