Interpreting Time-Varying Dynamic Bayesian Networks for Earth Climate Modelling

Enrique Valero-Leal enrique.valero@upm.es
Pedro Larrañaga pedro.larranaga@fi.upm.es
Concha Bielza mcbielza@fi.upm.es
Universidad Politécnica de Madrid, Spain

Abstract

Bayesian networks tend to be considered as transparent and interpretable, but for big and dense networks they become harder to understand. This is the case of non-stationary, and more generally time-varying dynamic Bayesian networks, as the relations change over time and cannot be represented with a single template model. We introduce methods to explain how the model evolves qualitatively over time, and quantify these changes. In addition, we offer a functional implementation for time-varying dynamic Bayesian networks that includes our explainability proposals and some extensions that are targeted to simplify the networks in the specific field of climate sciences.

Keywords: Bayesian networks; explainable artificial intelligence; climate science

1. Introduction

Explainable artificial intelligence (XAI) is an emerging field, due to the increasing number of black-box models that are in our daily life, that aims towards making machine learning models more comprehensible. This allows us to better understand how our model works and even extract knowledge from it. In the literature, we can find works that point towards the use of inherently transparent models for XAI (Rudin, 2019), which is the approach taken in this work using Bayesian networks rather than trying to explain black-box models.

Specifically, we are interested in studying how can we make non-stationary Bayesian networks interpretable, as they tend to have a great number of nodes and connections. We believe that this has very promising directions, as most of the current XAI state of the art focuses on static models. We also show its potential impact in the field of climate sciences, which has been previously explored using (static) Bayesian networks. Some of our solutions are implemented alongside a library for learning time-varying dynamic Bayesian networks.

In Section 2, we review the current state of the art on the topic. Next, in Section 3, we present our methodological advances in non-stationary Bayesian network interpretability and the implementation made, and in Section 4 some experiments are shown. Finally, in Section 5, we draw conclusions about our work and discuss possible future lines of research.

2. Background

2.1 Bayesian Networks

A Bayesian network (Pearl, 1988; Koller and Friedman, 2009) $\mathcal{B} = (\mathcal{G}, \boldsymbol{\theta})$ is a probabilistic graphical model that encodes a joint probability distribution (JPD) $P(X_1, ..., X_n)$ over a

set of variables $\boldsymbol{X} = \{X_1, ..., X_n\}$. Qualitatively, a Bayesian network is a directed acyclic graph $\mathcal{G}$ that represents the conditional (in)dependences between the variables $\boldsymbol{X}$ and, quantitatively, Bayesian networks factorise the JPD using the vector of parameters $\boldsymbol{\theta} = (\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_n)$, storing only local conditional probability distributions (CPDs) of each node X_i given its parents in the graph, $\boldsymbol{\theta}_i = P(X_i|\boldsymbol{Pa}_{X_i}), \forall X_i \in \boldsymbol{X}$.

We can model continuous distributions using continuous Bayesian networks. One of the most well-known types are the Gaussian Bayesian networks (Lauritzen and Wermuth, 1989), which factorize a multivariate Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}; \boldsymbol{\Sigma})$ such that the conditional density of each node is a Gaussian whose mean is a lineal combination of its parents.

2.2 Dynamic Bayesian Networks

Dynamic Bayesian networks (DBNs) (Dean and Kanazawa, 1989; Murphy, 2002) are an extension of Bayesian networks that let us model time series and perform inference. In a discrete-time dynamic Bayesian network, the time dimension is discretized into slices and a time stamp is added to each variable X_i, having then X_i^t with $t \in \{0, 1, ..., T\}$, for a time horizon T. Therefore, the JPD of the process encodes a set of variables over $T + 1$ different time instants $\boldsymbol{X}^{0:T} = (\boldsymbol{X}^0, \boldsymbol{X}^1, ..., \boldsymbol{X}^T)$, where $\boldsymbol{X}^t = \{X_1^t, X_2^t, ..., X_n^t\}$. Each X_i^t is represented by a node in the network.

Generally these networks are stationary, which allows for *template-based representation*, as we have a prior network and T redundant transition networks. However, many real world processes are difficult to represent using stationary networks, since the relations among the variables may change significantly over time, and thus more complex models such as *non-stationary dynamic Bayesian networks* (ns-DBNs) (Robinson and Hartemink, 2008) are required. These models are DBNs in which the structure, the parameters or both may change at a given time. Thus, we need to describe more than one transition model $P_i(\boldsymbol{X}^t|\boldsymbol{X}^{0:t-1}), i \in \{1, 2, ..., K\}$, where K is the number of transition models.

To study the problem of gene regulation, Song et al. (2009) introduce *time-varying dynamic Bayesian networks* (TV-DBN), which can be defined as ns-DBNs in which we have a different transition model for every time instant. These networks are assumed to have a first Markovian order, no intra-slice arcs are allowed and the density of each node given its parents is also assumed to be Gaussian. To train a certain instant t, the rest of instants are used as well, giving more weight to the closer instances using a Gaussian kernel function. A wider kernel bandwidth results in more equalized weights and vice versa.

2.3 Bayesian Network Comparison

Given two Bayesian networks, we might be interested in measuring the differences between them, attending to their structures and their probability distributions.

De Jongh and Druzdzel (2009) offer an excellent review on the topic of comparing Bayesian network structures. Two of the most interesting measures reviewed are the *global distance* (Colace et al., 2004), that measures the numbers of arcs that are directed in the same direction in both network structures $\mathcal{G}_1$ and $\mathcal{G}_2$, and the *Hamming distance*, which in the context of Bayesian networks measures the number of changes that have to be done to a Bayesian network structure to get the other one. Note that the global distance, although its naming, is actually a similarity measure. The distances are formally presented in Equations

(1) and (2).

$$Global(\mathcal{G}_1, \mathcal{G}_2) = \frac{\sum arcs\ oriented\ equally}{\sum shared\ arcs + \sum different\ arcs} \tag{1}$$

$$Hamming(\mathcal{G}_1, \mathcal{G}_2) = \sum arcs\ oriented\ oppositely + \sum different\ arcs \tag{2}$$

A way to compare probability distributions (and Bayesian networks, by extension) are the *f-divergences* (Rényi, 1961), functions that measure the distance between two probability distributions P and Q over the same set of variables $\boldsymbol{X}$. Some popular instantiations of f-divergence are the Hellinger distance (Hellinger, 1909), the Kullback-Leibler divergence (Kullback and Leibler, 1951) and the Jensen-Shannon divergence (Lin, 1991). We are interested in the latter, as it is a symmetrization of the widely used Kullback-Leibler divergence (Equation (3)) that is actually a distance. Its formula can be seen in Equation (4).

$$KL(P||Q) = \sum_{\boldsymbol{x}} P(\boldsymbol{x}) log \frac{P(\boldsymbol{x})}{Q(\boldsymbol{x})} \tag{3}$$

$$JSD(P||Q) = \frac{1}{2} KL(P||M) + \frac{1}{2} KL(Q||M) \tag{4}$$

where M is the arithmetic mean between P and Q, $M = \frac{1}{2}(P + Q)$.

For two Gaussian distributions, the Jensen-Shannon divergence yields a closed formula if we take as the density m the geometric mean (instead of the arithmetic) of f and g, $M_G(f, g)$ (Nielsen, 2019). The probability density function of m is therefore:

$$m(\boldsymbol{x}) = \frac{M_G(f(\boldsymbol{x}), g(\boldsymbol{x}))}{\int_{-\infty}^{\infty} M_G(f(\boldsymbol{x}), g(\boldsymbol{x})) d\boldsymbol{x}} = \frac{f(\boldsymbol{x})^{0.5} g(\boldsymbol{x})^{0.5}}{\int_{-\infty}^{\infty} f(\boldsymbol{x})^{0.5} g(\boldsymbol{x})^{0.5} d\boldsymbol{x}}.$$

3. Explainable TV-DBNs

In this section, we aim to provide model explanations that show how a ns-DBN changes over time, giving a simplified, visual and interactive explanation of the concept drift in the data. First, we present our theoretical proposal on how to explain the changes in both the structure and parameters and, next, we describe the library developed to test our methods.

3.1 Changes in the Network Structure

We can measure the changes in the structure using the global and Hamming distances discussed in Section 2.3 by comparing pairs of transition networks. We can then find and plot the points in which the concept drift is more pronounced and verify if the changes between transition networks are smooth. This allows us to understand *where* the changes are happening (explainability at a more macro level). We can also study visually *how* the transition networks change, i.e., interpretability at a more micro level. In order to achieve the latter goal, we will provide a visual explanation of the previously defined distances.

To visualize the global distance we can draw a transition network only with the arcs that are oriented equally in both transition networks, i.e., the logical AND between the arcs of the networks. This lets us visualize the *persistent* arcs of the network. We refer to the resulting graph as *global graph* (see an example in Figure 1).

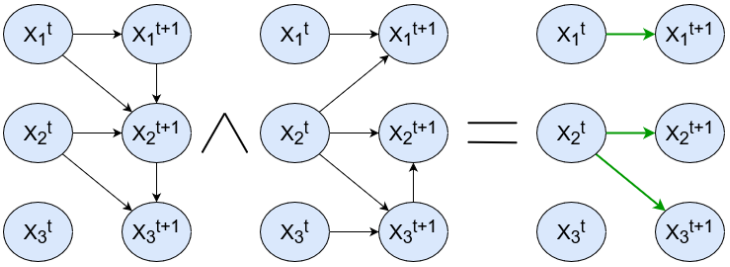


Figure 1: DBN global graph

The Hamming distance, intuitively, counts the number of errors, i.e., arcs missing in any of the two networks in relation with the other and arcs that are present, but incorrectly oriented. Similarly, we can plot an output transition network with just the arcs that are wrong when comparing both networks, visualizing then the *contingent* arcs in the transition networks. The graph obtained is referred to as the *Hamming graph*, and can be formally obtained by applying the XOR operator to the set of arcs of both transition networks.

These solutions can be extended to compare more than two transition networks, thus showing tendencies over time instead of just two given time points. It is specially interesting to see which edges persist for a long time in the network. Nonetheless, it might happen that if we try to compute the global graph for a large set of transition networks, we might find out that there are no persistent arcs. To solve this issue, we might include some kind of tolerance threshold. If set to $\delta \in [0, 1]$, it means that, if at least $100 \cdot \delta\%$ of the input graphs contain a certain arc, that will be considered persistent. To have a visual clue of which edges are more or less persistent we can set different thickness and/or colour saturations to the arcs. Another interesting idea is to compute the global graph of every consecutive k transition networks across the whole network, thus obtaining what we refer to as *summary network*, that summarises the whole model into a reduced set of transition networks.

3.2 Changes in the Probability Distribution

To study the the changes in the JPD of a pair of transition networks we can use the Jensen-Shannon divergence. We propose to decompose its computation into the sum of the Jensen-Shannon divergence of each pair of nodes, one of each network, using the chain rule. Since the Kullback-Leibler divergence can be decomposed using the chain rule and the Jensen-Shannon divergence is defined as a sum of the former (see Equation (4)), we can also decompose the latter using the chain rule.

In addition, we need to establish a common ancestral ordering in both networks to be able to compare them using the chain rule with the Jensen-Shannon divergence. In some cases, it can be impossible to find a common ancestral ordering for two transition networks, but there will always be one if both networks have a certain structure defined in this work and referred to as *sequential k-partite graph* with the same independent sets.

Definition 1 *A sequential k-partite DAG is a directed k-partite graph in which the independent sets of nodes are numbered from* 1 *to* k *and no arc is allowed from any node* X_i *in the independent set* i *to any node* X_j *in the independent set* j *iff* $i > j$.

A transition network of Markovian order τ with only inter-slices arcs is a type of sequential k-partite DAG. If we are working with continuous variables (as with TV-DBNs), we also need to compute the geometric mean of the two densities. To the best of our knowledge, no simple formula is available in the literature, so we provide one in Theorem 2.

Theorem 2 *Given two Gaussian densities $\mathcal{N}(\mu_f, \sigma_g)$ and $\mathcal{N}(\mu_f, \sigma_g)$ referred to as $f(x)$ and $g(x)$, then the geometric mean of such density functions follows a normal distribution $\mathcal{N}(\mu_m, \sigma_m)$, where*

$$\mu_m = \frac{\mu_f \sigma_g^2 + \mu_g \sigma_f^2}{\sigma_f^2 + \sigma_g^2} \qquad \textit{and} \qquad \sigma_m = \sqrt{\frac{2\sigma_f^2 \sigma_g^2}{\sigma_f^2 + \sigma_g^2}}.$$

Proof First, let us prove that the n-power of a Gaussian density function is also a Gaussian and find its mean and variance. The full formula for such function to the n-power is:

$$\big(f(x)\big)^n = \left(\frac{1}{\sqrt{2\pi}\sigma} e^{\frac{(x-\mu)^2}{2\sigma^2}}\right)^n = \left(\frac{1}{\sqrt{2\pi}\sigma}\right)^n \left(e^{\frac{(x-\mu)^2}{2\sigma^2}}\right)^n$$

We can rewrite the right part of the product as follows:

$$\left(e^{\frac{(x-\mu)^2}{2\sigma^2}}\right)^n = e^{n\frac{(x-\mu)^2}{2\sigma^2}} = e^{\frac{(x-\mu)^2}{2\frac{\sigma^2}{n}}} = e^{\frac{(x-\mu)^2}{2\left(\frac{\sigma}{\sqrt{(n)}}\right)^2}}$$

This proves that $(f(x))^n$ is a scaled Gaussian that represents the normal $\mathcal{N}(\mu, \frac{\sigma}{\sqrt{n}})$. Since Bromiley (2003) proved that the product of two Gaussian densities is another Gaussian with known mean and variance, we can deduce the values of the mean and variance of the geometric mean, which are stated in the theorem. ■

3.3 A Library for Learning TV-DBNs and Explaining Them

To the best of our knowledge, there is not any available library for learning, infere, nor explain TV-DBNs. Therefore, we have implemented the library `tvdbn` in R (R Core Team, 2020), built on top of the already existing library `bnlearn` (Scutari, 2010).

While R is not an object-oriented programming language, we offer a UML-like class diagram (Figure 2) to illustrate the organization of our library. We define the classes Tvdbn and Tvdbn.fit, that extend the `bnlearn` classes Bn and Bn.fit respectively, and represent the structure of a TV-DBN and a fitted TV-DBN respectively. Both classes add two attributes that represent the number of time points and the names of the variables. We also separated the functionality in different modules in order to modularize our library. The main ones are the learning, visualization, inference and explainability modules.

Regarding the explainability, we can highlight the following functionality:

- Computing the Hamming distance of a pair of transition networks,
- Finding the persistent arcs of a TV-DBN or of a set of transition networks.

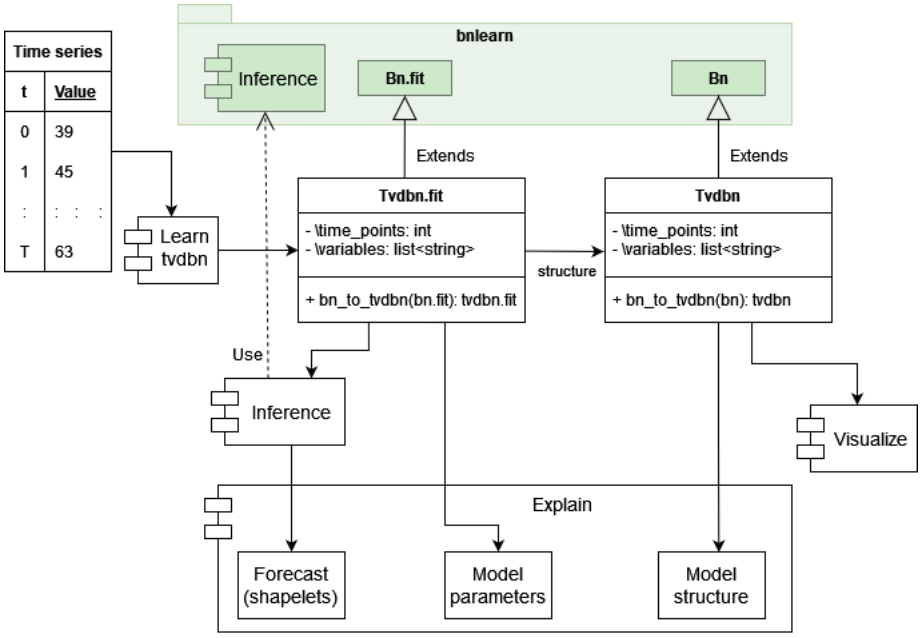


Figure 2: tvdbn library class diagram

- Computing the summary network of a TV-DBN.
- Display only the nodes in which we are interested and, optionally their parents and children. This is referred as *trimmed network*.

We also add an option to learn from data that is gridded to apply these methodologies to the database NCEP/DOE. We introduce a penalty in the learning process that increases with the Euclidean distance between two coordinates, similar to how we give more weights to the instances that are closer in time to the instant that we are learning.

This idea is modelled as follows. Let $\boldsymbol{c}_{X_i}$ be a bi-variate vector that denotes the coordinates in the space of variable X_i. When learning X_i^t, we will introduce a penalty in the $l1$-regularization process for the coefficient X_j^t be given by the inverse value of a bivariate Gaussian function with mean $\boldsymbol{\mu} = \boldsymbol{c}_{X_i}$ and a user-specified covariance matrix $\boldsymbol{\Sigma}$ at the point $\boldsymbol{c}_{X_j}$. A higher variance will result in a more even spatial penalty across the variables.

4. Experimental Results

4.1 Initial hypotheses

In our experiments, we aim to validate our proposals related to TV-DBN explainability from an empirical and practical point of view. We focus on the explanations of the changes in the structure, as the graph can provide much visual and simpler explanations, which is usually desired by the end-user. We formulate the following initial hypotheses that can be clustered into three different groups, depending on their variable of study.

- Directly related to explainability:

1. The Hamming distance can be used to study where the biggest changes in the network are produced and, as such, indirectly study the concept drift.
2. The global and Hamming graphs and the summary network help us identify the relations that persist across time and study the concept drift at the micro level.

- Related to how the parameters affect the characteristics of the network which, in turn, can help us to quantify how interpretable a network is (a simple and smooth network is will be easier to interpret):

3. In TV-DBNs, when the changes are smooth in the data, the smaller the Hamming distance is, the larger the percentage of persistent arcs will be, as we expect the consecutive transition networks to be very similar to each other.
4. The less uniform the spatial regularization is (low Gaussian variance), the smaller the Hamming distance will be and the number of persistent arcs will be higher.

- Related to performance of the network. Although this is not one of the main contributions of the paper, we decided to briefly evaluate how the newly introduce spatial penalty might improve the performance of the network:

5. We expect the spatial penalty to increase the accuracy for the test data, as we are introducing meaningful aprioris in our model that may help to better generalize.
6. A larger spatial regularization bandwidth will result in a shorter execution time, as we will guide the model towards learning more meaningful relations.

4.2 Datasets

An example of time series can be found in the field of climate and, as such, we decide to illustrate our proposal using two datasets obtained from the NCEP/DOE Reanalysis II database (Kanamitsu et al., 2002), which contains observations every 6 hours from January 1979 to May 2022 from different variables (temperature, humidity,...) both at a range of pressure levels and at the surface of the Earth.

We will focus only on the temperature at 850hPa and limit the latitude range from 30°N to 80°N and longitudes that range from 30°W to 60°E (Europe) and the observation time to twenty years, from 2001 to 2020. To reduce the great number of features, we subsample the points of the grid, using a total of 45 coordinates distributed in a 9×5 grid.

For the first dataset, we will consider as time slice each of the 12 months of the year and for the second one the time slices will be each of the 31 days of March. We aggregate observations using the mean. The difference in smoothness between the transition networks of both TV-DBNs will allow us to get more meaningful results.

4.3 Experiments

We aim to study how the size of the kernel bandwidth (and the spatial penalty if introduced) affects a set of metrics of the obtained networks in order to validate hypotheses 3 to 4.

To carry out the experiments, we start selecting a kernel bandwidth that is approximately $\frac{T}{7}^2$, where T is the length of the series. This value is used in the original TV-DBN article in the experiments (Song et al., 2009). We will select also higher and lower values.

We will use 5-fold cross-validation to study our proposal, training with 80% of the data and validating with 20% at each fold, for every value of the kernel bandwidth and spatial penalty. The measures that are subject of our study are the following:

1. Mean squared error (MSE) of the forecast starting from the first time instant to the last one (sliding window), considering the error of every variable at every time step.
2. Time: The time that it takes to learn the TV-DBN.
3. Arcs: The average number of arcs per transition network.
4. Hamming: The average Hamming distance between two adjacent transition networks.
5. Avg. pers. arcs: The average percentage of persistent arcs in each transition network.
6. Total pers. arcs: The average percentage of arcs that persist from the set of all arcs between two transition networks of the TV-DBN.

Tables 1 and 2 show these average values and their standard deviation for the networks learned with no spatial regularization. Regarding the MSE, we find that higher kernel bandwidths lead to a better performance in March daily climate dataset, whereas the network trained with the monthly climate dataset performs better with lower kernel bandwidths.

Kernel bandwidth	MSE	Time (sec.)	Arcs	Hamming	Avg. pers. arcs (%)	Total pers. arcs (%)
0.05	0.11 ± 0.03	15.87 ± 0.63	15.18 ± 12.38	27.71 ± 16.60	0.00 ± 0.00	0.00 ± 0.00
5	0.18 ± 0.06	68.40 ± 3.60	294.68 ± 79.72	189.33 ± 45.25	2.71 ± 2.12	0.34 ± 0.09
50	0.55 ± 0.17	49.74 ± 3.60	466.67 ± 75.73	61.44 ± 17.21	47.31 ± 6.69	10.71 ± 1.17
500	0.88 ± 0.49	49.15 ± 3.94	525.00 ± 79.36	11.09 ± 4.66	91.80 ± 1.88	23.77 ± 3.78
50000	0.91 ± 0.52	49.43 ± 4.31	531.85 ± 83.97	0.36 ± 0.93	99.59 ± 0.59	26.17 ± 4.69

Table 1: Monthly climate dataset measures

Kernel bandwidth	MSE	Time (sec.)	Arcs	Hamming	Avg. pers. arcs (%)	Total pers. arcs (%)
0.02	1.01 ± 0.33	45.16 ± 1.04	98.71 ± 23.57	138.42 ± 23.33	0.00 ± 0.00	0.00 ± 0.00
2	0.98 ± 0.32	133.20 ± 3.00	236.14 ± 44.08	182.23 ± 42.08	18.45 ± 3.51	2.08 ± 0.19
20	0.95 ± 0.31	121.22 ± 3.00	235.04 ± 27.46	56.55 ± 15.44	36.39 ± 4.38	4.17 ± 0.14
200	0.94 ± 0.30	97.80 ± 3.00	257.81 ± 18.91	14.74 ± 4.84	57.86 ± 4.83	7.33 ± 0.37
20000	0.94 ± 0.30	94.20 ± 3.00	272.71 ± 22.33	0.36 ± 0.68	98.35 ± 0.74	13.24 ± 1.20

Table 2: March daily climate dataset measures

As formulated in hypothesis 3, the Hamming distance decreases with a larger kernel bandwidth, since we are giving a more uniform weight to the instances and thus each adjacent transition network will be more similar. Nonetheless, the monthly climate network presents some spurious behaviour regarding the Hamming distance. Parallel to this, we can observe that the average number of persistent arcs also increases with the kernel bandwidth, for the same reason as before.

To test the spatial penalty, we decided to fix the kernel bandwidth to 50 and 250 for the monthly climate dataset and to 20 and 200 for the daily climate dataset. Those are

not necessarily the values that yield better performance, but they will allow to observe how the spatial penalty can improve (or worsen) the network performance, as with these kernel values there is a lot of space for improvement. Tables 3 and 4 show the results for the networks learned using the spatial penalty. The spatial penalty column represents the standard deviations of the bivariate Gaussian that determine the penalty for each feature based on the distance. A standard deviation of "Infinite" (Inf) means that the penalization is uniform (which is the default case presented in the previous experiments).

Kernel bandwidth	Spatial penalty deviation	MSE	Time (sec.)	Arcs	Hamming	Avg. pers. arcs (%)	Total pers. arcs (%)
50	5	0.54 ± 0.18	40.38 ± 1.19	573.87 ± 68.12	77.22 ± 19.55	45.82 ± 5.03	12.84 ± 0.82
	10	0.57 ± 0.19	46.09 ± 1.85	630.00 ± 83.33	90.64 ± 20.65	43.65 ± 5.45	13.39 ± 1.07
	20	0.50 ± 0.18	50.01 ± 1.46	548.40 ± 61.89	71.89 ± 15.60	44.74 ± 5.41	11.98 ± 0.89
	50	0.50 ± 0.19	52.83 ± 1.62	491.57 ± 59.34	63.00 ± 13.17	46.79 ± 6.00	11.23 ± 1.17
	Inf	0.50 ± 0.22	54.28 ± 1.70	460.37 ± 55.66	56.16 ± 12.55	50.59 ± 6.69	11.35 ± 0.93
250	5	0.61 ± 0.25	37.74 ± 1.48	621.35 ± 51.22	21.78 ± 5.31	85.16 ± 1.39	26.13 ± 2.45
	10	0.62 ± 0.23	43.01 ± 2.29	687.72 ± 53.00	24.87 ± 6.69	84.81 ± 2.19	28.80 ± 2.59
	20	0.58 ± 0.25	47.97 ± 1.53	589.97 ± 36.18	19.96 ± 5.03	86.78 ± 2.09	25.28 ± 1.88
	50	0.62 ± 0.26	50.75 ± 1.52	540.33 ± 36.01	18.75 ± 4.63	85.72 ± 2.36	22.87 ± 1.87
	Inf	0.62 ± 0.28	51.54 ± 2.03	503.95 ± 37.22	17.05 ± 4.99	86.03 ± 2.68	21.40 ± 1.81

Table 3: Monthly climate dataset measures (spatial network)

Kernel bandwidth	Spatial penalty deviation	MSE	Time (sec.)	Arcs	Hamming	Avg. pers. arcs (%)	Total pers. arcs (%)
20	1	0.94 ± 0.30	53.19 ± 3.10	149.95 ± 19.93	25.96 ± 11.26	54.65 ± 7.26	3.98 ± 0.18
	5	0.95 ± 0.30	81.00 ± 1.80	180.39 ± 28.02	35.56 ± 14.47	49.17 ± 7.75	4.28 ± 0.13
	10	0.95 ± 0.30	101.40 ± 4.80	192.57 ± 28.92	40.06 ± 14.73	45.13 ± 6.98	4.20 ± 0.20
	20	0.95 ± 0.30	108.60 ± 10.20	203.63 ± 32.11	43.77 ± 14.31	42.85 ± 6.69	4.21 ± 0.17
	Inf	0.95 ± 0.31	121.20 ± 3.00	235.04 ± 27.46	56.55 ± 15.44	36.39 ± 4.38	4.17 ± 0.14
200	1	0.94 ± 0.29	57.39 ± 1.67	175.40 ± 10.84	7.73 ± 4.29	69.96 ± 3.80	6.04 ± 0.27
	5	0.94 ± 0.29	78.00 ± 1.80	201.23 ± 14.12	9.97 ± 4.56	66.78 ± 4.83	6.61 ± 0.26
	10	0.94 ± 0.29	88.20 ± 3.60	207.52 ± 13.78	10.44 ± 4.82	65.80 ± 4.57	6.72 ± 0.24
	20	0.94 ± 0.29	99.00 ± 2.40	217.20 ± 15.88	11.08 ± 4.43	63.56 ± 4.67	6.79 ± 0.25
	Inf	0.94 ± 0.30	97.80 ± 3.00	257.81 ± 18.91	14.74 ± 4.84	57.86 ± 4.83	7.33 ± 0.37

Table 4: March daily climate dataset measures (spatial network)

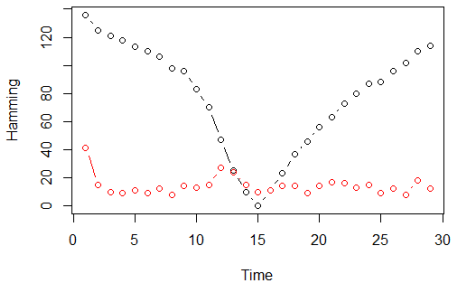
Some medium values for the spatial penalty (around 20 for this dataset) actually show a small decrease in MSE, partially confirming hypothesis 5. We can see how a more pronounced spatial regularization results in a shorter learning time for both networks, learning up to twice as fast compared with learning without penalty, confirming hypothesis 6.

The number of arcs shows a different behaviour in each network. In the monthly climate network, a more pronounced penalty usually results in more arcs (except when the penalty becomes too pronounced), whereas in the March daily climate network the number of arcs actually decreases with a higher spatial penalty (lower variance value). The same phenomenon occurs with the Hamming distance and the percentages of persistent arcs.

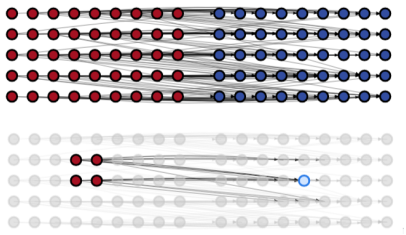
4.4 Practical demonstration

In order to prove the usefulness of our proposal and discuss hypotheses 1 and 2, we will briefly show how it can be used to learn and understand time series.

We can learn the networks `tvdbn_monthly` and `tvdbn_daily` with the function `learn_tvdbn`, and then study the Hamming distance between every pair of transition networks (red dots and lines) and between the 15th and the rest (black dots and lines). This can be done with the function `hamming_changes`, and the results are shown in Figure 3.



(a) Hamming distances for the March daily climate network



(b) Above, global graph of the March daily climate network. Below, highlighting of a persistent relationship

Figure 3: Some examples of functionality of the `tvdbn` library

Since the networks are too big to visualize, we will simplify the monthly network computing its global graph (function `global_graph`), which lets us find relationships present between all the months (see Figure 3b). The nodes are plotted using `graphviz` respecting the real spatial location of the coordinates that they represent. In future versions, we may overlap a map to better visualize which node represents which region.

Another example of functionality is to compute the summary network of the March daily network roughly aggregating by weeks and then computing the Hamming graph of the first and last transition networks (code shown below).

```
graph_weekly = summary_network(tvdbn_daily, frequency = 8)
plot_spatial_tvdbn(hamming_graph_2g(transition_network_graph(graph_weekly, 1),
    transition_network_graph(graph_weekly, 3)))
```

4.5 Discussion

We accept hypotheses 1 and 2 directly related to explainability, as we believe that the proposals improve the understanding of the network and add useful exploratory features. However, it would be interesting to have the opinion of a control group or of a climate scientist in order to better validate our proposal, as explainability is inherently subjective.

Regarding the simplicity and characteristics of the network, the experiments confirm with great confidence hypothesis 3, whereas hypothesis 4 is refuted, since the impact of the spatial penalty in Hamming distance is primarily determined by the smoothness in the data and it only holds in scenarios where the changes are smooth.

Concerning the performance, both hypotheses 5 and 6 are accepted, although we believe that some more experimentation is needed to accept with greater confidence the former.

In addition, we have exposed an explainability trade-off present in the TV-DBNs: a network that changes more smoothly across time usually tends to have more arcs and vice

versa. We have also proven that we can build a TV-DBN that retains its accuracy but with less arcs using the spatial penalty if the changes in the data are smooth enough.

The practical demonstration shows that our proposal allows to explore these types of very large networks and potentially draw conclusions regarding climate sciences. However, while we provided a new framework to understand these models, drawing conclusions about the climate is out of the scope of this work due to our lack of expert knowledge.

5. Conclusions

Our contributions can be divided into methodological, technological and applications. First, we presented methodological advances concerning model explanations for ns-DBNs and TV-DBNs. They have validated using our implementation, the main technological advance of this work, although other minor scripts have been developed as well. Finally, we showed how our advances could be applied in the field of climate sciences.

In future works, we expect to expand our implementation to also compute differences in the probability distributions and obtain more results. Additionally, many of our proposals, while useful, are merely aesthetics, since we cannot perform inference over a global graph or a summary network. We would like to build a functional version of the proposals to simplify both the model and inference. Even though the original TV-DBN proposal ensures smooth changes in the network, it would be interesting to further analyse whether these changes are a result of an actual concept drift in the data or a byproduct of the multimodality in the learning process. A promising technique is to use a multiple start approach.

Acknowledgments

This research has been partially funded by the Spanish Ministry of Science and Innovation through the PID 2019-109247GB-I00 project and to the BBVA Foundation (2019 Call) through the "Score-based nonstationary temporal Bayesian networks. Applications in climate and neuroscience" project.

References

P. Bromiley. Products and convolutions of Gaussian probability density functions. Technical report, School of Medicine, University of Manchester, 2003.

F. Colace, M. De Santo, M. Vento, and P. Foggia. Bayesian network structural learning from data: An algorithms comparison. In *International Conference on Enterprise Information Systems*, volume 2, pages 527–530, 2004.

M. De Jongh and M. J. Druzdzel. A comparison of structural distance measures for causal Bayesian network models. *Recent Advances in Intelligent Information Systems, Challenging Problems of Science, Computer Science Series*, pages 443–456, 2009. Springer.

T. Dean and K. Kanazawa. A model for reasoning about persistence and causation. *Computational Intelligence*, 5(2):142–150, 1989.

E. Hellinger. Neue begründung der theorie quadratischer formen von unendlichvielen veränderlichen. *Journal für die reine und angewandte Mathematik*, 1909(136):210–271, 1909.

M. Kanamitsu, W. Ebisuzaki, J. Woollen, S.-K. Yang, J. Hnilo, M. Fiorino, and G. Potter. NCEP–DOE AMIP-II reanalysis (r-2). *Bulletin of the American Meteorological Society*, 83(11):1631–1644, 2002.

D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. The MIT Press, 2009.

S. Kullback and R. A. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951.

S. L. Lauritzen and N. Wermuth. Graphical models for associations between variables, some of which are qualitative and some quantitative. *The Annals of Statistics*, 17(1):31–57, 1989.

J. Lin. Divergence measures based on the Shannon entropy. *IEEE Transactions on Information Theory*, 37(1):145–151, 1991.

K. P. Murphy. *Dynamic Bayesian Networks: Representation, Inference and Learning*. PhD thesis, University of California, Berkeley, 2002.

F. Nielsen. On the Jensen–Shannon symmetrization of distances relying on abstract means. *Entropy*, 21(5):485, 2019.

J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2020. URL `https://www.R-project.org/`.

A. Rényi. On measures of entropy and information. In *Proceedings of the 4th Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*, pages 547–561. University of California Press, 1961.

J. Robinson and A. Hartemink. Non-stationary dynamic Bayesian networks. *Advances in Neural Information Processing Systems*, 21:1369–1376, 2008.

C. Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215, 2019.

M. Scutari. Learning Bayesian networks with the bnlearn R package. *Journal of Statistical Software*, 35(3):1–22, 2010.

L. Song, M. Kolar, and E. P. Xing. Time-varying dynamic Bayesian networks. In *Proceedings of the 22nd International Conference on Neural Information Processing Systems*, pages 1732–1740, 2009.