MATCHING HIDDEN NON-MARKOVIAN MODELS: DIAGNOSING ILLNESSES BASED ON RECORDED SYMPTOMS

Claudia Krull and Robert Buchholz and Graham Horton
Otto-von-Guericke University
P.O. Box 4120, 39016 Magdeburg, Germany
E-mail: {claudia.krull|robert.buchholz|graham.horton}@ovgu.de

Abstract

Discrete stochastic models (DSM) can be used to accurately describe many natural and technical processes. The simulation algorithms usually require the system parts of interest to be completely observable in order to analyze the model. Hidden non-Markovian models (HnMM) have been applied successfully to the analysis of partially observable systems. They can determine the unobserved most likely system behavior that caused an observed output. The analysis can be done by the state space-based Proxel algorithm, which on-the-fly generates the reachable model state space at discrete points in time. In the current paper, we compute the unconditional probability of a given model having produced a given output. This can be used to find the most likely one of different possible system configurations to produce the given output. In our application we want to find the illness that most likely caused the recorded symptoms of a patient. Experiments are performed to determine the accuracy and limitations of the applicability of the approach. This paper increases the application area of HnMM analysis twofold. We can now perform model matching tasks for HnMM, and we have tested an application example from medical diagnosis.

INTRODUCTION

Discrete stochastic models (DSM) are widely used in industry today and can represent many manufacturing, natural, technical and other processes. Usually simulations are performed with fully parameterized models, which require a known and fully observable system. However, some real systems are not fully observable, only through their interaction with the environment. The internal state of the machine or process is not directly detectable, but the system generates observable output depending on the internal state. These models can be characterized as partially observable systems (Buchholz et al. 2010). Hidden Markov models (HMM) can model and analyze hidden systems, but they are restricted to discrete-time Markov chains (DTMCs) and thus to memoryless processes. This only allows for a very rough approximation of the runtime behavior of many real processes. The recently developed hidden non-Markovian models strive to relieve this problem by enabling the formal description of partially observable discrete stochastic systems with time-dependent processes (Krull and Horton 2009).

We have proposed using the Proxel algorithm (Horton 2002, Lazarova-Molnar 2005) for analyzing HnMM, which is based on the method of supplementary variables. The Proxel algorithm explores all possible system developments in given discrete time steps and quantifies them with their probability. Recent research (Buchholz et al. 2010) has shown that by using Proxels one can compute the most likely system behavior that produced a given output, when the specification of the HnMM is known. But what if several model configurations are possible? An example is a medical diagnosis system, which is described in detail later in this paper. The symptoms of a patient can be regarded as the output of a hidden model of the illness progression. However, it is not known which illness caused the symptoms, and thus the task is to find the most likely illness to have caused the observed symptoms. The question posed is now: Given a patient's symptoms and their time of occurrence or detection, what is the most likely illness he or she is suffering from? This can then help the physician to determine a promising treatment to apply or a medication to administer to the patient.

The abstract task is to determine the probability of a given model, given a specific sequence of output symbols. This corresponds to the model matching used in pattern recognition, a major application areas of HMM. Enabling the solution of this task for HnMM will broaden the number of applications that HnMM can be used for and increase their practical applicability.

STATE OF THE ART IN HNMM AND THEIR ANALYSIS

Hidden non-Markovian Models (HnMM) are an extension of Hidden Markov Models (HMM) (Fink 2008). The main enhancement of HnMM is the inclusion of time behavior and the shift of focus from the states to

the state transitions, since these are often the objects of interest in discrete stochastic systems. This also required shifting the symbol emissions from the states to the state transitions. The time-dependent transitions of a HnMM are described by continuous distribution functions, such as Normal or Weibull.

An HnMM can be described by a 6-tuple (S, C, V, A, B, Π), with the set of states S, the set of state transitions C, the set of output symbols V, the time-dependent transition matrix $A(t)$, the emission probability matrix B and the initial probability vector Π. This formal description is derived from HMM and has been adapted for HnMM, to allow for any kind of discrete stochastic models as hidden system description (Krull and Horton 2009).

The time dependence was also incorporated in the output symbol sequence O (trace) by attaching a time stamp to each symbol emission. The internal system behavior of a HnMM can be described by the sequence of state changes Q (path) with the corresponding time stamps. The path may be longer than the trace, because not every state change has to result in a symbol emission; however, every symbol emission is caused by a state change. This last condition has to be relaxed to reflect the medical diagnosis application example.

Proxel-based Analysis of HnMM

To analyze the newly developed paradigm of HnMM, we have adapted the original HMM algorithms to the new requirements Krull and Horton (2009). This was only possible for specific model properties, for example requiring the models to regenerate after every state transition. Since we developed a general modelling paradigm we are also interested in general analysis algorithms. One promising candidate is the state space-based Proxel algorithm (Horton 2002, Lazarova-Molnar 2005).

The algorithm tracks all possible system developments in discrete steps creating so-called probability elements (Proxels) and discovers possible development paths. Furthermore, it turns a model containing arbitrary continuous distribution functions into a DTMC (Bolch et al. 1998). This analogy to path analysis of HMM led us to apply Proxel-based analysis to HnMM.

The analysis of HnMM using Proxels has been described for example in (Buchholz et al. 2010). The task in that paper was to determine the most likely hidden behaviour that caused a given output trace, given a system with known specification. This is also known as the decoding task in HMM Fink (2008).

APPLICATION EXAMPLE: PATIENT DIAGNOSIS

In this paper we want to test the application of HnMM analysis to a field outside of engineering and manufacturing. The example used throughout the paper is the diagnosis of a patient's illness based on a sequence of recorded symptoms of that patient. We specified two different types of illness models: one describing the progression of an abstract illness and the other describing the progression of the common cold or an influenza infection. In this section we introduce the specification of the HnMM for that application. The basic time unit assumed in all of the models is one day.

Abstract Illness We assume that an abstract illness can progress in up to three successively severe stages, healing is possible in every stage. Each stage lasts for a random amount of time, described by an arbitrary continuous probability distribution. The time to healing depends on the total illness duration and is also random. Tests are performed every day: a blood test, urine tests and taking the patients temperature. Blood test and urine test can result in a negative, inconclusive or positive result and the patient may or may not have fever. Each stage has different probabilities for exhibiting each of the symptoms, the more probable the symptoms, the more severe the illness stage.
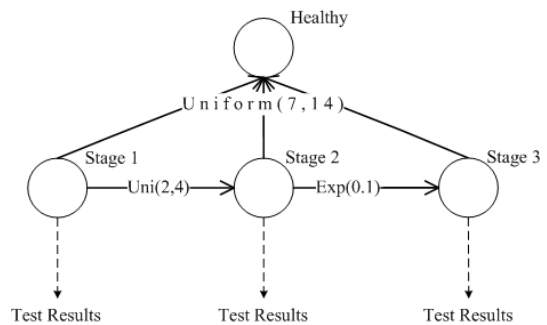

Figure 1: An HnMM Describing the Progression of an Abstract Illness and the Resulting Symptoms

Different abstract illnesses are characterized by different stage models, with individual stage durations and individual probabilities for causing certain symptoms in each stage. Figure 1 shows an HnMM describing the progression and symptom probabilites for one such abstract illness. Tables 1 and 2 show the stage durations and symptom probabilites for each of the three illnesses. While disease one has different symptom probabilities from the other two, disease two and three can only be distinguished by their stage durations.

Cold and Flu We assume, that the common cold and an influenza infection progress in two stages before healing, the first phase being the more severe of the two. We have chosen to consider fever, cough and body aches as the symptoms to be recorded, each of these will be recorded every day. An influenza infection has a more severe and a more prolonged first infection stage. The common cold has a milder progression indicated by

Table 1: Symptom Probabilities and Stage Duration of Disease One

		Stage 1	Stage 2	Stage 3
Fever	yes	0.1	0.5	0.8
	no	0.9	0.5	0.2
Urine	negative	0.8	0.6	0.5
	inconclusive	0.1	0.3	0.3
	positive	0.1	0.1	0.2
Blood	negative	0.7	0.5	0.2
	inconclusive	0.2	0.3	0.3
	positive	0.1	0.2	0.5
	Duration	U(2,4)	Exp(0.1)	U(7,14)

Table 2: Symptom Probabilities and Stage Duration of Diseases Two and Three

		Stage 1	Stage 2	Stage 3
Fever	yes	0.3	0.7	0.9
	no	0.7	0.3	0.1
Urine	negative	0.8	0.7	0.7
	inconclusive	0.2	0.2	0.2
	positive	0.0	0.1	0.1
Blood	negative	0.7	0.5	0.2
	inconclusive	0.2	0.3	0.3
	positive	0.1	0.2	0.5
Duration disease two	N(3,0.5)	Exp(0.3)	N(10,2)
Duration disease three	N(6,1)	Exp(0.3)	N(12,2)

smaller probabilities to develop the symptoms; cough, fever and aches. The two state models for influenza and common cold are shown in Figures 2 and 3.
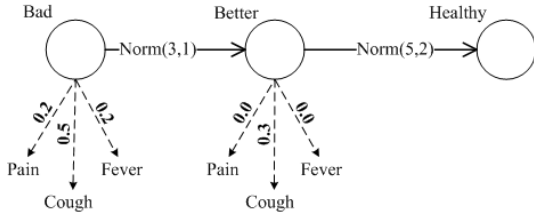

Figure 2: An HnMM Describing the Progression of the Common Cold and the Resulting Symptoms

We assume that the models describing the progression of these illnesses are known. Question: Based on the recorded symptoms of a specific patient, what is the illness that he is most likely suffering from? To answer this, the Proxel algorithm has to be adapted slightly.

Adaptions for Diagnosis Example - Decouple Emissions from State Changes

One difference of the above HnMMM to the definition from (Krull and Horton 2009) is that the state transitions of the illness progression do not cause symbol output. It is easy to reason that the progression from one stage of the illness to another does not cause a blood test. The symptom probabilities depend on the stage of the illness, which means that symbols are emitted based on the current state of the system, as in the original HMM paradigm (Fink 2008). The difference to HMM
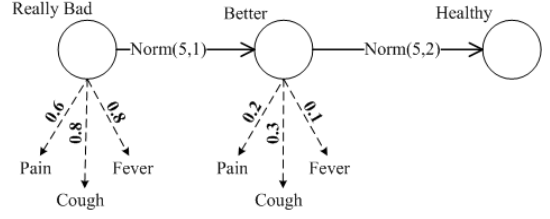

Figure 3: An HnMM Describing the Progression of an Influenza Infection and the Resulting Symptoms

is, that the emitted symbols in our case still have a time stamp, which can take on any value. Similarly, a detected symbol emission does not cause or indicate a state transition, effectively decoupling them.

The Proxel algorithm can be used in its original specification. The necessary adaption is that when a symbol is detected, the set of Proxels for the corresponding time step is modified as follows:

- Proxels that represent discrete states which could not have emitted the given symbol are discarded.

- The probabilities of the remaining Proxels are scaled by the particular symbol's emission probability in the discrete state of each Proxel.

The resulting reachability graph contains all reachable model states, now under the condition of having observed the given symbol sequence. When asking for the most likely model to have produced a given symbol sequence, we no longer need the information of the specific generating path. Therefore the path information can be omitted from the Proxel, speeding up the analysis algorithm considerably. The next section describes the use of the result of this Proxel algorithm in model matching.

MODEL MATCHING IN HNMM

Model matching for HnMM can be performed in the same way as speech recognition using the Forward algorithm in HMM (Fink 2008). In speech recognition, word models are assigned a probability to have produced a recorded piece of speech. The most probable word model is the most likely meaning of the audio recording. The modified Proxel algorithm can also compute the probability with which a given model has produced a given output sequence. The model with the largest probability should be the most likely model to have caused a given sequence.

In the application example, the Proxel algorithm can compute the probability with which a given sequence of symptoms is caused by a specific illness. This probability corresponds to the sum of the unconditional probabilities of all Proxels valid at the end of simulation time. These Proxels represent all possible end points of the patient's illness, under the condition of observing the given sequence of symptoms.

The unconditional probabilities of the different illness models to have produced the given symptom sequence can be compared and the illnesses ranked according to their likelihood. We assume that the largest probability represents the most likely illness to have caused these symptoms. This knowledge can be used to select a treatment or medication.

The procedure for HnMM matching is the following:

1. For each possible model or model configuration, the total probability of generating the given sequence is calculated as described in the previous section.

2. The models are ranked according to these unconditional probabilities.

3. The model with the largest probability of generating the given sequence represents the most likely system to have produced the output.

We applied this model matching approach in different experiments, described in the following section.

Knowledge of likely system behavior can be used to determine promising courses of action to take. The model that has been determined to be the most likely one for the system can be used to simulate the future system development. Knowledge about a likely system setup can also be compared to the manufacturer's specifications. When large deviations are detected, further steps can be taken, such as scheduling maintenance or supporting claims towards the manufacturer.

EXPERIMENTS WITH PATIENT DIAGNOSIS EXAMPLE

This section describes the validation and performance experiments performed for the Proxel-based HnMM matching algorithm and patient diagnosis application example. For test purposes we created models of the different illnesses and their progression in AnyLogic (Borshchev 2007). We then generated several different sequences of symptoms for each of the different models.

Due to inherent ambiguities within and similarities among the models, the illness model that was used to create a specific symptom sequence is not always the most likely one. The experiments are aimed at answering the following questions: How does less / more information (less, more frequent symptom data) influence the matching accuracy and algorithm performance? More specifically, how does matching accuracy develop when increasing the similarity of the models? If the matching errors are due to inherent ambiguities or algorithmic problems is currently very hard to determine, since it is still unclear how to determine inherent model ambiguities. The answers to these questions will lead to a statement of the overall feasibility of the proposed method for model matching in this particular application setting.

We will be using the following quantities as measures of our algorithms matching accuracy (Bramer 2008): accuracy , precision, recall and F1-measure. Symbol explanation: P positives (traces caused by the illness), N negatives (traces not caused by the illness), correctly classified traces: TP true positives and TN true negatives, incorrectly classified traces FP false positives and FN false negatives.

$$
\begin{aligned}
accuracy &= \frac{TP + TN}{P + N} \\
precision &= \frac{TP}{TP + FP} \\
recall &= \frac{TP}{P} \\
F1 - measure &= \frac{2 * precision * recall}{precision + recall}
\end{aligned}
$$

The algorithms performance will not be evaluated in a separate experiment. In all of the test cases, the algorithm runtime was within a few seconds, even with the smallest evaluation time step chosen ($\Delta t = 0.1day$). Smaller evaluation time steps were not necessary, because the choice of evaluation time step had little effect on the matching accuracy. Furthermore, the traces are of limited length, since the symptoms are recorded only for the relatively short period of time, during which the patient is ill, resulting on traces of at most 100 symbols.

Measurement Frequency Experiment

The goal of this experiment was to determine the effect of the symbol frequency on the matching accuracy. The symbol frequency in the application example was determined by the measurement interval of the various symptoms. We used the illness example with three distinct abstract illnesses as our test set. We created two separate test sets, each containing five traces for each illness. These 15 traces were then tested against each of the three illnesses. We varied the interval between subsequent measurements (blood test, urine test, taking temperature) from 0.25 days (0.5 days in test set two) to 2.0 days. The illness with the highest absolute probability for generating the given trace was then assumed to be the one to have caused that trace.

With increasing measurement intervals, the number of correctly classified traces decreases, and the class assignment of a trace seems to become arbitrary. Table 3 shows precision, recall and F1-measure for the different measurement intervals. It shows that the matching accuracy decreases with increasing measurement intervals. Figure 4 shows the fraction of correctly classified traces when the measurement frequency decreases. For a measurement interval of 0.25, almost all traces are classified correctly. Doubling the time between two measurements decreases the matching accuracy considerably. However,

Table 3: Accuracy Measures for Different Measurement Intervals for First Test Set

Interval	Precision	Recall	F1-Measure
0.25	1.0	1.0	1.0
0.5	0.84	0.8	0.818
1.0	0.61	0.6	0.606
2.0	0.59	0.6	0.594

the size of this decrease depends on the test set. Matching accuracy of the first test set declines more steeply than that of the second one.
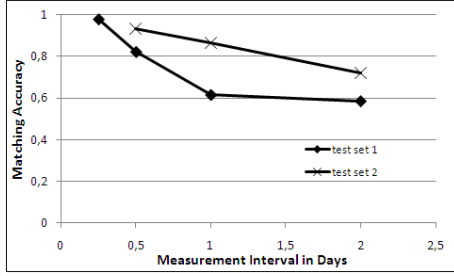

Figure 4: Matching Accuracy with Increasing Measurement Interval for Different Test Sets

The decrease in matching accuracy with decreasing symbol frequency demonstrated in this section is to be expected. It is due to a decrease in the amount of information available on the system behavior, thus also decreasing the ability to differentiate between traces created by different illness models. This decline is not due to shortcomings in the algorithm itself, but only to a decreasing amount of information available on the patient.

Symptom Reduction Experiment

The goal of this experiment was to determine the effect of the number of different symptoms on the illness matching accuracy. We used the illness example with three distinct abstract illnesses as our test set. We created two separate test sets, each containing five traces for each illness. These 15 traces were tested against each of the three illnesses. We varied the number of different measurements available by deleting one or two of the symptoms from the traces, fixing the measurement interval at 0.25 days. The illness with the highest absolute probability for generating the given trace was assumed to be the one to have caused the symptoms.
Figure 5 shows the development of the fraction of correctly classified traces when the number of available symptoms decreases. When all three symptoms are available in a trace, almost all traces are classified correctly. When the number of symptoms available is reduced to two, the matching accuracy decreases slightly. When only one of the three symptoms is available, the accuracy declines considerably, while not each of the three different symptoms leads to the same steep de-
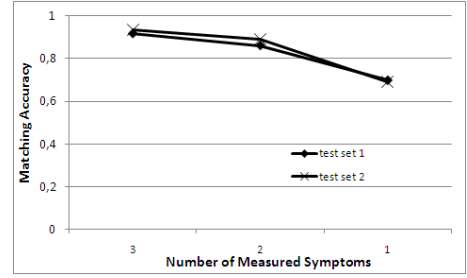

Figure 5: Matching Accuracy with Decreasing Number of Symptoms being Recorded for Different Test Sets

Table 4: Symptom Probabilities for Midpoint Model

		Stage 1	Stage 2
	cough	0.65	0.3
midpoint	fever	0.5	0.05
	pain	0.4	0.1

cline. Both test set showed almost the same behavior. The symptom fever distinguishes better between the illnesses, probably because it has only two possible outcomes, but the exact reason for this different ability to distinguish is a subject of further research.
The reduction of matching accuracy observed in this experiment is again as expected. Decreasing the amount of information available also decreases the ability to differentiate between traces created by different models.

Ambiguity Experiment

The goal of this experiment was to determine the effect of model ambiguity on the matching accuracy. We used the models of the common cold and the influenza as baseline, because the algorithm was able to match these traces even with large measurement intervals with an accuracy of 0.9. We created an intermediate model by changing the stage durations and symptom probabilities of the common cold model towards those of the influenza. For each of the different levels of similarity, ten traces were created for the influenza model, the midpoint model and the common cold model.
The stage durations and symptom probabilities for the common cold (*Cold*) and the influenza infection (*Flu*) are given in Section . The stage durations for an intermediate point (*midpoint*) are $Stage1 \sim N(4,1)$ and $Stage2 \sim N(5,2)$. Table 4 shows the symptom probabilities in each stage for an intermediate point (*midpoint*). Figure 6 shows the effect of increasing model similarity on the matching accuracy for the given example. The x-axis reflects the increasingly similar stage durations, where *Flu-Cold* having the largest difference and *Flu-Flu* the same stage durations for both models. The different colored bars similarly reflect the increasingly similar symptom probabilites. The small increase in matching accuracy when changing the symptoms to the mid-
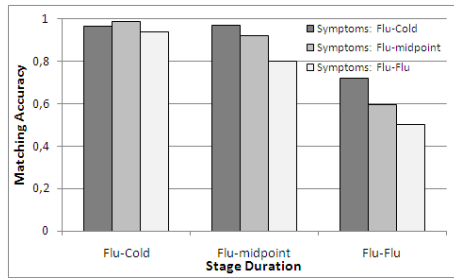
Figure 6: Accuracy for Increasing Model Similarity

point and keeping the original stage durations is probably due to random effects in the two different test sets. The diagram shows that with increasing similarity it becomes increasingly difficult to determine the correct illness. The matching accuracy seems to be more sensitive to a change in stage durations than to a change in symptom probabilities. However, that might be due to the test sets considered. The decline in matching accuracy in this experiment is due to increasing ambiguity between the different models.

CONCLUSION AND OUTLOOK

The paper presents a new application of HnMM from medicine. A patient's internal state of illness is unknown and regarded as source of the observed symptoms. The analysis is done by on-the-fly state space-based simulation only regarding the paths that could have resulted in the observed trace. The algorithm is successfully applied to finding a specific model's probability for a given symptom trace and finding the most likely model to have produced the given symptom trace.

The algorithm's matching accuracy is tested in the experiments section. The proposed method is able to accurately determine the correct illness for a given sequence of symptoms. The algorithm exhibits an expected decrease in matching accuracy, when the amount of information available is reduced or the model ambiguities are increased. The computational effort for the matching procedure is neglectable in this application example, due to the limited size of the traces and models.

Model matching is tested for the first time for HnMM, broadening the range of analysis tasks that can be solved for HnMM. The application example requires symbols being emitted depending on the current state at arbitrary points in time, not as before at state changes, thus further broadening the range of application areas.

Future Work

The application example presented in this paper is a diagnosis based on symptoms. We modeled an illness from an engineer's point of view. We do not know whether discrete stochastic models as we used them can represent illness progression in general or in specific cases. We still have to discuss the proposed approach with practitioners and build models based on field data or incorporate other important aspects that we have so far disregarded. Furthermore, the approach will most likely work for applications from engineering and manufacturing, because these systems can be represented accurately by discrete stochastic models.

We are also attempting to sompare the proposed approach with established data mining or classification techniques. However, it is not yet clear, which methods are applicable to the systems specified in this paper. We are interested in incorporating interventions from outside the model, which affect the future development of the system. This can be interesting in the medical domain, when incorporating the effect of treatments and medications. In general, we need to test the applicability of the approach to more complex systems, testing boundaries of applicability and extend the model matching to different application domains.

References

Bolch G.; Greiner S.; de Meer H.; and Trivedi K.S., 1998. *Queuing Networks and Markov Chains*. John Wiley & Sons, New York.

Borshchev A., 2007. *Multi-Method Simulation Modeling using AnyLogic*. INFORMS Roundtable Fall Meeting, Seattle. Www.anylogic.com XJ Technologies.

Bramer M., 2008. *Principles of Data Mining*. Springer-Verlag London ltd.

Buchholz R.; Krull C.; Strigl T.; and Horton G., 2010. *Using Hidden non-Markovian Models to Reconstruct System Behavior in Partially-Observable Systems*. In *3rd International ICST Conference on Simulation Tools and Techniques*.

Fink G.A., 2008. *Markov Models for Pattern Recognition*. Springer, Berlin, Heidelberg.

Horton G., 2002. *A New Paradigm for the Numerical Simulation of Stochastic Petri Nets with General Firing Times*. In *Proceedings of the European Simulation Symposium 2002*. SCS European Publishing House, 129–136.

Krull C. and Horton G., 2009. *HIDDEN NON-MARKOVIAN MODELS: FORMALIZATION AND SOLUTION APPROACHES*. In *Proceedings of 6th Vienna Conference on Mathematical Modelling, Vienna, Austria*. 682–693.

Lazarova-Molnar S., 2005. *The Proxel-Based Method: Formalisation, Analysis and Applications*. Ph.D. thesis, Otto-von-Guericke-University Magdeburg.