

---

# A Probabilistic Relational Model for Characterizing Situations in Dynamic Multi-Agent Systems

Daniel Meyer-Delius<sup>1</sup>, Christian Plagemann<sup>1</sup>, Georg von Wichert<sup>2</sup>, Wendelin Feiten<sup>2</sup>, Gisbert Lawitzky<sup>2</sup>, and Wolfram Burgard<sup>1</sup>

<sup>1</sup> Department for Computer Science, University of Freiburg  
{meyerdel,plagem,burgard}@informatik.uni-freiburg.de

<sup>2</sup> Information and Communications, Siemens Corporate Technology  
{georg.wichert,wendelin.feiten,gisbert.lawitzky}@siemens.com

**Abstract.** Artificial systems with a high degree of autonomy require reliable semantic information about the context they operate in. However, state interpretation is a difficult task. Interpretations may depend on a history states and there may be more than one valid interpretation. We propose a model for spatio-temporal situations using hidden Markov models based on relational state descriptions, which are extracted from the estimated state of an underlying dynamic system. Our model covers concurrent situations, scenarios with multiple agents, and situations of varying durations. In this work we apply our model to the concrete task of traffic analysis.

## 1 Introduction

Continuous monitoring and understanding of the state of the system is an essential ability for autonomous agents since this allows them to act at a more intelligent level. Consider, for example, a driver assistance application, that indicates the driver if a dangerous situation is developing, or a surveillance system at a train station or an airport that recognizes suspicious behaviors. Such applications need not only to be aware of the state, but must also be able to interpret it in order to act rationally.

State interpretation, however is not an easy task. It not only requires inferring the state of the system at a given time, but also the analysis of previous states. That is, temporal context is also needed. Interpretation is also difficult since the state of a system can be associated to multiple valid interpretations. We define a *situation* as a possible interpretation for a sequence of system states and propose a framework for modeling and recognizing situations in dynamic, multi-agent and uncertain systems.

Some related approaches includes, for example, the work from Liao *et al.* (2005), which is based on relational Markov networks for sensor-based activity

recognition. Activities are represented using a *relational activity model* which describes temporal and spatial information, as well as transitions. Our approach is more related to the ones of Patterson *et al.* (2003) and Subramanya *et al.* (2006) in the sense that we also hierarchically extend a dynamic Bayesian network of the state model to jointly represent state and situation. However, our approach introduces an abstraction layer that separates the model of the world from the situation models.

## 2 Framework for Modeling and Recognizing Situations

Dynamic and uncertain systems can in general be described using dynamic Bayesian networks or DBNs (Dean and Kanazawa (1989)). DBNs consists of a set of random variables that describe the system at each point in time  $t$ . The state of the system at time  $t$  is denoted by  $x_t$  and  $z_t$  represents the obtained sensor measurements. The dynamics between two states  $x_{t-1}$  and  $x_t$  are described by the conditional probability  $p(x_t|x_{t-1})$ . This transition model describes the probability of  $x_t$  being the current state given that  $x_{t-1}$  has been the previous state. It is assumed that observations  $z_t$  at time  $t$  depend only on the current state:  $p(z_t|x_t)$ . This conditional distribution is called the sensor model and describes how the observations are affected by the actual state of the system.

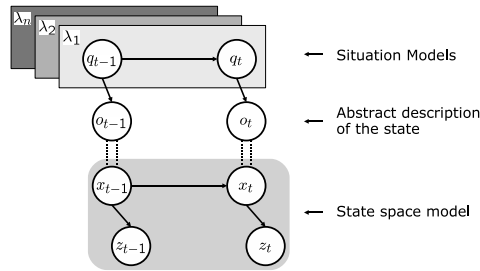
Our proposed framework for modeling and recognizing situations, which is described in the following sections, introduces an abstraction layer between the continuous state space model and the situations models. This layer avoids an explosion in the complexity of the DBN that would be needed to model situations by use of additional state variables.

The abstraction layer also provides modularity to the framework. Situation tracking and state estimation are handled separately. The system is modeled using a DBN and Bayes filtering is used for state estimation. Bayes filters address the problem of estimating the state  $x_t$  of a dynamical system from sensor measurements. The key idea is to recursively estimate the posterior probability density over the state  $x_t$  at time  $t$  from the gathered observations. Among the concrete implementations of the Bayes filter we find Gaussian filters like the Kalman filter and nonparametric filters like the particle filter.

Figure 1 depicts our proposed framework for situation recognition. At each time step  $t$ , an abstract representation of the estimated state is generated. A set of situation models is then evaluated against this abstract representation. If a model assigns a positive probability to the abstract representation of the state, then an instance of the corresponding situation occurs.

## 3 Modeling Situations

Based on the DBN model of the system presented in the previous section a situation can be defined as a sequence of states  $x_{t:t+n}$  with a meaningful



**Fig. 1.** Overview of the framework.

interpretation. The state  $x_t$  of the system is an assignment of values to the state variables that describe the system at any given point in time  $t$ . For the purpose of modeling and recognizing situations, the state space of the system will be represented at a more abstract level and reasoning about state sequences will be lifted to this abstract space.

### 3.1 Relational State Representation

For the abstract representation of the state of the system, relational logic will be used. In relational logic, an *atom*  $r(\tau_1, \dots, \tau_n)$  is a  $n$ -tuple of terms  $\tau_i$  with a relation symbol  $r$ . A *term* can be either a variable  $R$  or a constant symbol  $r$ . Relations can be defined over the state variables or over features that can be directly extracted from them. Table 1 illustrates possible relations defined over the *distance* and *bearing* state variables in a traffic scenario.

**Table 1.** Example distance and bearing relations for a traffic scenario.

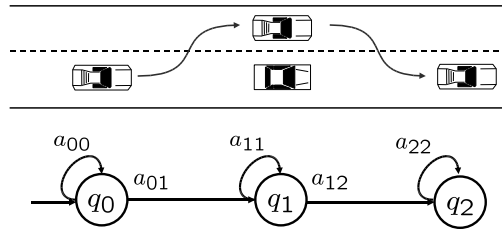
Relations over Distance		Relations over Bearing	
<code>equal(R, R')</code>	[0 m, 1 m)	<code>in_front_of(R, R')</code>	[315°, 45°)
<code>close(R, R')</code>	[1 m, 5 m)	<code>right(R, R')</code>	[45°, 135°)
<code>medium(R, R')</code>	[5 m, 15 m)	<code>behind(R, R')</code>	[135°, 225°)
<code>far(R, R')</code>	[15 m, ∞)	<code>left(R, R')</code>	[225°, 315°)

An *abstract state* is a conjunction of logical atoms (see also Cocora *et al.* (2006)). Consider for example the abstract state  $q \equiv \text{far}(R, R'), \text{behind}(R, R')$ . It represents all states in which a given car is *far* and *behind* another given car. An instance of an abstract state is a grounded abstract state, that is, an abstract state that contains no variables, for example  $\text{far}(c_0, c_1), \text{behind}(c_0, c_1)$ , where  $c_0$  and  $c_1$  are two constant symbols that represent two different cars.

### 3.2 Situation Models

For modeling situations, Hidden Markov models or HMMs (Rabiner (1989)) are used to describe the sequences of abstract states that correspond to a

given situation. HMMs are temporal probabilistic models for analyzing and modeling sequential data, and represent the state of the system using a single discrete random variable. In our framework we use HMMs whose states correspond to conjunctions of relational atoms, that is, abstract states as described in the previous section. The state transition probabilities of the HMM specify the allowed transitions between these abstract states. In this way, HMMs specify a probability distribution over sequences of abstract states.



**Fig. 2.** *passing* maneuver and corresponding HMM.

To illustrate, consider a *passing* maneuver like the one depicted in Figure 3.2 where a reference car is passed by a faster car on the left hand side. The maneuver could be coarsely described in three steps: the passing car is behind the reference car, it is left of it, and it finally is in front. Using the relations presented in Table 1, an HMM that describes this sequences could have three states, one for each step of the maneuver:  $q_0 = \text{behind}(\mathbf{R}, \mathbf{R}')$ ,  $q_1 = \text{left}(\mathbf{R}, \mathbf{R}')$ , and  $q_2 = \text{in\_front\_of}(\mathbf{R}, \mathbf{R}')$ . The transition model of this HMM is depicted in Figure 3.2. It defines the allowed transitions between the states. Observe how the HMM specifies that when in the second state ( $q_1$ ), that is, when the passing car is left of the reference car, it can only remain left ( $q_1$ ) or move in front of the reference car ( $q_2$ ). It is not allowed to move behind it again ( $q_0$ ). Such a sequence would not be a valid *passing* situation according to our description.

Observe too that the states of the HMM correspond to abstract states, that is, relational descriptions containing variables. In order to recognize a concrete instance of a situation, the variables in the abstract states need to be grounded. In the example above, the variables in the abstract states  $q_0$ ,  $q_1$ , and  $q_2$  have to be replaced by the constants that represent the two vehicles involved. The process of instantiating and tracking situation HMMs is detailed in the next section.

## 4 Recognizing Situations

The idea behind our approach for tracking situations is to find, at each time step  $t$ , the situation HMMs that explain the current state of the system. If

a situation HMM assigns a positive probability to the current state of the system, then an instance of the corresponding situation is said to occur.

The general algorithm is as follows. At every time step  $t$  the state of the system  $x_t$  is estimated and is represented as a conjunction of grounded relational atoms with an associated probability. Based on this relational description, all situation HMMs are grounded, that is, the variables in the abstract states of the HMM are replaced by constant terms. If a grounded HMM assigns a non-zero probability to the current relational description of the system state, the situation HMM can be instantiated. Once a situation HMM has been instantiated Bayes filtering is used to update its internal state. As long as the situation HMM assigns a positive probability to the sequence of system states, in other words, as long as it is a consistent sequence, the instance of that particular situation is considered active. Thus, at each time step  $t$  the algorithm keeps track of a set of active situation hypothesis, based on the sequence of relational descriptions of the estimated system state.

#### 4.1 Representing Uncertainty at the Relational Level

At each time step  $t$ , our algorithm estimates the state  $x_t$  of the system. The estimated state is usually represented through a probability distribution which assigns a probability to each possible hypothesis about the true state. In order to be able to use the situation HMMs to recognize situation instances, we need to represent the estimated state of the system as a grounded abstract state using relational logic.

A fundamental question is how to convert the uncertainties related to the estimated state  $x_t$  into appropriate uncertainties at the relational level. We assign to each relation the probability mass associated to the interval of the state space that it represents. The resulting distribution is thus a histogram that assigns to each relation a single cumulative probability. Such a histogram can be thought of as a piecewise constant approximation of the continuous density. The relational description  $o_t$  of the estimated state of the system  $x_t$  at time  $t$  is then a grounded abstract state where each relation has an associated probability.

The probability  $P(o_t|q_i)$  of being in a grounded abstract state  $q_i$  and observing  $o_t$  is computed as the product of the matching terms in  $o_t$  and  $q_i$ . In this way, the observation probabilities needed to estimate the internal state of the situation HMMs and the likelihood of a given sequence of observations  $O_{1:t} = o_1, \dots, o_t$  can be computed.

#### 4.2 Situation Model Selection using Bayes Factors

The algorithm for recognizing situations keeps track of a set of active situation hypothesis at each time step  $t$ . We propose to decide between models at a given time  $t$  using Bayes factors for comparing two competing situation HMMs that explain the given observation sequence. Bayes factors (Kass and

Raftery (1995)) provide a way of evaluating evidence in favor of a probabilistic model as opposed to another one and can be used to compare two competing situation models. The Bayes factor  $B_{1,2}$  for two competing models  $\lambda_1$  and  $\lambda_2$  is computed as

$$B_{12} = \frac{P(\lambda_1|O_{1:t})}{P(\lambda_2|O_{1:t})} = \frac{P(O_{1:t}|\lambda_1)P(\lambda_1)}{P(O_{1:t}|\lambda_2)P(\lambda_2)},$$

that is, the ratio between the likelihood of the models being compared given the data. The Bayes factor can be interpreted as evidence provided by the data in favor of a model as opposed to another one. One possible interpretation for the Bayes factor is suggested by Jeffreys (1961).

In order to use the Bayes factor as evaluation criterion, the evidence, that is, the observation sequence  $O_{1:t}$  must be the same for the two models being compared. To solve this problem we propose a solution used in Bioinformatics for sequence alignment (Durbin et al. (1998)) and extend the situation model with a simple *world* model to account for the missing part of the observation sequence. This *world* model is analogous to the bigram models that are learned from the corpora in the field of natural language processing (Manning and Schütze (1999)). By using the extended situation model, we can use Bayes factors to evaluate two situation models even if they were instantiated at different points in time.

## 5 Experimental Results

Our framework was implemented and tested in a traffic scenario using a simulated 3D environment. *Gazebo*, a 3D extension of *Player/Stage* (Collett *et al.* (2005)) was used as simulation environment. The scenario consisted of several autonomous vehicles with simple behaviors and one reference vehicle controlled by a human operator. When tracking situations, random noise was added to the pose of the vehicles to simulate uncertainty at the state estimation level. The goal of the experiments is to demonstrate how our framework can be used to model and successfully recognize different situations in dynamic multi-agent environments.

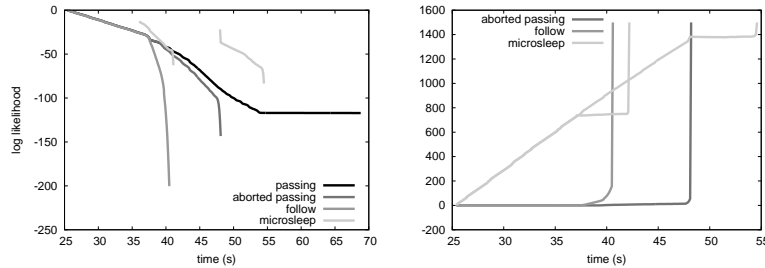
Four different situations relative to a reference car were considered:

1. The *follow* situation describes the reference car driving behind another car on the same lane.
2. The *passing* situation describes the reference car starting behind another car on the same lane, passing it on the left side, and ending up in front of it on the same lane.
3. The *aborted passing* situation is analogous to the *passing* situation, but instead of ending up in front of the other car, the reference car ends up behind the other car again.

4. The *microsleep* situation describes an uncorrected drift (to the left or to the right) from the straight driving condition.

The structure and parameters of the corresponding situation HMMs were defined manually. The relations considered for these experiments were defined over the distance and position of the cars relative to the reference vehicle. The absolute orientation of the car was also considered for the definition of the *microsleep* situation.

Figure 5 (right) plots the likelihood of the observation sequence for a passing maneuver (see the example in Section 3.2) according to the four different situation models. Observe how the algorithm successfully instantiated and tracked the different instances of the different situation models. For example, in the figure it can be seen how two different instances of the *microsleep* situation were instantiated: (1) the reference car changes to the left lane (time  $\sim 32$ s), and (2) it changes back to the initial lane (time  $\sim 48$ s). Observe that when a situation model stops providing a good explanation for the observation sequence, the likelihood starts to decrease rapidly until it becomes zero and the instance of the situation is deleted. For example, the *follow* situation expects the reference car to remain behind the other car. When the reference car changes to the left lane, the likelihood of the observation sequence with respect to the *follow* situation starts to decrease rapidly. Similarly, when the reference car changes to the right lane in front of the other car, the likelihood of the observation sequence with respect to the *aborted passing* situation also starts to decrease.



**Fig. 3.** (Left) Likelihood of the observation sequence for a passing maneuver according to the different situation models, and (right) Bayes factor in favor of the *passing* situation model against the other situation models.

The Bayes factor in favor of the *passing* situation model compared against the other models is depicted in Figure 5 (left). Relative to the *passing* situation model, the other models provide a worse explanation for the observations, and the evidence in favor of the *passing* situation model, increases. When both models provide an equally likely explanation, the evidence in favor of a model ceases to increase. Figure 5 (left) shows how Bayes factors can be

used to make decisions between competing situation models by using some meaningful interpretation of the values.

## 6 Conclusions and Further Work

We presented a general framework for modeling and recognizing situations. Our approach uses a relational description of the state space and hidden Markov models to represent situations. An algorithm was presented to recognize and track situations in an online fashion. The Bayes factor was proposed as evaluation criterion between two competing models. Using our framework, many meaningful situations can be modeled. Experiments demonstrate that our framework is capable of tracking multiple different situation hypothesis in a dynamic multi-agent environment.

## References

- COCORA, A. and KERSTING, K. and PLAGEMANN, C. and BURGARD, W. and DE RAEDT, L. (2006): Learning Relational Navigation Policies. *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.
- COLLETT, T. and MACDONALD, B. and GERKEY, B. (2005): Player 2.0: Toward a Practical Robot Programming Framework. *In: Proceedings of the Australasian Conference on Robotics and Automation (ACRA 2005)*.
- DEAN, T. and KANAZAWA, K. (1989): A Model for Reasoning about Persistence and Causation. *Computational Intelligence*, 5(3):142-150.
- DURBIN, R. and EDDY, S. and KROGH, A. and MITCHISON, G. (1998): *Biological Sequence Analysis*. Cambridge University Press.
- JEFFREYS, H. (1961): *Theory of Probability (3rd ed.)*. Oxford University Press.
- KASS, R. and RAFTERY, E. (1995): Bayes Factors. *Journal of the American Statistical Association*, 90(430):773-795.
- LIAO, L. and FOX, D. and KAUTZ, H. (2005): Location-Based Activity Recognition using Relational Markov Networks. *In Proc. of the International Joint Conference on Artificial Intelligence (IJCAI-05)*.
- MANNING, C.D. and SCHÜTZE, H. (1999): *Foundations of Statistical Natural Language Processing*. The MIT Press.
- PATTERSON, D. J. and LIAO, L. and FOX, D. and KAUTZ, H. (2003): Inferring High-Level Behavior from Low-Level Sensors. *In Proc. of the International Conference on Ubiquitous Computing (UBICOMP-03)*.
- RABINER, L. (1989): A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257-286.
- SUBRAMANYA, A. and RAJ, A. and BILMES, J. and FOX, D. (2006): Recognizing Activities and Spatial Context Using Wearable Sensors. *In Proc. of Conference on Uncertainty in AI (UAI)*.