

Phase Detection in Medical Context: Overview and Outlook

Patrick Philipp

Vision and Fusion Laboratory
Institute for Anthropomatics
Karlsruhe Institute of Technology (KIT), Germany
p.philipp@kit.edu

Technical Report IES-2017-05

Abstract: To provide assistance functions, e.g. in context of surgical interventions, the use of a phase detection plays an important role. For instance, by assessing the progress of an on-going surgery, a tailored (i.e. context sensitive) decision support for medical practitioners can be carried out. The optimization of a workflow, e.g. by comparing recorded data with a pre-defined target model, is another application example. Subsequently, a phase detection provides opportunities to prevent errors, injuries, negligence or malpractices in the medical context. In this work, an overview of notable model approaches for a phase detection in medical context is presented. Based on this, further suggestions for future models are proposed.

1 Introduction

In contemporary medicine, the use of advanced computer-based assistance (including both: hardware and software) becomes increasingly important [PFHB16]. E.g. the global market for medical robotics and computer-assisted surgical equipment is projected to grow to 6.8 billion dollars by 2021 (using a five-year compound annual growth rate of 11.3%) [McW17].

As part of a computer assisted surgery (CAS), assistance functions can be realized to enable a decision support for medical practitioners [KWN⁺15]. Thereby, a decision support opens up a field of optimization, e.g., concerning the prevention of errors, negligence, injuries or (as a result) malpractices.

In this context, a surgical phase detection plays an important role. Namely, because by assessing the progress of an on-going surgery, a tailored (i.e., context sensitive) and interactive decision support during an intervention can be enabled. In doing so, there is not only a passive dissemination (e.g. distribution via print media) of support (e.g. medical guidelines) – which has only little effect on the actual practitioners behavior [FL92, SGM⁺11]. Moreover, a tailored decision support allows for filtering physically available information in order to be operationally effective by preventing an informational overload of medical practitioners [JL83, KSF⁺13, LDC⁺13].

The optimization of a workflow, e.g. by comparing recorded data with a predefined target model, is another application example of a phase detection. This is especially relevant as surgeries have been identified as an important source of improvement to the hospital efficiency [Her03].

In this work, an overview of notable model approaches for a phase detection in medical context are presented. Based on this, further suggestions for elaborating future models are proposed. This contribution is structured as follows: first in Section 2, the constituents of the phase detection problem are elaborated. In Section 3 notable modeling approaches are presented and summarized in Section 4. Section 5 focuses on future implications for models concerning a phase detection and, finally, a conclusion is drawn in Section 6 on page 16.

2 Constituents of the Problem

Figure 2.1 depicts a workflow of a surgery using an UML activity diagram [OMG11]. This formalism is also used in previous research concerning the modeling of medical workflows [PFHB15a, PFHB15b, Phi16, PFHB16, PFB17].

The start of the activity “Surgery” is symbolized by a solid circle (initial node), whereas the end of the activity is given by a double circle (activity final). Performed actions are shown as rounded rectangles, while the control flow is represented by arrows (directed edges). In this example, the actions are performed sequentially – i.e. one after another.

The notation elements “...”, the red rectangle as well as the red arrows are added as visual aids – they are not part of the UML specification [OMG11]. Thereby

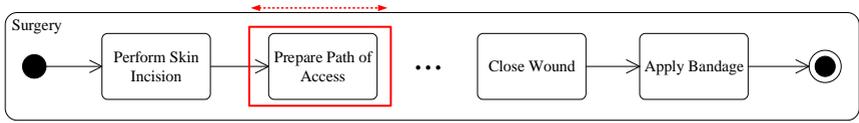


Figure 2.1: Figure shows a pre-modeled workflow of a surgery. Goal of a phase detection is to match a pre-modeled workflow with the current conditions in the operating room.

“...” symbolize a sequences of arbitrary actions which are not shown in the Figure for simplification. The red rectangle highlights the current predicted action, which changes over time.

Goal of a phase detection is to match online a pre-modeled workflow with the current situation in the operating room. It means in effect (cf. Figure 2.1) that we want to know, e.g., is the surgeon currently performing the skin incision or is he already preparing the path of access.

To reach this goal, feature values x are made available by sensors in the operating room. Such values are e.g. “number of persons at the table”, “number of used instruments” and so forth. These values are bundled by a feature vector \mathbf{x} which is defined as a column vector whose elements are these feature values x – i.e.

$$\mathbf{x} = (x^1, \dots, x^D)^T.$$

Whereby D represents the number of features, i.e. the dimension of the resulting feature space as well as of the corresponding feature vector. To save space, vector \mathbf{x} is shown horizontal (i.e. transposed, indicated by T).

In each time step a new feature vector \mathbf{x} with index t is observed by the technical system in the operating room:

$$\mathbf{x}_t = (x_t^1, \dots, x_t^D)^T.$$

As a consequence, there is an observation sequence of feature vectors

$$\mathbf{x}_{1:t} = \mathbf{x}_1, \dots, \mathbf{x}_t.$$

The observation sequence $\mathbf{x}_{1:t}$ is available to the technical system, whereas the current phase is not. I.e. there is a gap between the observation sequence $\mathbf{x}_{1:t}$

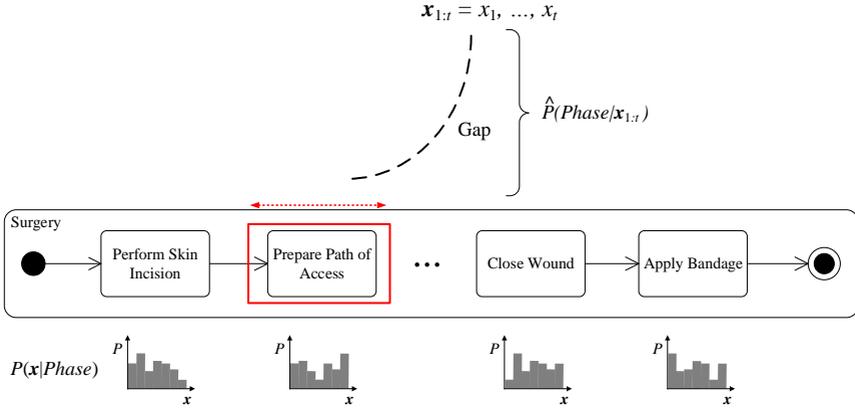


Figure 2.2: Figure depicts the gap between a sequence of feature vectors and the surgical phase. The latter is hidden to the technical system (w.l.o.g.). To bridge the gap between them, a probability distribution P can be used to specify how likely it is to emit a certain feature vector being at a specific phase (i.e. $P(\mathbf{x}|\text{Phase})$). A Hidden Markov Model (HMM) can be used to infer the probability of being in a phase given the observation sequence.

and the surgical phase which is not directly visible (i.e. hidden) to the technical system (w.l.o.g. cf. Figure 2.2).

In traditional approaches like Hidden Markov Models (HMM), a probability distribution P is used to specify how likely it is to emit a certain feature vector being at a specific phase (cf. Figure 2.2 bottom row).

Using further parameters of the HMM (cf. Section 3.3), vice versa, the probability of being in a Phase given the observation sequence can be inferred. As a consequence the gap between the observation sequences and a specific phase is bridged. Further model characteristic details are elaborated in the following sections.

3 Modeling Approaches

In the following section notable modeling approaches for a medical phase detection are elaborated. Thereby, the different characteristics of the models are illustrated.

3.1 Random Forest (RF)

A RF is a model which derives its decision from a set of classifiers and therefore belongs to the so called ensemble methods [Rok10, Zho12]. A RF predicts the class of a feature vector $f(\mathbf{x})$ by a weighted sum [Bre01]

$$f(\mathbf{x}) = \sum_{n=1}^N \alpha_n t_n(\mathbf{x}).$$

Thereby a set of decision trees (i.e. a forest) is used:

$$\{t_n(\mathbf{x}) : n = 1, \dots, N\}.$$

These classifications results are equally weighted, i.e. $\alpha_n = (N)^{-1}$. Conceptually, a RF utilizes randomization in two ways:

Firstly, the decision trees $t_n(\mathbf{x})$ are trained by randomly sampling the training set $\mathcal{D} = \{\mathbf{x}_1, \dots, \mathbf{x}_M\}$ with known class memberships $\omega(\mathbf{x}_i)$ for $i = 1, \dots, M$. The idea behind this is, to lower the variance by averaging over a large number of trees which are highly uncorrelated (bootstrap aggregation aka bagging).

Secondly, during the training of a decision tree, for each node only a random subset of feature values are considered for splitting the tree. I.e. the size of the considered feature vector \mathbf{x} in each split is $d < D$. Based on a quality metric (e.g. entropy or genie) and under consideration of d feature values, the split with the highest quality is calculated. Stopping criteria of the training algorithm of a tree is, e.g., that a minimum size of a leaf is reached. I.e. the split of a set of feature vectors would result in subsets that are too small.

For classification, each decision tree $t_n(\mathbf{x})$ of the RF receives the same feature vector \mathbf{x} and maps it to a class $\omega \in \Omega$. The corresponding a-posteriori probability is given by:

$$\hat{P}(\omega|\mathbf{x}) = \frac{1}{N} \sum_{n=1}^N [t_n(\mathbf{x}) = \omega]. \quad (3.1)$$

Please note that $[\cdot]$ represents a predicate mapping, i.e. $[\cdot]$ has a value of 1 iff the corresponding expression evaluates to true.

The predicted class $\hat{\omega}$ is then given by [Bre01]:

$$\hat{\omega} = \arg \max_{\omega \in \Omega} \hat{P}(\omega | \mathbf{x}). \quad (3.2)$$

The use of the RF model for a medical phase detection by the application example of a cholecystectomy (removal of the gallbladder) is elaborated in [SOP⁺14]. 7 phases are considered in context of this surgery. The training set comprises 9 surgical interventions. Nominal features are the use of eight specific instruments (yes/no), the state of the surgical light (on/off) as well as the state of the room lights (on/off). The authors claim that an RF is suitable, because it is highly applicable for multi-class problems and it is able to predict phases in a-typical order. The latter is a consequence of the fact that a RF does only consider a single feature vector for classification (cf. Equations (3.1) and (3.2)). That means in effect, that a feature vector is classified without taking the order of phases into account.

The trained model achieved an accuracy of around 69% for the 7 classes using cross-validation with a leave-one-out iterator. Classes with similar characteristics (which differ only in the sequence of occurrence) can not be distinguished. In [SPG⁺16] the authors also get comparable results for a simplified 7-phase hip replacement surgery with a high confusion for phases which are highly discriminative to the order of phases.

This is a prototypical behavior of a classifier that does not take the sequence of feature vectors into account. Therefore this model is, out of the box, not suited regarding the constituents of the problem. Nevertheless, to reduce the conceptual drawback, such a model can be embedded into a sequential model. One simple form of such a model is, e.g., a deterministic automaton.

In [KSW⁺16] this concept (cf. Figure 3.1) is used on two application examples: Firstly, a pancreatic resection (removal of the pancreas) comprising 12 classes and a training set of 11 surgeries. Secondly a adrenalectomy (removal of the adrenal glands) with 9 classes and a training set of 5 surgeries. Considered features are the use of a specific instrument, the action performed and the anatomical structure.

The use of this sequential framework (cf. Figure 3.1) is like shifting a window over a workflow (cf. Figure 2.1) comprising the current phase, and additionally,

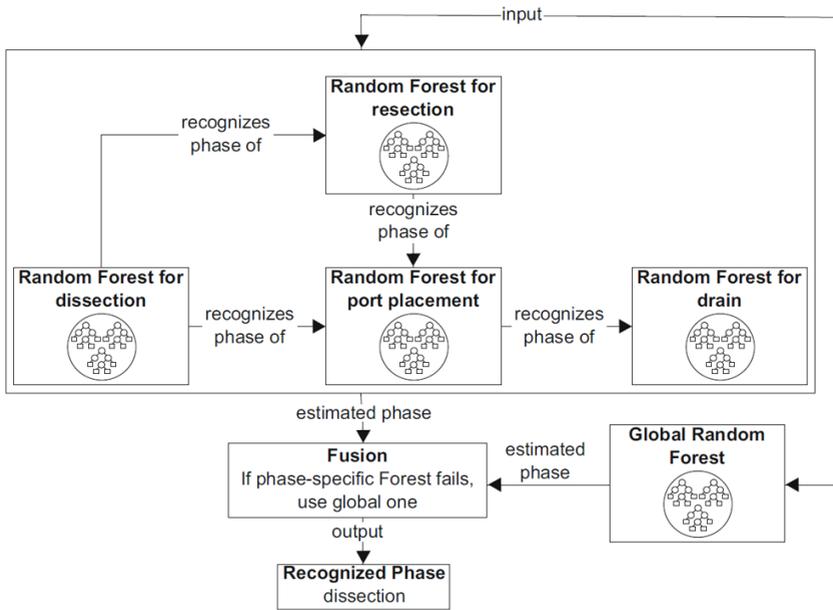


Figure 3.1: Figure depicts the utilization of a random forest (RF) embedded into a simple sequential model. A RF is used to predict the current and possible next phases. (e.g. dissection with possible next phases resection and port placement). If a next phase is predicted, a transition is made. As a result, another RF is used to discriminate between the new current phase and possible next phases. If a transition is made by mistake, the model tends to get lost. Therefore a global RF is used to discriminate between all possible phases in order to be able to reset the sequential model. Modified from [KSW⁺16].

possible next phases. A window-specific classifier then discriminates between the phases inside the window [Phi16, Phi17]. The window is shifted forward as soon as a next phase is predicted.

The modeling approach in [KSW⁺16] achieved an accuracy around 70%. Drawback is the use of deterministic transitions, which makes it necessary to implement strategies to reset the sequential model (cf. Figure 3.1).

3.2 Stochastic Petri Net (SPN)

There are a whole range of reasons for considering Petri Nets [Pet62] as a modeling tool for dynamic aspects of a process [vdA96, vdA98]. With respect to the application example, Petri Nets can be of use because of their formal semantics and the abundance of analysis techniques.

The net structure [Rei13b] of a Petri Net is given by the tuple

$$NST = (\mathcal{P}, \mathcal{T}, \mathcal{F}),$$

where \mathcal{P} is the set of places and \mathcal{T} is the set of transitions

$$\begin{aligned}\mathcal{P} &= \{p_i : i = 1, \dots, |\mathcal{P}|\}, \\ \mathcal{T} &= \{t_i : i = 1, \dots, |\mathcal{T}|\},\end{aligned}$$

so that

$$\mathcal{P} \cap \mathcal{T} = \emptyset.$$

The flow relation \mathcal{F} reflects the connection of places and transitions (and vice versa):

$$\mathcal{F} \subseteq (\mathcal{P} \times \mathcal{T}) \cup (\mathcal{T} \times \mathcal{P}).$$

Consequently, the net structure NST of a Petri Net is a directed bipartite graph (see Figure 3.2). To model the dynamic behavior of the system, so called “tokens” are introduced (black dot in Figure 3.2). The distribution of tokens on the set of places represents the state of the Petri Net. It is also called marking [Rei13b].

A Stochastic Petri Net (SPN) is defined as a tuple

$$SPN = (NST, \Lambda),$$

whereby $\Lambda = \{\lambda_i : i = 1, \dots, |\mathcal{T}|\}$ is the set of shifting rates λ_i which are assigned to transitions t_i . These shifting rates are distributed exponentially [Mol81].

Figure 3.2 depicts a SPN on the left. Circles represent places p_i and squares represent transitions t_i . The state of the SPN is given by the current distribution of tokens (marking). In the reachability graph the states are represented by vectors, whereby each entry i represent the number of tokens at a place p_i . That means,

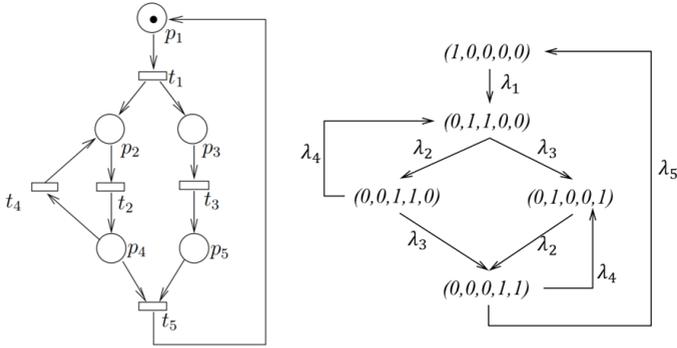


Figure 3.2: Figure depicts a SPN on the left. Circles represent places p_i and squares represent transitions t_i . The state of the SPN is given by the current distribution of tokens (marking). In the reachability graph the states are represented by vectors, whereby each entry i represent the number of tokens at a place p_i . The Figure visualizes that a SPN represents a Markov Process whereby, inter alia, the firing rates λ_i of states Z_i of the reachability graph corresponds to the transition probabilities a_{ij} of a (discrete) Markov Process. Modified from [B⁺02].

the state $(1, 0, 0, 0, 0)$ of the reachability graph represents the state of the SPN depicted on the left. Figure 3.2 visualizes the fact that a SPN actually represents a Markov Process. I.e. the states of the reachability graph of the SPN correspond to the states of a Markov Process and the firing rates λ_i of states Z_i of the reachability graph correspond to the transition probabilities a_{ij} of a (discrete) Markov Process. That is, inter alia, because the firing rates are exponentially distributed (cf. memorylessness) and therefore the Markov Property is satisfied [B⁺02].

The use of Markov Models with observable states is suitable in cases where the states of the modeled system are directly accessible. E.g. in [PFHB15a] a decision support system for the diagnosis of two complex cancerous diseases is modeled using Petri Nets. The involved dialog system allows for a direct access to the necessary features. Clearly, this model can also be used as sequential model to embed a model like a RF instead of a deterministic automaton (cf. Section 3.1). Nevertheless, with respect to the constituents of the problem, this model has to be extended, e.g. by introducing hidden states.

3.3 Hidden Markov Model (HMM)

A Hidden Markov Model (HMM) [RJ86] is a Markov Model using a Markov Process with unobserved (i.e. hidden) states. That means, in contrast to Markov Processes, where the state of the model at a time step t is known for certain (i.e. directly observable), in a HMM an observer can only access emissions which are generated by a hidden (i.e. not observable) state. To enable this, state transition probabilities and furthermore emission probabilities are part of a HMM.

Formally, a (discrete) HMM is defined as a 5-tuple

$$\lambda = (S, V, A, B, \pi),$$

whereby $S = \{s_1, \dots, s_n\}$ is the set of hidden states and $V = \{v_1, \dots, v_m\}$ is the set of emissions for each hidden state (i.e. output vocabulary). Moreover, there is a transition matrix $A \in \mathbb{R}^{n \times n}$ encoding the transition probabilities from a current state s_i to a next state s_j by the matrix entry (a_{ij}) . Matrix $B \in \mathbb{R}^{n \times m}$ encodes the emission probabilities of a state s_i by the corresponding row entries $(a_{ij}) : j = 1, \dots, m$. Finally, $\pi \in \mathbb{R}^n$ encodes the initial state probabilities, i.e. the probability that a state is the starting state.

The use of a HMM for a medical phase detection has been elaborated in [BPFN08] by the application example of a cholecystectomy (removal of the gall-bladder). 14 phases are considered in context of this surgery. The training set comprises 12 surgical interventions. Nominal features are the use of 17 specific instruments (yes/no). The authors present a 14-state HMM and a merged HMM. In the latter, similar states are transformed into one single state. The models achieved an accuracy of around 86% and 93% respectively, using a complete cross-validation.

In [PBF⁺08] a cholecystectomy comprising 14 phases is considered, too. The training set comprises 11 surgical interventions. Nominal features are the use of 18 specific instruments (yes/no) including the state of an optical device (inserted/not inserted). The best accuracy achieved using a complete cross-validation is around 92% – but for computing this value, a tolerance of 5 seconds before and after the ground truth definition of a phase is set. Additionally, the authors state that phases with very short durations are poorly recognized which is a result of the inertia of the model.

A HMM can be seen as a common sequential model considering the constituents of the problem (cf. Section 2). Indeed, it can be a suitable model for problems where data is not independent and identically distributed, i.e. a feature vector \mathbf{x} does depend on the feature vectors seen before. Nevertheless, the structure of this model is not well suited to include expert knowledge about emission probabilities. That is because, inter alia, the corresponding probability distribution can not be factorized and therefore grows exponentially with the number of possible emissions. To overcome this drawback, the generalization of a HMM, namely a Dynamic Bayesian Network can be used.

3.4 Dynamic Bayesian Network (DBN)

A Bayesian Network (BN) is a probabilistic graphical model (PGM), combining graph theoretic approaches with approaches of probability theory. Consequently, a BN over random variables $X^{0:N} := X^0, \dots, X^N$ is given by a pair

$$B = (G, P).$$

Whereby G corresponds to a directed, acyclic graph

$$G = (V, E),$$

and

$$P(X^{0:N}) = \prod_{n=0}^N P(X^n | \text{Pa}(X^n)),$$

corresponds to a joint probability distribution [KF09].

Graph G is used to define dependencies between random variables $X^{0:N}$. It is also known as the structure of the BN. The vertex set V represents the set of random variables, while a directed edge $V_i \rightarrow V_j$ of the set of edges E represents a direct dependency between two variables. A missing edge symbolizes the independence of these two variables.

The joint probability distribution is given by the product of all conditional probability distributions associated with the vertices of G . It is also known as the parameters of the BN. Here, $\text{Pa}(X^n)$ denotes the set of parents of a random variable X^n . Graphically, this corresponds to vertices having a directed edge pointing to

X^n 's vertex. Please note, if $\text{Pa}(X^n) = \emptyset$, a random variable X^n is a root node of the BN, and $P(X^n | \emptyset) = P(X^n)$ gives the a-priori probability.

A Dynamic Bayesian Network (DBN) is an extension of a BN, also taking the temporal dependencies of variables into account [Mur02]. A DBN is given by a pair

$$\text{DBN} = (B_0, B_{\rightarrow}),$$

where the BN B_0 uses $P(X_0^{0:N})$ to specify the a-priori probability distribution over random variables $X^{0:N}$ in a time step with index 0.

Furthermore, B_{\rightarrow} specifies the conditional probability distribution over discrete time steps t by using

$$P(X_t^{0:N} | X_{t-1}^{0:N}) = \prod_{n=0}^N P(X_t^n | \text{Pa}(X_t^n)).$$

Thereby $\text{Pa}(X_t^n)$ denotes the set of X_t^n 's parents in the corresponding graph. The parents can be in the same time slice (e.g. representing instantaneous causation) or the previous one (i.e., we assume the model to be first-order Markov). In the latter case, arcs point to time slices with ascending index, reflecting the causal flow of time [Mur02].

Figure 3.3 depicts on the left a DBN structure that represents a HMM. Thereby the root node represents a surgical phase and the child node represents the emission of feature vectors \mathbf{x}_t given a phase, i.e. $P(\mathbf{x}_t | \text{Phase}_t)$. By introducing a conditional independence between the feature values (cf. Figure 3.3, right side), a naive DBN can be constructed. The corresponding probability distribution of the children is given by $\prod_{n=1}^N P(\mathbf{x}_t^n | \text{Phase}_t)$. It can easily be seen that in the first case the number of parameters grows exponentially, whereas in the second case the number of parameters grows linearly. For example, let's assume that there are 7 distinct phases to predict and there is a feature vector \mathbf{x}_t with dimension $D = 3$ having discrete feature values with 4 different characteristics. Then, the corresponding conditional probability $P(\mathbf{x}_t | \text{Phase}_t)$ of a HMM is given by $(4 \cdot 4 \cdot 4) \cdot 7 = 448$ parameters from which 441 have to be specified because the probabilities sum up to 1. But $\prod_{n=1}^N P(\mathbf{x}_t^n | \text{Phase}_t)$ is given by $(4 + 4 + 4) \cdot 7 = 84$ parameters from which 63 parameters have to be specified. Further details can also be found in [PFB17].

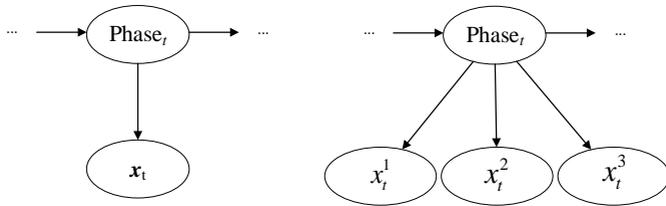


Figure 3.3: Figure shows on the left a HMM represented by a DBN structure. There is a root node representing the surgical phases and a child node representing the emissions given a phase. The parameters needed to specify a conditional probability distribution of this child node grow exponentially with the number of used features, i.e. with the size of the feature vector \mathbf{x}_t . On the right side of the figure a DBN structure is shown which takes advantage of conditional independence of feature values x_n^t to reduce the growing of parameters: in the depicted case the number of parameters linearly grows with the size of \mathbf{x}_t .

This reduction of parameters can be useful in case only a small amount of training data is available [KF09]. Furthermore, the special structure of the network supports expert-based parametrization and therefore to incorporate the knowledge of experts into the models – which also can be useful when dealing with small amount of training data and in context of translation and/or fusion mechanisms of workflow models [PFHB15b, PFHB16, PFB17].

In [PSG⁺16] a DBN model is elaborated by using the application example of a simplified total hip replacement. 7 phases are considered in context of this surgery and the training set comprises 12 surgical interventions. Features are the positioning of the sergeants (position A / position B), the number of used instruments (0/1/2) as well as the number of persons at the operating table (0/1/2/3/4). Using a complete cross-validation, the model achieves an accuracy around 85%. Concerning an earlier publication using RF [SPG⁺16] the sequence modeling greatly reduces confusion of time depended feature values.

4 Overview

The model approaches discussed in Section 3 are briefly summarized in Table 4.1. Thereby a Random forest (RF) shows prototypical behavior of a model

which omits the sequence of feature vectors x_t during classification. Although this drawback can be addressed, e.g. by embedding such a classifier into a sequential model, a RF can generally be categorized as non-sequential. Subsequently RF do not allow to model hidden states, which can be necessary to represent the fact that an observer can not directly access the state of a modeled system (cf. Section 2). Furthermore a RF is a discriminative model and allows for an online classification. The latter is necessary to be able to provide a phase detection on the fly (cf. Section 2). The model allows for a data-driven training but lacks of a suitable interface to incorporate expert knowledge which can be especially of use if only a small amount of training data is available.

A Stochastic Petri Net (SPN) represents a Markov Process (cf. Section 3.2). This model is suited for representing sequential dependencies for systems in which internal states are visible to an observer (e.g. Dialogue Systems). Furthermore, SPN is a generative approach allowing for an online classification. A data-driven training is possible, e.g. by using genetic algorithms. Finally, a knowledge-based parametrization of a SPN is possible, too – in practice the size of the model can cause limitations. In such a case, an extension of the model by using additional concepts, e.g. Coloured Petri Nets, can become necessary.

A Hidden Markov Model (HMM) can be seen as a common and well researched sequential model. A HMM allows for a direct modeling of hidden states which suits the considered application example (cf. Section 2). It is a generative approach and is able to classify online. Nevertheless, the structure of this model is not well suited to incorporate expert knowledge about the emission probabilities. That is because, inter alia, the corresponding probability distribution can not be factorized and therefore grows exponentially with the number of possible emissions.

To overcome this drawback, the generalization of a HMM, namely a Dynamic Bayesian Network (DBN) can be used. It extends the model by the ability to express the state space in a factored form and not only as a single random variable. This allows for the reduction of parameters and consequently facilitates the improvement of modularity and interpretability. This opens up a practicable way to incorporate expert knowledge into a DBN. Furthermore, concerning Kalman Filter Models, a DBN allows for arbitrary probability distributions (not only for

	RF	SPN	HMM	DBN
Sequential Model	□	■	■	■
Hidden States	□	□	■	■
Generative Approach	□	■	■	■
Online Classification	■	■	■	■
Data-driven Training	■	■	■	■
Knowledge-based Parametrization	□	■	□	■

Table 4.1: The Table briefly compares different modeling approaches discussed in this work. The symbols ■ / □ are used to specify if a property is present / not present.

unimodal linear-Gaussians). A DBN is a promising approach because it combines a reasonable tradeoff between expressiveness and complexity, and includes probabilistic models that have proved to be successful in practice (e.g. HMM).

5 Outlook

Recently, the use of Artificial Neural Networks is elaborated. E.g. a Convolutional Neural Network (CNN) for a medical phase detection is used in [LZL⁺16] by the application example of a resuscitation with only 3 phases. The training set comprises 20 workflows. Features are extracted from depth image recordings and from sound recordings. Furthermore, the model is embedded into a simple sequential model represented by thresholds which is similar to a finite automaton. Nevertheless, the model achieved an accuracy of 80%.

In [TSM⁺17] a cholecystectomy comprising 7 phases is considered. The training set comprises 80 surgical interventions. The authors use a CNN to extract features from video recordings of endoscope. The model is embedded into a HMM, which achieved an accuracy of 82%. Finally, in [LZZ⁺17] the authors elaborated the application example of a resuscitation with 35 classes and a training set of 42 video recordings. Also a CNN is used for feature extraction – in this case, the CNN is combined with a recurrent neural network using long-short term memory (LSTM). Features are extracted from depth image records, sound recordings and via a passive RFID (radio-frequency identification) system which tracks the medical instruments. The model achieves an accuracy of around 94%.

In view of these results, a further study of Artificial Neural Networks with respect to the constituents of the problem (cf. Section 2) can be considered as useful. Further elaboration is needed e.g. concerning the incorporation of expert knowledge (cf. [HML⁺16, PBFB17]) into the models, the challenge of a small amount of available data for training and the explainability of the models output [RSG16].

6 Conclusion

In this work, we discussed notable modeling approaches for a surgical phase detection. The models have different characteristics and are differently suited concerning the constituents of the problem. Considering a small amount of training data and a preferably expert-based modeling, Dynamic Bayesian Networks seems to be a suited and long-standing solution. Nevertheless, upcoming approaches, facilitating Artificial Neural Networks, have recently shown promising classification results in the field of surgical phase detection. To trim these models to be accessible to expert-based knowledge and be able to deal with the fact that typically only a small amount of real training data is available, could be a good starting point for approaches which enrich the current state of the art.

Bibliography

- [B⁺02] Falko Bause et al. *Stochastic Petri nets – An introduction to the theory*. Citeseer, 2002.
- [BPFN08] Tobias Blum, Nicolas Padoy, Hubertus Feußner, and Nassir Navab. Modeling and online recognition of surgical phases using hidden Markov models. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 627–635. Springer, 2008.
- [Bre01] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.
- [FL92] E Field and K Lohr. *Guidelines for Clinical Practice: From Development to Use*. National Academies Press, 1992.
- [Her03] C Herfarth. Lean surgery through changes in surgical work flow. *British Journal of Surgery*, 90(5):513–514, 2003.
- [HML⁺16] Zhiting Hu, Xuezhe Ma, Zhengzhong Liu, Eduard Hovy, and Eric Xing. Harnessing deep neural networks with logic rules. *arXiv:1603.06318*, 2016.

- [JL83] Joseph P Joyce and George W Lapinsky. A history and overview of the safety parameter display system concept. *IEEE Transactions on Nuclear Science*, 30(1):744–749, 1983.
- [KF09] Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.
- [KSF⁺13] Michael Kranzfelder, Christoph Staub, Adam Fiolka, Armin Schneider, Sonja Gillen, Dirk Wilhelm, Helmut Friess, Alois Knoll, and Hubertus Feussner. Toward increased autonomy in the surgical OR: needs, requests, and expectations. *Surgical endoscopy*, 27(5):1681–1688, 2013.
- [KSW⁺16] Darko Katić, Jürgen Schuck, Anna-Laura Wekerle, Hannes Kenngott, Beat Peter Müller-Stich, Rüdiger Dillmann, and Stefanie Speidel. Bridging the gap between formal and experience-based knowledge for context-aware laparoscopy. *International journal of computer assisted radiology and surgery*, 11(6):881–888, 2016.
- [KWN⁺15] HG Kenngott, M Wagner, F Nickel, AL Wekerle, A Preukschas, M Apitz, T Schulte, R Rempel, P Mietkowski, F Wagner, et al. Computer-assisted abdominal surgery: new technologies. *Langenbeck's Archives of Surgery*, 400(3):273–281, 2015.
- [LDC⁺13] Cristian A Linte, Katherine P Davenport, Kevin Cleary, Craig Peters, Kirby G Vosburgh, Nassir Navab, Pierre Jannin, Terry M Peters, David R Holmes III, Richard A Robb, et al. On mixed reality environments for minimally invasive therapy guidance: systems architecture, successes and challenges in their implementation from laboratory to clinic. *Computerized Medical Imaging and Graphics*, 37(2):83–97, 2013.
- [LZL⁺16] Xinyu Li, Yanyi Zhang, Mengzhu Li, Shuhong Chen, Farneth R Austin, Ivan Marsic, and Randall S Burd. Online process phase detection using multimodal deep learning. In *Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), IEEE Annual*, pages 1–7. IEEE, 2016.
- [LZZ⁺17] Xinyu Li, Yanyi Zhang, Jianyu Zhang, Shuhong Chen, Ivan Marsic, Richard A Farneth, and Randall S Burd. Concurrent activity recognition with multimodal CNN-LSTM structure. *arXiv preprint arXiv:1702.01638*, 2017.
- [McW17] Andrew McWilliams. Medical robotics and computer-assisted surgery: The global market. *BCC Research*, HLC036G, 2017.
- [Mol81] Michael Karl Molloy. *On the integration of delay and throughput measures in distributed processing models*. University of California, Los Angeles, 1981.
- [Mur02] Kevin Patrick Murphy. *Dynamic Bayesian networks: representation, inference and learning*. PhD thesis, University of California, Berkeley, 2002.
- [OMG11] OMG. OMG Unified Modeling Language(OMG UML) Superstructure Version 2.4.1, 2011.
- [PBF⁺08] Nicolas Padoy, Tobias Blum, Hubertus Feussner, Marie-Odile Berger, and Nassir Navab. On-line recognition of surgical activity for monitoring in the operating room. In *AAAI*, pages 1718–1724, 2008.
- [PBFB17] Patrick Philipp, Johannes Bleier, Yvonne Fischer, and Jürgen Beyerer. Towards a surgical phase detection using Markov logic networks. *Radermacher, Klaus (Ed.): CAOS*

- 2017, *17th Annual Meeting of the International Society for Computer Assisted Orthopaedic Surgery. Papers*. Online resource: June 14-17, 2017, Aachen, Germany. (*EPiC Series in Health Sciences 1*), pp. 288-294, 2017.
- [Pet62] Carl Adam Petri. *Kommunikation mit Automaten*. PhD thesis, Universität Bonn, 1962.
- [PFB17] Patrick Philipp, Yvonne Fischer, and Jürgen Beyerer. Expert-based probabilistic modeling of workflows in context of surgical interventions. *CogSima 2017, IEEE Conference on Cognitive and Computational Aspects of Situation Management*, 2017.
- [PFHB15a] Patrick Philipp, Yvonne Fischer, Dirk Hempel, and Jürgen Beyerer. Framework for an Interactive Assistance in Diagnostic Processes Based on the Translation of UML Activities into Petri Nets. In *Proceedings of ISHI 2015 – International Symposium on Health Informatics and Medical Systems: CSCI 2015. International Conference on Computational Science and Computational Intelligence*, pages 732–737. IEEE Conference Publishing Services, 2015.
- [PFHB15b] Patrick Philipp, Yvonne Fischer, Dirk Hempel, and Jürgen Beyerer. Modeling of clinical practice guidelines for interactive assistance in diagnostic processes. In *WorldComp 2015, World Congress in Computer Science, Computer Engineering, and Applied Computing : HIMS 2015, International Conference on Health Informatics and Medical Systems, July 27-30, Las Vegas, Nevada, USA*, pages 3–9. CSREA Press, 2015.
- [PFHB16] Patrick Philipp, Yvonne Fischer, Dirk Hempel, and Jürgen Beyerer. Framework for an interactive assistance in diagnostic processes based on probabilistic modeling of clinical practice guidelines. In *Emerging Trends in Applications and Infrastructures for Computational Biology, Bioinformatics, and Systems Biology*, pages 371–390. Elsevier, 2016.
- [Phi16] Patrick Philipp. Framework for modeling medical guidelines based on the translation of UML activities into YAWL. Technical report, KIT Scientific Publishing, Karlsruhe, 2016.
- [Phi17] Patrick Philipp. Combining YAWL and DBNs for surgical phase detection. Technical report, KIT Scientific Publishing, Karlsruhe, 2017.
- [PSG⁺16] Patrick Philipp, Luzie Schreiter, Johannes Giehl, Yvonne Fischer, Joerg Raczowsky, Markus Schwarz, Heinz Woern, and Jürgen Beyerer. Situation detection for an interactive assistance in surgical interventions based on dynamic Bayesian networks. *CRAS 2016, 6th Joint Workshop on New Technologies for Computer/Robot Assisted Surgery*, 2016.
- [Rei13b] Wolfgang Reisig. *Understanding Petri Nets*. Springer, Heidelberg, 2013.
- [RJ86] Lawrence Rabiner and B Juang. An introduction to hidden Markov models. *ieee assp magazine*, 3(1):4–16, 1986.
- [Rok10] Lior Rokach. *Pattern classification using ensemble methods*, volume 75. World Scientific, 2010.
- [RSG16] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Why should I trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144. ACM, 2016.

- [SGM⁺11] Earl Steinberg, Sheldon Greenfield, Michelle Mancher, et al. *Clinical Practice Guidelines We Can Trust*. National Academies Press, 2011.
- [SOP⁺14] Ralf Stauder, Ashi Okur, Loïc Peter, Armin Schneider, Michael Kranzfelder, Hubertus Feussner, and Nassir Navab. Random forests for phase detection in surgical workflow analysis. In *International Conference on Information Processing in Computer-Assisted Interventions*, pages 148–157. Springer, 2014.
- [SPG⁺16] Luzie Schreiter, Patrick Philipp, Johannes Giehl, Yvonne Fischer, Joerg Raczkowski, Markus Schwarz, Jürgen Beyerer, and Heinz Woern. Situation detection for an interactive assistance in surgical interventions based on random forests. in *Proceedings of International Journal of Computer Assisted Radiology and Surgery (CARS)*, pages 115–116, 2016.
- [TSM⁺17] Andru P Twinanda, Sherif Shehata, Didier Mutter, Jacques Marescaux, Michel de Mathelin, and Nicolas Padoy. Endonet: A deep architecture for recognition tasks on laparoscopic videos. *IEEE transactions on medical imaging*, 36(1):86–97, 2017.
- [vdA96] Wil van der Aalst. Three good reasons for using a Petri-net-based workflow management system. In *Proceedings of the International Working Conference on Information and Process Integration in Enterprises*, pages 179–201. World Scientific, 1996.
- [vdA98] Wil van der Aalst. The application of Petri nets to workflow management. *Journal of circuits, systems, and computers*, 8(01):21–66, 1998.
- [Zho12] Zhi-Hua Zhou. *Ensemble methods: foundations and algorithms*. CRC press, 2012.