Contents

Document history

Version	Date	Author (Unit)	Description
v0.3	1/9 2014		First draft finished

1 Executive summary

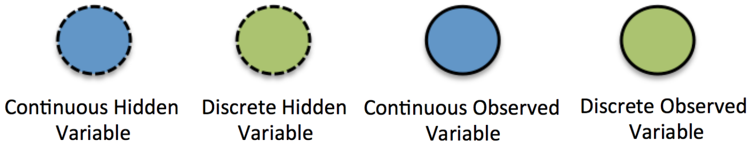

Continuous Hidden Variable
Discrete Hidden Variable
Continuous Observed Variable
Discrete Observed Variable

Figure 1: PreliminariesNotation

2 Preliminaries

2.1 Dynamic Bayesian Networks

2.2 Data analysis

As already commented in the introduction, the data analysis detailed here will be used to test some assumptions supporting the models elicited by the experts in the different use cases, and also to complement our understanding about the nature of the problem we are modelling. The set of tools employed for this purpose allow us to get insights about some simple and basic aspects of the structural and the distributional assumptions present in a dynamic Bayesian network (DBN).

Structural assumptions: sample correlograms and partial correlograms

A DBN mainly aims to model complex multivariate time series. By using sample correlogramas and sample partial correlograms, we will try to test if the available data supports the temporal correlation between variables assumed by the DBN model, i.e., the temporal links between variables. However, these tools will only allow us to look at univariate time series, what strongly limits the reach of the extracted conclusions. But, at least and as we will see later in the different use-cases, this analysis will give us some interesting insights which usually can not be elicited from experts.

Sample correlogram: Let $x_1, ..., x_T$ be a univariate time series. The *sample autocorrelation coefficient* at lag v is given by

$$\hat{\rho}_v = \frac{\sum_{t=1}^{T-v}(x_t - \bar{x})(x_{t+v} - \bar{x})}{\sum_{t=1}^{n}(x_t - \bar{x})^2}$$

where $\bar{x}$ is the sample mean. The plot of $\hat{p}_v$ versus v for $v = 1, ..., M$ for some maximum M is called the *sample correlogram* of the data.

Sample partial correlogram: Let us denote by X_t to the random variable associated to X taking values at time t. We can build the following regression problem:

$$X_t = a_0 + a_1 X_{t-1} + a_2 X_{t-2} + ...a_{v-1} X_{t-v-1}$$

> In addition, let $e_{i,v}$ denotes the residuals of this regression problem (i.e., the error when estimating X_t using a linear combination of $v-1$ previous observations). The *sample partial autocorrelation coefficient* of lag v, denoted as $\hat{\theta}_v$, is the standard sample autocorrelation between the variable X_{t-v} and these residuals. Intuitively, the sample partial autocorrelation coefficient of lag v can be seen as the correlation between X_t and X_{t+v} after having removed the common *linear* effect of the data in between.

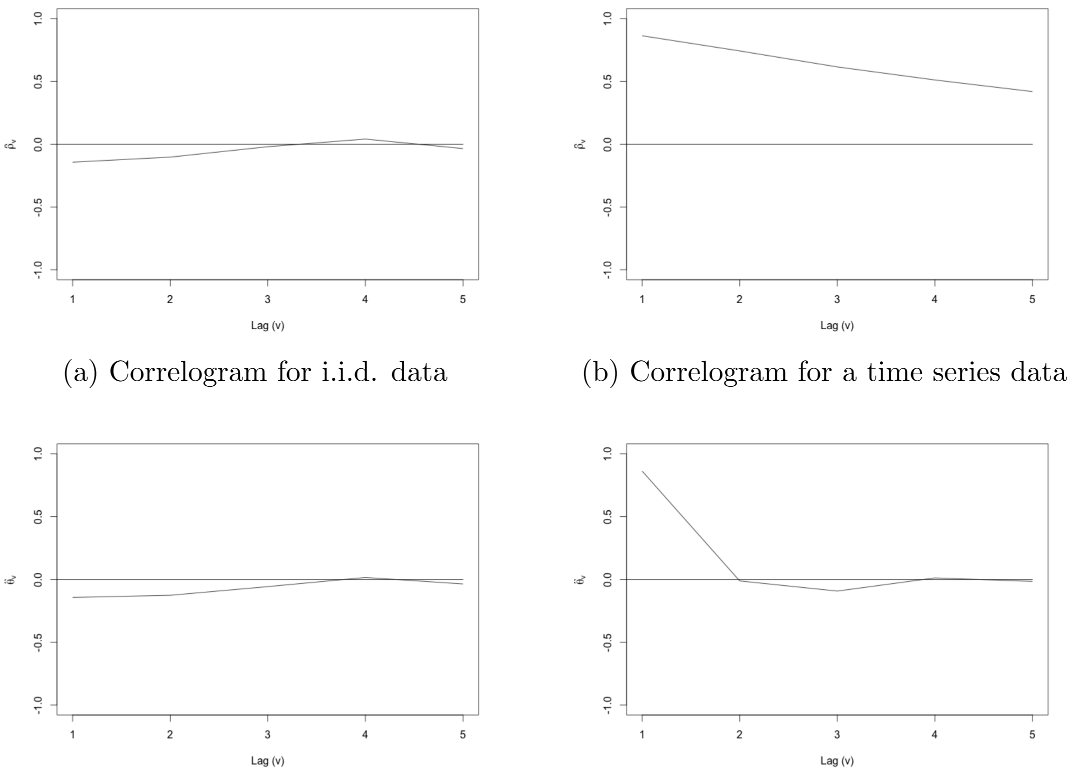
Sample correlograms can be interpreted as a way to measure the strength of the following unconditional dependences: $X_t \not\perp X_{t+v}$ for some lag $v \geq 1$. When $\hat{\rho}_v$ is close to zero, this strongly indicates that there is an unconditional independence between X_t and X_{t+v}. However, when $\hat{\rho}_v$ is close to either 1 or -1, this strongly indicates that there is a correlation or dependency between X_t and X_{t+v}. But, again, we should never forget that we are making linear relationships and normality assumptions.

Figure 2 shows an example of how a sample correlogram looks like for two kind of data sets: First, a sequence of 50 data samples i.i.d. according to a Gaussian distribution with zero mean and unit variance $x_t \sim N(0,1)$ (see Figure 2(a)); and second, a sequence of 50 data samples distrubuted as $x_t = x_{t-1} + \epsilon$, $x_0 = \epsilon$, such that $\epsilon \sim N(0,1)$ (see Figure 2(b)). As can be seen, for the i.i.d. data the correlogram has always values close to zero for all the lags. However, for time series data, the correlogram clearly identifies the presence of a temporal relationship in the data. As expected, the correlation decreases with the size of the lag, and how quickly it decreases depends on the strength of the temporal relationship.

Similarly, we plot in Figure 2(c) and in Figure 2(d) the sample partial correlograms for the same two data sequences presented above. In the case of i.i.d. data, we can see again that the partial correlogram does not show any sign of partial correlation between the data sequence samples. However, for time series data, the partial correlogram takes a high value for $v = 1$ and then is null for v higher than 1. Sample partial correlogram can be interpreted as a way to measure the strength of the following conditional dependence: $X_t \not\perp X_{t+v} | X_1, ..., X_{t+v-1}$ for some lag $v \geq 1$. Accordingly, the sample partial correlogram correctly identifies that we have the following conditional independencies: $X_t \perp X_{t+2} | X_{t+1}$ in the considered time series data. Therefore, sample partial correlogram can be seen as a tool to test the order of the Markov chain generating a time data sequence, with all the same caveats expressed for the sample correlogram.

Distributional assumptions: Histograms and Bivariate Distributions

With the tools described in this section we tried to get insights about the conditional distribution probabilities of the proposed models. The first basic tool that we employed



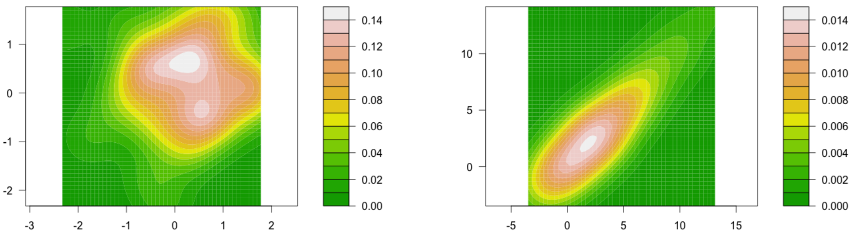
(a) Correlogram for i.i.d. data

(b) Correlogram for a time series data

(c) Partial correlogram for i.i.d. data

(d) Partial correlogram for a time series data

Figure 2: Examples of sample correlograms and sample partial correlograms for i.i.d. and time series data.



(a) i.i.d. data (b) Time series data

Figure 3: Bivariate contour plots for a set of i.i.d. data and for a time series data.

for this purpose was the histograms. However this tool, although quite useful in a static context, is quite limited in dynamic models. For example, let assume we have a time series $x_1, \ldots, x_T$ and our histogram shows that the empirical distribution of the variable when we aggregate the data samples over time looks like a mixture of Gaussian distributions. There are two simple possibilities that can give rise to this finding: i) X_t is distributed according to a mixture of Gaussians where each Gaussian component depends on X_{t-1}; ii) there is a discrete hidden variable that influences X_t but which is not observed and it is the one that generates the different mixture components. As shown in this example, histograms are difficult to interpret in dynamic models, but we are going to use them when we think that they can be of some help.

The other tool that we are going to use to get insights about the conditional distributions of the model is the contour plot of the empirical bivariate distribution of X_t versus X_{t-1}. These contour plots can show many relevant information such as if there are linear relationships between variables or if we can assume they are normality distributed, etc. In Figure 3, we plot the bivariate contour plot for the i.i.d data and the time series data previously employed when describing the sample correlograms. As can be seen, the bivariate contour plot of the time series clearly show a linear relationship between X_t and X_{t-1} and how they can be assumed to be distributed according to a bivariate normal with a covariance that displays some degree of correlation. In the case of i.i.d. data, the contour plot looks like quite different.

3 Preliminary Models

3.1 Daimler Models

Daimler use-case is based on two application scenarios [**?**]: i) early recognition of a lane change manoeuvre; and ii) earlier prediction of the need for a lane change based on relative dynamics between two vehicles driving in the same lane at different speeds.

The first scenario has been previously addressed by Daimler [?]. The main result of this previous work was a static object oriented Bayesian network [?] able to detect a manoeuvre 0.6s before execution. The goal now is to extend the prediction horizon for manoeuvre recognition at least 1-2 seconds to further improve the quality of the on board adaptive cruise control. As we will explain later, this improvement is expected to be achieved by a dynamic extension of this previously proposed static model. We built on this previously proposed static model for two main reasons: although with a limit prediction horizon, this static model has proven to be very robust for this task and it is considered to be the gold-standard for this problem in Daimler; and, also, because the developed models in this project are expected to be integrated in a ECU [?], and the advances made about this respect for the static model [?] can be exploited during the integration in the ECU of their dynamic counterparts.

3.1.1 Early recognition of a lane change manoeuvre

The basic settings of this application scenario are as follows. Let us suppose we are driving our car, which will be referred to as the EGO vehicle, in a highway. This EGO vehicle is equipped with a video camera, radar and some on-board sensors. Using the data provided by these sensors, the challenge consists on making an early recognition of a manoeuvre either by the EGO or another relevant car in the traffic scene (OBJ). In total, the system is expected to recognise the following set of manoeuvres (a visual description of them is given below in Figure 4):

1. **Object-CutIn**: A vehicle is moving to the lane where the EGO vehicle is placed.
2. **Object-CutOut**: A vehicle that was driving in front of the EGO is leaving the EGO's lane.
3. **Object-Follow**: There is no lane change. The EGO is driving and there is some other vehicle in front.
4. **Lane-Follow**: There is no lane change. The EGO is driving and there are no other vehicles in front.
5. **EGO-CutIn**: The EGO vehicle is moving right-direction to a new lane already occupied by another vehicle.
6. **EGO-CutOut**: The EGO vehicle is leaving left-direction the lane where it was driving.

Instead of working with the raw data from the video, radar and on-board sensors, the manoeuvre recognition system uses the so-called "object data", which contains "high level" representations or features describing the "traffic scene" such as EGO's speed, distance between EGO and another vehicle in front, etc.

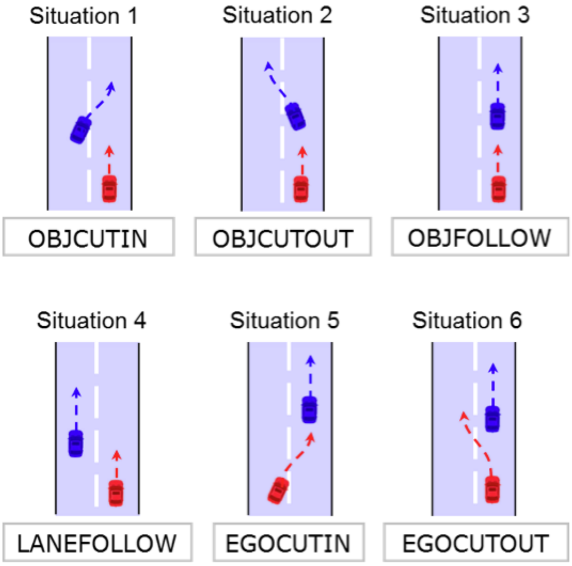


Figure 4: Different maneuvers which should be identified by the AMIDST system. Red blocks represent the EGO vehicle and blue blocks represent other vehicles in the scene.

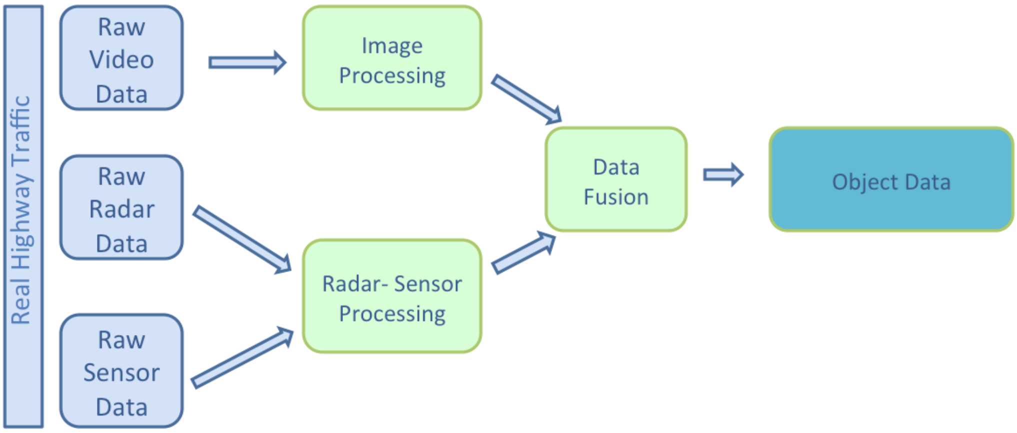
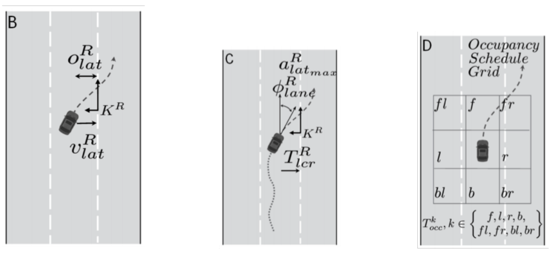


Figure 5: Daimler's Data Flow.



Lateral Evidence Trajectory Occupancy Grid

Figure 6: The three main dimensions of the situation features [**?**].

Figure 5 contains a visual description of the current data flow used to create this "object data". As can be seen in this figure, in a first step the raw data coming from the video, radar and sensors is preprocessed. In a second step this preprocessed data is fused and the high-level or "object data" describing the traffic scene is obtained.

As commented before, using the resulting "object data", Daimler has developed a probabilistic graphical model [**?**] which is able to recognize an ongoing manoeuvre around 0.6 seconds before the manoeuvre really takes place. This probabilistic approach is based on modelling the problem in different abstraction layers.

The static-OOBN model

The model described here was presented in [**?**] as an object-oriented Bayesian network (OOBN) [**?**] for addressing the problem of early recognition of a lane change manoeuvre (application scenario 1). This model works with the so-called "object data". This data mainly consists of a set of measured and/or computed signals or situation-features denoted by S (e.g.. EGO speed, EGO lateral velocity, speed of a car in-front, etc., see [**?**] for further details) describing the traffic scene. The situation features used for manoeuvre recognition are structured along three main dimensions: lateral evidence (LE), trajectory (TRAJ), and occupancy schedule grid (OCCGRID). A visual description is given in Figure 6. They are referred to as the three hypotheses of possible lane change manoeuvre. The lateral evidence hypothesis considers the lateral offset and the lateral velocity of the car and accounts for the lateral movement of the car. The trajectory hypothesis tries to account for the evidence about the car's trajectory by using the measures of the angle of the car and the estimated time to crossing the line. Finally, for the occupancy grid hypothesis it is collected data that allow to identify if the surroundings of the car are going to be occupied by some other vehicle in the traffic scene.

The general structure of this OOBN model consists of a number of abstraction levels as detailed in Figure 7. :

Class Sensor Measurement: This class represents objects at the lowest level of the

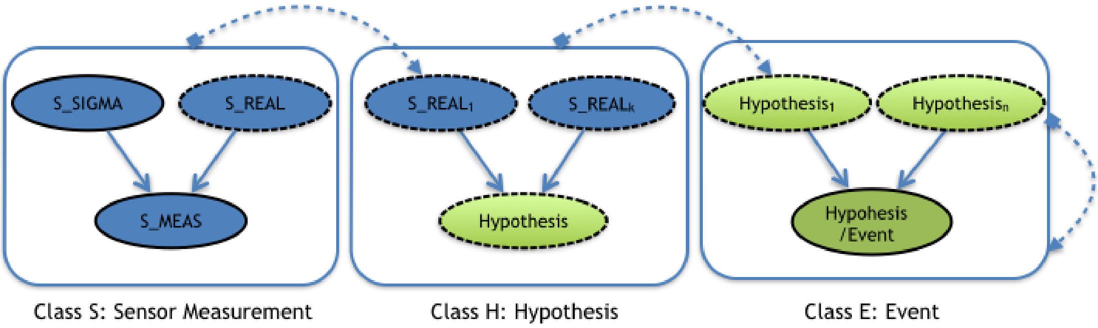


Figure 7: Static-OOBN model for the prediction of an event (maneuver) [?].

OOBN. It models the so-called *measured data* which are the observations characterising a situation. They are acquired from sensors and computations (see Figure 5). The structure is the common one found in a standard Kalman filter model to account for sensor noise or fault: S_MEAS refers to the sensor-reading value, S_REAL to the real value and S_SIGMA to the uncertainty in the measurements. So, the sensor-reading of a measured variable is conditionally dependent on random changes in the real value under measurement (S_REAL) and sensor noise/fault (S_SIGMA). In this problem, and due to the particular data flow in Daimler, observations about the measured value S_MEAS but also about the uncertainty of the measurement S_SIGMA are given in the object data.

Class Hypothesis: This class is in a higher level and directly depends on the real values S_REAL obtained in the previous class. These real sensor values are used to evaluate different hypothesis such as lateral evidence, trajectory and occupancy grid.

Class Event: This class is at the top level of the modelling. It allows to model high-level hypothesis based on low-level hypothesis in a recursive way. This class also includes the variable or the event representing the the possible traffic manoeuvres of the own and neighbour vehicles. This event is modelled based on previous high-level hypothesis.

Finally, in Figure 8, we show a concrete fragment of the OOBN model related the modelling of the lateral evidence (LE) hypothesis. As can be seen, this hypothesis depends on the lateral offset to a lane marking, O_LAT_REAL, and on the lateral velocity, V_LAT_REAL, of the car. Both measures are estimated from their measured values, O_LAT_MEAS and V_LAT_MEAS, and from the estimated uncertainty of the measurement,O_LAT_SIGMA and O_LAT_SIGMA. This part the OOBN is used to model the growing probability for the lateral evidence to cross the lane marking, based on the vehicle coming closer to the lane marking and the increase of its lateral velocity.

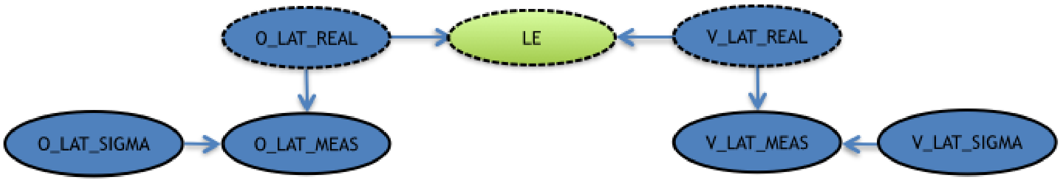


Figure 8: Static BN fragment for the LE hypothesis.

As denoted in Figure **??** and 8, S_REAL variables are continuous variables. However, this modelling has significant impact when making inference because we have discrete childs with continuous parents and, in consequence, the conditional probability distribution of the hypotheses given then S_REAL variables does not fall inside the conditional linear Gaussian family [**?**]. One possible solution used in [**?**] is to discretize the S_REAL variables. In this case, we have that S_MEAS, S_SIGMA are always observed so their potentials can be collapsed/combined with the potential of the respective discretized S_REAL variables. Then, all the inferences in this probabilistic model can be made by only using discrete potentials [**?**].

The dynamic-OOBN model

The above described static OOBN is able to detect a manoeuvre 0.6 seconds before execution. As detailed in the DoW [**?**], the goal is to extend the prediction horizon for manoeuvre recognition to at least 1-2 seconds (max. $4-5$ seconds ahead) before the actual lane marking crossing. Moreover, this early detection of the manoeuvre should not be at the expense of the prediction accuracy; as specified on Daimler's DoW [**?**], the area under the ROC curve (AUC) should be greater than 0.96 for 1 second and greater than 0.9 for 2 seconds.

Figure 9 shows an example of the current performance and limitations of the static-OOBN model. They correspond to two randomly selected sequences in which we keep track of the behaviour of two cars, the EGO and the OBJECT. In these figures we plot the evolution of different time-steps for lateral velocity and lateral offset to a lane marking in ongoing Object-CutOut and Object-CutIn manoeuvres (as sketched in Figure 4). The vertical black bar indicates the moment in which the manoeuvre has been recognised by the static-OOBN. The manoeuvre is finished at the end of the series, which coincides with the actual moment of changed lane. The black curve corresponds to the lateral offset and lateral velocity of the EGO car, which is just following the lane (LF), and the green curve corresponds to the values of the Object car performing the Object-CutOut/Object-CutIn manoeuvres. As expected for lane follow (LF), the lateral velocity of the EGO fluctuates around zero (i.e., EGO car is just driving inside its lane)

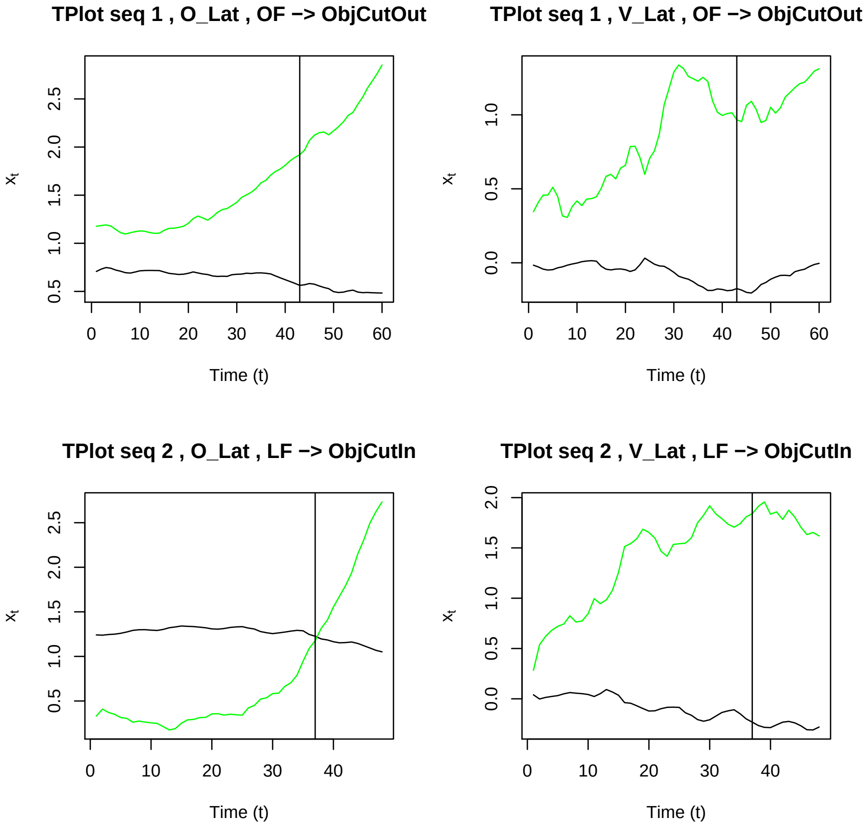


Figure 9: Daimler time plots of the lateral velocity and offsets for EGO (black line) and OBJECT (green line) in two randomly selected sequences of an ObjCutOut and an ObjCutIn. The x-axis corresponds to the different timesteps and the y-axis to the lateral offset on the left-hand side graphs and lateral velocity on the right-hand side graphs.

and the lateral offset of the lane marking is almost constant all the time. However, when we look at the lane change behaviour of the OBJ car (green lines), we easily see a quite different behaviour. Firstly, we observe that the lateral velocity is much higher indicating a lateral movement. Similarly, we also observe how the lateral offset steadily increases, what clearly indicates that the Object car is leaving its current lane in both manoeuvres.

Although the manoeuvre is clearly identified before it completes in both cases, it is desired to predict it *further* in advance. Looking at the evolution of lateral offset and velocity in Figure 9, we could argue that the detection of the manoeuvre can be performed earlier (i.e. when the lateral offset and lateral velocity starts to increase more dramatically and consistently for *several* timesteps). This is one of the basic pieces of

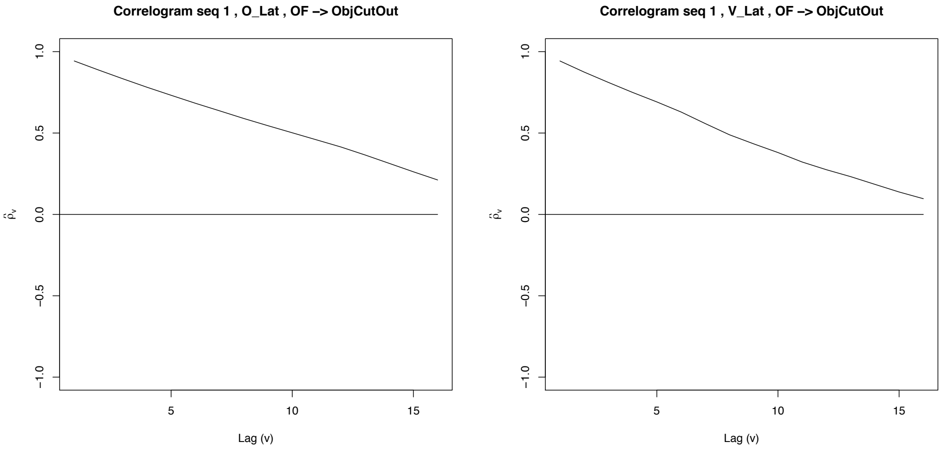


Figure 10: Correlograms for Lateral velocity and offset. The x-axis represents the lag v or time difference, and the y-axis the sample autocorrelation coefficient of lag v, $\hat{\rho_v}$, that is, the correlation between variables at time t and at time $t + v$ (See Section 2 for more details).

evidence that motivates the development of the dynamic version of the static-OOBN model [**?**]. Another piece of evidence appears when we look at the sample correlograms for lateral velocity and offset (that is, the correlation of the data with lagged values of themselfves) plotted in Figure 10. There we can see a strong temporal dependency (the correlation coefficients are high) for the two variables when we consider a *small* number of time steps between the temporal observations. Hence, the use of a dynamic model to make predictions about the evolution of the lateral velocity and lateral offset in a near future seems to be reasonable according to this analysis. And our main assumption is that the use of dynamic Bayesian network models (see Section 2) will allow us to build a more accurate and reliable system for manoeuvre recognition.

In any case, let us note however that the prediction horizon for manoeuvre recognition is going to be limited in order to avoid false positives (i.e. recognizing a lane change manoeuvre when the driver was just performing some random lateral movement). In case of a false positive, such as an erroneously Object-CutIn for instance, the adaptive cruise control would react with an unnecessary break, so they ought to be avoided. It is then required to find a good balance between the rate of false positives and false negatives.

As explained in Section 2, the dynamic extension involves copies of the static OOBN for different number of time steps in the time window. Figure 11 shows an example for the LE hypothesis, where the nodes for O_LAT_REAL and V_LAT_REAL are temporal clones defining the share belief state between consecutive time steps, and hence creating a first order Markov process. This model has been proposed in [**?**] by some of the AMIDST partners. The first order assumption is a standard assumption in this kind of models and helps to simplify the posterior inference and learning process. Fortunately,

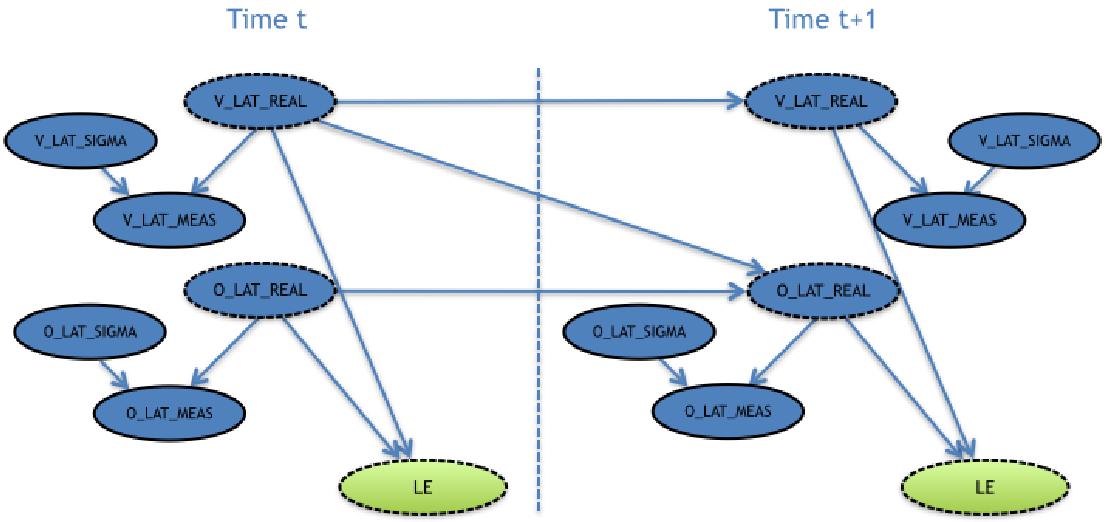


Figure 11: Daimler dynamic fragment for the LE hypothesis.

if we build a partial-correlogram as the ones plotted in Figure 12, we can see that this assumption seems reasonable, because the influence of the lateral offset a time $t-1$ on the lateral offset at time $t+1$ given the lateral offset at time t is close to null (i.e. the value at $v=2$ for the lag is close to null in Figure 12). And the same reasoning applies to the lateral velocity.

As described in [**?**], the proposed dynamic BN (DBN) aims to incorporate the trend of change for the real values, where their physical relations are represented as causal dependencies between the time steps dt. For instance, in Figure 11 the transition function of O_LAT_REAL at time t, denoted by $O(t)$, is modeled as a Gaussian distribution. Its mean is affected by $O(t-1)$, and by V_LAT_REAL at time $t-1$, denoted by $V(t-1)$:

$$O(t) = O(t-1) + V(t-1)dt + \epsilon \tag{1}$$

where ϵ denotes a white noise $\mathcal{N}(0, \sigma^2)$ which is assumed to be small. In order to corroborate the validity of this distributional assumption, we also analysed the hypothesis $O(t) - O(t-1) = \Delta O = V(t-1)dt + \epsilon$ on our data. Figure 13 shows the plot and contour plots for V and ΔO, which show that the assumption of linear relationship with Gaussian noise might not be very far from reality.

Finally, in Figure 14 we show a rough overview of the final structure of this dynamic Bayesian network[1], which shows how the temporal connection is only made on the top nodes involving the situation-features in consecutive time steps. The final event or

[1] Full details can not be given for confidentiality reasons

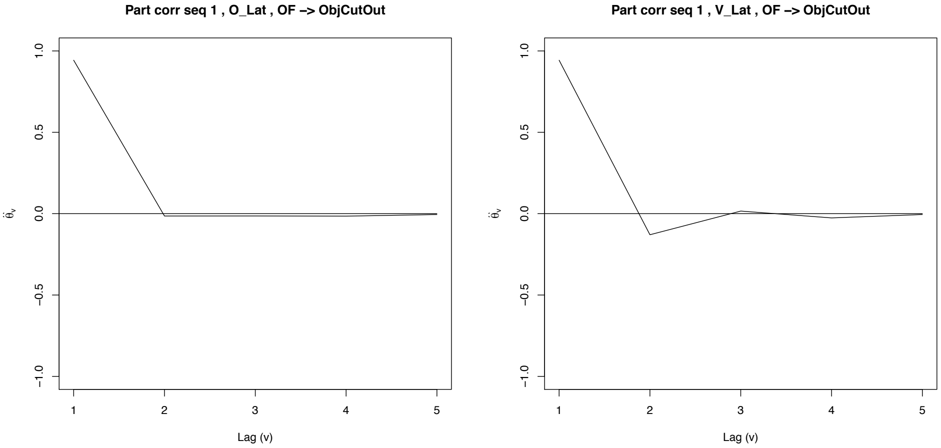


Figure 12: Partial-correlograms for lateral velocity and offset. The x-axis represents the lag v or time difference, and the y-axis the partial autocorrelation coefficient of lag v, $\ddot{\theta}_v$, that is, the correlation between variables at time t and at time $t+v$ after having removed the common linear effect of the data in between. (See Section 2 for more details).

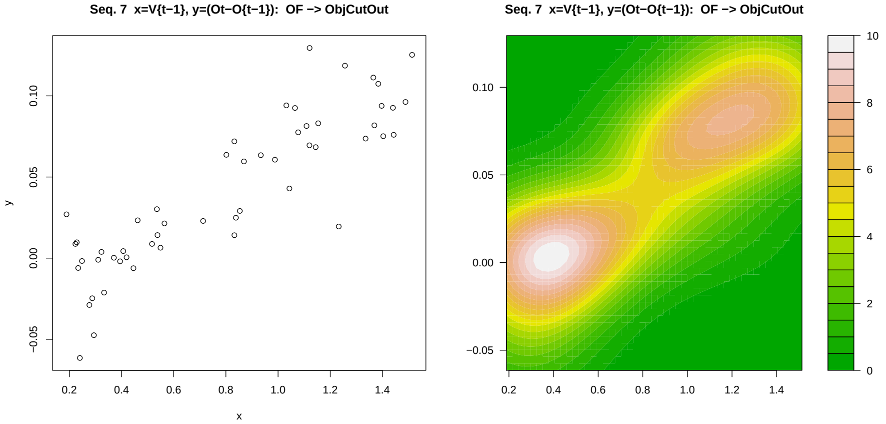


Figure 13: Time plot for $V(t-1)$ vs $O(t)-O(t-1)$. Linear correlation can be observed.

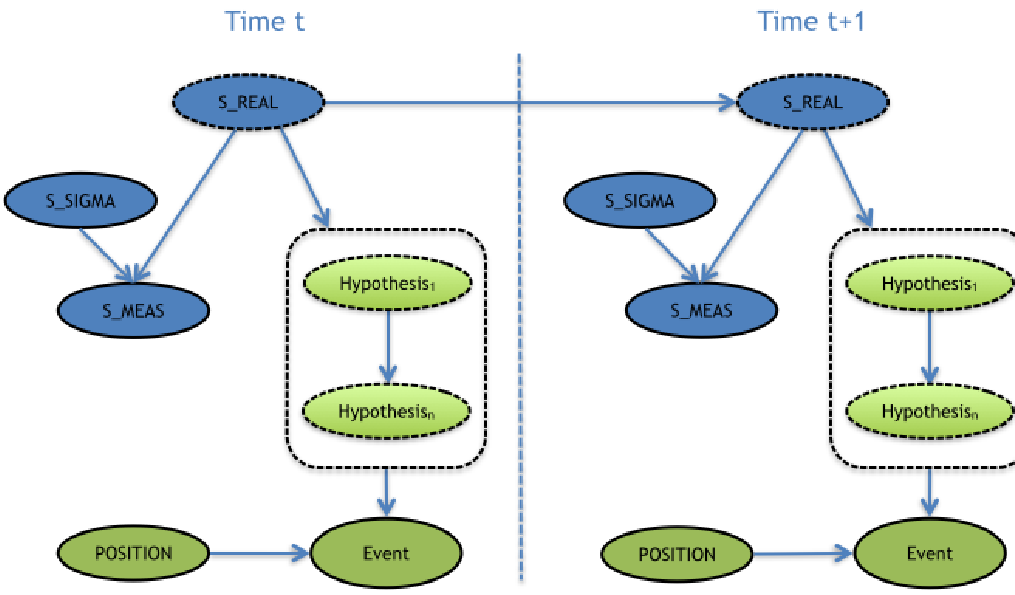


Figure 14: Daimler dynamic model with several hypothesis.

manoeuvre prediction is then determined by the combination of these hypotheses and the position of the OBJ with respect to the EGO car.

Similarly to what happened with the static version of this OOBN, for the dynamic model in this current form we have that the conditional probability distribution of the LE hypotheses given the V_LAT_REAL and O_LAT_REAL variables does not fall in the conditional linear Gaussian family [?] because we have discrete childs with continuous parents. Similarly to the static case, a possible approach to deal with that is the discretization of the V_LAT_REAL and O_LAT_REAL variables. And, again, we would obtain a model where all the inferences can be implemented over discrete potentials because the remaining continuous variables S_MEAS and S_SIGAM are always observed.

3.1.2 Earlier prediction of the need for a lane change based on relative dynamics

Earlier prediction of manoeuvre intentions could be achieved before any development of the trend for lateral evidence has been observed. A first indication of possible lane change intention can be observed through the relative dynamics between one vehicle (EGO or OBJECT) and the vehicles in front of it on the same lane. If a slower vehicle is driving in front of the own vehicle on the same lane, this could be an indication of the need for a lane change. Because to continue its safe driving, the approaching vehicle should either break and reduce its speed to the speed of the vehicle in front or, alternatively, it should change

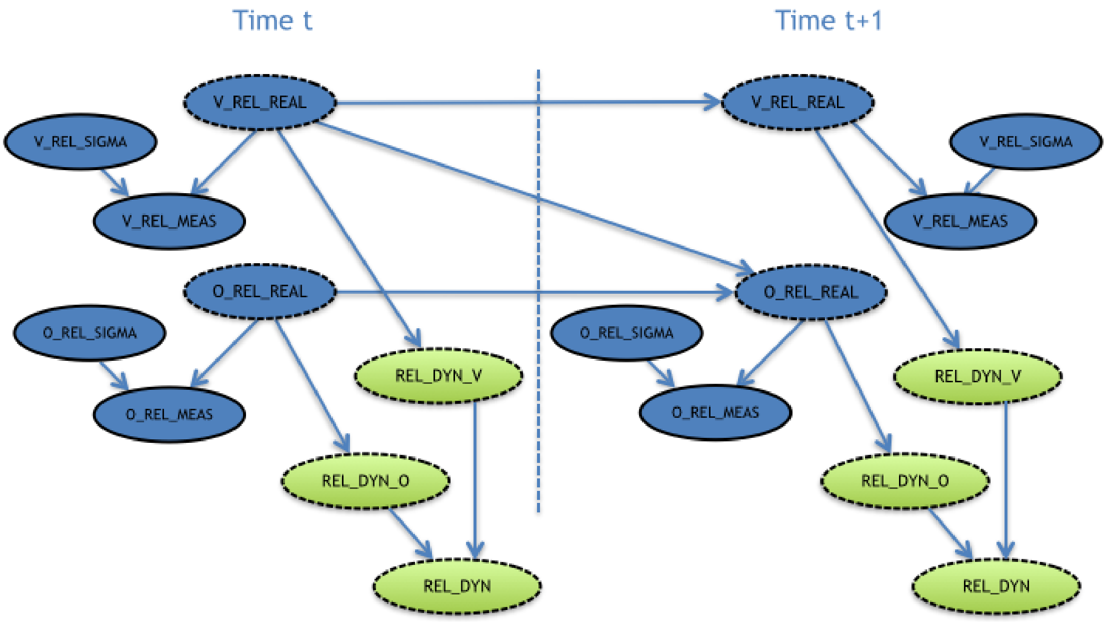


Figure 15: Daimler Temporal Model with relative dynamics. V_REAL and O_REAL refers to the relative velocity and relative distance between the EGO and the car in front, repectively. The other variables with _MEAS and _SIGMA suffixes refers to the measured values and the uncertainty of the measurements, respectively, as commented in the previous section for other sensor readings.

to the neighbour lane, if the neighbour lane is free and no other vehicle is approaching with a higher speed than the own vehicle. A continued safe manoeuvre (of type “lane follow” or “lane change”) is modelled by estimating the TTC (TimeToCollision) to the vehicle in front (on the same lane) or to eventually approaching vehicle (on the neighbour lane). For safe manoeuvre, TTC should be bigger than 1 second, if the own vehicle wants to change to the neighbour lane or if it needs to break to ensure safe driving on the same lane (“lane follow”) . Our assumptions is that this approach will allow us to further increase the prediction horizon for manoeuvre recognition (up to 5 seconds).

Similarly to the models described in Figure 11, a new dynamic OOBN is defined with the hypothesis “relative dynamics” (REL_DYN), as shown in Figure 15. Due to confidentiality reasons, we can not give at this stage of the project further details about this model. We can see however that this model have the same general structure that the dynamic models proposed for the previous application scenario.

Discussion and future models

The above included proposals are preliminary models designed in line with the expert knowledge facilitated by Daimler. They all try to balance a good level of expressiveness and efficiency during inference.

In the case of the dynamic extension, another alternative is to consider not only the temporal dependences between consecutive time steps for the sensor measurements but also, or instead, consider the temporal links between a hypothesis at consecutive time steps. It seems natural, for instance, that the probability of observing lateral evidence (LE) for the EGO car should be higher given that there was LE at the previous time step. And similarly for the rest of the hypotheses, such as TRAJ or OCCGRID and the final hypothesis or event that identifies the type of manoeuvre. However and since these hypothesis nodes are placed at the bottom of the DBN; as opposed to the sensor measurements, which are placed at the top; adding these type of dependences would greatly increase the complexity of inference in the resulting network because the number of dependencies will be much higher as the DBN is unfolded through the time.

Another possible extension that we envision is the consideration of an extra hidden variable to model acceleration for the dynamic model. We believe that assuming that lateral velocity varies according to a constant acceleration can lead to inaccuracies. Figure 16(a) shows the behaviour of lateral velocity for a given sequence, that in this case corresponds to an EGO-CutIn manoeuvre. At the beginning of the sequence, acceleration is close to constant, but from around time steps 30 to 50 a higher acceleration value should be taken into account. The contour plot on Figure 16(b) shows a large density of points around lower and higher values of V where the acceleration is constant, but just isolated points in between. Hence, we believe that the dynamic model could benefit from an extra hidden variable to represent acceleration as displayed in Figure 17. Thus, the following equation would be considered for velocity at consecutive time steps:

$$V(t) = V(t-1) + A(t-1)dt + \epsilon \tag{2}$$

3.2 CajaMar Models

3.2.1 Introduction

There are two tasks to be solved for Cajamar's use case (see D1.2 and DoW). The main one is the estimation of the *probability of default*, defined as the probability that a credit operation will end up in default within two years. The secondary task consists in obtaining good customer profiles in terms of risk, so that marketing campaigns can be specifically targeted to these low risk customers.

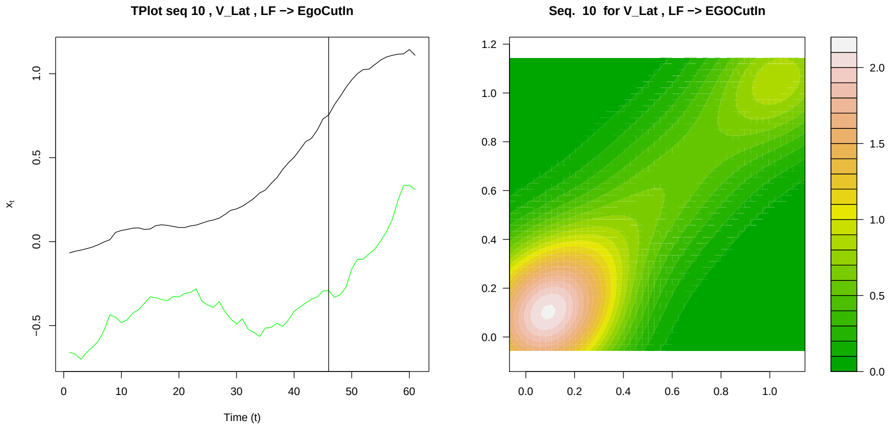


Figure 16: Time and contour plots for $V(t)$ vs $V(t-1)$. Varying acceleration should be considered to capture the variations in the data.

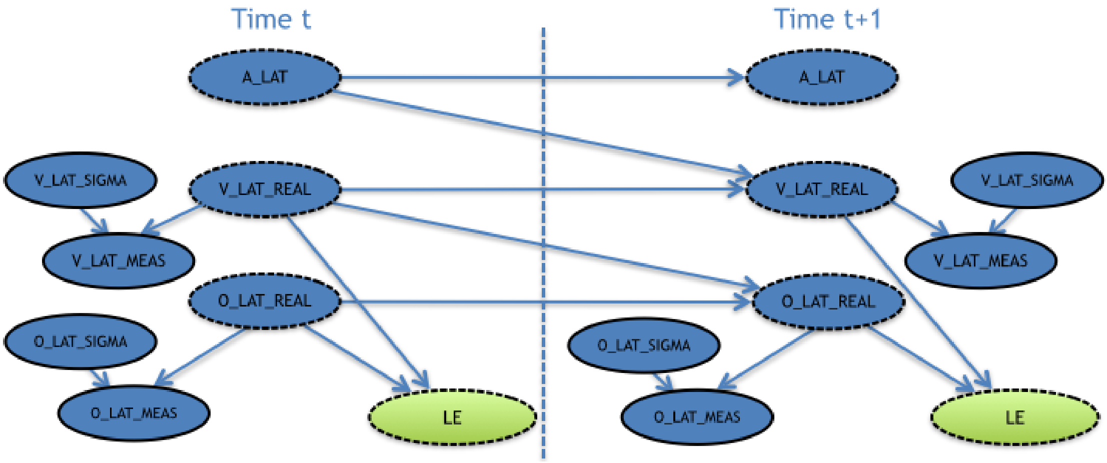


Figure 17: Daimler dynamic fragment for the LE hypothesis with a hidden node for acceleration.

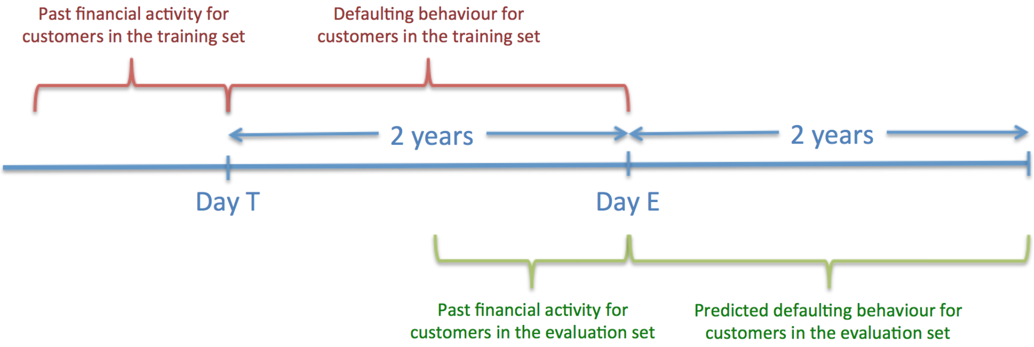

Past financial activity for customers in the training set
Defaulting behaviour for customers in the training set
2 years
2 years
Day T
Day E
Past financial activity for customers in the evaluation set
Predicted defaulting behaviour for customers in the evaluation set

Figure 18: Time-line of the generation of the evaluation and training data sets.

3.2.2 Predicting probability of default

Introduction of the problem

In any commercial bank, every time a customer applies for a loan, the bank experts evaluate the customer's risk profile before making their decision.

At Cajamar, this risk evaluation protocol is implemented by evaluating whether the client is going to default or not within the following two years. If a client is labelled as defaulter, he/she will remain defaulter in the database at least two more years. At the moment of writing, this decision is supported by an automatic supervised classification model (i.e. a logistic regression) which takes information about the recent financial activity of the customer at CajaMar, as well as information about the recent past paying behaviour of the customer provided by other Spanish financial institutions, and make predictions using this information about the probability that the client will default during the following two years.

The methodology currently employed does not assume dependence structure among the variables. Even though this model is quite simple, current predictions are made using only a set of 27 variables out of the more than 1000 that are available. Updates in risk predictions are made on a monthly basis, whereas the predictive model is only updated after several years. These low update frequencies are selected partly due to limitations in the available commercial software and the computing resources.

The objective is to daily update the risk evaluation for every customer of the bank. This daily evaluation will be made by creating two data sets: the model training data set and the model evaluation data set (see Use Case 1 in D1.2). How these data sets are created gives us some insights about the nature of this risk prediction problem. Figure 18 shows the time-line of the generation of the evaluation and training data set, which is further explained below. First, let denote the current time as t and 2 years back as k. and

- **Model Evaluation Data Set:** The evaluation data set is created at time t. This data set contains a record for every client to be evaluated. As in any standard evaluation data set in supervised classification settings, each record will contain the values of the predictors variables for this customer. The predictor variables refers to the financial activity and the paying behaviour of the customers in a recent past and their socio-demographic data.

  The recent financial activity refers to attributes of the customer such as "account balance", "number of credit card operations", etc. in the last 180 days [2]. These attributes usually change daily for a customer, then they are encoded by introducing a set of variables for each attribute that contains the value of this attribute in the last 180 days from day E. I.e. if the financial activity of the customer is detailed by F attributes, $180 \cdot F$ predictive variables are included in this data set.

  In the case of past paying behaviour, the attributes refers to attributes such as the paying behaviour of the customer with other financial institutions or other companies (phone companies, electricity companies, public bodies, etc.). A monthly record of the last 36 months from day T is considered when building the evaluation data set. So, if there are P variables referring to the paying behaviour of a customer, $36 \cdot P$ variables are included as information about the past paying behaviour.

  The dataset for the evaluation of customers is detailed in Table 2.

	Days			Semester			
Time t	$\mathbf{X}^{(t-180)}$	$\dots$	$\mathbf{X}^{(t-1)}$	$\mathbf{Y}^{(t-6)}$	$\dots$	$\mathbf{Y}^{(t-1)}$	$\mathbf{Z}$
Client$_1$							
$\vdots$							
Client$_n$							

Table 1: Evaluation dataset at time t. Attributes for financial activity, paying behavior and sociodemographic information are denoted as $\mathbf{X}$, $\mathbf{Y}$ and $\mathbf{Z}$, respectively.

The task of the supervised classification model is to take each of the records (i.e. customers) of this data set and compute the probability of defaulting within the following two years. Finally, the risk table in the system is updated (see Table 19).

Time t	Risk of defaulting
Client$_1$	$\hat{r_1}$
$\vdots$	$\vdots$
Client$_n$	$\hat{r_n}$

Figure 19: Table containing the risk of defaulting for every client.

[2] This limit is fixed by the Bank of Spain

If at some point the probability of defaulting of some customers rises above some predefined threshold, the bank can try to take preventive actions to reduce the chances that this customer finally defaults in some of his/her loans.

- **Model Training Data Set:** The training data set is built in the same way than the evaluation data set. They contain the same set of predictive variables with the main difference that they refer to different time points. If the predictive variables in the evaluation data set are built for day E, the predictive variables in the training data set are defined for the day $T = E$ - 2 years (i.e. two years before in time). So the training data set contain one record for each customer of the bank and the predictive variables refers to the financial activity and paying behaviour of this customer in the recent past before day T (i.e. the previous 180 days or the previous 36 months).

  The dataset for training/updating the model is detailed in Table 2.

	Days			Semester				
Time t	$\mathbf{X}^{(k-180)}$	...	$\mathbf{X}^{(k-1)}$	$\mathbf{Y}^{(k-6)}$	...	$\mathbf{Y}^{(k-1)}$	$\mathbf{Z}$	Defaulter$^{(t)}$
Client$_1$								
$\vdots$								
Client$_n$								

Table 2: Training dataset at time t with $k = t - 2$ years. Attributes for financial activity, paying behavior and sociodemographic information are denoted as **X**, **Y** and **Z**, respectively. Note that the column *Defaulter* at time t is labelled as false only if the customer has not been defaulter during the last 2 years.

  Unlike the evaluation data set, each record of this data set contains a class label indicating if this customer is defaulter o non-defaulter. To decide if a customer is defaulter or not, we look at the 2 years data between day T and current day E. If during these two years the client has regularly pay each of his/her loans and the other financial institutions does not provide any evidence of defaulting, the client is labelled as non-defaulter. Otherwise, it is labelled as defaulter (see Data Characteristics document for more detailed information about this respect).

  Using this data set, the bank fits a logistic regression model, which is then used to update the probability of defaulting of each of the clients in the evaluation data set. Checks if the dataset currently used is the same?

Static Model

In this first approach we do not explicitly consider some of the dynamics of the problem: we do not model that a customer can be non-defaulter and defaulter at different moments in time (e.g. one customer can be creditworthy and, after some time, go bankrupt due to unemployment). Instead, we consider a static prediction model where given the financial

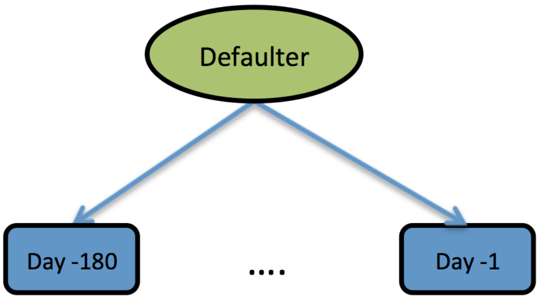


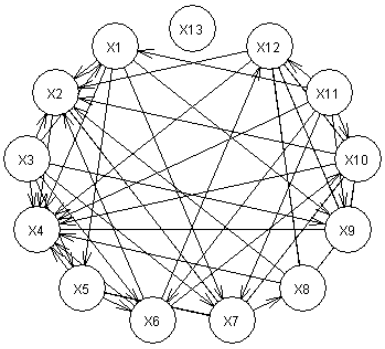
Figure 20: Global structure of the static model. Each blue box represents a set of variables measures during the same day. The variables within a box can be connected (e.g. according to a tree structure and, globally, conforming a TAN).

behaviour of the client over a recent past, it predicts whether the client will default or not within the next 2 years. Similarly to what it is done in the current CajaMar approach.

Figure 20 shows the general structure of this static model. Each yellow box represents a set of F variable measurements for a particular day. For the shake of simplicity, for now on and in the graphical models depicted, we do not explicitly show the variables that refer to monthly information during the last 36 months. The green node is the class variable that represents the probability that a customer will default within the next two years.

The variables within a box can be connected (e.g. according to a tree structure and, globally, conforming a TAN). Figure 21(a) shows how indeed there exist dependences between some of the variables in the data for a particular day. Please replace this BN by another one with the name of some of the variables. Figure 21(b) shows the contour bivariate plot for variables V1 and V2, showing a clear linear relationship between the two variables. This might even indicate that one variable is just a replica of the other variable or contain quite similar information, and their joint contribution will not only make learning and inference computationally more expensive, but it could lead to poorer performance []. This would contribute to support the need to explore suitable feature selection techniques as specified in Use Case 2 of Cajamar's Requirement analysis.

It is also of major interest to analyse the type of density probability distributions to use in the proposed model. Figures 22(a) and (b) show the density histogram for the values of a continuos attribute that measures the end-of-day balance for defaulters and non-defaulters respectively. The density curves represent a credible approximation using mixture of Gaussian distributions, which is the case for most of the other predictive attributes. Note that for defaulters, the values of the end-of-day balance are overall lower compared to those corresponding to non-defaulters.



Contour plot to be included with (ideally) strong linear correlation between variables (maybe class-conditioned?)

Figure 21: .

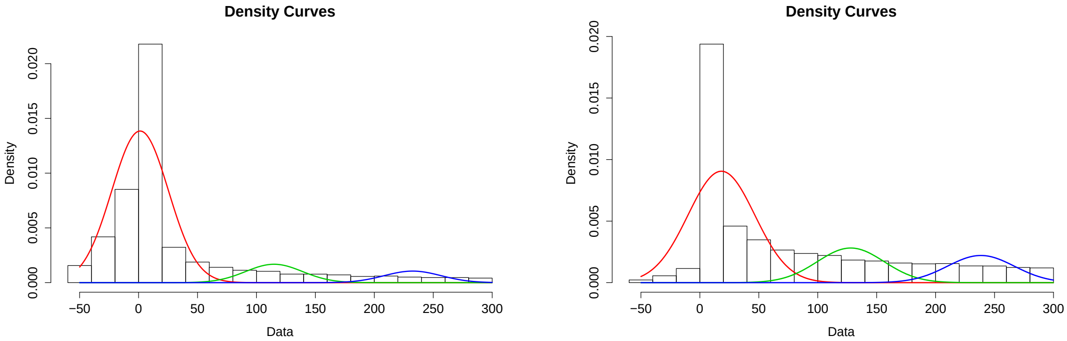


Figure 22: Mixture of Gaussian approximation of *end-of-day balance* for defaulter (left) and non-defaulter (right).

Check please!There exist however discrete (non-ordinal) predictive attributes which impose a limitation in the structure of the Conditional Gaussian network, since discrete attributes cannot have continuous parents. This might not be a problem for the semi-naive Bayesian network to be considered, and it is certainly not a problem for naive Bayes. Nonetheless, if this limitation were found to be problematic in future stages of the project then other families of probability distributions will be explored, such as Mixture of Truncated Exponentials[] or Mixture of Truncated Basis Functions[], which can cope with more generic Bayesian structures at a higher computation cost.

In summary and given the original relational database, the process of building this static model consists of the following steps:(Should this be included?).

- Construct a single flat table, containing information on time windows of *180 days*.
- Build a semi-naive BN classifier (e.g NB or TAN).
- Update risk profiles using the static classifier.

Dynamic Model

In this second approach, we will consider the dynamic structure of the problem. These dynamics are present because the behaviour of the customers evolves over time (e.g. the account balance is continuously changing from one month to another, also the income levels, etc.) as well as the label as a defaulter or non-defaulter customer (e.g. customers can be creditworthy and, after some time, go bankrupt because they have lost their job). Analysing some of the data, we can actually see that if a customer was a defaulter at day t, the probability of being a defaulter at day $t+1$ changes from p (prior probability in the static model) to p' (transition probability in the dynamic model). The reason for this dramatic change is that once a client is a defaulter, he/she will be a defaulter for some time, and the static model is unable to represent this effect. And the way we have defined the problem it actually does not matter much, does it?

Figure 23 represents the global idea of the proposed temporal model. It can be compactly represented by a dynamic Bayesian network made of components as the one displayed in Figure 24. D_t represents the class variable at time slice t (i.e. defaulting or non-defaulting client). Each feature variable at time t, denoted as X_t, is linked to the same variable at time $t+1$, X_{t+1}. Although this is a reasonable assumption for most of the variables, this first Markov order relationship however might prove insufficient for some of the variables.[We include the following comments to justify the introduction of memory variables. We think that we need to introduce memory variables if we have evidence that first order Markov relationships does not hold and we need to account for information coming from the past. This evidence could be obtained for a partial corrrelogram.]. Figures 25 and FigY (FigY should show a couple of partial correlograms that do not drop to zero at lag 2) show the partial correlograms for different continuous predictive variables. For the variables in Figure 25, the partial correlograms drop to

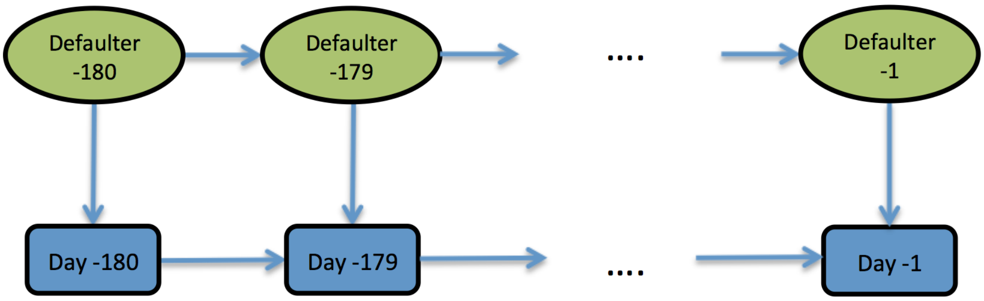


Figure 23: Global structure of the dynamic model. Each yellow box represents a set of variables measures during the same day. The variables within a box can be connected (according to a tree structure and, globally, conforming a TAN) as well as variables between two consecutive days. Red box refer to the possibility that client is defaulter and are temporal connected.

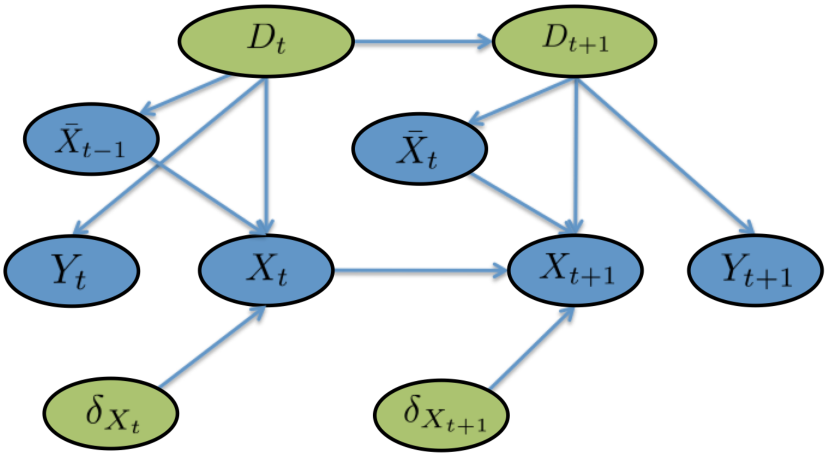
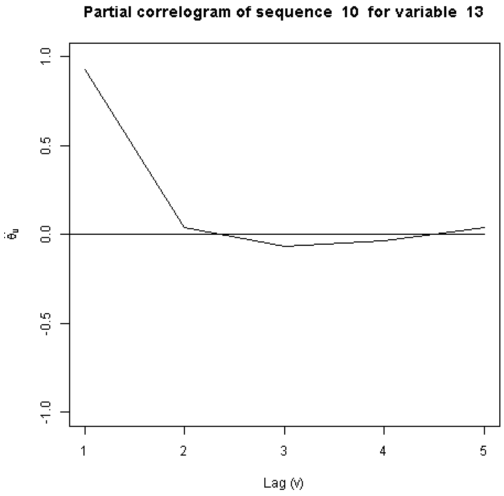
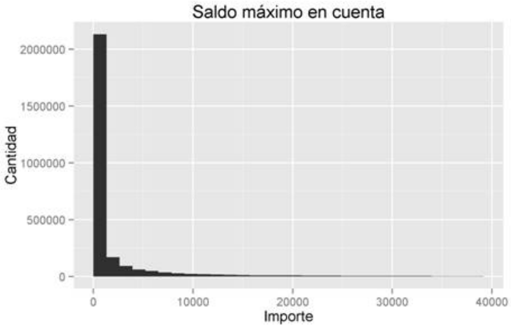


Figure 24: Basic component of the structure of the dynamic model.



Partial correllogram for another variable

Figure 25: Partial correllogram for variables A and B. A first order Markov assumption seems reasonable.



Outstanding amount without zeros

Figure 26: Frequency histogram of the historical monthly outstanding amount

almost zero for a lag equal to 2, making the first Markov order assumption a reasonable one. However, for the variables in Figure FigY the partial correllogram takes more time to drop to zero, which might indicate that past samples still have an influence on the current sample given the previous one. To mitigate this effect, a *memory variable*, $\bar{X}_t$, that represents the average value of X during the last 180 time slices (days) is included.

Finally, an indicator variable δ_{X_t} may be included if the variable is such that is observed many times at point 0. This is the case for payments made by credit card or the historical monthly outstanding amount on the account for instance, whose value can be equal to zero for a large number of days for most customers, as shown in Figure 26. Figure 26(a) displays the histogram of this variable including all values, and 26(b) when the zero values are not considered.

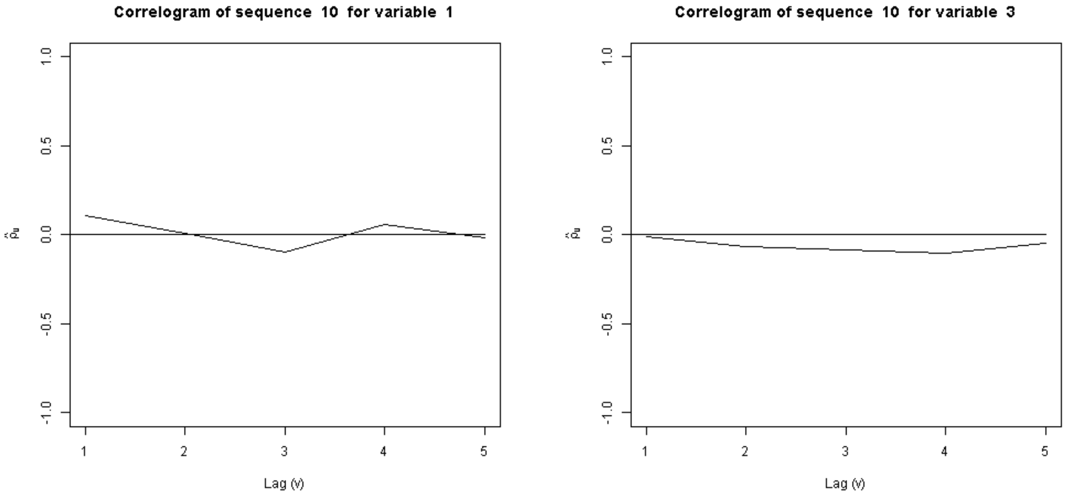


Figure 27: Correllogram for variables 1 and 3. No temporal dynamic shown.

On the other hand, there exist some variables that do not display any type of dynamic behaviour. This is for instance the case for variables 1 and 3 (please replace with the names of the vars.), whose correllograms on Figure 27 show values very close to zero for all lags. These variables are hence not linked through consecutive time steps, which are represented by Y_t and Y_{t+1} in Figure 24. What about drawing only one Y? aren't we including the socio-economic variables pointing at the class variable?.

In summary, the process of building this dynamic model from the original relational database consists of the following steps:(Should this be included?).

- Construct *1 flat table* for each day.
- Build a *dynamic* BN classifier (e.g. NB or TAN like structure extended in a dynamic fashion).
- Update risk profiles using the dynamic classifier.

3.2.3 Low risk profile extraction

Introduction of the problem

The marketing department at Cajamar periodically launches marketing campaigns for the recruitment of new products by customers (i.e. a new credit card, an insurance, ...). The success of these campaigns depends greatly on the client group to which the campaign has finally targeted. It is also crucial to reduce as much as possible unnecessary expenses focused on non-potential customers.

For this purpose, the marketing group proceeds as follows. First, they filter the customers using their own marketing models and also, in collaboration with the risk department, select a subset of predictors considered relevant to be part of the final profile (mainly sociodemographic variables). For example, the age of a client is a relevant predictor when designing campaigns to attract customers for a death insurance.

After obtaining this first group of customers for the campaign and determining the set of relevant attributes, the model proposed in Section 3.2.2 will be used to identify the profiles of the less risky clients using the most probable explanations (MPE) method (*defaulter* variable is evidenced to *No*) . Thus, the target customers previously filtered can be now grouped and ranked using these profiles.

Static model

As pointed out before, mainly sociodemographic variables will determine the customer profiles used in the campaigns. These variables are mostly static as they do not change frequently over time (e.g. marital status, sex, type of job, ...). Hence, it has more sense to explore a profile extraction solution based on a static model as the one depicted in Fig. 28. A solution could be the use of augmented Bayesian classifiers like TAN, k-DB to reduce the search space required for general structures.

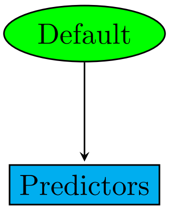


Figure 28: Static Bayesian network for the profile extraction. The variables within the blue box can be connected to conform an augmented Bayesian classifier.

Note that past information about the variables are not considered for this problem as it was for predicting the probability of defaulting in Section 3.2.2. In this model the latest information about the variables are considered instead.

For the profile extraction, the model not necessarily must have a classifier structure but it can have a general structure instead. However, what is desirable is to avoid using a NB structure for this task. The reason is that once the *defaulter* variable is evidenced to *No*, the predictors become independent and the analysis would be poorer (MPE in this case would correspond to the maximum probability value for each variable individually).

Dynamic model

A dynamic model for the profile extraction makes no sense as the variables used are mainly sociodemographic and their values do not change over time.

We could include some plots showing that sociodemographic variables does not change over time?

3.3 Verdande Models

As has been pointed out in Delivrable D1.2, three main tasks have to be addressed for Verdande use case:

1. The automatic detection of drillstring vibrations and abnormal torque states which aims to diagnose better the shapes of the wellbore and the shape of the equipment, make better decisions on how to manage the well, and thereby reduce the non-productive time.
2. The semi-automatic labelling which aims to compute and output a normality score for a considered drilling situation. This should be performed taking into account of the temporal dynamics of the drilling process and continuously adapting to changes in the incoming streaming data. This further reduces the non-productive time and improves the data quality.
3. The automatic formation detection which aims to provide a probability table for the formations at depth of the MWD (measurements while drilling) tool in real-time. Once again, this should be performed taking into account of the temporal dynamics of the drilling process and continuously adapting to changes in the incoming streaming data. The evaluation process includes measuring the deviation between the detected lithology chart and the lithology chart that is interpreted after the well is drilled.

For all tasks, the model must deal with both continuous and discrete observed random variables. Moreover, the oil-well data to be used presents the following characteristics:

- Having a dynamic that is put together by *long-term* patches (ranging from a couple of hundred observations and into the thousands). Inside a patch, the data is typically fairly *stable* and with low noise, even if this is not always the case. Between the patches the data can vary a lot. A *patch of data* typically corresponds to the implementation of one activity (like drilling, connection tripping in/out, etc.). Verdande has systems in place to detect activities (and even more important: the change of activities). The models that we make will probably be better at working inside a single patch. I?m calling these *local models.* Models designed for a broader span will be denoted *global.*

- Inside one activity, many of the attributes can be strongly correlated. The observed correlation between variables X and Y can either be instantaneous (i.e., $\mathrm{corr}(X_t, Y_t)$ significant), or delayed to some extent (i.e., exposed through the correlation $\mathrm{corr}(X_t, Y_{t+k})$ for some fixed k).

- Physical models can be used to understand why these correlations are there, but not to quantify them. In addition, sometimes the strength, and even the sign, of the correlation may change from well to well.

3.3.1 Detection of drill string vibrations

3.3.2 Semi-automatic labelling

3.3.3 Automatic formation detection

3.3.4 Discussion and future models

4 AMIDST Model Class

4.1 Introduction and Notation

One of the main goals of this project [] was the definition of a general model class which the following features: (i) it has to be applicable to the three main use-cases; (ii) it should be general enough to apply to other future similar use-cases; (iii) and it should contain certain structure that can be exploited to make this model class valid for scalable inference and learning in massive data streams.

With these three aims in mind, we started the definition of the general model class by finding commonalities between the different models presented in Section **??**. With this aim, we introduce some new "graphical" notation to ease this process. This new graphical notation is based on the use of subnetworks modules (?), they are represented by square boxes and refers to a subset of nodes with some kind of interconnection between them (when looking at the examples, we think this concept will be much clear). Following the same notation we used for nodes, square boxes rounded by dashed lines refers to a subnetwork which is not observable (i.e. composed by hidden variables); square boxes rounded by continuous lines refers to a subnetwork where the nodes are observed; when the square box is coloured, it means that all the nodes of the subnetwork are discrete (green color) or continuous (blue color). Square boxes can also be nested to represent smaller subcomponents inside a bigger subcomponent. In Figure ? we give a visual description of this notation.

We present now the models of the three use-cases but using this previously introduced notation.

Figure 29: Graphical Notation

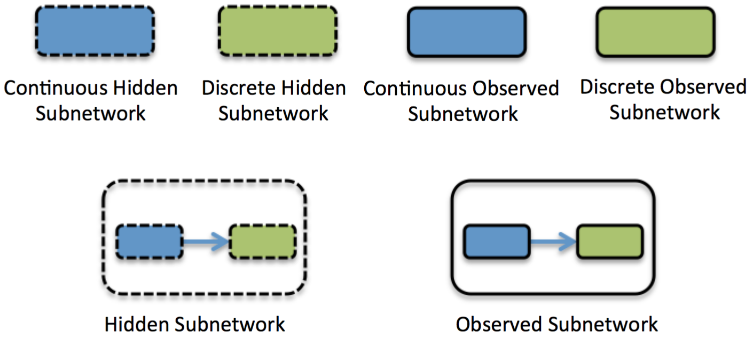
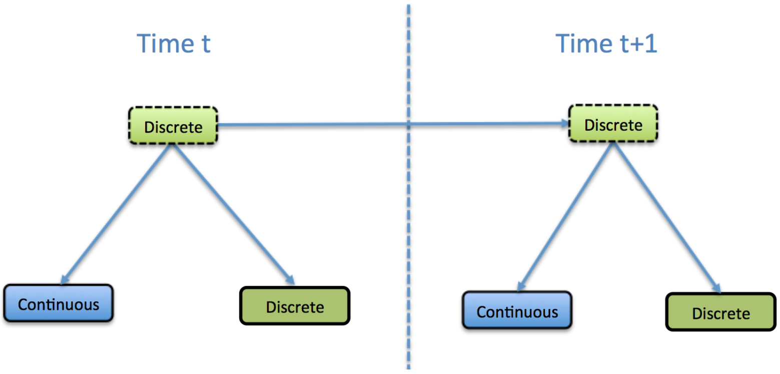

Continuous Hidden Subnetwork
Discrete Hidden Subnetwork
Continuous Observed Subnetwork
Discrete Observed Subnetwork
Hidden Subnetwork
Observed Subnetwork

Figure 30: Daimler Model Class

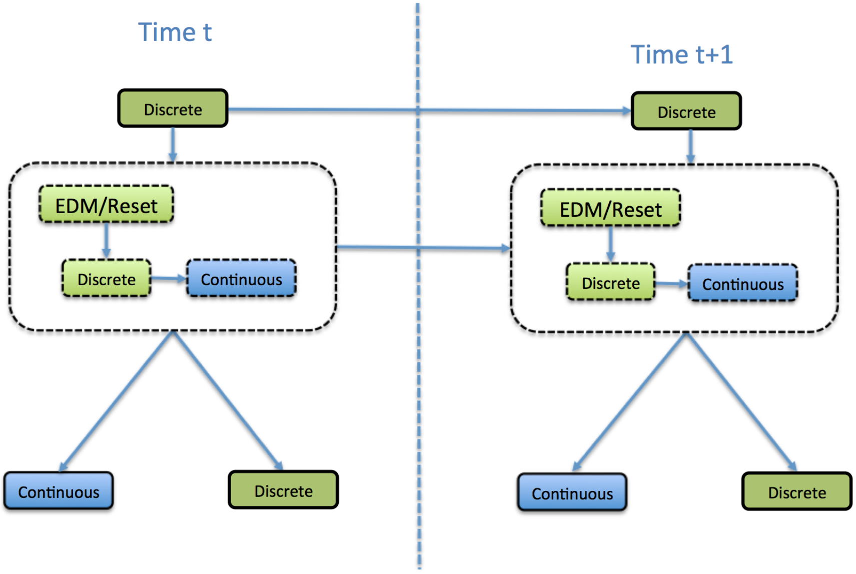


Daimler Model Class

Figure 30 shows the general structure of the Daimler's models. The model is structured as a stationary dynamic Bayesian network. The static model which is repeated through the time is composed by three elements: a set of hidden discrete nodes on top; and two observed sub-networks, one continuous and one discrete, which directly depends of the hidden sub-network. This dynamic BN is only temporally connected through the hidden sub-network. In consequence, the future and past time slices of our dynamic BN are conditionally independent given the hidden sub-network corresponding to the present time. Additionally, inside a time slice, the observed continuous and discrete sub-networks are also conditionally independent given the hidden-subnetwork.

We also have more structure that can be exploited during inference and which is not reflected in this meta-network. Specifically, we have that the hidden and observed subnetworks have a poly-tree structure. Then, the static BN as well as the dynamic BN both have poly-tree structure and this can be exploited during inference.

Figure 31: Verdande Model Class



Verdande Model Class

Caja Mar Model Class

4.2 AMIDST Model Class

Figure 32: CajaMar Model Class

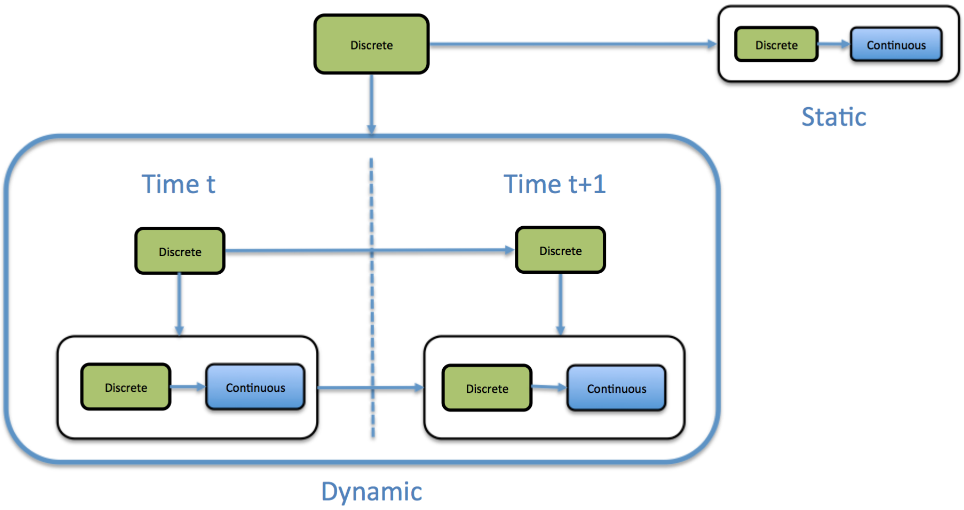
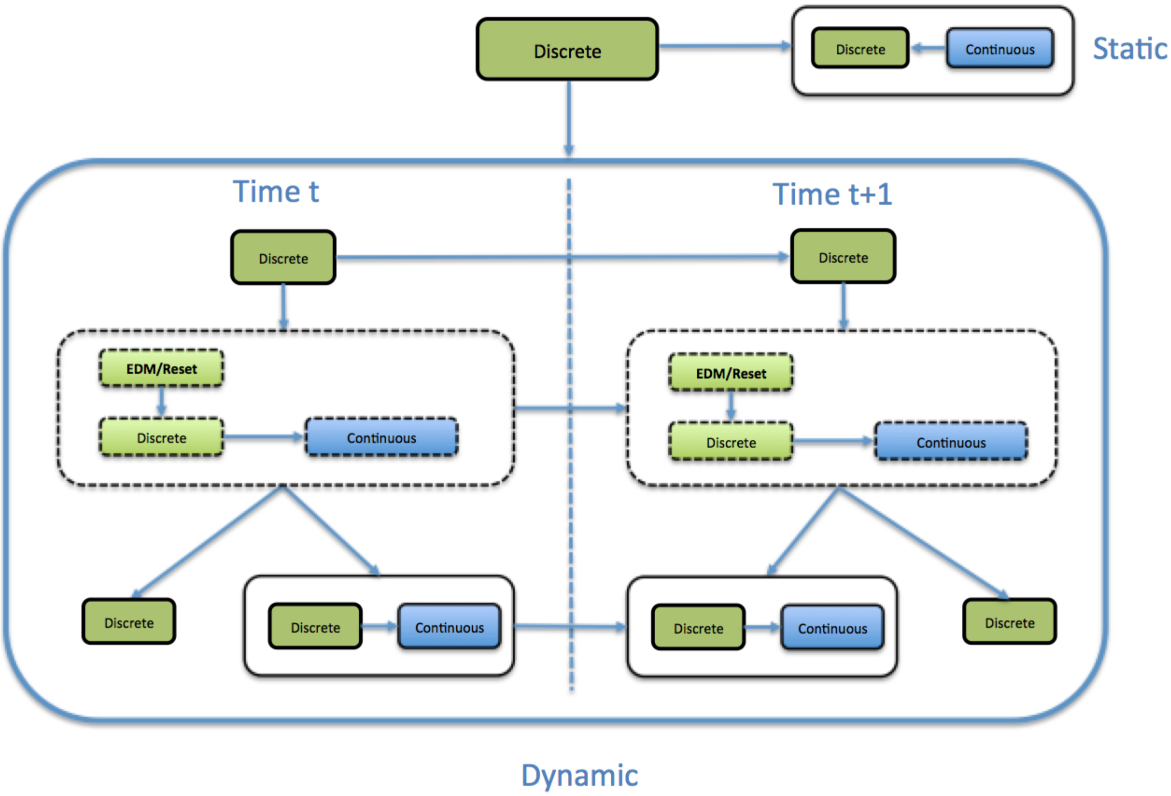


Figure 33: CajaMar Model Class


Discrete
Discrete
Continuous
Static
Time t
Time t+1
Discrete
Discrete
EDM/Reset
Discrete
Continuous
EDM/Reset
Discrete
Continuous
Discrete
Discrete
Continuous
Discrete
Continuous
Discrete
Dynamic