Contents

1 Executive summary **3**

2 Preliminary Models **4**

2.1 Daimler Models 4

2.1.1 Introduction 4

2.1.2 The static-OOBN model 7

2.1.3 The dynamic-OOBN model 8

2.2 CajaMar Models 12

2.2.1 Introduction 12

2.2.2 Predicting probability of default 13

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

2.1.1 Introduction

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; the developed models 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 of their dynamic counterparts.

Description of Application Scenario 1

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 of the EGO or another relevant car in the traffic scene (OBJ). In total, the system is expected to recognised the following set of manoeuvres (a visual description of them is given below in Figure 1):

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 by another vehicle.
5. **Object-Follow**: There is no lane change. The EGO is driving and there is some other vehicle in front.

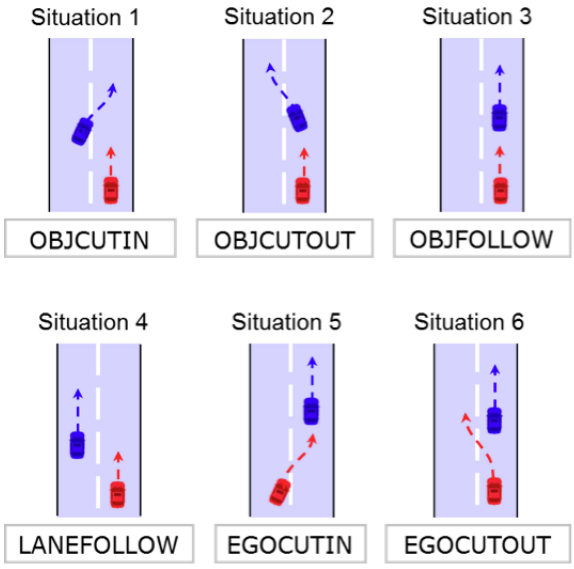


Figure 1: Different maneuvers which should be identified by the AMIDST system. Red blocks represents the EGO vehicle and blue blocks represents other vehicles in the scene.

6. **Lane-Follow**: There is no lane change. The EGO is driving and there is not any other vehicle in front.

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 2 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 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 layers as shown in Figure 3.

The sensor data is modelled in the first step. 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.

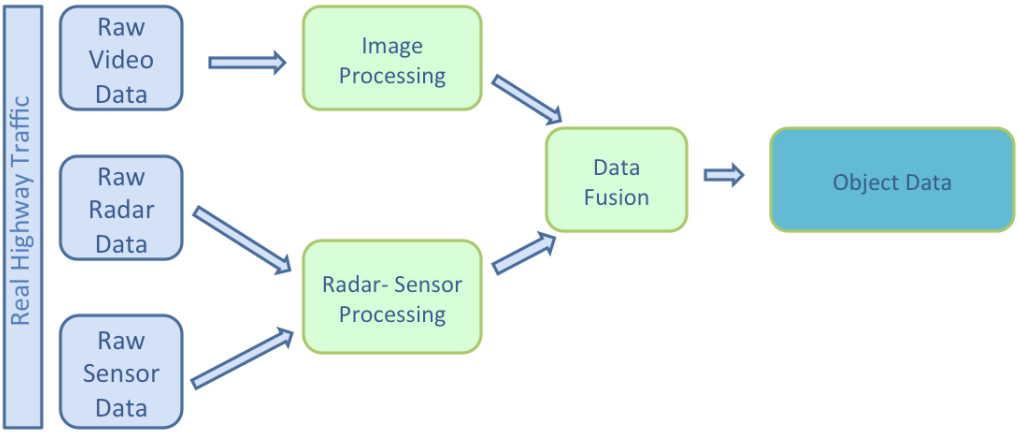


Figure 2: Daimler's Data Flow.

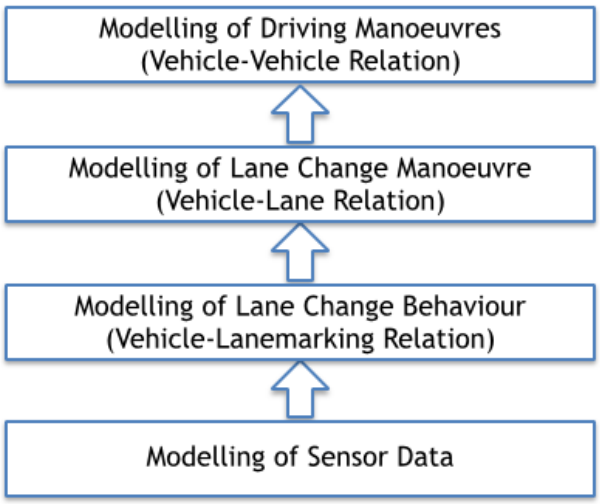


Figure 3: Hierarchical layers for the recognition of driving manoeuvres.

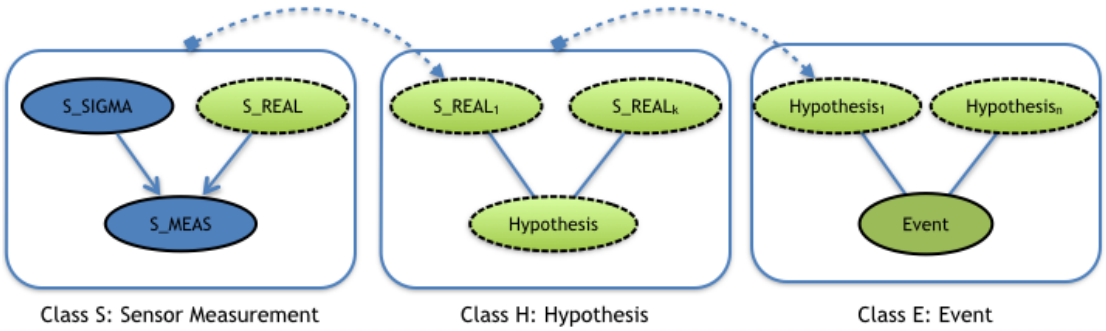


Figure 4: Static-OOBN model for the prediction of an event (maneuver) [**?**].

2.1.2 The static-OOBN model

As commented above, this model will work 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 whole modelling is structured in hierarchical layers as detailed in Figure 3 and it has been previously implemented [**?**] using an object-oriented Bayesian network (OOBN) [**?**].

The general structure of this OOBN model consists of a number of abstraction levels (see Figure 4): all measured and/or computed signals S_MEAS are handled with their uncertainties S_SIGMA. These are represented as object classes at the lowest level (class S) of the OOBN. The real values S_REAL of evidence signals are then used at the next level of the hierarchy to evaluate the hypotheses (class H in Figure 4). The combined evaluation of several hypotheses results in the prediction of events, class E. In our case, the events are modelling traffic manoeuvres of the own and neighbour vehicles.

As commented above, the observations characterising a situation are acquired from sensors and computations (see Figure 2) and, in consequence, they are regarded as *measured data*. If the measurement instrument is not functioning properly (due to sensor noise or fault), then the sensor-reading (S_MEAS) and the real variable (S_REAL) under measurement need not to be the same. This fact imposes the causal model structure as shown in the first part on Figure 4. 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 manoeuvre recognition are structured along three main dimensions: lateral evidence (LE), trajectory (TRAJ), and occupancy schedule grid (OCCGRID). They represent the three hypotheses (see Figure 4), which are modelled by the corresponding OOBN-fragments [**?**]. The BN fragment for the hypothesis LE is shown in Figure 5. Its conditional probability distribution is represented by a sigmoid

Figure 5: Static BN fragment for the LE hypothesis.

(logistic) function. This 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 (modelled by O_LAT_MEAS) and the increase of its lateral velocity (modelled by V_LAT_MEAS).

The right-hand square on Figure 4 abstractly shows how these hypotheses are combined into events, which in our automotive scenario correspond to the different driving manoeuvres: lane follow, lane change (cut-in, cut-out), expressed for ego and surrounding objects [**?**].

2.1.3 The dynamic-OOBN model

The above described static OOBN is able to detect a manoeuvre 0.6s before execution. As detailed in the DoW [], 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. Moreover, this early detection of the manoeuvre should be at not expenses of the prediction accuracy, as indicated on Daimler's use cases [**?**], 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 6 shows an example of the current performance, and limitations, of the static-OOBN model. In these figures we plot the evolution on different time-steps for lateral velocity and lateral offset to a lane marking in an ongoing Object-CutOut and Object-Cutin manoeuvres (see Figure 1). The vertical black bar in these figures 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 line corresponds to the lateral offset and lateral velocity of the EGO car, which is just following the lane (LF), and the green line 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) 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

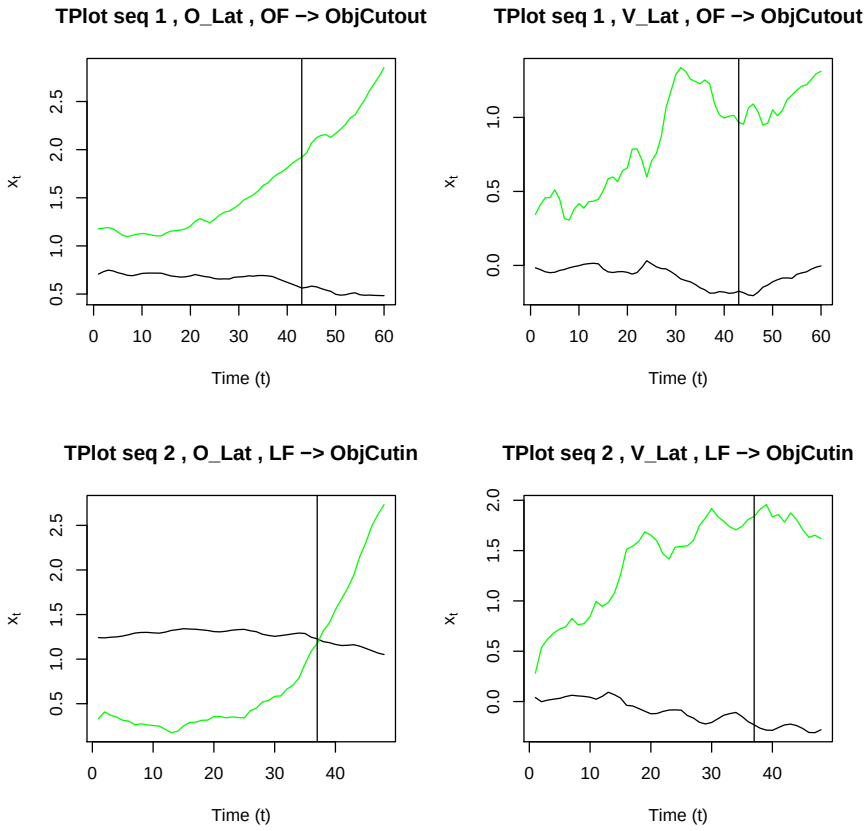


Figure 6: Daimler Time Plot

how the lateral offset steadily increases, what clearly indicates that the Object car is leaving its current lane in both manoeuvres.

the

Although the manoeuvre is clearly identified before it completes in both cases (approx. 0.6 seconds in advance), it is desired to predict it 1-2 seconds in advance. Looking at the evolution of lateral offset and velocity in Figure 6, we could argue that the detection of the manoeuvre can be performed earlier (i.e. when the lateral offset and lateral velocity starts to increases consistently). This is one of the basic piece of evidence that motivated the development of the dynamical version of the static-OOBN model [**?**]. Another piece of evidence comes when we look at sample correlograms for lateral velocity and offset (that is, the correlation of the data with lagged values of themselfves) plotted in Figure 7. There we can see as there is a strong temporal dependency (the correlation coefficients are high) for the two variables when we consider not many time steps between the temporal observations. So, the employment of a dynamic model to make predictions

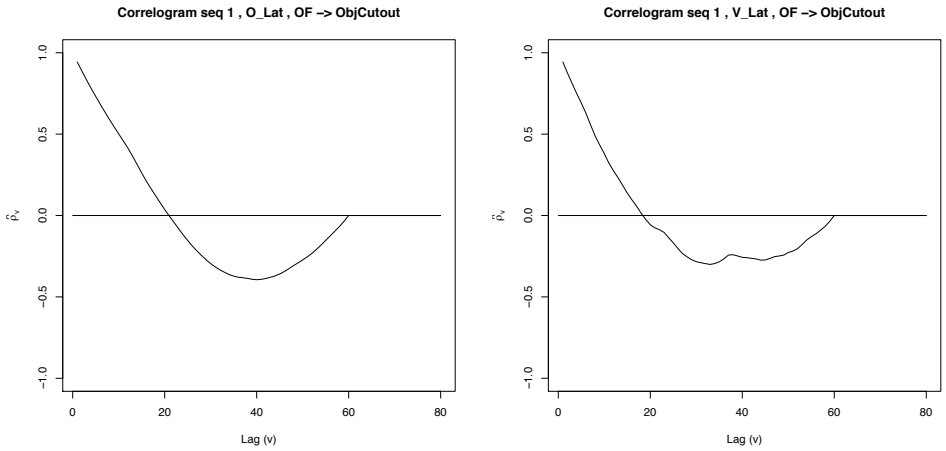


Figure 7: Correlograms for Lateral velocity and offset.

about the evolution of the lateral velocity and lateral offset in a near future seems to be reasonable with this analysis. So, our main assumption is that the use of dynamic Bayesian network models (see Section **??**) 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.

As explained in Section **??**, the dynamic extension involves copies of the static OOBN for different number of time steps in the time window. Fig. 8 shows an example for the LE hypothesis, where the two top nodes are temporal clones defining the share belief state between consecutive time steps, and hence creating a first order Markov process. This one the model 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, if we build a partial-correlogram as the ones plotted in Figure 9, we can see that this assumptions seems to reasonable because the influence of O_LAT_{t-2} on O_LAT_t given O_LAT_{t-1} is close to null (i.e. the value at $v = 2$ is close to null in Figure 9). Same applies to V_LAT.

As described in [**?**], the proposed dynamic BN (DBN) aimed to incorporate 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. 8 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 $t-1$, $v(t-1)$:

$$O(t) = O(t-1) + v(t-1)dt + N \tag{1}$$

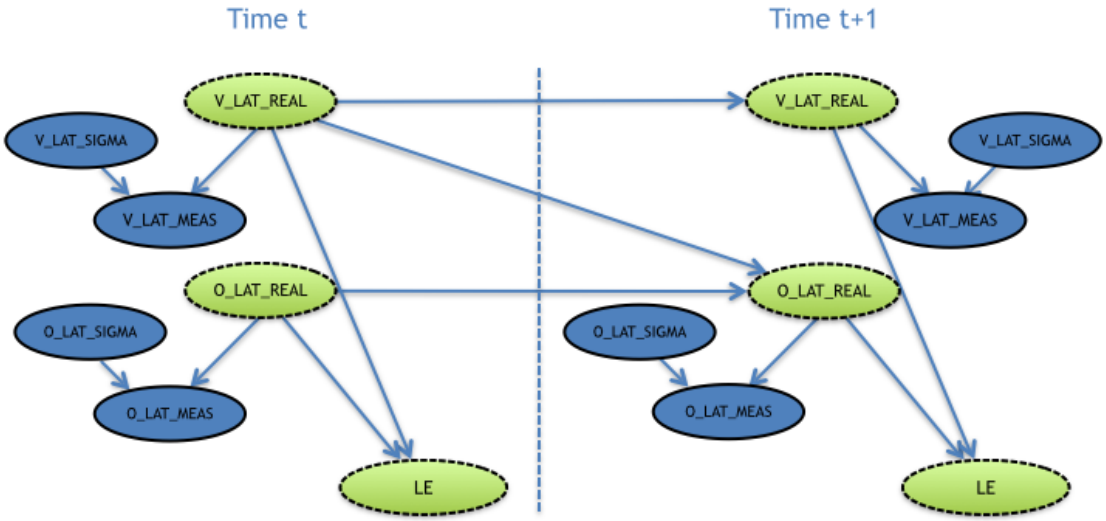


Figure 8: Daimler Temporal fragment for the LE hypothesis.

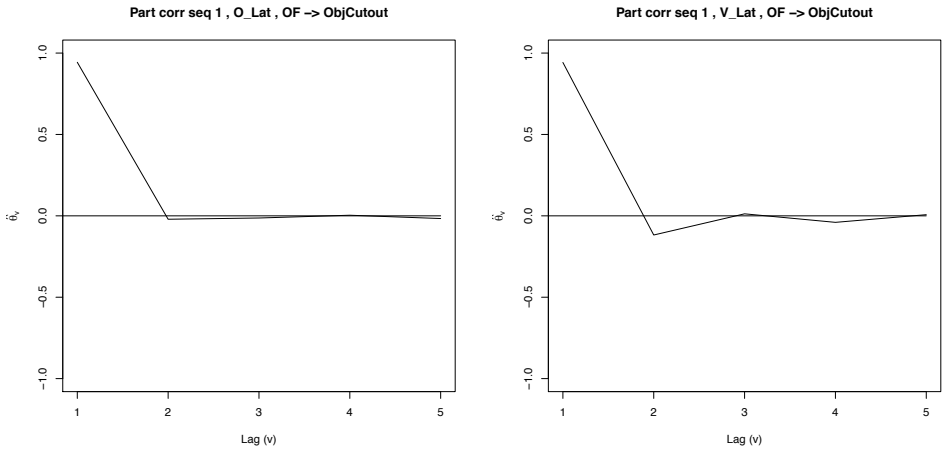


Figure 9: Partial-Correlograms for Lateral velocity and offset.

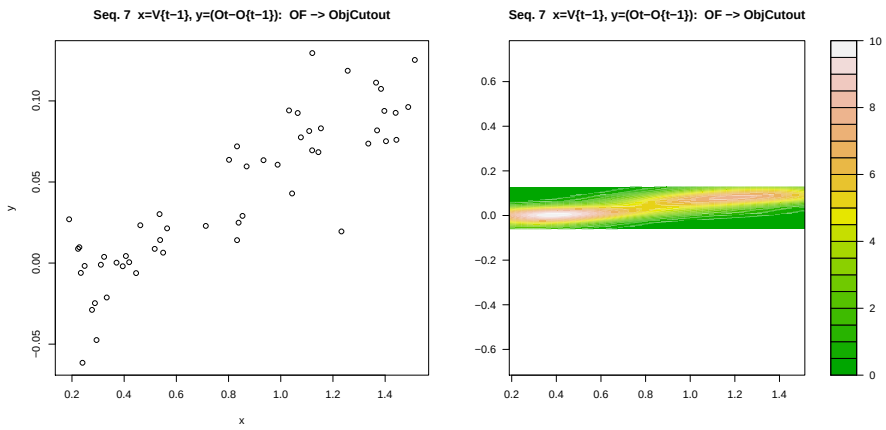


Figure 10: Time plot for $v(t-1)$ vs $O(t)-O(t-1)$. Linear correlation can be observed.

where N denotes a white noise $N(0, \sigma^2)$ which is assumed to be small. In order to corroborate the validity of this distributional assumption, we also analysed the hypothesis $O(t) - O(t-1) = \Delta O = v(t-1)dt + N$ on our data. Figure 10 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.

Additionally, we believe that even earlier prediction of manoeuvre intentions could be achieved 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), and this approach will be further explored in future stages of the project.

Finally, in Figure 11 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 manoeuvre prediction is then determined by the combination of these hypotheses and the position of the OBJ with respect to the EGO car.

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 in default within two years. The secondary task consists in

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

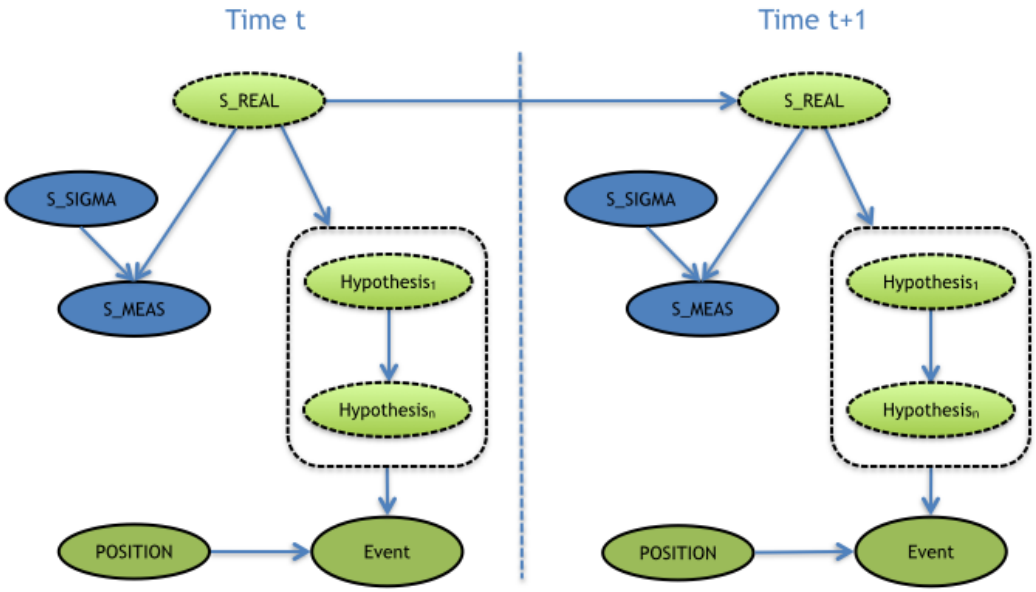


Figure 11: Daimler Temporal model with several hypothesis.

obtaining good customer profiles in terms of risk, so that marketing campaigns can be specifically targeted to these low risk customers.

2.2.2 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 CajarMar, this risk evaluation protocol is implemented by evaluating whether the client is going to default or not within the following two 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

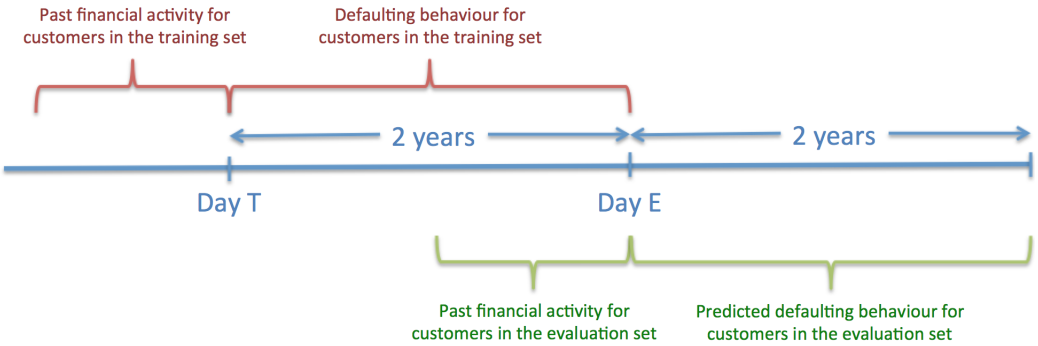


Figure 12: Time-line of the generation of the evaluation and training data sets.

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 12 shows the time-line of the generation of the evaluation and training data set, which is further explained below.

- **Model Evaluation Data Set:** We denote by day E to the day at which this evaluation data set is created. This data set contains a record for every client to be evaluated at that day E. 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-demographics 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

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

a customer, $36 \cdot P$ variables are included as information about the past paying behaviour.

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. 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).

  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.

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 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 13 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.

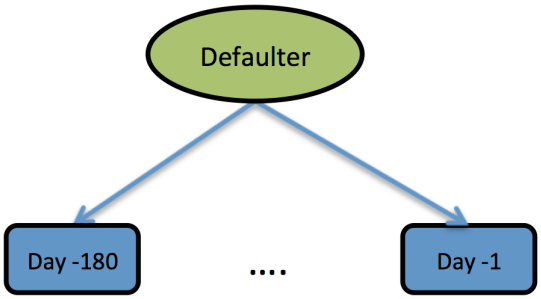


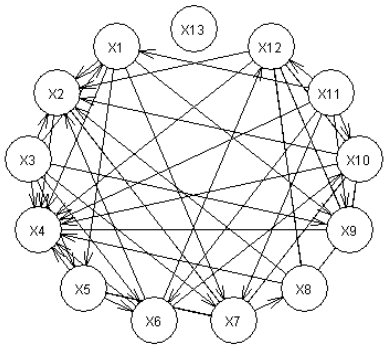
Figure 13: Global structure of the static model. Each blue box represents a set of variables measures during the same day. The variables within a box can be connected (e.g. according to a tree structure and, globally, conforming a TAN).

The variables within a box can be connected (e.g. according to a tree structure and, globally, conforming a TAN). Figure 14(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 14(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 15(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.

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



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

Figure 14: .

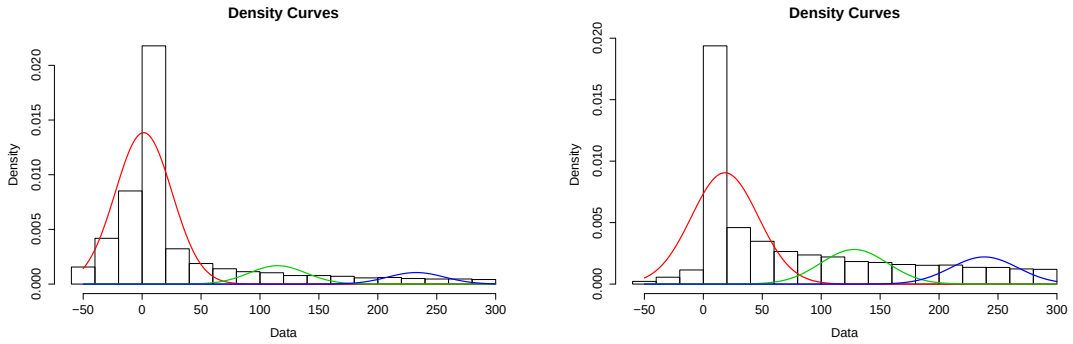


Figure 15: Mixture of Gaussian approximation of *end-of-day balance* for defaulter (left) and non-defaulter (right).

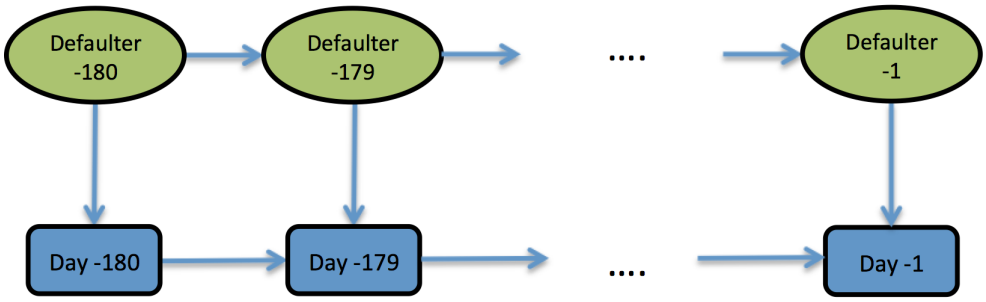


Figure 16: Global structure of the dynamic model. Each yellow box represents a set of variables measures during the same day. The variables within a box can be connected (according to a tree structure and, globally, conforming a TAN) as well as variables between two consecutive days. Red box refer to the possibility that client is defaulter and are temporal connected.

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 16 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 17. 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

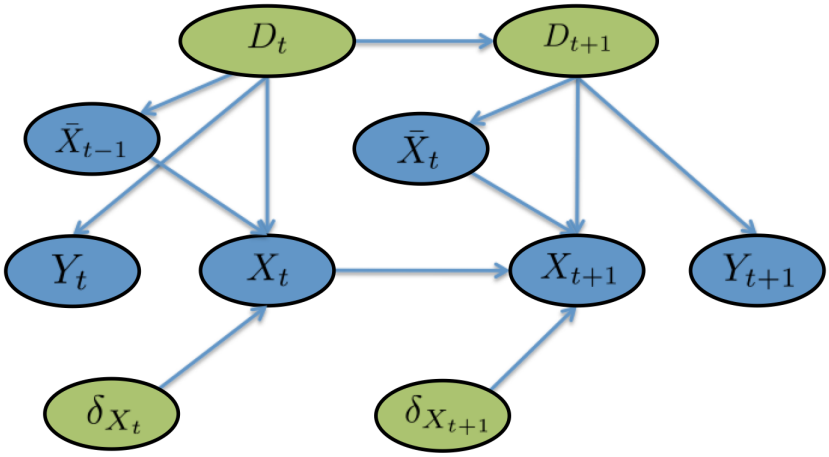
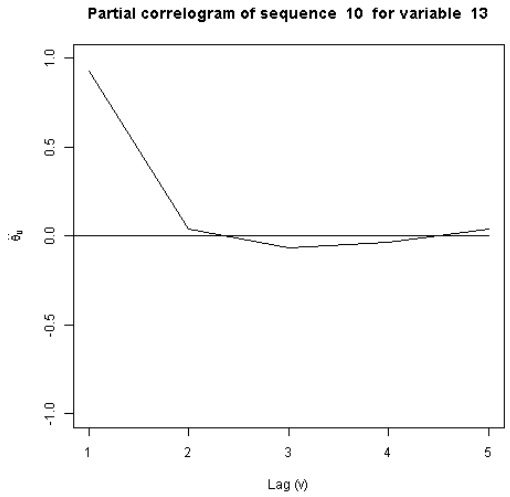


Figure 17: Basic component of the structure of the dynamic model.

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 18 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 18, the partial correlograms drop to 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.

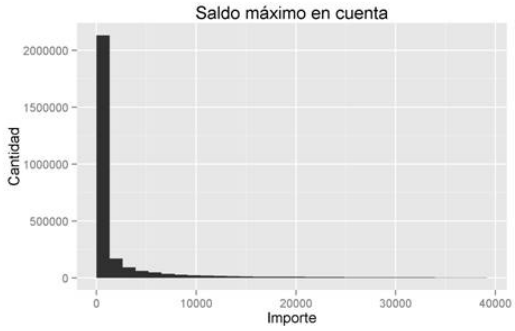
Finally, an indicator variable δ_{X_t} may be included if the variable is such that is observed many times at point 0. This is the case for payments made by credit card or the historical monthly outstanding amount on the account for instance, whose value can be equal to zero for a large number of days for most customers, as shown in Figure 19. Figure 19(a) displays the histogram of this variable including all values, and 19(b) when the zero values are not considered.

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 20 show values very close to zero for all lags. These variables are hence not linked through consecutive time steps, which are



Partial correllogram for another variable

Figure 18: Partial correllogram for variables A and B. A first order Markov assumption seems reasonable.



Outstanding amount without zeros

Figure 19: Frequency histogram of the historical monthly outstanding amount

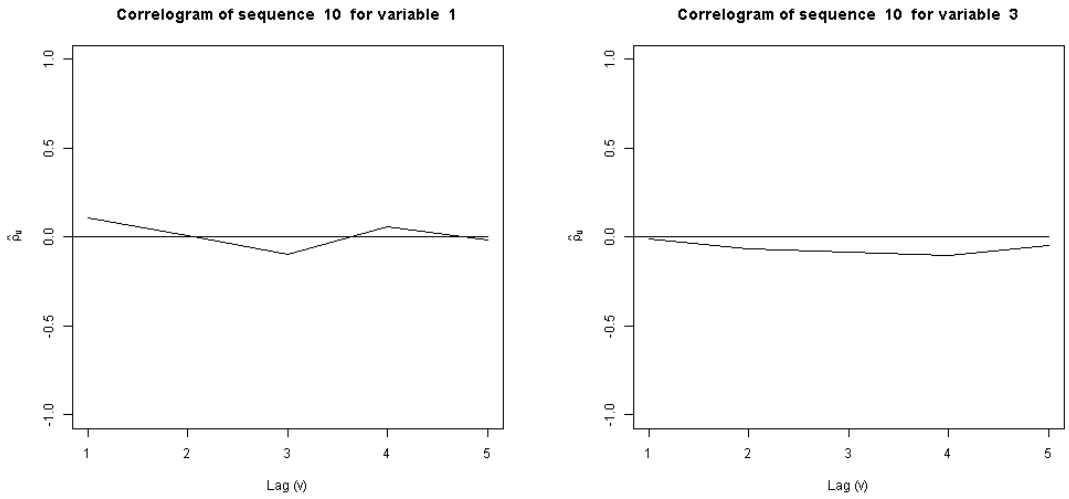


Figure 20: Correllogram for variables 1 and 3. No temporal dynamic shown.

represented by Y_t and Y_{t+1} in Figure 17. 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.