Change Detection of Model Transitions in Proxel Based Simulation of CHnMMs

Dávid Bodnár(✉), Claudia Krull, and Graham Horton

Institut für Simulation und Graphik,
Otto-von-Guericke-Universität Magdeburg, Magdeburg, Germany
david.bodnar@st.ovgu.de

Abstract. To analyze discrete stochastic models, Virtual Stochastic Sensors were developed at the Otto-von-Guericke-University Magdeburg. This procedure makes it possible to reconstruct the behavior of a broader class of hidden models, like Conversive Hidden non-Markovian Models, in a very efficient way. One assumption of this approach is that the distribution functions, which describe the state changes of the system, are time-homogeneous. However, this assumption is not always true when it comes to real world problems.

To overcome this limitation, the paper presents an algorithm where the concept of Virtual Stochastic Sensors was extended with statistical tests to continuously evaluate the parameters of a Conversive Hidden non-Markovian Model and the current results. If needed, the tests stop the execution of the behavior reconstruction and reevaluate the model based on the current knowledge about the system.

The project showed that detecting the change and adjusting the model is possible during the behavior reconstruction, improving reconstruction accuracy. The method was tested using four types of distribution functions, three of which showed very good results. By using this new algorithm, one is able to construct adaptive models for behavior reconstruction without additional conceptual effort. In this way, loss of modeling accuracy due to abstractions in the modeling process can be balanced. Another possible application appears in the case of long time investigations. The change detection method can be invoked after a given period to reevaluate the system model and make the relevant adjustments if needed.

Keywords: Virtual Stochastic Sensor · Conversive Hidden non-Markovian Model · Drift · Change detection

1 Introduction

Virtual Stochastic Sensor (VSSs) were introduced in [4] and are able to reconstruct the behavior of partially observable processes in discrete stochastic systems. In previous research involving VSSs, the model was always based on previous knowledge about the system, which was considered to be relatively accurate.

© Springer International Publishing AG 2017
N. Thomas and M. Forshaw (Eds.): ASMTA 2017, LNCS 10378, pp. 32–46, 2017.
DOI: 10.1007/978-3-319-61428-1_3

However, in the case of real world applications or long time analysis of bigger systems, not all details can be taken into account. Another source of error is, that processes which require human interaction or intervention cannot be considered to be time-homogeneous. This can lead to serious inaccuracy over time, especially in large systems.

To overcome this limitation, the proxel-based solution algorithm of the Conversive Hidden non-Markovian Models (CHnMMs) was combined with two stochastic tests, the Kolmogorov-Smirnov test (KS test) [6] and the Wald-Wolfowitz runs test [12]. These tests are used to monitor the current state of the reconstructed behavior and evaluate the results. The goal is to detect changes in the system configuration compared to the model, as well as deviations during the execution of the reconstruction. The changes which we will try to detect in this research are parameter shifts in the distribution functions in the hidden part of the system. The type of the distributions is considered to be not changeable. Neither is the actual system state space. This paper is a feasibility analysis based on the master's thesis [2] of one of the authors. The goal of the analysis is to find out whether indication and elimination of such non-obvious changes during the behavior reconstruction are possible or not. It is expected that by using such an algorithm one is able to construct adaptive models. With this, a more realistic connection between the model and the real world can be created and maintained over the lifetime of the model, even with evolving system conditions, without requiring additional effort during the model parametrization.

2 Related and Previous Work

As we live in the world of the Internet of Things (IoT), there is an increasing demand for data to be acquired by sensors. However, there are situations when the required information cannot be measured directly. In these cases, virtual sensors [13] and sensor fusion are used to try to get accurate data on the given system. These sensors use mathematical rules to construct their output. A virtual sensor combined with stochastic models results in a VSS [4] which utilizes stochastic knowledge about a given system. One mathematical model that can be used to model the stochastic relationship between the input data and the results of the VSS is a CHnMM. The proxel-based analysis method can then be used to conduct behavior reconstruction for the VSS in order to acquire information in the form of a statistically relevant estimate of non-measurable system parameters.

In this section, a brief overview will be given of the previous work on VSSs and the needed statistical tests.

2.1 Proxels-Based Analysis

The proxel method [10] was introduced in [7] and is able to construct and analyze the state space of discrete stochastic models represented by Stochastic Petri Net (SPNs) in an efficient and controlled way. A proxel (probability element) is the smallest addressable unit of the reachable state space of the stochastic system at a given moment in time. A proxel is a container which contains all relevant information about a possible state:

$$P_{\mathrm{x}} = (m, \boldsymbol{\tau}, p, t) \tag{1}$$

The stored information are the marking of the current discrete system state (m), the transition age variables ($\boldsymbol{\tau}$), the probability of this discrete state and age combination (p), the current time of the analysis (t) and any additional information which might be relevant for the application like the generating path, utilization of a defined state, etc.

As the analysis runs, the proxel algorithm tracks the possible states and the necessary information from one time step to another. The computation is done in discrete time steps. This means that the time of analysis is discretized with a given granularity and the system is considered to be time homogenous between two time steps. The probability of a possible state change is defined by the Hazard Rate Function (HRF) in Eq. (2), where τ is the age of the current state change. The HRF can be understood as the current rate of the given state change if it did not happen yet.

$$\mathrm{HRF}(\tau) = \frac{\mathrm{PDF}(\tau)}{1 - \mathrm{CDF}(\tau)} \tag{2}$$

During the behavior reconstruction, a proxel holds a possible actual system state with a possible history in the given time step. In the next time step, the possible states are computed from the current state by computing the HRF of the active transitions. Impossible or very unlikely states are pruned away from the proxel tree to dampen state space explosion and maintain acceptable computation time. Using this iterative method, the reachable state space of an SPN is generated in discrete time steps, which can then be analyzed to obtain information such as system performance parameters.

2.2 Conversive Hidden non-Markovian Model

The Hidden Markov Models (HMMs) are well-known statistical models, which assume independent state changes and can be observed through symbol emissions which occur with a given probability. Krull [8] created the so-called Hidden non-Markovian Models (HnMMs), where one can use arbitrary continuous distribution functions and concurrent activities to define connection and dependence between events.

The concept of HnMM in combination with the proxel method can be used for solving the evaluation and decoding tasks for more general models than HMMs. In the case of evaluation one tries to compute the probability that a given observation sequence (trace) was generated by the model. While in the case of decoding one tries to find the most likely state sequence (path) which generated a given observation sequence (trace). The proxel algorithm needs to be modified to not just generate the reachable model state space, but to compare this state space to the observed trace. Only the proxels, which could have produced the trace are retained and analyzed further. The thus modified proxel algorithm results in all possible system paths, in discrete steps, which could have generated the observed trace. Therefore, HnMM combined with a proxel-based solution algorithm can be used for behavior reconstruction of partially observable discrete stochastic systems [8]. This can be used for example for gesture recognition [5].

To increase the efficiency of VSSs, the concept of CHnMMs was introduced and analyzed by Buchholz in [3]. The CHnMM is a subclass of the HnMM, where every state change of the hidden part of the system results in a symbol emission, making state changes easier to track. By using this subclass of the HnMMs, one is able to save a lot of computation time and perform more accurate behavior reconstructions by utilizing a continuous time proxel-based solver algorithm instead of a discrete one. During the feasibility analysis, the project was restricted to CHnMMs to be able to easily analyze and compare the results.

2.3 Statistical Hypothesis Testing

Statistical hypothesis testing [1,6,12] is a group of mathematical methods to observe the significance of a statistically relevant statement, commonly referred as the null hypothesis, by using sample sets and assuming a given certainty that the result of the test is right. As it is a well-known and commonly used tool it will not be discussed in details in this paper. The following two statistical tests were used in the current research.

Kolmogorov-Smirnov Test. The KS test [6] is a well-known non-parametric test, which is used for comparing probability distributions. In our case, the test computes the maximal deviation between an Empirical Distribution Function (EDF) and a theoretical Cumulative Distribution Function(CDF) on a regular grid. If this maximal deviation between them exceeds a given threshold, then the test fails and one is able to say that the given sample was not drawn from the given theoretical distribution function. This test works more accurately if there is a difference [11] in the mean between the given distribution functions. However, in the case of deviation in the variance the test can commit a type II error in rare cases.

Wald-Wolfowitz Runs Test. The Wald-Wolfowitz runs test [12] is another well-known non-parametric test which can be used for validating the randomness of a binary sequence. As it is defined in [12] one can convert a continuous random sequence into a binary sequence by using the median value of the continuous variable and defining the values above and below the median as the same state. In this test, one counts the so-called runs in the binary sequence, which is the number of subsequences where the system stays in the same state. If the number of runs differs significantly from the expected number, then the test fails, indicating that the sequence is not random. This can be used for indicating changes in the variance of the dataset effectively [11].

3 Implementation

In a HnMM the states are connected through transitions, which describe the possible state changes in the system. The firing times of these transitions are characterized with given distribution functions. As already mentioned, the change detection algorithm focuses on the parameters of these distribution functions during the behavior reconstruction.

The basic idea behind the change detection algorithm is a sliding window approach where the already described statistical tests evaluate a given amount of last firing times for every single transition in the model. If a change is detected on a transition, then the execution of the behavior reconstruction is stopped and the new parameters are estimated. After the estimation, the algorithm resets the time to the last valid state of the results and continues the computation.

3.1 Restrictions to the Transitions Behavior

Before diving into the details of the different experiments and the results, the limitations of the project need to be discussed. As already mentioned, the project was limited to the investigation of CHnMMs because of the clearer and easier analysis. The reason is that CHnMMs produce a symbol emission with every state change.

The drift recognition was limited to a change in the parameters of the distribution function because specific machines, processes, and occurrences tend to have a very specific type of distribution. In this way, the assumption was introduced that the expert, constructing the model, is able to guess the type of the distribution accurately. However, during the model construction, one tends to have only a limited amount of information on the system. This knowledge also tends to be stationary, not taking any kind of change into account. Let us assume that one gets information about a production line where extreme precision and high qualification is needed. During the model construction time, a qualified key worker gets sick and is temporarily replaced by a less qualified one. In many cases, the data analyst would not notice this change, but the constructed model and thus the behavior reconstruction might already get inaccurate. Considering this, the drift correction was limited to small, non-obvious changes in the system. Dramatic changes should be easily noticed in most cases during runtime, such as a machine in the production line stopping for a longer period of time.

The goal of the project is therefore to fine tune the system during execution time and prevent a constant need of manual adjustment of the system distribution parameters.

3.2 Change Detection Algorithm

For easier understanding, the change detection algorithm was visualized on the flowchart shown in the Fig. 1.

To be able to use statistical tests for the validation of the model, firstly, one should acquire some data about the system. To do this, a sliding window

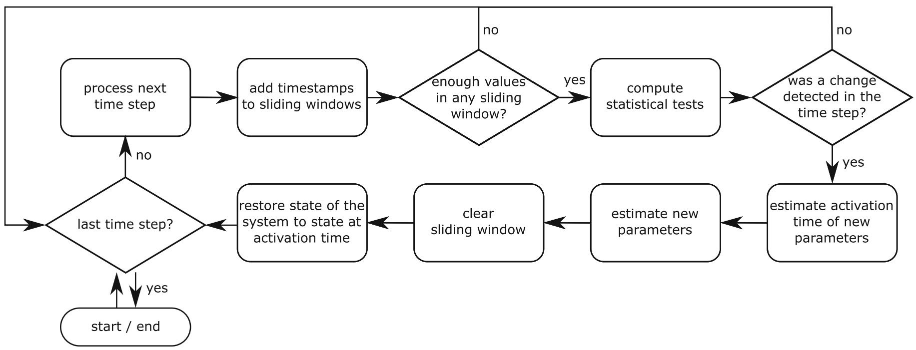

Fig. 1. Flowchart of the change detection algorithm

approach is used. An independent window is defined for all system transitions (state changes) in the model. In these windows, the last n firing times are stored. When one of the windows is filled, the change detection algorithm is activated for the given transition, and the statistical tests determine for the given transition whether a change was detected or not.

The size of this sliding window has a huge impact on the algorithm. If the size is too small, then there might not be enough information present for the statistical tests to perform efficiently. While if the size is too large, then one might get problems with the required memory. Another problem might occur in the case of larger window sizes because the algorithm does not start as long as the sliding window is not filled. This can cause more computations to be performed with the wrong parameters until the algorithm finds the next change.

After the statistical tests are activated, for all active transitions and for every single proxel, the statistical tests are computed. If one of them fails, then a change is indicated for the given transition on the proxel in that timestep. Of course, if every change detected on a single proxel would result in a new estimation of the model, then the computation would never end. That is why a threshold is introduced, which can be seen in the Eq. (3). In the equation, $n_{\mathrm{c,k}}$ is the number of proxels where a change was detected in the timestep and n_{k} is the number of proxels in the timestep. This threshold determines that a change is only recognized as change if the sum of the probabilities of all proxels where a change is detected ($\sum p_{\mathrm{c,i}}$) exceeds half of the probabilities of all proxels ($\sum p_{\mathrm{l}}$) in the time step (t_{j}) normalized with the confidence level $(1-\alpha)$.

$$\frac{\sum_{\mathrm{i}=0}^{n_{\mathrm{c,k}}} p_{\mathrm{c,i}}(t=t_{\mathrm{j}})}{\sum_{\mathrm{l}=0}^{n_{\mathrm{k}}} p_{\mathrm{l}}(t=t_{\mathrm{j}})} \geq \frac{0.5}{1-\alpha} \tag{3}$$

If a change is detected, then the algorithm makes a parameter estimation for every single proxel. The estimated parameters (ρ_{i}) are weighted with the probability of the given proxel (p_i) and normalized. This can be seen in the Eq. (4). Of course, there are also proxels where no change is detected. For these proxels, the original distribution parameters are weighted with the proxel probability.

The algorithm also needs to compute the activation time of the new parameters. The activation time is the time when the detected shift in the distribution parameters likely took place. The computation is similar to the estimation of the new distribution parameters. The unique activation time (δ_i) is assigned to every proxel, based on the oldest element in the sliding window to the given transition. These activation times are weighted with the proxel probability and normalized as it can be seen in the Eq. (5).

$$\rho = \frac{1}{\sum_{\mathrm{l}=0}^{n_\mathrm{k}} p_\mathrm{l}} \sum_{\mathrm{i}=0}^{n_\mathrm{k}} \rho_\mathrm{i} p_\mathrm{i} \tag{4}$$

$$\delta = \frac{1}{\sum_{\mathrm{l}=0}^{n_\mathrm{k}} p_\mathrm{l}} \sum_{\mathrm{i}=0}^{n_\mathrm{k}} \delta_\mathrm{i} p_\mathrm{i} \tag{5}$$

After the new parameters and the activation time is computed, the algorithm assigns the change to a global array which tracks the activation of these. After that, a reset signal is sent to stop the execution of the behavior reconstruction, go back in time before the activation of the new parameters and recompute that part of the past. To speed up the computation, a ring buffer is implemented which stores the model states periodically. After a reset, only the last valid state needs to be restored and the computation can continue.

3.3 Implementation Challenges

During the implementation, two major challenges were discovered. Distribution functions that only have finite support and a type of aliasing effect, which occurred because of the sliding window approach, causing virtual distributions during the execution. In the following, these will be discussed briefly.

Distribution Functions with Finite Support. Distribution functions with finite support, like a uniform distribution, might cause problems for the change detection algorithm. In these cases the HRF drops to zero if a transition fires outside the strictly defined boundaries of the distribution function. In these cases, the proxel tree would die out and there is no way to continue the analysis.

To overcome this limitation, these distribution functions are altered in that way that a small amount of probability is redistributed from them into newly constructed tails which come from an equivalent normal distribution. The probability of the equivalent normal distribution without the tails is the same as the original finite support distribution without the redistributed probability.

In this way, the HRF will produce small non-zero probabilities outside the support of the original distribution function. Of course, this results in a larger number of proxels with very small probability during the behavior reconstruction, but they are pruned away in the next time step because the probability difference between a firing inside and outside the borders of these newly defined distributions is pronounced.

Virtual Distribution Functions. Virtual distributions occur because the sliding window contains data from two different distribution functions at the same time. Assume that at the beginning of the behavior reconstruction the model is accurate and the sliding windows are filled with samples of the given distribution functions. Then a sudden change occurs. This does not immediately result in a detected change, but when it does, the window still includes samples from the old distribution but has already some from the new one.

As the type of the distribution is assumed to be the same, the parameter estimation would result in a distribution somewhere between the old and the new one with a bit more variance. This is a virtual distribution because it only occurs due to the transient phase. The occurrence of these virtual distributions would not be a problem, if they would not cause an endless loop to happen. The parameters of the virtual distribution cannot fulfill the criteria of the statistical tests. Therefore, the algorithm is not able to leave this state because it always tries to reestimate the same parameters without moving forward.

To overcome this limitation, the algorithm is forced to move forward. After the sliding windows are filled with the required amount of samples, the change detection algorithm does not get activated for a short time. This can be defined as a given amount of time or the number of new symbols which needs to be processed before.

4 Experiments and Results

To make the analysis easy to handle and evaluate, a test scenario was needed. The quality tester example (Figs. 2 and 3) was introduced by Buchholz in [3] and was used during the research in the field of CHnMMs and VSSs. In this example, two independent production lines are merged before a quality tester. The quality tester should indicate when the error rate increases on one of the production lines. It was shown in [3] that this can be done. In our case, we assume that source 0 on the production line is altered in the middle of the analysis (upgrade, maintenance, etc.) and the question is, whether the behavior reconstruction is capable of updating the model accurately and reducing the misclassification error at the same time. The misclassification error denotes the portion of produced items in a trace whose source is incorrectly reconstructed in the path.

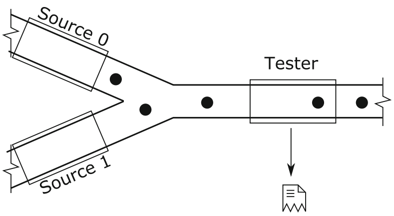

Fig. 2. Visualization of quality tester [3]

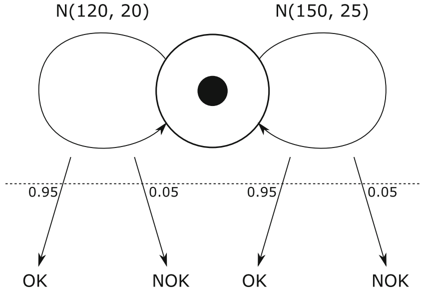

Fig. 3. Model of the quality tester

4.1 Experiment Setup

In the experiments, two parameters needed to be analyzed because of their significant impact on the results. These were the size of the sliding window and the pruning threshold of the proxel-based solution method.

In the case of the size of the sliding window, as it can be seen in the Fig. 4, a relatively broad range was analyzed between 100 and 250. With small sizes, the algorithm was not robust enough to withstand noise and the transient phase produced a type of numerical oscillation. However, the computation took only 1–10 min. With large sizes, the computation took significantly longer. At the size of 250, it even reached 6 h. In addition, the test set was found to be too short sometimes. A relatively good trade-off between noise and speed can be found around 200–210, where the computation takes about 50–65 min. In our test cases, the amount of 200 timestamps was chosen as window size.

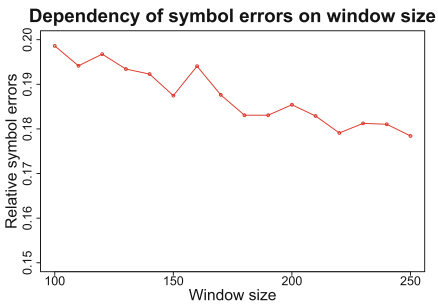

Fig. 4. Relative misclassification error for different window sizes

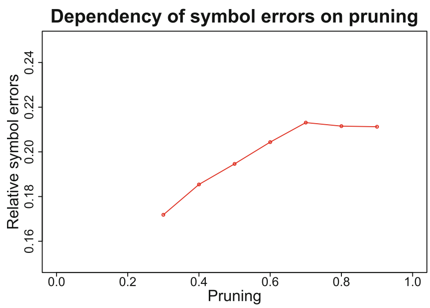

Fig. 5. Relative misclassification error for different pruning thresholds

In our case, we use a relative threshold, meaning that for every time step, all proxel probabilities are compared to the most probable one. If the probability is below a given fraction of this maximum probability, then the proxel is pruned away. This parameter has a huge impact on the computation speed, the needed storage and the amount of significant information gained from the behavior reconstruction. The pruning threshold, as it can be seen in the Fig. 5, was tested between 30% and 90%. The computational times varied between a day and 30 s during these experiments. At 40% the algorithm was executed with an acceptable computation time of about an hour without losing a significant amount of information so during the tests the pruning threshold was held at that level, if possible.

4.2 Results

The experiments were done by using multiple types of distributions with a wide parameter set: Normal, uniform, gamma, and lognormal. Because of the restrictions on the length of this paper, only the results of the test with the normal distributions will be introduced in more detail. For more detailed results on the other distributions, please refer to [2].

The quality criteria for these tests were the following:

– Reconstruction accuracy in terms of misclassification error.
– Accuracy of the distribution parameters for the transition with change.
– Accuracy of the distribution parameters for the transition without change.

In the following heat maps, light green means that there is no error or no change compared to the original value, while red indicates an increase in error/-positive parameter deviation and dark blue a decrease in error/negative parameter deviation.

Normal Distribution. The normal distribution is a well-known continuous distribution function often used for representing randomness in examples. The parameters are easy to compute and to estimate, the results are exemplary so that is why it was chosen to represent the results in this paper.

The relative misclassification error needed to be analyzed in two different ways. The results of the first experiment can be seen in the Fig. 6. In this figure, one sees the relative change in the overall misclassified symbols with different distribution parameter changes compared to the case without change detection, when the change happens in the middle of the time protocol. As one can see, there is no significant improvement. In addition to that, in the case of the most dramatic mean change during the analysis, one can notice a decrease in accuracy of about 2–10%. Here the problem lies in the experiment setup. The behavior reconstruction did not spend enough time in the stationary phase at the end of the execution. In this way, there was not enough time to neutralize the effect of the transient phase on the misclassification error.

To have a better overview of this phenomenon, a second experiment was performed. This time, another sequence was generated with the new parameters of the stationary phase and two standard proxel-based solver algorithms were run without the change detection algorithm. In these two runs, the sequence was analyzed with the original and the new model of the HnMM. The difference in the misclassified symbols can be seen in the Fig. 7. The second experiment shows that the algorithm is able to reduce the misclassification error significantly. The

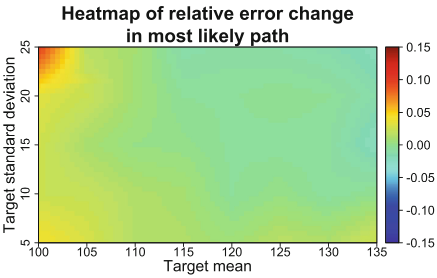

Fig. 6. Relative misclassification error (Color figure online)

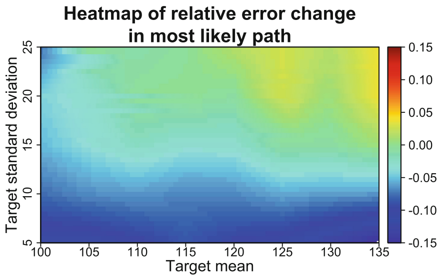

Fig. 7. Relative misclassification error if change happens at the start of the behavior reconstruction (Color figure online)

gain in cases with small variance already reaches 10–15%. One can observe an about 2–3% decrease in accuracy in some cases in the upper right corner. In these cases, the two normal distribution functions nearly merge, so the results are nearer to a random guess than to a significant result.

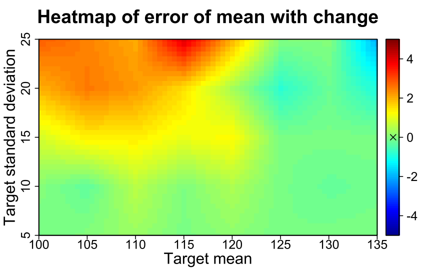

Fig. 8. Error in the mean of the transition with change (Color figure online)

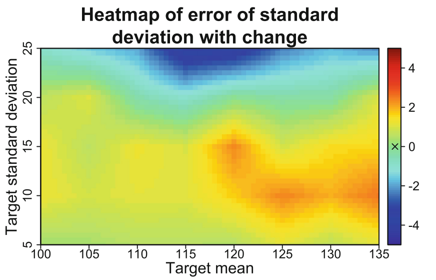

Fig. 9. Error in the standard deviation of the transition with change (Color figure online)

The Figs. 8 and 9 show the error in the distribution parameters at the end of the change detection algorithm. One can notice an error in the upper left corner of the mean diagram. This was caused by the experiment setup. The larger drift in mean with a higher variance had a longer transient phase and some of the computations did not reach the end distribution until the end of the experiment. The same effect can be noticed in the case of the standard deviation in the upper part of the diagram. The red area in the middle of the diagram is an effect of the two distribution functions merging because of the change. Despite these smaller deviations, the results can be considered to be accurate if one takes the relatively short reconstruction time and the drift of the parameters into account. In 75% of the cases, the algorithm performed very accurately in reaching the mean parameter and slight inaccuracies occur only with higher standard deviations. The same accuracy is achieved for the standard deviation in about 50% of the cases.

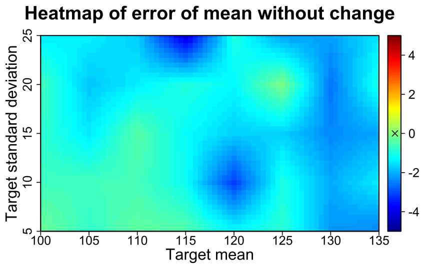

Fig. 10. Error in the mean of the transition without change (Color figure online)

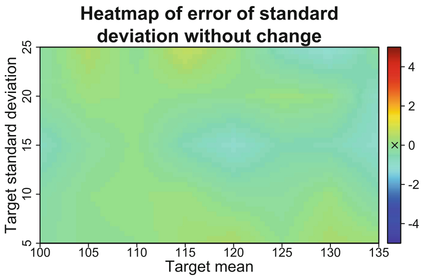

Fig. 11. Error in the standard deviation of the transition without change (Color figure online)

In the Figs. 10 and 11 one can observe the robustness of the algorithm. The noise created by the transient phase indicated some very small change in the transition without change but these changes tend to be compensated in the stationary phase at the end of the behavior reconstruction. The results of the normal distribution show that there is a significant improvement in the misclassification error of the HnMM. The distribution parameters were reached accurately and the transitions are robust enough to withstand the noise of the transient phase accurately.

Lognormal Distribution. The lognormal distribution was chosen as a possible experiment because the parameters are still easy to estimate but the distribution function is not symmetric and some shape change is possible during the change detection. The results were very similar to the ones with the normal distribution.

Uniform Distribution. The uniform distribution was chosen to be an experiment because the reparametrization of distribution functions with finite support needed to be tested. A part of the probability in the Probability Density Function (PDF) was redistributed into the tails as already described. Of course, the HRF computation of the proxel-based solver algorithm was also modified in the same way. The results with the newly constructed distribution exceed the results of the normal distribution when it comes to the accuracy of the parameters. However, it performed slightly worse in the case of the misclassification error.

Gamma Distribution. The gamma distribution is a continuous distribution function with a pronounced shape change depending on the parameters. The gamma distribution also includes the exponential and the chi-squared distribution functions. In this case, the results became very bad. The change detection algorithm almost never reached the required parameters. This was caused by multiple problems. First of all, a more dramatic change in the parameters resulted in a drastic shift of the mean and the variance, which resulted in a longer generated input sequence as the mean got lower. There were much more changes detected, so the computation became very long. This means that some of the experiments on the sample grid ran for more than a day.

Unfortunately, in many cases, the change detection algorithm estimated the parameters for one of the transitions very inaccurately. Sometimes the mean of the estimated parameter set was so high that it made impossible for the transition to fill the sliding window before the end of that part of the experiment. In other cases, the distributions just started to switch places and could not reach the steady state until the end of the analysis. The gamma distribution can be considered not to be solvable by the change detection algorithm in the current stage.

5 Conclusion

5.1 Summary of the Project

The experiments show that the constructed change detection algorithm is able to track a change in the distribution parameters accurately. During the analysis, a gain of about 10–15% in the accuracy of the behavior reconstruction was achieved.

The biggest disadvantage of the algorithm is that it consumes a lot of resources and in some cases takes very long to compute. The memory, required for the computation, jumps from about 50–200 MBs for normal behavior reconstruction to a value of about 2–20 GBs depending on probability difference of the concurrent possible paths. The reason for this huge difference is the implemented ring buffer. Of course, reading and writing chained lists into a ring buffer consumes a lot of time. This is the main reason, that the computation time, required for the algorithm, jumped from a couple of minutes to a value between 10 min and 6 h. Like the required memory, the execution time is very strongly depending on the probability difference of the concurrent possible paths. To overcome this limitation, the change detection algorithm is suggested to be used periodically by very long term behavior reconstructions combined with a general proxel-based solver. By a periodic call of the change detection algorithm, the current model is evaluated and if there is no deviation detected, then the change detection is disabled until the next call to save resources.

Given the results in this paper, a possible change detection algorithm is presented for CHnMMs, despite the inaccurate results in the case of the gamma distribution. By using a more accurate parameter estimator, the algorithm could be a general solution for evaluating the system model during a proxel-based solver algorithm. The new algorithm makes it possible to automatically correct slight inaccuracies in the distribution parameters implemented in the system model unintentionally. Additionally, the algorithm is able to handle model simplifications in a more accurate way by adjusting the system parameters during the behavior reconstruction. In this way, one is able to construct more realistic behavior reconstruction models by using the introduced change detection algorithm.

5.2 Future Research Possibilities

There are multiple possibilities for further development of the algorithm. First of all, a more general and more accurate estimation algorithm might save additional computation time and improve the results of the change detection algorithm. In this feasibility test the project is restricted to CHnMMs. A feasibility test for general HnMM is still needed, which can be built on the basis of the current project. Last but not least, a more efficient ring buffer algorithm might solve the resource shortage. The algorithm should be able to identify key moments during the reconstruction time and save only the states in the ring buffer which are needed in the future with a higher probability. Not just memory consumption can be reduced significantly, but also the computational power of saving and restoring the proxel tree, which would lead to a more acceptable execution time.

5.3 Potential Real World Applications

The algorithm can be used for evaluating the system model in all CHnMMs behavior reconstruction analyses which are long enough to be suitable for the change detection. Of course, this might make the execution time significantly longer, so that the already suggested periodic call might be an option to consider for these cases. The algorithm can also be used for observing the parameter changes of the hidden part of the system model over the execution time. In this way, one should be able to understand the system in a better way and there is a possibility to identify reasons and causes more accurately. The change detection algorithm might be a very good tool to use during behavior reconstruction of production lines and factories like the one in [9]. These processes tend to involve a large amount of human interaction, and this means that changes might occur more often than in other processes. As a result, this change detection algorithm can add a significant gain compared to a standard proxel-based solution algorithm and might help in combination with other information to better understand the processes.

References

1. Bamberg, G., Baur, F., Krapp, M.: Statistik. Oldenbourg Verlag München, München (2011)
2. Bodnár, D.: Change Detection of Model Transitions in Proxel Based Simulation of CHnMMs. Otto-von-Guericke-Universität, Magdeburg (2016)
3. Buchholz, R.: Conversive Hidden non-Markovian Models. Otto-von-Guericke-Universität, Magdeburg (2012)
4. Buchholz, R., Krull, C., Horton, G.: Virtual stochastic sensors: how to gain insight into partially observable discrete stochastic systems. In: The 30th IASTED International Conference on Modelling (2011)
5. Dittmar, T., Krull, C., Horton, G.: A new approach for touch gesture recognition: Conversive Hidden non-Markovian Models. J. Comput. Sci. **10**, 66–76 (2015)
6. Hartung, J., Elpelt, B., Klsener, K.H.: Statistik. Oldenbourg Verlag München, München (2009)

7. Horton, G.: A new paradigm for the numerical simulation of stochastic petri nets with general firing times. In: Proceedings of the European Simulation Symposium, ESS 2002 (2002)
8. Krull, C., Horton, G.: Hidden non-Markovian Models: formalization and solution approaches. In: 6th Vienna International Conference on Mathematical Modelling (2009)
9. Krull, C., Horton, G., Denkena, B., Dengler, B.: Virtual stochastic sensors for reconstructing job shop production workflows. In: 2013 8th EUROSIM Congress on Modelling and Simulation (EUROSIM), pp. 276–281 (2013)
10. Lazarova-Molnar, S.: The Proxel-Based Method: Formalisation, Analysis and Applications. Otto-von-Guericke-Universität, Magdeburg (2005)
11. Magel, R.C., Wibowo, S.H.: Comparing the powers of the Wald-Wolfowitz and Kolmogorov-Smirnov tests. Biom. J. **39**(6), 665–675 (1997)
12. NIST: NIST/SEMATECH e-Handbook of Statistical Methods (2013). http://www.itl.nist.gov/div898/handbook/index.htm
13. Wilson, E.: Virtual sensor technology for process optimization (1997). Presentation at the ISSCAC 1997