

Evolutions Detection Using Markov Models

Arnaud Paris, Adnen El-Amraoui, Nathalie Cislo, and Selma Arbaoui and Nacim Ramdani

Abstract—We consider a specific case of the anomaly detection problem where the considered system is unknown but observable. The aim is to detect if this system changes or not for a long period, and, if possible, to characterize these evolutions. The proposed method consists of two steps. The first one is to learn the system thanks to a Hidden Markov Model with explicit duration of states, using observations of the system at different moments. The second step is to compute distances between these models to detect the evolutions. Due to the constraints of the learning algorithm and computational time, we proposed a new method to compute distances between Markov models, based on the Kullback-Leibler divergences. The method is tested on a trial evolving model. These results are discussed and limits of the method are highlighted.

Index Terms—Evolution detection, distance measure, markov models, model comparison.

I. INTRODUCTION

Anomaly detection is a class of problem which correspond to find if a set of data is conform to an expected behavior of the system, and otherwise, define the subset which not respect the expected behavior. These problems and used methods are presented in a survey [1], with different application domains. Different variants exist, for example, the study of anomaly detection in the specific case of temporal evolution [2].

We consider a specific case of the anomaly detection problem where the expected behavior is unknown and can evolve in time. The problem is not to find outliers data but to find when the system evolves and his behavior change. This problem corresponds to anomaly detection of data with previous data. A further problem is to characterize the evolution.

If the system is complex and unknown, we suppose three hypotheses about this model in order to detect and characterize these evolutions.

The former is that the system follows a Markov Process, i.e. the states s_t of the system at a time step t depends only on the previous state at $t-1$ and not at all of the history ($P(s_{t+1} = x | s_t, s_{t-1}, \dots) = P(s_{t+1} = x | s_t)$). Markov processes are used in anomaly detection [3] and in modelization of complex systems [4].

The second hypothesis supposes that the system emits, at each time step, a set of observations, depending on the system

state ($P(o_t | s_t)$ where o_t is an emission).

The third hypothesis assumes that the system can remain many time steps in a state before switching in another state. This duration time depends on the system state. This hypothesis allows studying the temporal evolution [2] and confirms a Markov Model. Transitions of states explain the temporal order of states. However, to study the duration of a state of the model, a semi-Markov model is usefully.

To take into account to these hypotheses, it is possible to describe the system as a Gs-GsHSMM (Gaussian-Gaussian Hidden Semi-Markov Model). However, a Gs-GsHSMM cannot modelize the unpredictable evolution of the model. That is why we use a learning algorithm to get the regular model.

In addition, a model can evolve over time in different ways. The elementary evolutions are on transitions, on emissions and on duration times in a state. A system evolving is a set of these different elementary evolutions.

The rest of the paper is organized as follows. Section II described the used models; the Gs-GsHSMM. Section III presents a state of the art of methods to compute distances between Markov models. Section IV presents the proposed methods. Finally and before concluding, Section V illustrates some experiments and discusses results.

II. Gs-GsHSMM

A. Description of the Model

Gs-GsHSMMs are models based on Markov processes. These models are defined by the tuple (Π, A, Δ, τ) where Π is the initial state distribution, A is the state transition probability distribution, Δ is the duration probability distribution and τ is the emission probability distribution. In our work, we do not take into account Π , which values are considered equally likely.

At each time step t , the system may be described by a state s_t included in a state space S . The matrix A gives the probability of transition a_{ij} , where $i, j \in S$ and a_{ij} is the probability of transition from states $s_t = i$ to the state $s_{t+1} = j$. Self-transitions are not allowed, thus $a_{ii} = 0 \forall i$.

The system state is not directly known, but at each time step, the system emits an observation o_t , in a probabilistic way according to the current state. Probabilities of emission are given by $\tau = \{\tau_i(k)\}$ where $\tau_i(k)$ is the probability that the observation k is emitted by the state i . The vector k is composed of binary and continuous values. Each element of the observation vector is modeled by a Gaussian distribution.

The transition between states is allowed only after a duration time d , defined in a probabilistic way according to the current state. This description allows us to define explicitly the duration times in a state, given by Δ , according to formulas (1) and (2). These duration times are

Manuscript received July 25, 2016; revised January 12, 2017. Thanks to the Urban Community of Bourges, the Generals Councils of Cher and of Loir et Cher, the Regional Councils of the Centre and of Limousin, the French Industry Ministry, the European fund for regional development FEDER (of Centre and Limousin) and the French Oseo (BPI) for their financial support within the framework of the E-monitor'âge project through the Competitiveness Clusters S2E2 and Elopsys.

The authors are with the Univ. Orléans, INSA-CVL, PRISME EA 4229, F45072, Orléans, France (e-mail: arnaud.paris1@univ-orleans.fr).

probabilistically defined by Gaussian distributions. These distribution functions are truncated because the learning algorithm must have a maximum duration time D_{max} . Thus these functions are defined in $[1; D_{max}]$. The probabilities distributions should, then, respect $\sum_{d=1}^{D_{max}} P(d|s) = 1 \forall s$. Gaussian functions are thus truncated and normalized.

$$\begin{cases} P(d_t|s_t = i) = \Delta_{i,d} \text{ if } s_t \neq s_{t-1} \\ d_t = d_{t-1} - 1 \text{ else} \end{cases} \quad (1)$$

$$\begin{cases} P(s_t|s_{t-1}, d_{t-1}) = P(s_t|s_{t-1}, A) \text{ if } d_{t-1} = 1 \\ P(s_t|s_{t-1}, d_{t-1}) = 1 \text{ if } d_{t-1} > 1 \text{ and } s_t = s_{t-1} \\ P(s_t|s_{t-1}, d_{t-1}) = 0 \text{ if } d_{t-1} > 1 \text{ and } s_t \neq s_{t-1} \end{cases} \quad (2)$$

B. Trial Model

To evaluate distance methods, we use an evolving model. This experience model is a Gs-GsHMM with 4 states. The transition matrices, initial and final, are presented by the Fig. (1). Modifications in the transition matrix consist of evolutions of probabilities, including probabilities that become null or, on the contrary, null probabilities become no-null.

The Fig. (2) presents evolution of the duration probabilities. This evolution affects the mean and the variance of the Gaussian functions that modelize the duration time probabilities.

The emission of the system consists of 5 values, 1 continuous and 4 binaries. Thus, tau consists of 20 Gaussian functions. Like for the duration time probabilities, these may evolve in mean and in variance.

Due to the model, three kinds of evolution can be detected, each bound to an element of the model. A method to compute distance between two models should take into account all these elements to correctly detect all kinds of evolution. Indeed, distributions of models can be very different, while just one parameter is different. If we supposed that an evolution of the system implies several parameters change, the constraint can be weakening.

Others constraints should to be take into account. To detect an evolution of the distance between two models in true situations, time to compute distances should to be acceptable.

Methods to compute distances between models are studied in the next section.

III. DISTANCES BETWEEN MODELS

A method to compute distances should respect 4 properties to define a metric system. The first property is that a distance has to be positively defined ($D(\lambda_1, \lambda_2) \geq 0$). The others properties are the commutativity ($D(\lambda_1, \lambda_2) = D(\lambda_2, \lambda_1)$), the reflexivity ($D(\lambda_1, \lambda_1) = 0$) and the triangular inequality ($D(\lambda_1, \lambda_2) + D(\lambda_2, \lambda_3) \geq D(\lambda_1, \lambda_3)$). Distances between Markov models do not always respects all these properties. For example, a no-symmetrical distance can be characteristic of a model that which can explain another model but without that the contrary is true [5].

Several methods exist to compute distance between two models of Markov, including HMM. Reference [6] describes a method based on the KL divergence in the specific case of

left-right HMM, used in speech recognition. The distance described by Falkhausen [5] uses also the KL divergence, with Viterbi. For this distance, it is necessary that models have same number of states. Reference [7] describes a method that uses an observation sequence, generates randomly, to compute relative entropy between models. The length of the sequence modifies results of the distance. To have an exact result, it is necessary to take a length close to infinity.

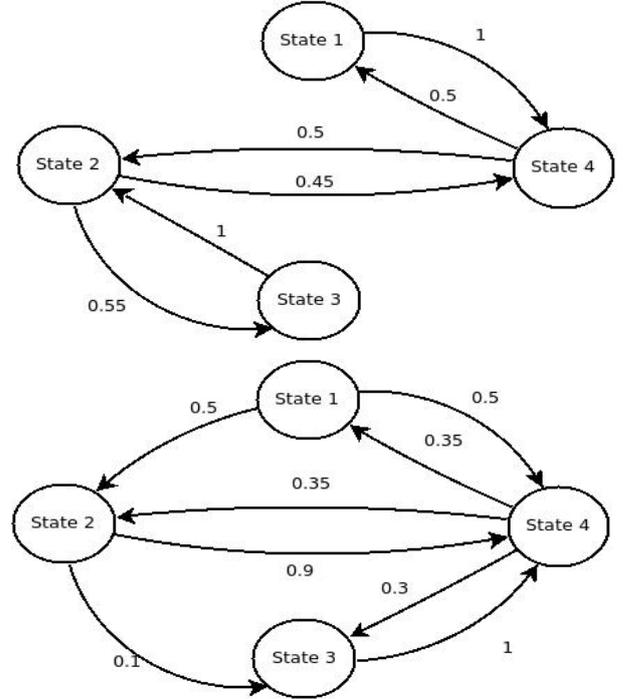


Fig. 1. Initial and final transition matrix.

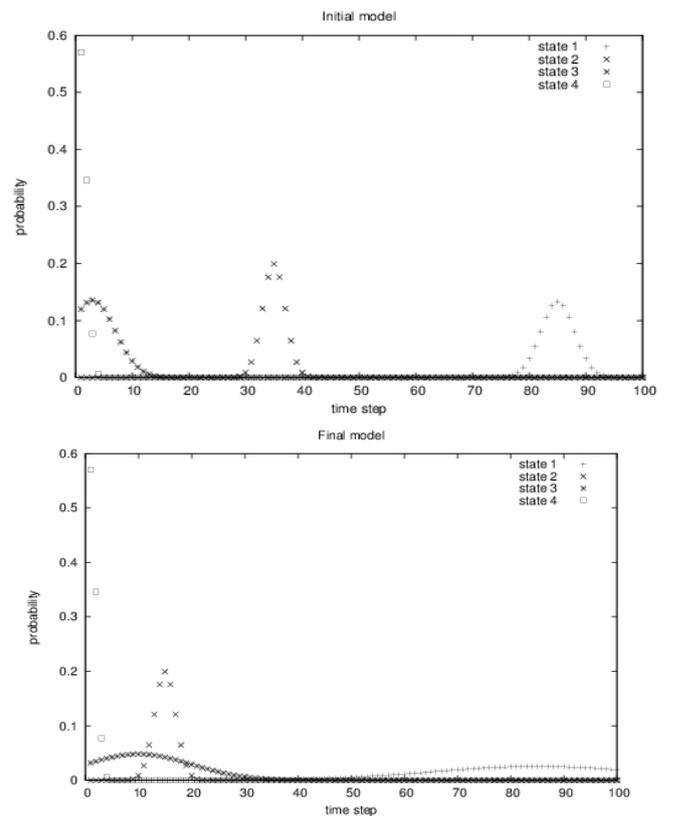


Fig. 2. Initial and final duration probabilities.

Others methods allows to compute the distance between two Markov models, especially for HMM. 4 of these methods are most described in the following part. Each method is based on a specific approach of the problem. The first method is based on a geometrical approach. The second one in generation of observations of models, the third on specific properties of Markov models and the last is statistical. Differences between these approaches are studied.

A. Kullback-Leibler and Bhattacharyya Hybridation Distance

The first distance consists to compute separately the distance between transition matrices A , which are probabilities, and the distance between the Gaussians functions of durations and observations.

To compute the distance between transition matrices, we used the KL divergence (Kullback-Leibler divergence), defined by $D_{KL}(P, Q) = \sum_i P_i \log(P_i/Q_i)$ with P and Q two discrete distributions of probabilities. The KL divergence is not symmetric, so we compute the distance like the mean of the two distances. It is possible to use others methods to compute the distance between two distributions, like divergences study [8].

The distance between two transition matrices is the sum, of distances, element by element, between the matrices. For observation probabilities and duration probabilities, defined by Gaussians functions, we use the Bhattacharyya distance. The distance of observations and durations is the sum of distances between equivalent Gaussians.

Learning methods do not respect order of states. However, the distance between two models λ_1 and λ_2 consists to compute all distances between λ_1 and all models given by inversion of order of states of λ_2 . Only the minimum distance is kept. Due to division by zero, the KL divergence between two models can be close to infinity, when distributions of these models are very close. To avoid that, we add a variable real, close to 0, at all null denominators.

B. KL Distance for Gs-GsHSMM

This distance is an application of the KL distance for HMM [9]. The distance between two models λ_1 and λ_2 is defined by $D(\lambda_1, \lambda_2) = (\log P(O|\lambda_1) - \log P(O|\lambda_2))/T$ with O a sequence of observations generated from the model λ_2 . during T time step. Differences with the original method to adapt at Gs-GsHSMM are to generate the observations sequence and to compute the log-likelihood.

To compute the distance, we used a model to generate an observations sequence. This sequence is defined in a probabilistic way. Thus, the distance between two models is not determinist and the result is different for two calculations. The length of the observations sequence is an important element to compute the distance. Its value should be close to infinity to have the exact result. So, the length should be enough large to give a correct result, but it mean also the computational time increase. The choice of length should take into account models, like the maximum duration times in a state, or the probabilities of transitions.

C. HSD

The HSD distance [10] is used to compute the distance between two Gaussian-HMM. The distance between two

models λ_1 and λ_2 is $HSD(\lambda_1, \lambda_2) = \int_x |F_1(x) - F_2(x)| dx$ with $F_i(x)$ is the stationary cumulative distribution of the model λ_i . $F_i(x)$ is defined by $F_i(x) = \int_x \sum_{j=1}^N \Pi_j b_j(x)$, with Π_j is the stationary probability of state j . For a Markov model, the stationary probabilities can be considered like the time in a state after a necessary number of iterations.

To adapt HSD at Gs-GsHSMM, two steps are needed.

In a first step, we take into account the duration probabilities. That is why the stationary probability is computes to take into account the duration probabilities. The stationary distribution is compute like defined by Barbu [11].

A second step is to take into account that each observation consists of several continuous values. To simplify calculation, authors proposed to not integrate the function, but to add up on a list of typical observations of models.

D. Statistical Distance

This distance is an adaption of the normalized statistical distance for HMM [4]. The distance between two models is defined like the joint probability that allows than models are statistically equivalent. First, joint probabilities of models are computed in order. Next, while models are not equivalent, events whose probabilities are not greater than the present joint probability are removed. The result is the joint probability when modified models are statistically equivalent.

To adapt the method at Gs-GsHSMM, several modifications are necessary.

The method needs to compute the stationary distributions of models, which are computed like for HSD, described previously.

The removed events are the transitions between states defined in the transition matrices. However, for the HSMM, self-transitions are not defined in the transition matrices. If duration times are important, the probabilities of self-transition are important too and it is not a problem. Else, it is necessary to compute self-transition probabilities to allow to remove duration times, if self-transitions are removed.

To compute the minimum number of time step needed to define if models are equivalent, we multiply the number of necessary steps defined with the transition matrices by the maximum duration time.

The last modification is to generate a sequence of states that takes into account the duration probabilities in each state. Take duration times in a state like one event reduces their significances. To be closer of models, it will be better to consider duration times like several events.

E. Comparison of Methods

We test these methods to compute distances for all the different elementary evolutions of the model. In a first step, we use the methods to compute directly the distance between the model λ_0 and the model λ_t .

If models are identical, results of methods are very different. The distances computed by the hybrid distance allow to detect all the kinds of evolutions, without noise. In the contrary, results given by the KL distance are defined in a probabilistic way. Thus, these results present a noise but it is possible to define important variations for all the kinds of

evolution. Mainly for the evolutions of the parameters Δ , those define observations. If HSD is also a probabilistic method, the noise is less important than for the KL distance. The evolutions of the model can be detected. The statistical method is the most interesting, but his computational cost is very higher than others methods. Results are also defined in a probabilistic way. The computed distances are very noisy and do not allow to detect evolutions of model.

This first set of trials allows to define several differences between the methods and their results. In this way, the main advantage of the hybrid method is to be determinist. On the contrary, the statistical method needs the most computation time and its results are very noisy. Due to this element, this method is leave for our application.

A second set of experiments consists to test influence of the learning algorithm in the distance methods. We want to identify noise due at the learning method, in the aim to verify if the distance methods can be used with learned models. In this way, several observations sequences are generated by the evolving model. These sequences are used to learn new models, and distances between these models and the first one are computed.

For the hybrid method, results are very noisy and it is not possible to detect any evolutions of the models. This method seems to be very affected by the learning method and so cannot be used with learned models.

The KL method allows a better detection of the evolutions than the hybrid distance. This method, using a stochastically sequence, seems to be less affected by the learning algorithm. If results are noisy, a global increased of distance can be detected.

Like for the hybrid method, results of HSD do not allow to detect the evolutions. The computed distances are very noisy and are not significant of evolution of models.

The KL distance is the most interesting. If the less important evolutions are not detected, this method is the less affected by the learning algorithm and the important evolutions can be detected correctly. Another advantage is it do not depend on the typology of models, i.e. the number of states or the maximum duration time. The only constraint is the number of observations should be the same. However, methods are tested with one model, and results can depend on method but depend also on model.

A limit of the KL distance is the result depends on the length of the observations sequence. If the length is not enough, results are very stochastic and not significant. On the contrary, the increased of the length implies an increase of the computational time needed. Due to these elements, a method is proposed, based on the KL distance. Aim is to decrease the computational time to find a convergent distance.

IV. PROPOSED METHOD: CONVERGENCE KL DISTANCE

The proposed method is based on the KL distance. To assure the convergence of the result, it is necessary to increase the length of the observations sequence used to compute the likelihood. Indeed, the KL distance depends on $P(O|\lambda_1)$ and $P(O|\lambda_2)$, that depend on T , the length of O . To decrease T and to assure convergence, the method consists of increase T until $|D_{KL1}(\lambda_1, \lambda_2) - D_{KL2}(\lambda_1, \lambda_2)| < \varepsilon$ true, with ε is close to 0 and the distances D_{KL1} and D_{KL2} are

computed with two different sequences of observations. This condition can be simplify according to the formula (3), with O_1 and O_2 two sequences randomly generated using the model λ_2 and t the number of time steps of the sequence O_i .

$$\left(\begin{array}{l} \log P(O_1|\lambda_1) - \log P(O_1|\lambda_2) \\ -\log P(O_2|\lambda_1) + \log P(O_2|\lambda_2) \end{array} \right) / t < \varepsilon \quad (3)$$

This condition allows that the results converge, if the likelihoods converge, like explained by formula (4). This condition is not true for all Markov models, but is demonstrated under several confines. In our application, the models are acquired by learning algorithm, so we can not assure these confines are true.

$$\lim_{t \rightarrow \infty} \log P(O_1|\lambda_i) = \lim_{t \rightarrow \infty} \log P(O_2|\lambda_i) \quad \forall i \text{ and } O_1 \neq O_2 \quad (4)$$

Thus, distance is defined by the formula (5), like the mean between the distances D_{KL1} and D_{KL2} .

$$D(\lambda_1, \lambda_2) = \left(\begin{array}{l} \log P(O_1|\lambda_1) - \log P(O_1|\lambda_2) \\ + \log P(O_2|\lambda_1) - \log P(O_2|\lambda_2) \end{array} \right) / (2t) \quad (5)$$

The method is based on computing of likelihoods of models, which can be computed in the forward part of the Baum-Welch algorithm, applied to Gs-GsHMM. A difference is this method needs to compute the likelihoods of two different models, for two different observations sequences. The length of these sequences is unknown, because one aim is to decrease it. It is not necessary to know all sequences, but just the current observation and the elements to generate future observations, i.e. the current state and the number of time steps before state changes. It is possible to compute at the same time all the likelihoods. The complexity of method is, in that case, $O\left(T(N_1(N_1 + D_1) + N_2(N_2 + D_2 + 2D_2k))\right)$, with N_i the number of states, D_i the maximum state duration, i the model, T the number of time step before convergence and k the number of Gaussians of the observations. If models have the same typology, it is possible to decrease the complexity at $O(TN(N + D))$.

If the method allows to compute the distance with the shortest required sequence, it is necessary that this sequence is enough long to avoid to have a specific case, when (3) is true but that the computed distance is wrong. For the first time steps, the computed likelihood evolves quickly, with very different results, then begins to converge. To avoid specific cases, the condition is not test during several time steps, which is dependent of the models.

The proposed method no respects the symmetrical property. If it is a drawback, it is possible to get a symmetric result through computing the mean of distances $D(\lambda_1, \lambda_2)$ and $D(\lambda_2, \lambda_1)$. Compute each distance independently of each others has a computing time cost. A solution is to compute the eight likelihoods in the same time with four distinct observations sequences.

V. RESULTS

The first trial consists to try the method on different

evolutions of the evolving model, in particular for try the good convergence of results. The Fig.(3) shows the evolution of distances between the original model and the variations of the evolving model.

For all kinds of evolution, variations are visible. For each variation, the noise is not null, and increases with the distance. However, global variations are more important than the noise, and evolutions can be detected. All the evolutions are not detected in the same way. With the trial model, the values of the most important distances for each kinds of evolution are very different. For the evolution of the transition matrix A , the most important distance is close to 1, when the distance for the observation parameter τ is 24.

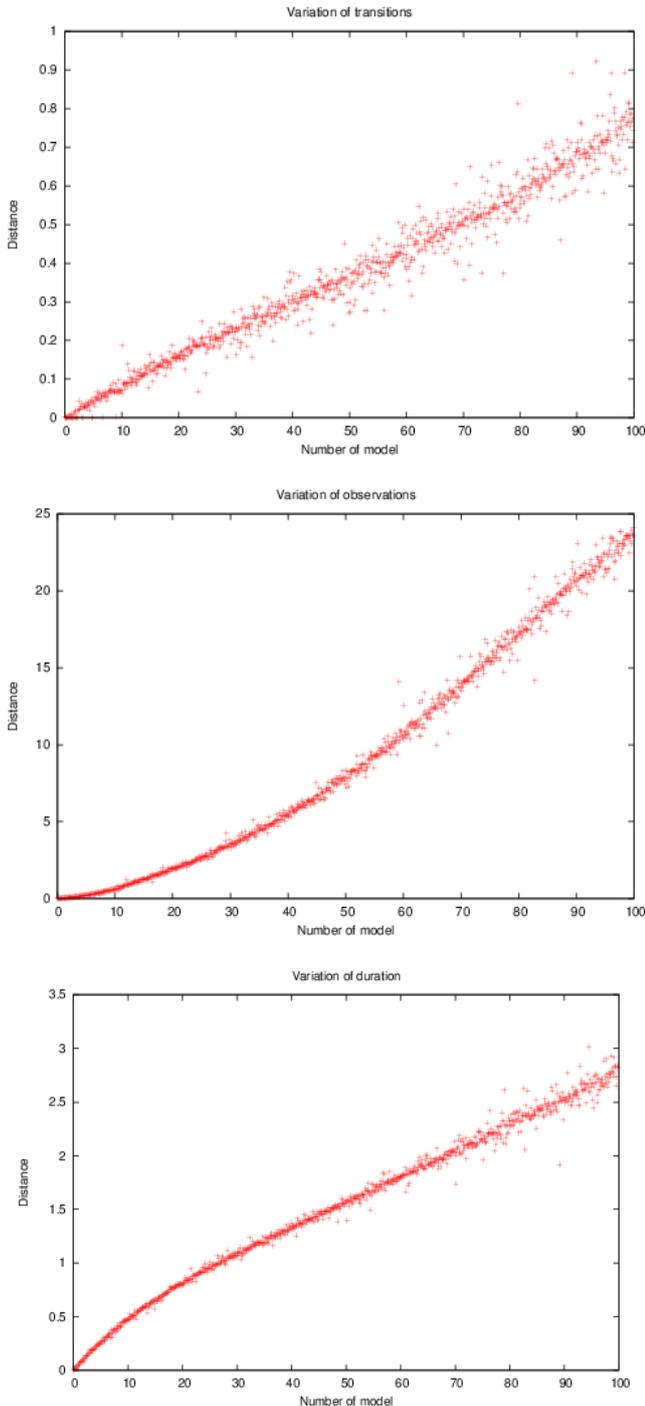


Fig. 3. Computed distance between model and variations of the model.

If evolutions between models can be detected, it is

necessary to evaluate the effect of the learning method on results. We compute distances between the original model and models learned. For the learning, we use sequences of observation generated using 100 evolution of the evolving model. We consider, to compute distance, a maximum of 10000000 time steps. The Fig. (4) shows the distances computed. The evolutions can be detected if distances are enough important to offset the noise due to the learning algorithm. The reflexivity property is not respected, this is a confirmation that the results are not significant if the correct distance is not sufficiently large.

An important aim is to reduce the number of time step to compute the distance. The Fig. (5) shows number of time step to compute distances. If the maximum value is reached one time, on the contrary, the minimum value is 1181 time steps. The average number of time step is 460901, which is correct, due at our trial model.

To confirm these results, we compute the distances of models learned with sequences of observations generated using the evolving model, with changes of one parameter at each time. The Fig.(6) shows results. In the case of the evolution of the observation parameter Δ , it can be detected. On the contrary, variations of A and τ are not detected.

The proposed method allows to compute the distance between two Gs-GsHSM, with a decreased of the noise, relative at the KL divergence, and of the computing time. This distance method can be used to detect the evolution of an unknown system, which can be learned. However, several elements should to be discussed.

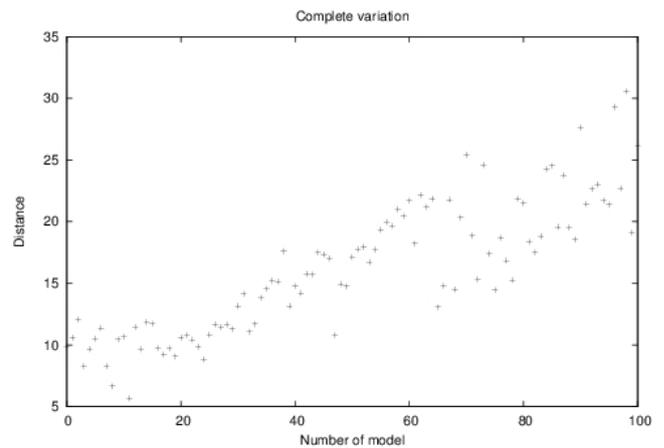


Fig. 4. Distance evolution for learned models.

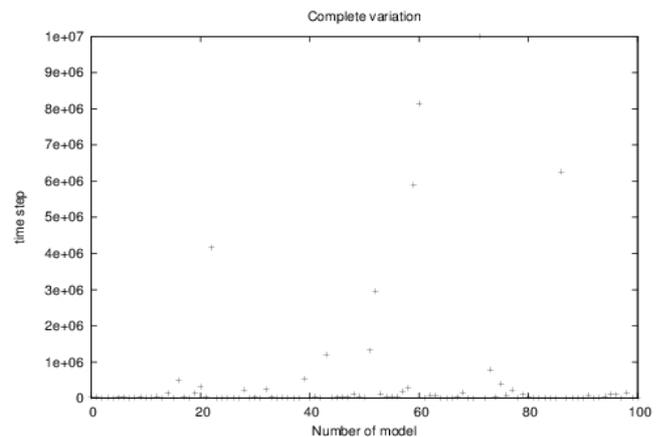


Fig. 5. Number of time steps to compute distances.

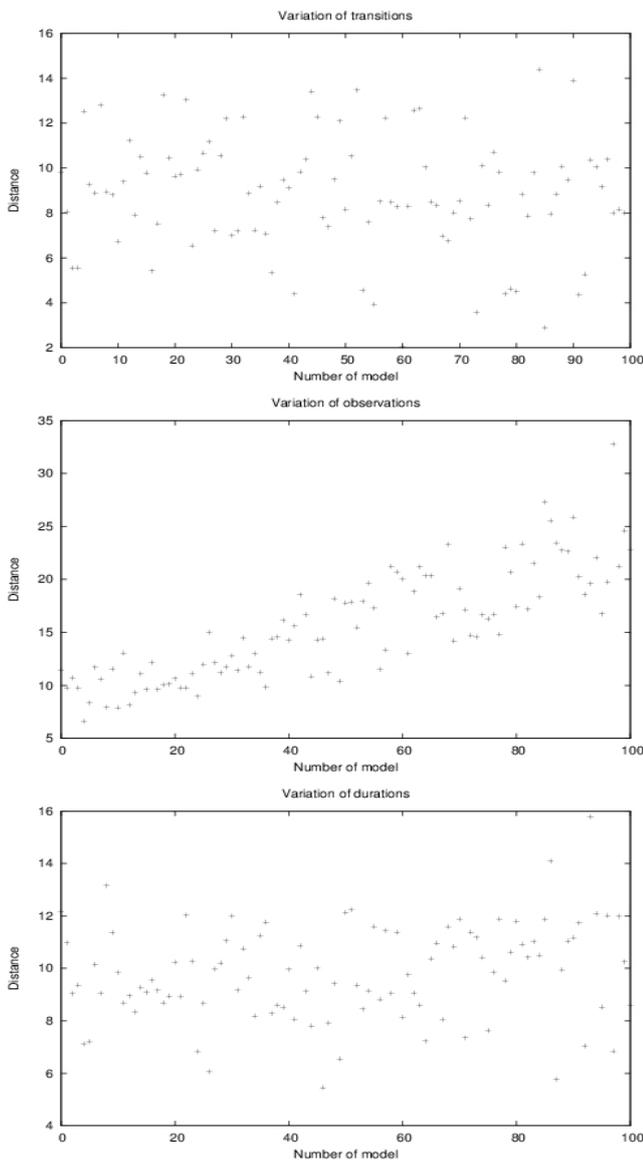


Fig. 6. Distances evolution for learned models with variation of one parameter.

All presented results depend on the used method but also depend on the trial model used. It is possible to have different results with an other model. For example, parameters A , τ and Δ do not have the same evolutions, so they can be detected differently with a different model.

The stochastically characteristic of method should be takes into account also. The experiment without learned models shows that the noise is not null. Due to this element, lowest distances cannot be detected.

An other limit is due to using the learning algorithm. If our method is most robust at the learning algorithm than the other methods, the problem is not solved. The learning method implies an important noise and results are already very variable. Due to this element, the lowest distances cannot be correctly computed. If the global variations can be detected, compute one distance is not enough to conclude.

VI. CONCLUSION

The proposed method allows to compute the distance between two systems to detect an evolution. Main advantages are to decrease the computing time and the stochastically characteristic of results. The method allows also to compute

the distance between models that have a different number of hidden states or maximum duration time. The constraint is that the number of observations is the same.

Due to the stochastic characteristic of the models, the deterministic property of the distance methods should be taking into account. If this characteristic is reduced for our method, this problem is not solved and the noise due to the learning algorithm stays important. So, the method should be update to decrease the influence of this noise.

REFERENCES

- [1] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection for discrete sequences: A survey," *IEEE Trans. on Knowledge and Data Engineering*, pp. 823-839, 2012.
- [2] M. Gupta, J. Gao, Aggarwal, C. Charu, and J. Han, "Outlier detection for temporal data," *Synthesis Lectures on Data Mining and Knowledge Discovery*, vol. 5, pp. 570-578, 2014.
- [3] N. Goernitz, M. Braun, and M. Kloft, "Hidden markov anomaly detection," in *Proc. the 32nd International Conference on Machine Learning (ICML-15)*, pp. 1833-1842, 2015
- [4] C. Lu, J. Schwiier, R. Craven, L. Yu, R. Brooks, and C. Griffin, "A normalized statistical metric space for hidden Markov models," *IEEE Trans. on Cybernetics*, vol. 43, pp. 806-819, 2013.
- [5] M. Falkhausen, H. Reininger, and D. Wolf, "Calculation of distance measures between hidden Markov models," *Int. Conf. Spoken Language Processings*, 1995.
- [6] J. Silva and S. Narayanan, "A statistical discrimination measure for hidden Markov models based on divergence," *Int. Conf. Spoken Language Processing*, 2004.
- [7] L. Xie, V. Ugrinovskii, and I. Petersen, "Probabilistic distances between finite-state finite-alphabet hidden Markov models," *IEEE Trans. on automatic contr.*, vol. 50, pp. 505-511, 2005.
- [8] F. Österreicher, "Csiszar's f-divergences-Basic properties," *Technical Report*, 2002.
- [9] L. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *IEEE Proceedings*, pp. 257-286, 1989.
- [10] J. Zeng, J. Duan, and C. Wu, "A new distance measure for hidden Markov models," *Expert Systems with Applications*, vol. 37, pp. 1550-1555, 2010.
- [11] V. Barbu, J. Bulla, and A. Maruotti, "Estimation of the stationary distribution of a semi-Markov chain," *Journal of Reliability and Statistical Studies*, vol. 5, pp. 15-26, 2012.



E-monitor'âge.

Arnaud Paris was born in Vitry-le-François, France on January 14, 1988. He received his B.S. degree in computer science from the University Paul Verlaine (Metz, France) in 2010 and his M.S. degree in computer science (RAR) from the University Nancy I (Nancy, France) in 2012.

He is currently the Ph.D. candidate at the University of Orleans in Bourges, Franc. He is, since July 2016, a research engineer for the project



Adnen El-Amraoui received his Ph.D. degree in process control from the Technical University of Belfort-Montbéliard (UTBM, France) in 2011. Actually and since September 2014, Dr. Adnen EL AMRAOUI is an associate professor at the University of Orléans (France). From Sept. 2013 to Aug. 2014 (1 year), he was a post-doctoral researcher in the French Engineering School «Ecole Centrale de Lille» (EC-Lille).

From Sept. 2009 to Aug. 2013 (4 years), he was an assistant professor at the French Engineering School (UTBM, France).

His research interests include combinatorial optimization, approximation, interval analysis, scheduling and production planning, supply chain management, transportation systems and maintenance.

Dr EL AMRAOUI is the first and the corresponding author of several publications in refereed journals (COR, CIE, EEAI, EJIE, 4-OR, JSSSE, SIC and so on) and in several IFAC and IEEE conferences.



Nathalie D. Cislo was born in Saint-Etienne, France, in 1971. She received her Ph.D. degree in computer science/robotics from the University Pierre and Marie Curie (UPMC – Paris VI, France) in 1999.

Since September 2002, she has been associate professor at the University of Orleans (France). From Sept. 2000 to Aug. 2002 (2 years) she was an assistant professor at University Joseph Fourier (Grenoble I, France). She was previously a post-doctoral researcher at INRIA (the French National Institute for computer science and applied mathematics, Grenoble, France). Her research interests include artificial intelligence, optimization and smart homes.

Dr. Nathalie D. CISLO has been an academic director in Corporate and Administrative Management (GEA) Department (University Institute of Technology, Bourges, France) since 2015, where she teaches Computer science and statistics.



Selma Arbaoui received her Ph D degree in computer science applied to social science from the University Pierre Mendes France (Grenoble II, France) in 1993.

Actually, she is an associate professor at the University of Orleans (France) in the laboratory PRISME.



Nacim Ramdani received the engineer degree from Ecole Centrale de Paris, Paris, France, in 1990, the Ph.D. degree from the University of Paris-Est Créteil, Paris, France, in 1994 and the habilitation degree in 2005.

Since September 2010, he has been a professor at the Université of Orléans (IUT de Bourges) and a member of the laboratoire PRISME. From 1996 to 2010, he was a maître de conférences with the University Paris-Est Créteil. He was affiliated with the LIRMM CNRS Montpellier during 2005–2010 and also on secondment with the INRIA during 2007–2009. His current research interests revolve around modelling, analysis, and estimation of non-linear and hybrid systems with applications to robotics, biomimetics and healthcare. He mainly focuses on interval methods and set computation techniques.

Prof. Ramdani is the Co-Chair of the workgroup on Set Computation Techniques within the French research group on Automatic Control (GDR MACS).