A Probabilistic Approach for Context Reasoning

Jörg Roth

Univ. of Applied Sciences Nuremberg
D 90489 Nuremberg, Germany
Joerg.Roth@Ohm-hochschule.de

Abstract: In this paper we present an approach for reasoning about continuous context variables. We introduce a probabilistic mechanism based on efficient geometric structures that avoid typical restrictions of existing approaches. Especially, we can model non-Gaussian distributions, negated statements and we can deal with external knowledge that is only accessible on demand.

1. Introduction

Many context variables are continuous quantities that can be described by a time-dependent numerical value. Typical examples are a user's heart beat rate, current geographic coordinates or the outside temperature. If such variables are not mapped to discrete symbolic values, reasoning cannot base on classical logic as introduced by artificial intelligence approaches. As such variables have to be measured with a certain measurement error, probabilistic approaches are thus appropriate.

Many probabilistic approaches have certain demands on the measurement error distribution. Usually they assume that all distributions are Gaussian and relations between two states can easily be expressed (e.g. by matrices). However, these assumptions often are not true in common context scenarios.

2. Related Work

Probabilistic approaches usually are based on the following consideration: given a certain state vector – what is the probability to get the specific list of measurements [DFG01]? In the case of continuous random variables, the probability of any single discrete event is in fact 0. Thus, the probabilities of all possible context states are represented by a *probability density function*. According to Bayes' rule, multiple measurements at a single point in time are processed using the *multiplication* of the corresponding densities. If a measurement relates two points in time, we use the *convolution* of densities. With multiplication and convolution, we can model most required probabilistic computations. Two existing approaches perform reasoning about context variables based on this consideration: Kalman filters and Particle filters.

The Kalman filter assumes a context state vector with arbitrary dimensions [DMC00, Kal60]. The state is unknown, but Gaussian distributed measurements indirectly reflect information about the state. Further, relations between two states (at two points in time) can easily be described by a matrix. A result of a reasoning step is expressed by a mean (the most probably state) and an error covariance matrix. The Kalman filter computations can be simplified to few matrix multiplications and one matrix inversion that even run on small computers or embedded systems.

Particle filters [HB04] use a set of particles; each presents a specific potential state. A particle contains a state vector and a weight, which reflects the probability density for this state, approximated by a Dirac delta density. Particle filters support a huge variety of densities. Increasing numbers of particles improve the precision, but also increase the required memory and processing time.

Both approaches have their drawbacks. They have certain demands on the measurement error distribution. Moreover, they cannot model negated knowledge about the state and assume the ultimate knowledge about the context to be available locally. These assumptions are often not true in context aware scenarios.

3. The Reasoning Approach

Our approach is based on the following assumptions: first, a single reasoning step processes up to two continuous context variables; each of it is time-dependent. A complete reasoning process may have multiple reasoning steps. Second, each variable is unrestricted, i.e. we cannot generally assume maximum or minimum values. As a consequence, we cannot easily use grids. Third, the context variables can effectively be measured. This means, we cannot deal with variables such as "the level of happiness" or "the current level of distraction" that only can be indirectly measured or even only estimated.

Typical examples for our intended context variables:

– Blood pressure and heart beat rate of a certain person are related quantities that allow reasoning about the current cardiovascular state. We can formulate certain relations between these quantities as well as relations between these quantities and the current time. We can effectively measure these context variables.
– The latitude and longitude values of the user's current geographic position are one of the most important context variables. We are able to collect several pieces of knowledge about the position. E.g., we may know that a certain position has to be on a road or we expect a position to reside inside a certain mobile phone cell. We can use reasoning to derive the most probable position based on many pieces of position knowledge.
– Quantities such as the outside temperature, humidity, noise level etc. are further context variables of our intended type.

Knowledge about a context state can either base on *measurements* or be *a-priori knowledge*. In addition, knowledge can describe *single points* in time or the relation between *two points in time*. We get four different types as presented in table 1.

Table 1: Types of knowledge about a context with examples

	measurements	**a-priori knowledge**
single point in time	current blood pressure; current GPS position	cars drive on roads; blood pressure is always lower than 250 mmHg
relations between two points in time	odometers that measure the driving distance	pedestrians have a maximum speed of 5km/h; the temperature does not drop more than 10^{o}C per hour

Note that the type *measurements/two points in time* currently is not considered in our approach, as context alteration between two points in time usually are not explicitly measured but are mapped to two single-point measurements.

Based on these pieces of knowledge we derive four types of *predicates*:

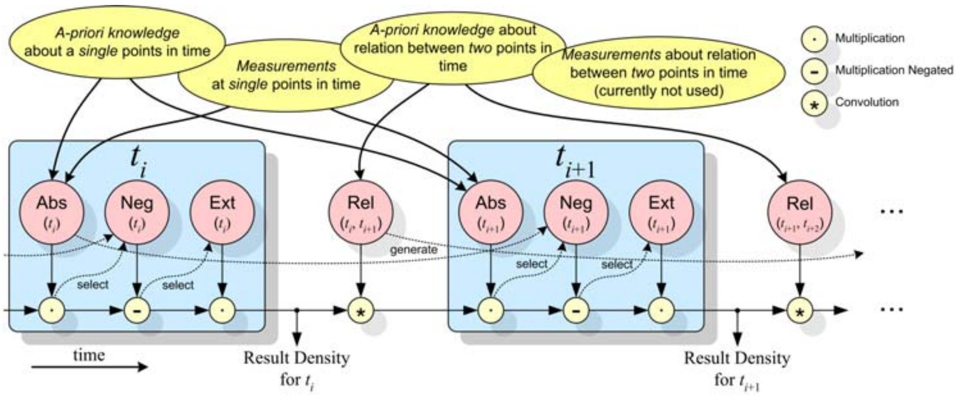
– *Absolute* predicates are e.g., "the position *is x*/*y*" or "the blood pressure *is x* mmHg".

– *Negative* predicates are e.g., "the position is *not* at home". Negative predicates are derived from former positive predicates. E.g. if a predicate once is true and a further measurement does not indicate this predicate any longer, the negation automatically is assumed.

– *External* predicates are absolute predicates that have to be externally downloaded from other databases at processing-time. E.g., the user drives on a street, but the map of the current position first has to be loaded from a huge street database. External predicates are looked up with the help of former results. E.g., only those road information is looked up, that cover an area with a probability greater than 0. As a benefit, not the world-wide roadmap has to be loaded, but only the roads in small areas.

– *Relative* predicates relate two points in time and are directly derived from a-priori knowledge (table 1, lower right).

Fig. 1 shows the connections between pieces of context knowledge and the respective predicates.



A-priori knowledge about a single points in time
Measurements at single points in time
A-priori knowledge about relation between two points in time
Measurements about relation between two points in time (currently not used)
Multiplication
Multiplication Negated
Convolution
t_i
t_{i+1}
Abs (t_i)
Neg (t_i)
Ext (t_i)
Rel (t_i, t_{i+1})
Abs (t_{i+1})
Neg (t_{i+1})
Ext (t_{i+1})
Rel (t_{i+1}, t_{i+2})
generate
select
select
select
select
...
...
time
Result Density for t_i
Result Density for t_{i+1}

Fig. 1: Data flow for the reasoning process

In addition to the two density operations *multiplication* and *convolution*, we introduced a third one: *multiplication negated* (see below). These three operations have to be effectively implemented. In our approach we represent densities with the help of *multipolygons with holes* (*mph*) that are common approximating geometric two-dimensional structures. An mph contains a list of polygons; each represents a coherent part of the surface. Each of it may contain polygons that represent the holes in the surface. We represent a density *f* with the help of mphs as follows

$$f(p) \approx \hat{f}(p) = \sum_{i=1}^{n} w_i \Lambda(p, mph_i) \qquad (1)$$

In this equation *p* describes a two-dimensional state vector (considered as a geometric point in a two-dimensional area), *n* denotes the number of areas that approximate the density, w_i denotes a constant weight of an area, mph_i denotes the geometric description of an area and Λ the characteristic function of an mph, i.e. $\Lambda(p, mph) = 1$ if $p \in mph$, 0 otherwise. The variables *n* and w_i are defined by the respective application and have to balance between the precision of the approximation and required computational time.

To be a density, $\hat{f}$ has to cover a volume of one. We further require $mph_i \cap mph_j = \varnothing$ for every $i \neq j$. We get an ordered list of areas by their weights. This requirement leads to an efficient implementation especially of the density multiplication and to an efficient resampling operation. Fig. 2 illustrates the density representation.

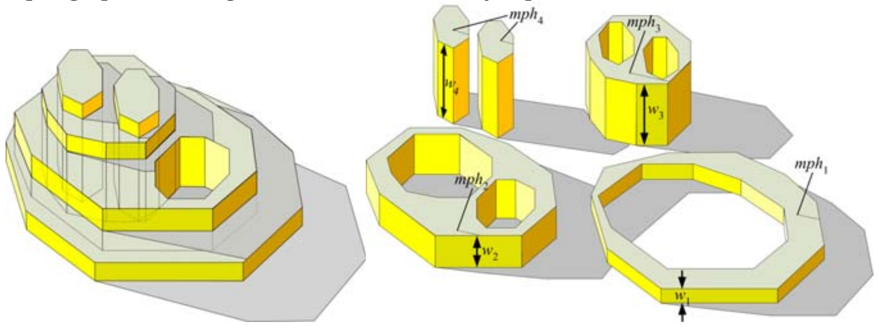


Fig. 2: Density approximation (left) and parts of the approximation (right)

With the help of this density representation we now can implement the required density operations. Here, we only provide the ideas of the most important operations:

Multiplication: For two approximated densities we use the equation

$$\begin{aligned}\hat{f}_1 \cdot \hat{f}_2 &= \hat{c} \cdot \left(\sum_{i=1}^{n_1} w_{1i} \Lambda(mph_{1i}) \right) \cdot \left(\sum_{j=1}^{n_2} w_{2j} \Lambda(mph_{2j}) \right) \\ &= \hat{c} \cdot \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} w_{1i} w_{2j} \Lambda(mph_{1i} \cap mph_{2j})\end{aligned} \qquad (2)$$

where $\hat{c}$ is a normalization factor. Only those mph_{1i}, mph_{2j} that overlap can contribute to the result, thus an efficient algorithm first tests this, before the actual intersection is computed. As the overlapping test knows efficient implementations (e.g. using bounding boxes), this approach is reasonable.

Multiplication negated: we only consider the negation of predicates that have a unique density inside a finite area and a zero density outside (a usual case). Let f_1 be a density and f_2 a density that should be negated. Then

$$f_{result} = c \cdot f_1 \cdot \bar{f}_2 \text{ where } \bar{f}_2(p) = \begin{cases} 1 & f_2(p) = 0 \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

Here, c is the normalizing factor. Note that $\bar{f}_2$ is not a density, as it does not produce an integral of one. Thus, we explicitly mark a density as negated and store the original non-negated density.

Convolution: We currently only convolve an arbitrary density with a circular density c, with a unique density inside the circle with radius r and 0 outside (which, e.g., describes a maximum movement per time). The convolution then simplifies to

$$\begin{aligned} \int \hat{f}(t_1, p - p_\Delta) \cdot c(p_\Delta, r) dp_\Delta &= \int \left(\sum_{i=1}^{n} w_i \Lambda(p - p_\Delta, mph_i) \right) \cdot c(p_\Delta, r) dp_\Delta \\ &= \sum_{i=1}^{n} w_i \left(\int \Lambda(p - p_\Delta, mph_i) \cdot c(p_\Delta, r) dp_\Delta \right) \end{aligned} \tag{4}$$

This means, we have to compute the inner integral and simply create a sum. The inner integral can be numerically approximated using the geometric *buffer* function [OGC06]. Table 2 shows all required density operations and the corresponding geometric mph operations used for the realization.

Table 2: Mapping of density operations to geometric operations

	mph ops							
density ops	*point in area*	$\cup$	$\cap$	\	*surface area*	*buffer*	*cen-troid*	*dist.*
prob. of pos.	×							
resample density		×			×			
multiplication			×					
convolution		×	×	×	×	×	×	
multiplic. negated			×	×				
centroid of density					×		×	
maxima of density		×					×	×
area with prob.>0		×						

A simulation illustrates our approach. The two-dimensional context state variables are latitude and longitude of the user's current geographic position (fig. 3). A walker promenades at a lakeside. Every minute, his mobile device tries to receive GSM and WLAN cell information. We assume the device can receive two GSM cells and one WLAN hotspot. In addition, we know that the user cannot reside inside the lake (e.g. does not row a boat). The knowledge about the position undergoes our reasoning process. In summary, the process executed 2 convolutions, 3 multiplications and 5 multiplications negated. Fig. 3 (bottom) shows the result. After 3 minutes a density represents the knowledge about the current position. We now can easily compute the most probable position (marked by an arrow).

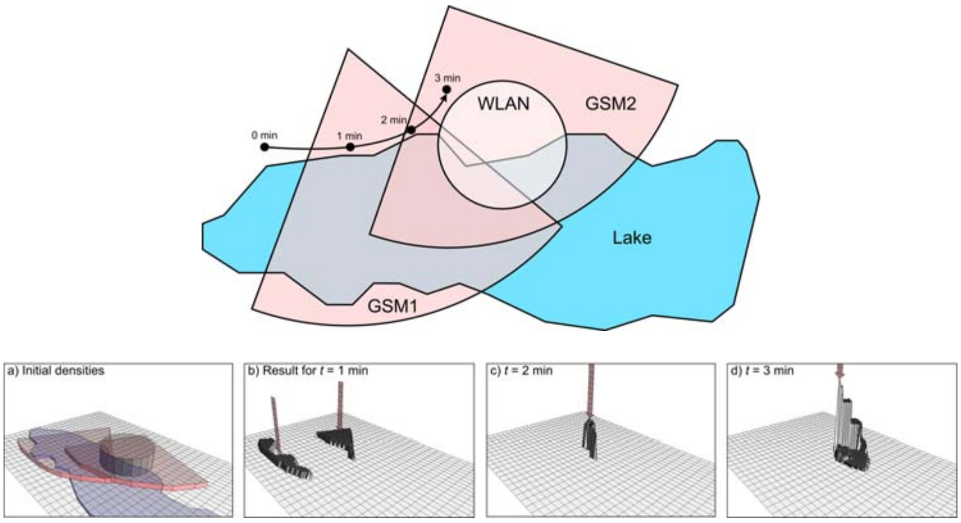


Fig. 3: A simulated scenario (top) and simulation results (bottom)

4. Summary

Our approach allows to effectively reason about continuous context variables with the help of a probabilistic mechanism. It heavily makes use of geometric operations widely available and efficiently implemented in many tool environments, software libraries and spatial databases. In particular, all required density operations (especially multiplication, multiplication negated and convolution) can be mapped to geometric operations.

References

[DFG01] Doucet, A., de Freitas N., Gordon N. (eds): Sequential Monte Carlo in Practice, Springer-Verlag, New York, 2001

[DMC00] Drolet, L., Michaud F., Côté J.: Adaptable sensor fusion using multiple Kalman filters. Proc. IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS), Takamatsu, Japan, 2000

[HB04] Hightower, J., Borriello G.: Particle Filters for Location Estimation in Ubiquitous Computing: A Case Study, in Proc. of the 6th Int. Conference on Ubiquitous Computing (Ubicomp 2004), Sep. 2004, 88-106

[Kal60] Kalman, R.: A new approach to linear Filtering and prediction problems. Transactions ASME Journal of Basic Engineering 82 (1960), 35-44

[OGC06] Open Geospatial Consortium Inc.: OpenGIS® Implementation Specification for Geo–graphic information - Simple feature access - Part 1: Common architecture & Part 2: SQL option. John R. Herring (ed.), 2006

[Roth07] Roth, J: Inferring Position Knowledge from Location Predicates, 3rd International Symposium on Location- and Context-Awareness (LoCA 2007) 20.-21. Sept. 2007, Oberpfaffenhofen, LNCS 4718, Springer Verlag, 245-262