Learning network structure using parameterized Dynamic Bayesian Networks

E.Auclair[1] N.Peyrard[1] R.Sabbadin[1]

[1] MIAT, Université de Toulouse, INRA, Castanet-Tolosan - France

Abstract

L'apprentissage de Réseau Bayésien Dynamique (RBD) consiste à apprendre la fois la structure et les paramètres du RBD. La plupart des approches d'apprentissage de RBD considèrent une représentation tabulaire des lois de probabilités conditionnelles, chaque table étant indépendante. Dans cet article nous proposons une méthode d'apprentissage dans le cas où ces tables sont paramétrées et que les paramètres sont partagés par les lois conditionnelles de toutes les variables. Dans ce cas, la propriété de décomposition de la vraisemblance en sous-problèmes indépendants pour l'estimation des tables par optimisation est perdue. L'approche proposé alterne entre une phase d'amélioration de l'estimation des paramètres, résolue par maximisation globale de la log-vraisemblance, et une phase d'amélioration de la structure apprise, modélisée comme un problème de programmation linéaire 0-1. La méthode est instanciée sur un problème d'apprentissage de la structure d'un réseau écologique. Les premières expérimentations, sur des petits réseaux, montre la faisabilité de l'approche.

Keywords

Réseaux Bayesiens Dynamiques paramétrés. Programmation linéaire en nombre entier, graphes avec labels, réseaux d'interactions écologiques

1 Introduction

The question of learning an interaction network between entities is a classical question in bioinformatics [13], ecology [9] or social science [7]. This problem has been formalized in the framework of Bayesian Networks (BN) [6], where the states of entities are modeled by the means of variables of finite domains, and the interactions between them are represented by the structure of the BN.

When the states of variables change through time, learning approaches based on *Dynamic Bayesian Networks* have also been proposed [5]. Learning a DBN amounts to learn both its *structure* (i.e. the conditional independences between the variables) and the parameters of its probability tables. This problem is now rather well formulated and several solution approaches exist [2][4]. These approaches consider tabular DBNs, where each individual transition probability table is learnt independently. This allows for very flexible models, however learning approaches must be adapted when some specific knowledge is used to describe a specific model. In this case, a new bayesian network model might be introduced, for instance, the Module network model when some parameters and parents are shared amongst several variables [12], or with other types of constrained parameters [10]. In ecological applications where the DBN is used to model the dynamics of species in interaction, the possible types of interactions are well-defined and in limited number: predator-prey, host-parasite,... Therefore, edges of the DBN are labelled according to the interaction type. Species confronted to the same types of interactions share the same transition probability tables. We introduce a new model for this type of constrained parameters DBN model, with parametrized transition probability tables, refered as learning *a parameterized DBN (P-DBN)*.

In the non-parameterized case, when the structure of the DBN is known, explicit formulas for the estimators of the transition probabilities are available from counts on data (e.g. [4]). Explicit solution to likelihood maximization is not available in the case of a P-DBN since the tables are not independent (they share some parameters). The structure learning step, for given model parameters, must also be handled differently. Score functions, combining a term measuring how a network fits a dataset and a penalty term on the model complexity (e.g. BIC [11]) are not relevant since the number of parameters of a P-DBN is independent of the graph structure. The BDe score [6] is not relevant either, due to the assumption of independence between parameters in the different tables which does not hold in a P-DBN.

We propose a generic procedure for learning the structure and parameters of a P-DBN. This approach iteratively improves estimates of the structure and labels of the network and of the parameters defining the transition probabilities.

Then, we consider the specific problem of learning an ecological interaction network from time series of species occurence. We develop a dedicated version of our procedure, where the parameters improvement phase is performed using a standard likelihood maximization algorithm, while the structure/labels improvement phase is modeled as a 0-1

integer linear program. The proposed approach is illustrated on fixed ecological networks. Our case study assumes that the network contain labelled edges.
The problem of learning a P-DBN and the generic solution procedure are presented in the following Section. The formalization of the problem of learning an ecological network is described in Section 3. Then in Section 4 we describe the two steps of the learning algorithm for this case sudy. Experimental results are presented in Section 5.

2 Parameterized Dynamic Bayesian Network learning

2.1 Dynamic Bayesian Network

Let us consider a set of n coupled random processes over T horizons, $\{(X_1^t)_{t=1,...T}, \ldots, (X_n^t)_{t=1,...T}\}$ and a a set of additional information designed as *covariates* that can affect the random processes. $X_i^t \in \{0,1\}$ is a random variable, taking values in a finite set, which represents the state of process i at time t. Then, denoting $X^t = \{X_1^t, \ldots, X_n^t\}$, a *Dynamic Bayesian Network (DBN)* [4] allows to represent concisely the joint probability $P(X^1, \ldots, X^T|a)$ of the process $(X^t)_{t=1,\ldots,T}$ knowing covariates a, under Markovian and stationary assumption, by exploiting local independence between the variables. Formally, the joint probability $P(X^1, \ldots, X^T|a)$ writes (for any T):

$$P(X^1, \ldots, X^T|a) = P(X^1|a) \prod_{t=1}^{T-1} P(X^{t+1}|X^t, a). \quad (1)$$

The local independences between variables are represented by two directed graphs:

- $\mathcal{G}_1 = (V_1, E_1)$ is a directed acyclic graph with vertices $V_1 = \{1, \ldots, n\}$ and edges $E_1 \in V_1^2$. It represents the dependence structure between the variables $\{X_1^1, \ldots, X_n^1\}$.
- $\mathcal{G}_\rightarrow = (V, E)$ is a bipartite graph between two sets of vertices both indexed by $\{1, \ldots, n\}$ and respectively representing the variables $\{X_1^t, \ldots, X_n^t\}$ and $\{X_1^{t+1}, \ldots, X_n^{t+1}\}$. In $\mathcal{G}_\rightarrow$, edges are directed from vertices at time t, to vertices at time $t+1$. Since the process is Markovian and stationary, the temporal independence between the variables are identical for all t.

Figure 1 shows the graphical representation of an example DBN with $n = 4$: $\mathcal{G}_\rightarrow$ is the left graph. Figure 1 (right) shows an equivalent representation of $\mathcal{G}_\rightarrow$, where nodes corresponding to variables X_i^t and X_i^{t+1} have been collapsed, and resulting self-loops have been omitted. This representation is more readable than the corresponding bipartite graph. Furthermore, it can also have a natural meaning with respect to the represented process, as will be the case with the ecological network case study we will describe in the section 3.
We classicaly define the set of *parents* of variable X_i^1 in $\mathcal{G}_1$ and of variable X_i^{t+1} in $\mathcal{G}_\rightarrow$ as: $Par(i, \mathcal{G}_1) = \{j, (j,i) \in E_1\}$ and $Par(i, \mathcal{G}_\rightarrow) = \{j, (j,i) \in E\}$.

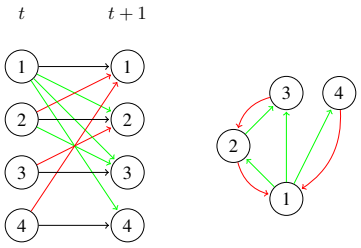

Figure 1 – DBN representation of an ecological interaction network.Left: Graph $\mathcal{G}_\rightarrow$ of the transition structure, where black edges describes the local species dynamic (survival or extinction) , green edges correspond to a positive influence, and red edges to a negative influence on the species dynamics. Right : Graph $\mathcal{G}$ of the associated ecological network

Then in a DBN, the initial probability distribution and the transition probability distribution are factored according to $Par(i, \mathcal{G}_1)$ and $Par(i, \mathcal{G}_\rightarrow)$:

$$P(X^1|a) = \prod_{i=1}^{n} P_i(X_i^1 | X^1_{Par(i,\mathcal{G}_1)}, a), \quad (2)$$

$$P(X^{t+1}|X^t, a) = \prod_{i=1}^{n} P_i(X_i^{t+1} | X^t_{Par(i,\mathcal{G}_\rightarrow),a}). \quad (3)$$

2.2 P-DBN

In order to complete the definition of a DBN, the tables of conditional probabilities, $P_i(X_i^1 | X^1_{Par(i,\mathcal{G}_1)}, a)$ and of transition probabilities $P_i(X_i^{t+1} | X^t_{Par(i,\mathcal{G}_\rightarrow)}, a)$ need to be specified, in addition to $\mathcal{G}_1$ and $\mathcal{G}_\rightarrow$. There may be a huge gain in space in representing individual tables $P_i(X_i^{t+1} | X^t_{Par(i,\mathcal{G}_\rightarrow)}, a)$ rather than direclty the joint transition probability $P(X^{t+1}|X^t, a)$. However, it is possible to save even more space, by using some domain-specific knowledge to specify that some individual transition probabilities are identical. This will be for instance the case when a limited number of interactions types between variables exist: then two individuals i and j, such that $Par(i, \mathcal{G}_\rightarrow)$ and $Par(j, \mathcal{G}_\rightarrow)$ correspond to the same influence have the same transition probabilities. Therefore we consider a labeled version of graph $\mathcal{G}_\rightarrow$, namely graph $\mathcal{LG}_\rightarrow = (V, (E_1, \ldots, E_L))$, where E_l is the set of edges with label l, from which we can define the sets of parents of type l, $Par^l(i, \mathcal{LG}_\rightarrow)$ of a vertex i. $Par(i, \mathcal{LG}_\rightarrow)$ is the union of such sets.
The space needed by the model can be reduced further by considering that the transition probability table of a variable X_i^t takes a pre-specified parameterized form $P_{i,\theta_\rightarrow}(X_i^{t+1} | X^t_{Par(i,\mathcal{G}_\rightarrow)}, a)$, where $\theta_\rightarrow$ is a (generally of small size) vector of parameters, shared by all individual transition probability tables.
With this representation, the transition probability function of a DBN can be modelled in a very concise way, by specifying only the (labelled) graph (sets $Par^l(i, \mathcal{LG}_\rightarrow)$) and the parameter vector $\theta_\rightarrow$. In Section 3, we will describe an example of such a parameterized transition function.

One of the advantages of using a parameterized representation of a DBN is that it may be learnt more efficiently from data (i.e. from observed trajectories of the variables(X_i^t)) than a non-parameterized representation.
In the next section, we describe a general approach to learn a parameterized DBN from data. In Sections 3 and 4 we will illustrate how it can be applied to an ecological network reconstruction problem.

2.3 Learning P-DBN

Learning a P-DBN amounts to learning both its independence structure $\mathcal{LG}_{\rightarrow}$, and its parameters, $\theta_{\rightarrow}$, from data D corresponding to a set of trajectories, $D = \{(x_i^t)\}_{i=1,\ldots,n;t=1,\ldots,T}$ and from a given value of covariates a. Recall that for a P-DBN, the number of parameters is fixed, independently of the graph $\mathcal{LG}_{\rightarrow}$. So we consider learning by maximizing the loglikelihood of the data directly, since a penalty term based on the model complexity would be meaningless. The likelihood $P_{\mathcal{LG}_{\rightarrow},\theta_{\rightarrow}}(D)$ of the data D for a given graph $\mathcal{LG}_{\rightarrow}$ and a set of values of parameters, $\theta_{\rightarrow}$, is defined as:

$$P_{\mathcal{LG}_{\rightarrow},\theta_{\rightarrow}}(D) = \prod_{t=1}^{T-1}\prod_{i=1}^{n} P_{\theta_{\rightarrow}}(x_i^{t+1}|x^t_{Par(i,\mathcal{LG}_{\rightarrow})}, a). \quad (4)$$

The learning problem amounts to finding $\hat{\mathcal{LG}}_{\rightarrow}$ and $\hat{\theta}_{\rightarrow}$ which jointly maximize $\log\left(P_{\mathcal{LG}_{\rightarrow},\theta_{\rightarrow}}(D)\right)$.
We will consider the general iterative procedure which provides a local maximum of $\log\left(P_{\mathcal{LG}_{\rightarrow},\theta_{\rightarrow}}(D)\right)$:
P-DBN Learning Procedure

$k \leftarrow 0$;
Choose an arbitrary $\mathcal{LG}_{\rightarrow}^{(0)}$;
Iterate
(1) $\theta_{\rightarrow}^{(k)} \leftarrow \arg\sup_{\theta\rightarrow} \log\left(P_{\mathcal{LG}_{\rightarrow}^{(k)},\theta_{\rightarrow}}(D)\right)$;
(2) $\mathcal{LG}_{\rightarrow}^{(k+1)} \leftarrow \arg\max_{\mathcal{LG}_{\rightarrow}} \log\left(P_{\mathcal{LG}_{\rightarrow},\theta_{\rightarrow}^{(k)}}(D)\right)$;
$k \leftarrow k+1$;
Until Convergence

Note that this algorithm is very general and the two steps (1) and (2) need to be specified to a given parameterised DBN learning problem, for the algorithm to be implemented. However, for all such implementations, the following proposition holds:

Proposition 1 *The procedure* **P-DBN Learning** *converges to a local maximum of* $\log\left(P_{\mathcal{LG}_{\rightarrow},\theta_{\rightarrow}}(D)\right)$.

Sketch of proof: First steps (1) and (2) jointly increase the log-likelihood :

$$\log\left(P_{\mathcal{LG}_{\rightarrow}^{(k+1)},\theta^{\rightarrow(k+1)}}(D)\right) \geq \log\left(P_{\mathcal{LG}_{\rightarrow}^{(k)},\theta^{\rightarrow(k)}}(D)\right), \forall k$$

Then, if for some k, $\mathcal{LG}_{\rightarrow}^{(k+1)} = \mathcal{LG}_{\rightarrow}^{(k)}$ in the algorithm, then $\theta^{\rightarrow(k+1)} = \theta^{\rightarrow(k)}$. In this case, the algorithm has converged. Finally, if $\mathcal{LG}_{\rightarrow}^{(k+1)} \neq \mathcal{LG}_{\rightarrow}^{(k)}$, $\mathcal{LG}_{\rightarrow}^{(k)}$ cannot be again a solution of another step $k' > k$. Since the possible set of graphs is finite, necessarily for some k we have $\mathcal{LG}_{\rightarrow}^{(k+1)} = \mathcal{LG}_{\rightarrow}^{(k)}$.

3 Learning an ecological network

3.1 Representing an ecological network as a directed labelled graph

An ecological network describes the relations between species in a given environment. These relations can be trophic (prey/predator), parasitic, competitive ... The knowledge of such a network can help a specialist to understand the effect of targeted conservation actions on the species interactions. In particular, management actions can be focused on some key species that will influence the whole network.
The structure of the interaction network can be modeled by a directed graph $\mathcal{G}$ with n nodes, one per species in the system. An edge from species i to species j represents an influence (positive or negative) of species i on the dynamics of j. An edge from i to j is labelled, to characterize the nature of the influence. It is labelled "+" if i has a positive influence on species j's survival probability. i can be a prey or a facilitator (if, for instance, i provides a shelter to j) of j. It is labelled "$-$" if i has a negative influence on species j's survival probability (i can be a predator, a competitor or a parasite of j).
Influences between pairs of species can be combined to define higher level, ecologically meaningful interactions. For instance, a pair of edges $(i \overset{+}{\rightarrow} j, i \overset{-}{\leftarrow} j)$ can describe a trophic or a parasitic relation, where j is a *specific predator/parasite* of i. The configuration with an edge $i \overset{+}{\rightarrow} j$ without $i \overset{-}{\leftarrow} j$ describes that i is a *facilitator* of j.
See Figure 1 (right) for an example ecological interaction network with 4 species.

3.2 DBN model for the species dynamics

In practice, not every relations are known. However, co-occurence data may be available during T consecutive time steps. For all $i \in \{1,\ldots,n\}$ and for all $t \in \{1,\ldots,T\}$, we observe the absence ($x_i^t = 0$) or the presence ($x_i^t = 1$) of the species i at time t. Information may also be available on whether the observed area is protected ($a^t = 1$) or not ($a^t = 0$). The objective is then to learn the edges of $\mathcal{G}$ and their labels.
To solve this learning problem, we model the species dynamics as a DBN. Each observation x_i^t is modeled as the realisation of a random variable $X_i^t \in \{0,1\}$ representing the absence ($X_i^t = 0$) or the presence ($X_i^t = 1$) of the species i at time t. Then from the graph $\mathcal{G}$ representing the interaction network, we define $\mathcal{LG}_{\rightarrow}$: there is an edge from X_i^t to X_j^{t+1} if and only if the edge from i to j exists in $\mathcal{G}$. In addition, all edges from X_i^t to X_i^{t+1} are present. Then we define $Par^+(i,\mathcal{LG}_{\rightarrow}) \subset \{1,..,n\}$, and $Par^-(i,\mathcal{LG}_{\rightarrow}) \subset \{1,..,n\}$ as respectively the set of species with positive and with negative influence on i. Therefore $Par(i,\mathcal{LG}_{\rightarrow}) = Par^+(i,\mathcal{LG}_{\rightarrow}) \cup Par^-(i,\mathcal{LG}_{\rightarrow}) \cup \{i\}$. See Figure 1 for an example of correspondance between an interaction network $\mathcal{G}$ and the corresponding graph $\mathcal{L}G_{\rightarrow}$ into the DBN model.

The global transition function of the DBN is given by eq (3). Each individual transition probability is defined in terms of the vector of parameters $\theta_{\rightarrow} = (\varepsilon, \rho^+, \rho^-, \mu)$ is ε, the probability of recolonization, i.e. of observing $x_i^{t+1} = 1$ when $x_i^t = 0$ (recolonization probability is assumed identical for all species). ρ^+ and ρ^- are the probabilities of success of positive and negative influences. $\mu \in [0, 1[$ is a penalty factor applied to recolonization and survival probability of species (assumed identical for all species) when the area is not protected.

In this model, it is assumed that a species survives if at least one positive influence succeeds and all negative ones fail[1]. In addition, we make two other assumptions. A species with empty $Par^+(i, \mathcal{LG}_{\rightarrow})$ (for instance a species at the bottom of the trophic chain) cannot disappear if it is protected and all the species in $Par^-(i, \mathcal{LG}_{\rightarrow})$ are absent. A species with non empty $Par^+(i, \mathcal{LG}_{\rightarrow})$ cannot survive if all species in $Par^+(i, \mathcal{LG}_{\rightarrow})$ are absent.

These assumptions act as hard constraints on the possible transitions of the system. They forbid some specific interactions between species, given observed data.

Now, let us describe the transition probabilities towards presence $P_i(X_i^{t+1} = 1 | X_i^t, a^t)$. All other transition probabilities can be derived from these. Let us consider first the case where the observed area is protected ($a^t = 1$). Two situations are possible.

(i) the probability for a species absent at t to recolonize the observed area at $t + 1$ is assumed independent of the presence of other species and is defined as:

$$P_i(X_i^{t+1} = 1 | X_i^t = 0, a^t = 1) = \varepsilon. \tag{5}$$

(ii) Then, let us denote $N_{i,+}^t = |\{j \in Par^+(i, \mathcal{LG}_{\rightarrow}), X_j^t = 1\}|$, and $N_{i,-}^t = |\{j \in Par^-(i, \mathcal{LG}_{\rightarrow}), X_j^t = 1\}|$ the number of species present at t and respectively with positive or negative influence on i. The probability for a species present at t to survive at $t + 1$ is the probability of success of at least one positive influence (if needed) and the probability of failure of all negative influences. It can be expressed as

$$P_i(X_i^{t+1} = 1 | X_i^t = 1, x^t_{Par^-(i,\mathcal{G}_{\rightarrow})}, a^t = 1) = \left(1 - \rho^-\right)^{N_{i,-}^t} \quad \text{if } Par^+(i, \mathcal{G}_{\rightarrow}) = \emptyset. \tag{6}$$

$$P_i(X_i^{t+1} = 1 | X_i^t = 1, x^t_{Par^+(i,\mathcal{G}_{\rightarrow})}, x^t_{Par^-(i,\mathcal{G}_{\rightarrow})}, a^t = 1) = \left(1 - \left(1 - \rho^+\right)^{N_{i,+}^t}\right)\left(1 - \rho^-\right)^{N_{i,-}^t} \quad \text{if } Par^+(i, \mathcal{G}_{\rightarrow}) \neq \emptyset. \tag{7}$$

When the considered area is not protected ($a^t = 0$), we assume that the transition probabilities (5), (6) and (7) are multiplied by μ, to account for the loss in recolonization/survival probability when species are not protected.

[1] The exact model will be described later.

4 Finding graph structure and parameters with maximum likelihood

We now describe how we derive an iterative algorithm for learning an ecological interaction network and the parameters of the species dynamics from the generic learning procedure of Section 2.3. The **P-DBN learning** procedure alternates between two steps. The maximization step of the loglikelihood of the parameters for a given graph structure uses the interior point method for non-linear programming [1]. For the second step, we formulate the maximization of the likelihood of the graph as an integer linear programming problem by introducing additional variables to describe the network and the links between the network and the data. Before detailing this second step, we describe the loglikelihood expression of the model. Even if the first step uses an approximate method, the property that the loglikelihood is increased at each step will hold, guaranteeing the convergence of the algorithm to a local maximum likelihood solution.

4.1 The model loglikelihood

To express the model loglikelihood, we define two types of species behaviour: type nb, for species that have a non empty $Par^+(i, \mathcal{LG}_{\rightarrow})$, and type b for the others (the *basal* species, which have no prey). We also define the quantity $R_{i,V}^{t,d^+,d^-}$ equal to 1 if the species i is of type $V \in \{nb, b\}$ and at time t, $N_{i,+}^t = d^+$, $N_{i,-}^t = d^-$ and 0 otherwise. By convention, for a species of type b, we set $N_{i,+}^t = 0$. We also assume that the maximum overall number of incoming edges of any node i is known, equal to k. With this definition, the loglikelihood of a dataset $D = \{x^1, \ldots, x^T\}$ can be expressed as follows[2]:

$$\log P_{\mathcal{LG}_{\rightarrow},\theta}(x^2, \ldots, x^T \mid x^1, a) = \sum_{i=1}^{n} score(i) \tag{8}$$

The $score$ is the contribution of a species i to the likelihood. It is expressed as follows :

$$score(i) = \sum_{t=1}^{T-1} (1 - x_i^t) \cdot \log\left(P_0^t(x_i^{t+1})\right) + \sum_{d^+=0}^{k} \sum_{d^-=0}^{k-d^+} x_i^t \cdot \log\left(P_{1,+}^{t,d^+,d^-}(x_i^{t+1})\right) \cdot R_{i,V_n b}^{t,d^+,d^-} + x_i^t \cdot \log\left(P_{1,b}^{t,d^+,d^-}(x_i^{t+1})\right) \cdot (1 - R_{i,V_b}^{t,d^+,d^-}).$$

For one species i at one time t, there is only one element which is non-zero (inside the sum over T). This element is itself a sum of three terms, only one of which is not equal to zero, the one corresponding to the probability of the transition from $x_i^t = 0$ to x_i^{t+1} ($P_0^t(x_i^{t+1})$) or $x_i^t = 1$ to x_i^{t+1} for basal species ($P_{1,b}^{t,d^+,d^-}(x_i^{t+1})$) or non-basal species ($P_{1,+}^{t,d^+,d^-}(x_i^{t+1})$). These probabilities write:

$$\log\left(P_0^t(x_i^{t+1})\right) = x_i^{t+1} \cdot a^t \cdot \log\epsilon + (1 - x_i^{t+1}) \cdot a^t \cdot \log(1 - \epsilon) + x_i^{t+1} \cdot (1 - a^t) \cdot \log(\mu\epsilon) + (1 - x_i^{t+1}) \cdot (1 - a^t) \cdot \log(1 - \mu\epsilon).$$

[2] $P_{\mathcal{LG}_{\rightarrow},\theta}(x^1)$ is not estimated.

And

$$\log\left(P_{1,V}^{t,d^+,d^-}(x_i^{t+1})\right) = x_i^{t+1}\cdot a^t\cdot\log\left(P_{1\to1}^{1V}(d^+,d^-)\right) + (1-x_i^{t+1})\cdot a^t\cdot\log\left(P_{1\to0}^{1V}(d^+,d^-)\right) + x_i^{t+1}\cdot(1-a^t)\cdot\log\left(P_{1\to1}^{0V}(d^+,d^-)\right) + (1-x_i^{t+1})\cdot(1-a^t)\cdot\log\left(P_{1\to0}^{0V}(d^+,d^-)\right),$$

where $P_{1\to x_i^{t+1}}^{a^tV}(d^+,d^-)$ is the probability to transition from $x_i^t=1$ to x_i^{t+1} for species i of type V under action a^t, when it has d^+ favourable and d^- unfavourable species extant. Those probabilities are described in (6) and (7).
The interest of expression (8) of the loglikelihood is that it is a linear function of the variables $\{R_{i,V}^{t,d^+,d^-}\}$, given the data $\{x_i^t\}$, $\{a^t\}$ and parameters $(\varepsilon,\rho^+,\rho^-,\mu)$ of the model. We will exploit it further in Section 4.2.

4.2 Graph structure estimation for given parameters

This step consists in learning a network that maximizes the loglikelihood, the parameters being fixed. We describe the network by two sets of binary variables: $\{g_{ij}^+\}_{1\le i,j\le n}$ and $\{g_{ij}^-\}_{1\le i,j\le n}$, such that $g_{ij}^+=1$ iff $j\in Par^+(i,\mathcal{LG}_\to)$ (Similarly for g_{ij}^-). So the optimization problem is that of optimizing expression (8) over variables $\{g_{ij}^-\}_{1\le i,j\le n}$ and $\{g_{ij}^-\}_{1\le i,j\le n}$. However, expression (8) does not depends explicitly on these variables. Indeed, it is an expression of the variables $\left\{R_{i,V}^{t,d^+,d^-}\right\}$ which are themselves functions of the variables g_{ij}^+ and g_{ij}^-, as well as of the observed data D. In the following we show how to define the binary variables $\left\{R_{i,V}^{t,d^+,d^-}\right\}$ from linear constraints involving the binary variables g_{ij}^+ and g_{ij}^-, as well as the data and some other auxiliary binary variables. So doing, we will have described a linear programming problem which can be solved by classical algorithms. This linear programming problem is decomposable into one problem per species. A linear programming problem for a species i describes the vertices received by i that maximise the quantity $score(i)$. The following variables are defined for a particular species i.
A first auxiliary binary (0-1) variable is added to identify whether the species i is basal or not: $h_i^nb = 1$ iff $Par^+(i,\mathcal{LG}_\to)\neq\emptyset, \forall i=1,\dots,n$.
This binary variable is defined by the following linear constraints, involving variables (g_{ij}^+):

$$h_i^nb \le \sum_{j=1}^n g_{ji}^+, \forall i\in\{1,..,n\}, \quad (9)$$

$$h_i^nb \ge g_{ji}^+, \forall(i,j)\in\{1,..,n\}^2. \quad (10)$$

Now, let us define the set of 0-1 variables $\left\{M_i^{t,d^+}\right\}_{i=1..n,t=1..T,d^+=0..k}$, where k is the maximum allowed number of parents of any label. $M_i^{t,d^+}=1$ iff the species i has at least $d+$ parents of label type $+$ extant at time t : $M_i^{t,d^+}=1$ iff $N_{i,+}^t\ge d^+$.
These binary variables are defined by the following linear constraints, involving variables $\{g_{ij}^+\}$ as well as the observed data:

$$M_i^{t,d^+}\cdot(d^++1)-\sum_{j=1}^n\left(g_{ji}^+\cdot x_i^t\right) \le 1, \quad (11)$$

$$M_i^{t,d^+}\cdot(k+1-d)-\sum_{j=1}^n\left(g_{ji}^+\cdot x_i^t\right) > -d^+, \quad (12)$$

M_i^{t,d^-} can be defined the same way.
Symmetrically, let us define the set of binary variables $\left\{\nu_i^{t,d^+}\right\}_{i=1..n,t=1..T,d=0..k}$, where $\nu_i^{t,d^+}=1$ iff the species i has at most d^+ parents of label type $+$ extant at time t: $\nu_i^{t,d^+}=1$ iff $N_{i,+}^t\le d$. These integer variables are defined by the following linear constraints, involving variables $\{g_{ij}^+\}$ as well as the observed data :

$$\nu_i^{t,d^+}\cdot(k+1-d^+)+\sum_{j=1}^n\left(g_{ji}^+\cdot x_i^t\right) \le k+1, \quad (13)$$

$$\nu_i^{t,d^+}\cdot(d^++1)+\sum_{j=1}^n\left(g_{ji}^+\cdot x_i^t\right) > d^+, \quad (14)$$

ν_i^{t,d^-} can be defined the same way.
Finally, we can define the set of 0-1 variables $\left\{\Lambda_i^{t,d^+}\right\}_{i=1..n,t=1..T,d=0..k}$, where $\Lambda_i^{t,d^+}=1$ iff the species i has exactly d parents of label type $+$ extant at time t: $\Lambda_i^{t,d^+}=1$ iff $N_{i,+}^t=d^+$. These variables are defined by the following set of constraints:

$$\Lambda_i^{t,d^+}-M_i^{t,d^+} \le 0, \quad (15)$$

$$\Lambda_i^{t,d^+}-\nu_i^{t,d^+} \le 0, \quad (16)$$

$$\Lambda_i^{t,d^+}-M_l^{t,d^+}-\nu_i^{t,d^+} \ge -1, \quad (17)$$

Λ_i^{t,d^-} can be defined the same way.
Now, we are ready to write the linear constraints defining the binary variables $R_{i,V}^{t,d^+,d^-}$. Recall that $R_{i,V}^{t,d^+,d^-}=1$ if and only if the species i is of type $V\in\{nb,b\}$ and has exactly d^+ parents of type $+$ and d^- parents of type $-$ extant at time t. Thus,

$$R_{i,nb}^{t,d^+,d^-}=1 \quad\text{iff}\quad h_i^nb=1,\Lambda_{i,+}^{t,d^+}=1,\Lambda_{i,-}^{t,d^-}=1.$$
$$R_{i,b}^{t,d^+,d^-}=1 \quad\text{iff}\quad h_i^nb=0,\Lambda_{i,+}^{t,d^+}=1,\Lambda_{i,-}^{t,d^-}=1.$$

Remark that in the second series of equations, $\Lambda_{i,+}^{t,d^+}=1\Rightarrow d^+=0$.
Finally, these variables are defined by the following set of linear constraints, $\forall t,i,V,l,d^+,d^-$:

$$R_{i,V}^{t,d^+,d^-} \le \Lambda_{i,l}^{t,d^l}, \quad (18)$$

$$R_{i,nb}^{t,d^+,d^-} \le h_i^+, \quad (19)$$

$$R_{i,b}^{t,d^+,d^-} \le 1-h_i^+, \quad (20)$$

$$R_{i,V}^{t,d^+,d^-} \ge \Lambda_{i,+}^{t,d^+}+\Lambda_{i,-}^{t,d^-}-1. \quad (21)$$

At this stage, we know every variable we need for the calculation of the likelihood. We can exploit further the LP

formulation to include expert knowledge on ecological networks, in the form of additional constraints. First, we impose that the species i has at most k parents. This is indicated by an additional set of constraints to the problem:

$$\sum_{j=1}^{n} \sum_{l \in \{+,-\}} g_{ij}^{l} \leq k, \forall i = 1, \ldots, n. \quad (22)$$

If the species i is non-basal, it will become extinct at time $t+1$ if they have no prey at time t. This can be forced by:

$$R_{i,nb}^{t,d^+=0,d^-} \cdot x_i^{t+1} = 0, \forall i, t, d^-. \quad (23)$$

If the species is basal, it will remain extant at time $t+1$ if they are extant at time t without any negative influence. This can be forced by:

$$R_{i,b}^{t,d^+,d^-=0} \cdot x_i^t \cdot (1 - x_i^{t+1}) = 0, \forall i, t, d^+. \quad (24)$$

Finally, we have modelled the problem of finding the ecological network structure which optimizes the likelihood as a 0-1 linear program, in variables $\{g_{ij}^l, M_{i,l}^{t,d}, \nu_{i,l}^{t,d}, \Lambda_{i,l}^{t,d}, R_{i,V}^{t,d^+,d^-}\}$ with constraints (9–24), and criterion (8). The total number of variables of a linear programming problem for a species is :

$$\left(3 \cdot n + 1 + T \cdot \left(\frac{k^2}{2} + \frac{3 \cdot k}{2} + 8\right)\right)$$

The number of constraints of this problem is :

$$\left(n + 1 + T \cdot \left(2 \cdot k^2 + 6 \cdot k + 21\right)\right)$$

5 Experimental results

We have tested the performance of the two steps of the algorithm, parameter estimation and graph structure learning, as well as the complete P-DBN learning algorithm on toy examples of ecological networks and data sets simulated from the DBN described in Section 3.2. The network we consider is the one represented on Figure 1, with a number of species $n = 4$. We assume that the species have been monitored for $T = 30$ time step and that the area was *unprotected* the 12 first years of observations ($a^{\{t=1,..,12\}} = 0$), and *protected* the 18 last years ($a^{\{t=13,..,30\}} = 1$). Different configurations were tested for $\theta = \{\varepsilon, \rho^+, \rho^-, \mu\}$: each parameter was set either to 0.2 (low influence) or to 0.8 (strong influence) for a total of 2^4 possible configurations. Each configuration is indexed is indexed lexicographically : $1(0.2, 0.2, 0.2, 0.2)$; $2(0.2, 0.2, 0.2, 0.8)$; $3(0.2, 0.2, 0.8, 0.2)$; ...; $15(0.8, 0.8, 0.8, 0.2)$; $16(0.8, 0.8, 0.8, 0.8)$.

5.1 Performance of P-DBN parameter estimation step

The estimation of the parameters is tested on data simulated based on the network on Figure 1. For each of the 16 parameters configurations described previously, 150 simulations of the species dynamics are realized and the parameters are estimated by the maximisation of the likelihood knowing the graph used to generate the data, using the fmincon function in Matlab. Some performance indicators such as the mean and the standard deviation of the estimators on the 150 simulations of each set of parameters are shown on figure 2. This estimation is quite satisfying. In average, the estimated parameters are quite close to the real parameters. However, when the value of a parameter is low, it is more difficult to estimate.

5.2 Performance of P-DBN structure learning step

The network $\mathcal{LG}_{\rightarrow}$ is the one of Figure 1. For each of the 16 parameters configuration, we ran 150 simulations of the species dynamics. In order to limit the size of the search space, we fixed the maximum number of parents for each species to $k = 2$. The optimization by linear programming is done using the software cplex.

For a given configuration of θ, we used different performance indicators we computed the average time (over the 150 simulations) taken for learning the structure, $P_{\hat{\mathcal{LG}}_{\rightarrow},\theta_{\rightarrow}}(D)$, and the average precision and recall for edges labeled $+$ and edges labeled $-$. These indicators are computed from the confusion matrix between $\mathcal{LG}$ and $\hat{\mathcal{LG}}$. In this case, the confusion matrix is a 3×3 matrix since for each oriented pair of species (i, j) either there is no edge, or an edge labeled $+$ or an edge labeled $-$. The precision measures the proportion of relations of a certain type correctly learnt amongst every relations of this type learnt. For instance, for the edges labeled $+$ it corresponds to the quantity : $\frac{+ \text{ edges correctly learnt}}{+ \text{ edges learnt}}$. The recall measures the proportion of relations of a certain type correctly learnt amongst every relation of this type that exist in the original graph. For instance, for the edges labeled $+$ it corresponds to the quantity : $\frac{+ \text{ edges correctly learnt}}{+ \text{ edges present in the } \textit{original graph}}$. For a graph of this size, it takes in average 35.6 seconds to learn the structure of a graph knowing the parameters of the model. The other indicators are represented in Figure 3.

For this small problem, the results are dependent on the parameters configuration. The quality of the learning, measured by the precision and recall, highly improves when the parameters ρ and ε increases and slightly when the parameter μ increases. Expert knowledge is still required to determine which precision and recall are acceptable to be applied on real data. However, we can assume that a recall and a precision of 70% or more is reasonable and covers a good part of the real graph, which is the case when all parameters are high.

5.3 Performance of the P-DBN learning algorithm

For the learning of both the graph structure and the estimation of the parameters, the data are simulated with the help of a graph inspired by the alaskan food web [8] that count 13 species. The maximum of parents in this graph is 4, which lead to a huge amount of variables as described in section 4.3. 100 simulations of species dynamics are realized with this graph, but for each of them, the structure

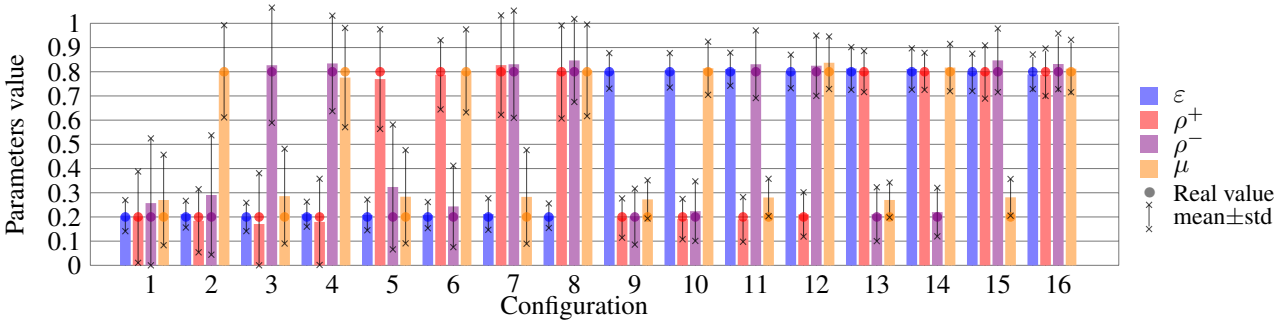

Figure 2 – Diagram of the mean value of the estimated parameters for 150 simulations for the 16 configurations. The dots represents the real value of the parameters, the interval between the cross corresponds to the difference between the mean value of the estimated parameters and the standard deviation of those estimations.

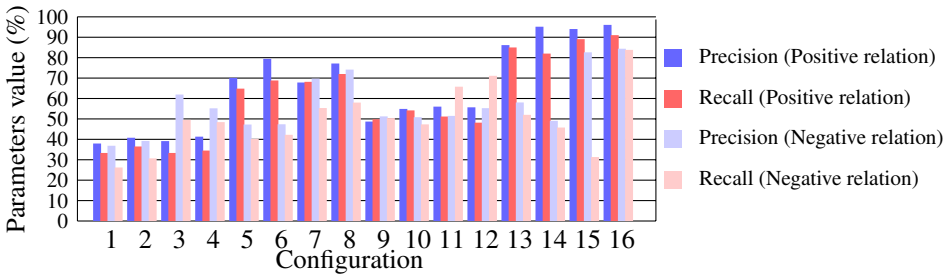

Figure 3 – Representation of the quality measures of the structure learning of a network by linear programming for each 150 simulations of each parameters configuration.

learning algorithm is performed with a limitation of maximum number of parents $k = 2$. The global quality of the

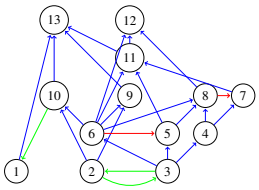

Figure 4 – Ecological network considered for testing the algorithm. " + " edges are represented in green and " − " edges are represented in red. For a more comprehensive representation, a blue edge from a species a to a species b means that there is a " + " edge between a and b and a " − " edge between b and a, which often represents a trophic relation.

algorithm can be measured by the similarities between the learnt graphs and the graph used to generate the data, with the help of the *precision* and the *recall* obtained by the cumulated confusion matrix on table 1. The algorithm takes an average of 3.68 simulations. The precision is 47.51% for the edges labelled " + " and 34.41% for the edges labelled " − ". The recall is 21.04% for the edges labelled "+" and 20.35% for the edges labelled "−". These results may not seem great. However, it is quite rare that an edge is

		Learnt graph			
		$\emptyset$	+	-	Total
Original graph	$\emptyset$	10860	487	853	12200
	+	1856	505	39	2400
	-	1761	71	468	2300
	Total	14477	1063	1360	16900

Table 1 – Cumulated confusion table. Count amongst every possible edges between two species from every learnt graph which label it have been learnt (" + ", " − " or no edge : $\emptyset$) and how it appears in the original graph that has generated the data

mislabelled. The errors are most often due to edges learnt while not existing, or not learnt when it should be. If we choose another aggregation, the results are more satisfying. If we consider a graph composed of the 50 most often learnt labelled edge amongst the 100 simulations, and compare it with the original graph, the precision is 83.33% for the + labelled edges and 53.85% for the − labelled edges. The recall is 83.33% for the + labelled edges and 60.87% for the − labelled edges. It seems that a single simulation leads to some errors, but the edges learnt by a majority of simulations are relevant.

6 Conclusion

In this paper, we proposed an approach to learn the structure of a parameterized Dynamic Bayesian Network (DBN), where the temporal dynamics is described by a limited number of parameters. It consists in a greedy iterative two-steps algorithm : *parameter estimation* computes the maximum likelihood estimates for a fixed structure, and *structure improvement* computes the structure of the DBN maximizing the loglikelihood, given a fixed set of parameters. The *parameter estimation* step uses a standard likelihood maximization algorithm, which may lead to a suboptimal set of parameters. The *structure improvement* step is modelled as a 0-1 integer linear program, which can also be solved using standard algorithms. This procedure is generic since the loglikelihood of a DBN can always be decomposed as a linear function of variables describing the graph structure, as in [3], and as soon as additionnal constraints on these variables are linear, ILP can be applied.

Still, for a specific parameterized DBN, an ILP model can be derived, requiring fewer variables. In particular, we considered the problem of learning an ecological interaction network from temporal data of presence/absence of species. In this case, there is a limited number of types of defined possible relations between the species, and the DBN of the species dynamics only has few parameters. We have applied the learning algorithm on data simulated on a quite small network with different sets of parameters' values. Parameters' estimation is satisfying in general, and all the more as the interactions are strong. The same conclusion holds for the structure learning step.

In order to keep the learning problem tractable, we limited the number of allowed parents in the learnt network. This limitation may not hold for the real graph. One possible empirical way to overcome this limitation is to learn several graphs from (non disjoints) sub-sets of the variables and then to aggregate these networks, into one with more parents. We have suggested a naive way to aggregate learnt networks that seems to lead to good results, still there is room for improving this method.

Solving the 0-1 ILP step may become rapidly intractable when the number of variables increases (the number of variables of the ILP increases polynomially with this number). The natural solution would be to apply continuous relaxation to solve an easier (tractable) LP problem, and then round the solution to obtain an approximate structure improvement step. Alternatively, the parameter estimation step could be performed directly from the continuous solutions of the LP, interpreted as "probabilities of presence" of links, in the spirit of the EM algorithm.

References

[1] Richard H Byrd, Mary E Hribar, and Jorge Nocedal. An interior point algorithm for large-scale nonlinear programming. *SIAM Journal on Optimization*, 9(4):877–900, 1999.

[2] David Maxwell Chickering. Learning bayesian networks is np-complete. pages 121–130, 1996.

[3] James Cussens. Bayesian network learning with cutting planes. In *UAI 2011, Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence, Barcelona, Spain, July 14-17, 2011*, pages 153–160, 2011.

[4] Nir Friedman, Kevin Murphy, and Stuart Russell. Learning the structure of dynamic probabilistic networks. In *Proc. of the 14th Conf. on Uncertainty in Artificial Intelligence (UAI'98)*, 1998.

[5] Zoubin Ghahramani. Learning dynamic bayesian networks. *Lecture Notes in Computer Science*, 1387:168–197, 1997.

[6] David Heckerman, Dan Geiger, and David M Chickering. Learning bayesian networks: The combination of knowledge and statistical data. *Machine learning*, 20(3):197–243, 1995.

[7] David Liben-Nowell and Jon Kleinberg. The link-prediction problem for social networks. *Journal of the American society for information science and technology*, 58(7):1019–1031, 2007.

[8] E McDonald-Madden, R Sabbadin, ET Game, PWJ Baxter, I Chadès, and HP Possingham. Using food-web theory to conserve ecosystems. *Nature Communications*, 7, 2016.

[9] Isobel Milns, Colin M Beale, and V Anne Smith. Revealing ecological networks using bayesian network inference algorithms. *Ecology*, 91(7):1892–1899, 2010.

[10] Radu Stefan Niculescu, Tom M Mitchell, and R Bharat Rao. Bayesian network learning with parameter constraints. *The Journal of Machine Learning Research*, 7:1357–1383, 2006.

[11] Gideon Schwarz. Estimating the dimension of a model. *The annals of statistics*, 6(2):461–464, 1978.

[12] Eran Segal, Dana Pe'er, Aviv Regev, and Daphne Koller. Learning module networks. In *Proceedings of the Nineteenth conference on Uncertainty in Artificial Intelligence*, pages 525–534. Morgan Kaufmann Publishers Inc., 2002.

[13] Jing Yu, V Anne Smith, Paul P Wang, Alexander J Hartemink, and Erich D Jarvis. Advances to bayesian network inference for generating causal networks from observational biological data. *Bioinformatics*, 20(18):3594–3603, 2004.