

Contents

<b>1</b>	<b>Executive summary</b>	<b>3</b>
1.1	CajaMar Models . . . . .	4
1.1.1	Introduction . . . . .	4
1.1.2	Predicting probability of default . . . . .	4

## Document history

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

---

# 1 Executive summary

## 2 Preliminary Models

### 2.1 Daimler Models

#### Introduction

The basic settings of the DAIMLER's use-case [1] are as follows. Let us suppose we are driving our car, which will be referred 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 problem consists in the early recognition a maneuver either of the EGO or another relevant car in the traffic scene. In total, the system is expected to recognize the following set of manoeuvres (a visual description of them is given below in Figure ??):

1. **Object-CutOut:** A vehicle that was driving in front of us is leaving the EGO lane.
2. **Object-CutIn:** A vehicle is moving to the lane where the EGO vehicle is placed.
3. **EGO-CutOut:** The EGO vehicle is leaving the lane where it was driving.
4. **EGO-CutIn:** The EGO vehicle is moving to a new lane already occupied for another vehicle.
5. **Object-Follow:** There is no lane change. The EGO is driving and there is some other vehicle in front.
6. **Lane-Follow:** There is no lane change. The EGO is driving and there is not any other vehicle in front.

The data used to address this problem do not contain raw data from the video, radar and on-board sensors. The manoeuvre recognition system directly works with 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 ?? 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.

Using this "object data", Daimler has developed a probabilistic graphical model [2] 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 layers as shown in Figure ??.

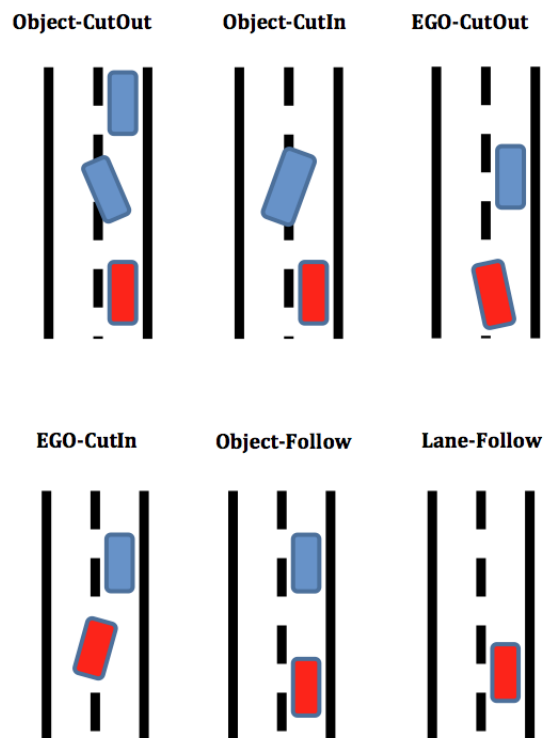


Figure 1: Different maneuvers which should be identified by the AMIDST system. Red blocks represents the EGO vehicle and blue blocks represents other vehicles in the scene. In the first four maneuvers, there is a lane change event or, under Daimler's terminology, a "Lane Marking Crossing" (LMC) event.

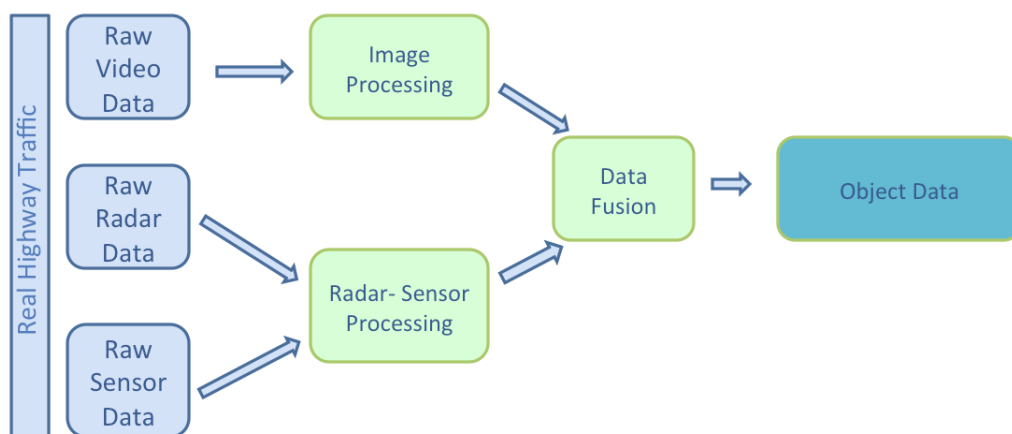


Figure 2: Daimler's Data Flow.

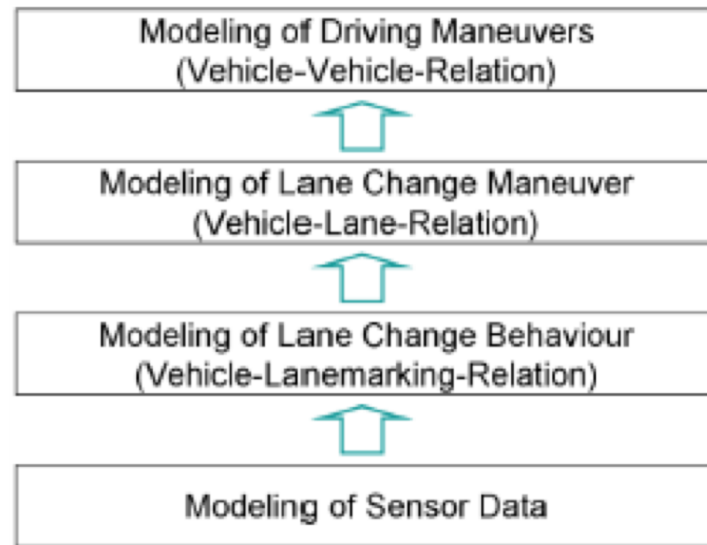


Figure 3: Hierarchical layers for the recognition of driving manoeuvres.

In a first step it is only modelled the sensor data. Using this layer, a new layer is created on top with the goal of detecting a lane change behaviour. The detection of a lane change behaviour allows the system to model the lane change manoeuvre in a higher layer. Finally, with this information, the system is able to identify the kind of driving manoeuvre which is taking place between a pair of vehicles.

### The static-OOBN model

As commented above, this model will work with the so-called “object data”. This data mainly consists on 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. And the whole modelling is structured in hierarchical layers as detailed in Figure ???. This hierarchical modelling was previously implemented in [] using an object-oriented Bayesian network (OOBN) [].

The general structure of this OOBN model consists of a number of abstraction levels (see Figure ??): all measured and/or computed signals  $S$  are handled with their uncertainties  $\sigma^2$ . These are represented as object classes at the lowest level (class  $S$ ) of the OOBN. The real values  $\mu$  of evidence signals are then used at the next level of hierarchy to evaluate the hypotheses (class  $H$  in Figure ??). The combined evaluation of several hypotheses results in the prediction of events, class  $E$ . In our case, the events are modelling traffic maneuvers of the own and neighbour vehicles.

As commented above, the observations characterizing a situation are acquired from sen-

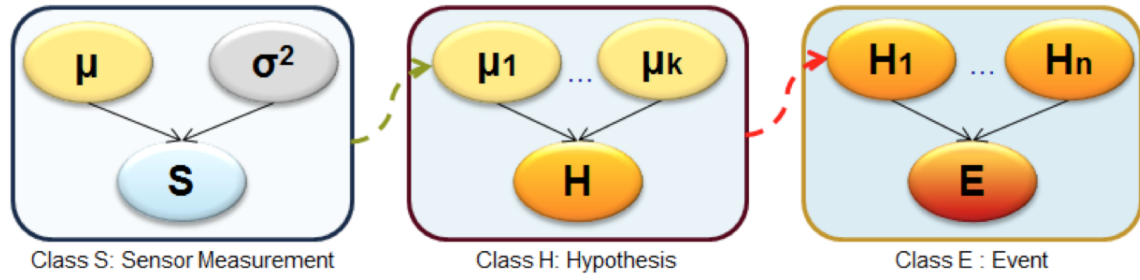


Figure 4: Static-OOBN model for the prediction of an event (maneuver).

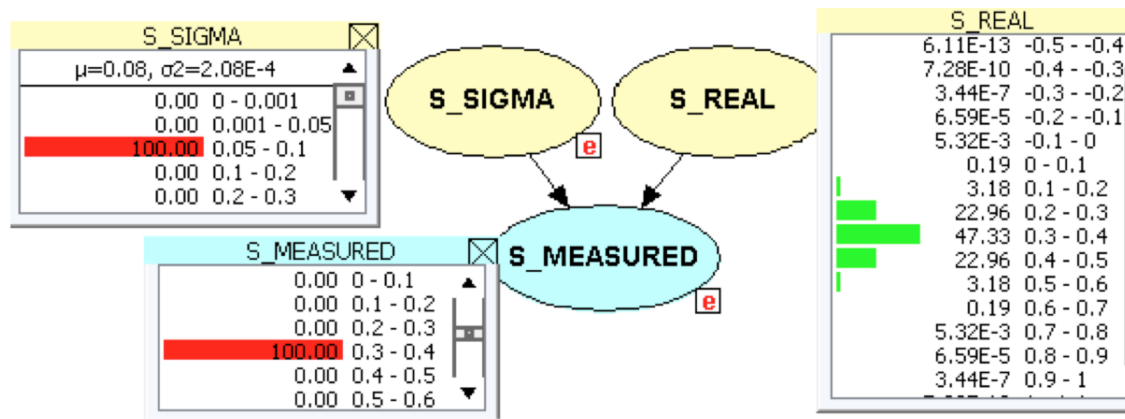


Figure 5: BN fragment for modeling of sensors uncertainties with a discrete “measurement” variable.

sors and computations (see Figure ??) and, in consequence, are *measured data*. If the measurement instrument is not functioning properly (due to sensor noise or fault), then the sensor-reading ( $S\_MEASURED$ ) and the real variable ( $S\_REAL$ ) under measurement need not to be the same. This fact imposes the causal model structure as shown in Figure ?. The sensor-reading of any measured variable is conditionally dependent on random changes in two variables: real value under measurement ( $S\_REAL$ ) and sensor fault ( $S\_SIGMA$ ).

The situation features used for maneuver recognition are structured along three main dimensions: lateral evidence (LE), trajectory (TRAJ), and occupancy schedule grid (OCCGRID). They represent the three hypotheses (see Figure ??), which are modelled by the corresponding OOBN-fragments. For more details see [13], [14]. The hypothesis LE is shown in Figure ?. Its conditional probability distribution is represented by a sigmoid (logistic) function to expresses the growing probability for the lateral evidence on crossing the lane marking, when the vehicle is coming closer to the lane

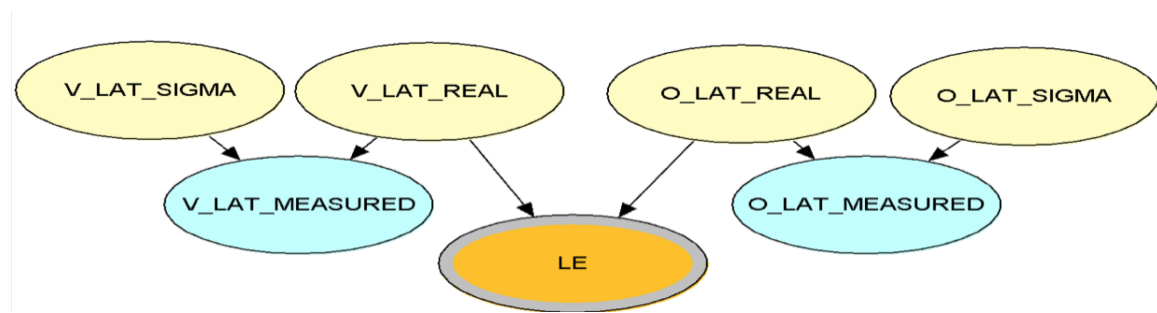


Figure 6: BN fragment for modeling of sensors uncertainties with a discrete “measurement” variable.

marking (modeled by *O\_LAT\_MEASURED*) by growing lateral velocity (modeled by *V\_LAT\_MEASURED*).

Figure ?? abstractly shows how these hypotheses are combined into events, which in our automotive scenario correspond to the different driving maneuvers: lane follow, lane change (cut-in, cut-out), expressed for ego and surrounding objects, see [12], [13].

### The dynamic-OOBN model

The above described static OOBN is able to detect a maneuver 0.6s before execution. The goal is to extend the prediction horizon for manoeuvre recognition at least to 1-2 seconds (max. 4-5 seconds ahead) before the actual lane marking crossing, which is of advantage for the adaptive cruise control. Most precisely, and as indicated in the Use Case 8 on the Requirement Analysis, the area under the ROC curve (AUC) should be greater than 0.96 for 1 second and greater than 0.9 for 2 seconds.

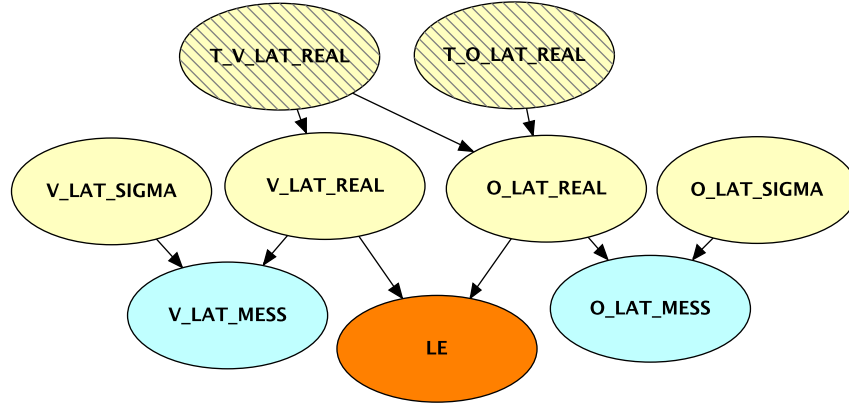
Figure [timedetection] shows the evolution on time for velocity and offset in an EGO\_CutOut manoeuvre. The vertical bar indicates the moment in which the manoeuvre has been recognised by the static OOBN. By taking the temporal properties of the data into account on the model, we should be able to predict the manoeuvre earlier on time.

Each manoeuvre can be considered as a process, developing in time, i.e., as data stream given by a time sequence of the transition from lane follow into lane change manoeuvre. The dynamic extension involves copies of the static OOBN for different number of time steps in the time window (e.g. see Fig. ?? where the two top nodes are temporal clone defining the share belief state between consecutive time steps creating a first order Markov process), if also the requirement on earlier prognostics of maneuver is to be satisfied.

1. **Dynamics on Lateral Evidence (LE)** A good starting point to model the dynamics of the data involves the variables that capture the lateral evidence for



Figure 7: Daimler Temporal Model



the different vehicles, given its relevance and simplicity. The dynamic BN (DBN) incorporates the trend of change for the real values, where their physics relations are represented as causal dependencies between the time steps  $dt$ , e.g. in Fig. ?? the transition function of O-LAT at time  $t$ ,  $O(t)$ , is modeled as a Gaussian distribution. Its mean is affected by  $O(t-1)$ , and by V-LAT at time  $t-1$ ,  $v(t-1)$ :

$$O(t) = O(t-1) + v(t-1)dt + N \quad (1)$$

where  $N$  denotes a white noise  $N(0, \sigma^2)$  due to possible acceleration term  $(adt^2)/2$ , which is assumed to be small for a time step in the order of 102 milliseconds.

The shaded nodes represent the development of the real values of observations over several time steps in the time window. Thus, their trend estimation contributes to the prediction of probability of transition from a lane follow to a lane change manoeuvre.

A DBN induces a number of constraints on the compilation of the network into a computational structure. One constraint relates to transferring the belief state from one time slice to the next where the belief state is the probability distribution over the variables shared by neighbouring time slices. In general, the belief state is transferred as a joint distribution. This means that approximate methods [?] may have to be considered for meeting the requirements of the target platform.

## 2. New hypothesis: Relative Dynamics (REL\_DYN)

Earlier prediction of manoeuvre intentions can be achieved even before any development of the trend for lateral evidence LE has been observed. A first indication of possible lane change intention can be observed through the relative dynamics between one vehicle (host or object) and the vehicles in front of it on the same lane.

Once again, the goal is to further increase the prediction horizon for manoeuvre recognition (up to 5 seconds).

We can include qualitatively new information based on driving experience, which indicates a need for a lane change if a slower vehicle is driving in front of the own vehicle on the same lane. 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 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”).

Figure [timedetectionRelDyn] shows the evolution on time for the velocity and distance in an EGO.CutOut manoeuvre. The vertical bar indicates the moment in which the manoeuvre has been recognised by the static OOBN. By taking the temporal properties of the relative dynamics into account on the DBN, we should be able to predict the manoeuvre even earlier on time.

By analogy to Fig. ??, the original OOBN has been extended with the hypothesis “relative dynamics” (REL\_DYN), as shown in Fig.???. This BN fragment models the hypothesis REL\_DYN with 3 states Left/Follow/RIGHT, utilising the independency assumption for the discrete variables V\_REL\_MEASURED and X\_REL\_MEASURED.

If we compare the structure of this network with that of Fig. ??, we can observe two additional nodes: REL\_DYN\_V\_REL\_OBJ and REL\_DYN\_X\_REL\_OBJ. They are the results of a modelling trick to simplify the EM-learning of parameters from data for the static BN fragment.

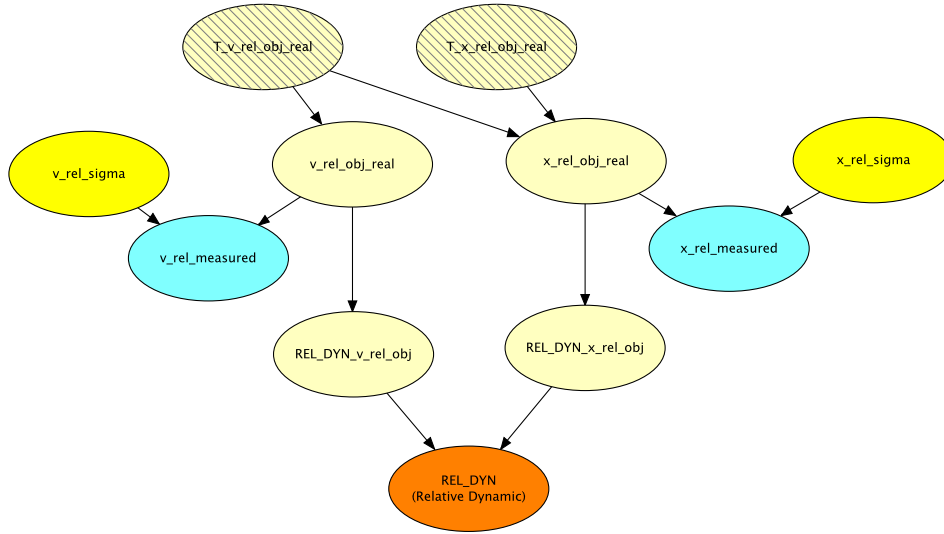
Note that the new REL\_DYN hypothesis introduced would require two instances in the OOBN, one for the relative dynamics of the EGO with the OBJ in front, and another one for the OBJ and another OBJ in front of it. Each REL\_DYN would indicate if the EGO and the OBJ cars are going to turn right, left or continue straight.

## 2.2 CajaMar Models

### 2.2.1 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 to a default in two years. The other task is to obtain

Figure 8: Daimler Temporal Model with relative dynamics



good customers profiles in terms of risk so that marketing campaigns can be specifically directed to low risk customers.

### 2.2.2 Predicting probability of default

The main task is predicting the probability that an operation will result in a default **2 years before** it actually happens. It is therefore a *supervised classification* problem, currently solved using *logistic regression* over *27 predictors*. These predictor variables (many of them manually built by CajaMar's experts) describes the financial behaviour of the customers in the last 180 days, which is a limit imposed by the Bank of Spain. The provided database is *imbalanced* -around 10% of defaulters-.

We will initially consider two different approaches.

#### Static Model

In this first approach we ignore 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, be in bankrupt for becoming unemployed). Here we just build a prediction model where given the financial behaviour of the client over the last 180 days, it predicts whether the client will default or not in 2 years.

Figure 1 shows the general structure of this static model. Each yellow 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). The model only considers variables referring to the last 180 days. The red node models the possibility that the customer is a defaulter in the next two years.

The process of building this static model would consist of the following steps:

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

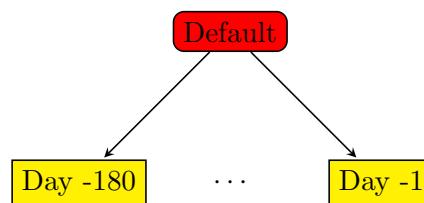


Figure 9: Global structure of the static model. Each yellow box represents a set of variables measures during the same day. The variables within a box can be connected (according to a tree structure and, globally, conforming a TAN).

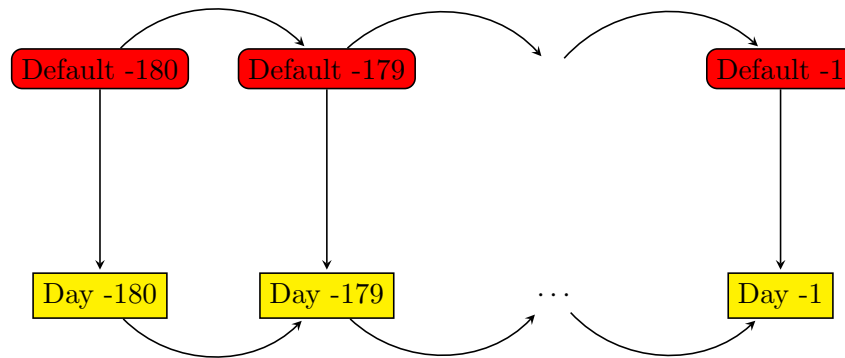


Figure 10: Global structure of the dynamic model. Each yellow box represents a set of variables measures during the same day. The variables within a box can be connected (according to a tree structure and, globally, conforming a TAN) as well as variables between two consecutive days. Red box refer to the possibility that client is defaulter and are temporal connected.

## Dynamic Model

In this approach we will consider the 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 month to another, the level of incomes, etc.) as well as the labelling as defaulter or non-defaulter customer (e.g. one customer can be creditworthy and, but after some time, be in bankrupt for becoming unemployed).

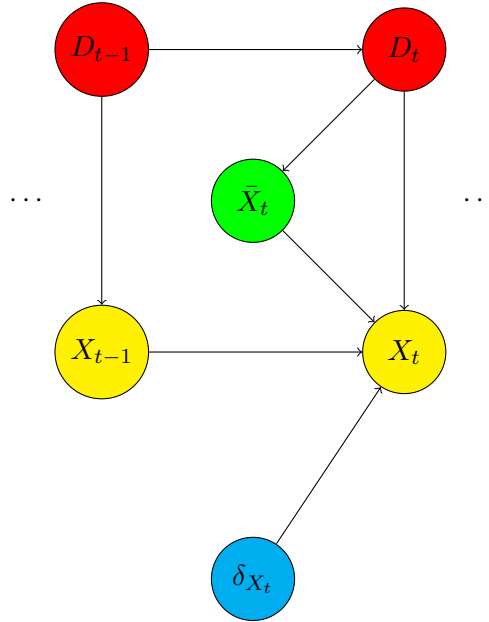


Figure 11: Basic component of the structure of the dynamic model.

Figure 2 represents the global idea of the temporal model. It can be compactly represented by a dynamic Bayesian network made of components as the one displayed in Figure 3.  $D_t$  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  $t_1$ :  $X_{t-1}$  as well as to a *memory variable*  $\bar{X}_t$  that represents the average value of  $X$  during the last 180 time slices (days). Finally, an indicator variable  $\delta_{X_t}$  may be included if the variable is such that is observed many times at point 0. This is the case as, for instance, payments by credit card, where many of the days can be equal to zero for most of the customers.

The process of building this static model would consist of the following steps:

- Construct *1 table* for each day.
- Build a *dynamic* BN classifier (NB or TAN like structure extended in a dynamic fashion).

- Update risk profiles using the classifier.

## Data Analysis